Survey on Enhanced Sequential Pattern Mining

Mehta Devshri[1] , Madhuri Vaghasia [2]
Student of Masters Engineering [1], Assistant Professor[2]
Department of Computer Science and Engineering, B H Gardi College of Engineering and Technology, India

***Abstract* - Data mining is the method or the activity of analyzing data from different perspectives and summarizing it into useful information. Among different tasks in data mining, sequential pattern mining is one of the most important tasks. Sequential pattern mining involves the mining of the subsequences that appear frequently in a set of sequences. Sequential pattern mining algorithms using a vertical representation are the most efficient for mining sequential patterns in long sequences, and have excellent overall performance. The vertical representation allows generating patterns and calculating their supports without performing costly database scans. i.e. only one database scan is used. However, a crucial performance bottleneck of vertical algorithms is that they use a generate candidate and test approach that can generate a large amount of infrequent candidates. To address this issue, we use pruning candidates based on the study of item co-occurrences. There is a new structure named CMAP (Co-occurrence MAP) for storing co-occurrence information.**

Index Term - Sequential Pattern, Minimum support, vertical database format, SPAM, Co-occurrence information

I . INTRODUCTION

Discovery of sequential patterns from large dataset is first introduced by Agrawal A sequence is a collection of an ordered list of item set where item set is a collection of unordered, non empty set of items. For a given set of sequence, to find a set of frequently occurring sub sequences. Any subsequence is said to be frequent if its support value greater or equal to minimum support or threshold value. We can define Support as occurrences of that item is present in the given number of sequences. Mining useful patterns in sequential data is a challenging task. Sequential pattern mining plays an important role in data mining and is essential to a wide range of applications such as customer purchase sequence , the analysis of web click-streams, ,biological data like DNA, protein sequence analysis, gen sequence analysis and e-learning data.

Several different types of algorithms are available for sequential pattern mining. Like GSP,SPADE, SPAM, CloSPan, LAPIN , Freespan, prefix Span, Apriori etc. Sequential pattern mining algorithm can be categorized using horizontal database format or vertical database format.

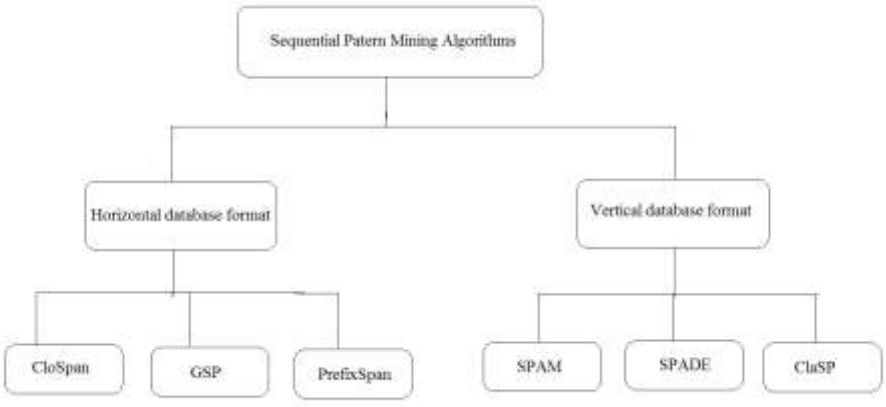

Fig 1. Sequential Pattern mining Algorithms

II. SPAM(Sequential PAttern Mining) ALGORITHM

SPAM is one of Apriori–all type of sequential pattern mining algorithm. It assumes that entire database used for algorithm completely fit into memory. SPAM algorithm generate the lexicographic tree.

2.1 Lexicographic Tree Generation

For the given sequences it generates a lexicographic tree . The root of tree always starts with empty string. The children node of it are formed by sequence-extended (S-Step) or an item set extended sequence(I-Step) .Let take one example of Protein sequences.

Sequence No.	Sequence
1	MKKV
2	KVM
3	MKKVM

Table 1 Protein Sequence

If minimum support value is 50%, then those sequence with support greater than or equal to minsup are called frequent subsequences. So in the given number of sequence MK, MKK, MKKV, KV, KKV, VM and KVM etc. are some of the frequent subsequences.

Let for sequence containing M, K, V, we have to generate a lexicographic analysis of tree then it starts with root node as {}.

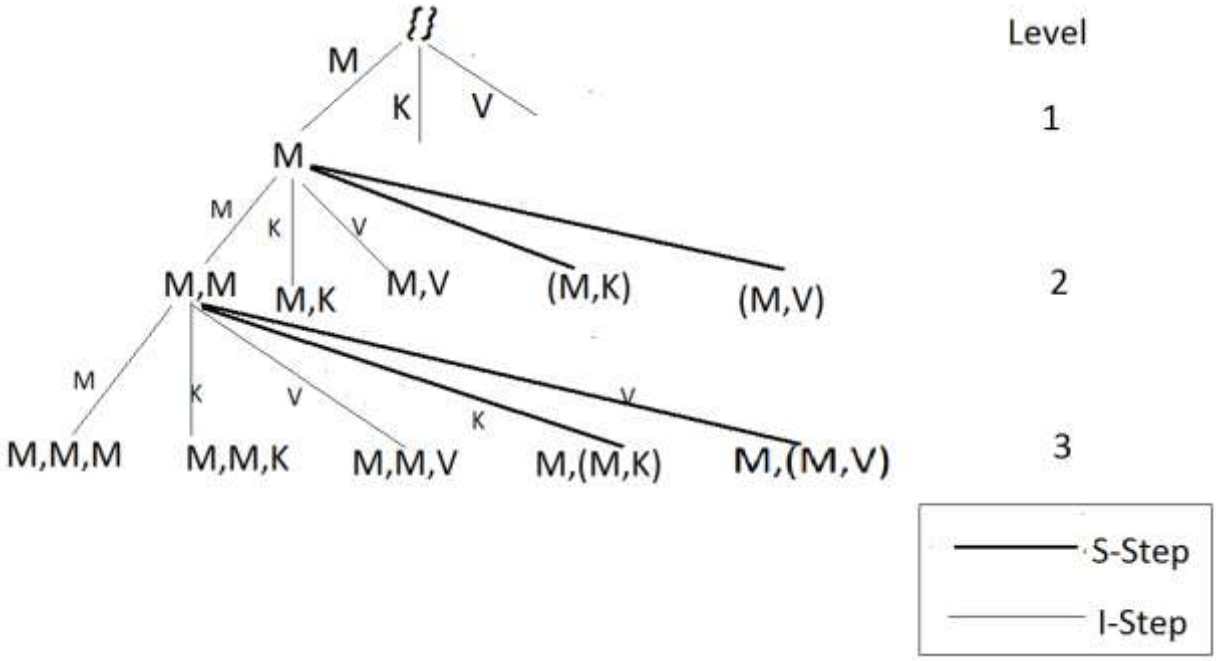

Fig. 2. Lexicographic tree for dataset in table1

At level 1 considers items M, K, V separately. Let us consider M item first, for candidate generation it will take sequence-extension step and generate different sequence as {M, M} {M, K} {M, V}.Similarly generates item set extended sequence by Item set extension step or I-Step as (M, K),(M,V) at level 2. Similarly with different items at level 3 it will generate a sequence of length 3. Once the whole tree is generated for discovering or searching user specified set of subsequence algorithm traverses depth first search strategy. In above tree each single letter represents the item and items between brackets are represents the item set.

- S-Step(Sequence- Extended Sequence) :- A sequence-extended sequence is a sequence generated by **adding a new transaction** consisting of single item to the end of its parents sequence in the tree.
- I-Step(Item -Extended Sequence):- A Item -extended sequence is a sequence generated by **adding an item** to the last item set in parents sequence in the tree.

2.2 Depth First Tree Traversal

SPAM traverse the sequence tree described above in a standard depth – first manner. At each node, support of each sequence extended child and each item set extended child is tested. Those nodes having support value greater than or equal to minimum support are stored and repeats DFS recursively on those nodes. Otherwise the node is not considered by the principle of Apriori.

For the support counting algorithm uses a vertical bitmap structure. How we create the vertical presentation is also explain in this paper.

III Data Representation

SPAM use vertical database format. Using vertical representation of the transaction database it enable more efficient counting. The basic idea of vertical representation is that one can express the transaction database as an inverted list that means foe each transaction one can have list of items that are contained in it.

Vertical representations are most efficient for mining sequential pattern in long sequence and have excellent overall performance. Vertical representations allows generating patterns and calculating their support without performing costly database scan i.e. it **does not require to scan the database more than once.**

As bitmap representation of sequence length maximum is 5 so vertical bitmap consists of 5 bit and as the number of sequences are 3 in table 1 bitmaps with 3 slots.

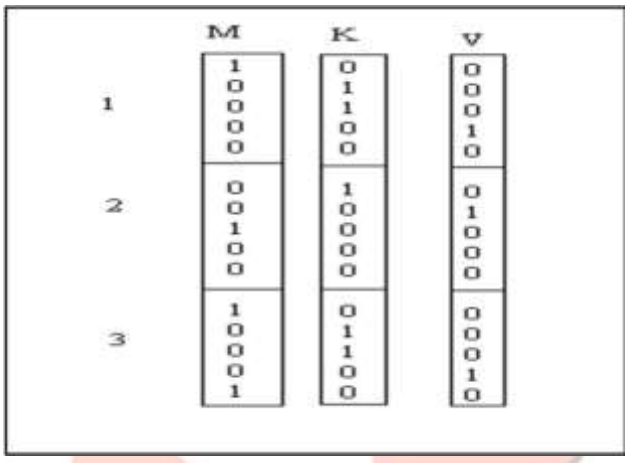

Fig. 3. Vertical bitmap representation for items in table 1

3.1 Candidate Generation

As fig 3 shows the vertical bitmap representation now we have to generate the candidate .Candidate will be generated using either S-Step process or I-Step process.

- For the bitmap presentation of S-Step, every bit after the first index of one is set to zero and every bit after that index position set to be one.

- For the bitmap representation of I-Step, newly added item set's bitmap are logically ANDed with the sequence generated from S-Step.

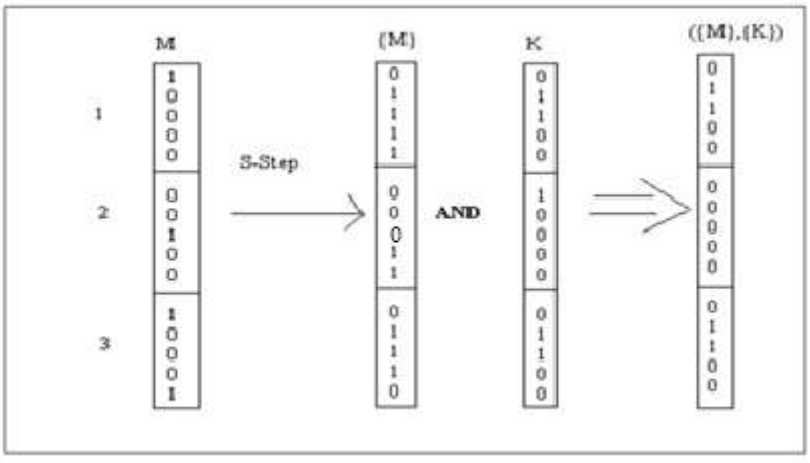

Fig. 4. Vertical bitmap representation for S-step

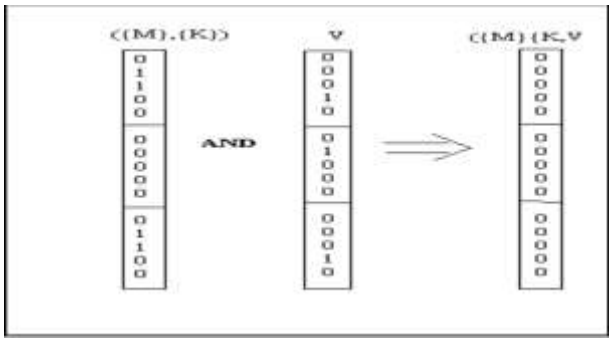

Fig 5. Vertical bitmap representation for I-step

To improve the performance of an algorithm pruning techniques are used with S-extension and I-extension of a node. S-step pruning technique prunes S-step children. For pruning it applies Apriori principle i.e. Let for the sequence({M}{M}),({M},{K}) and ({K}{V}) are given and if ({M}{K} is not frequent then ({M}{M}{K}),({M}{V}{K}) or ({M},{M,K}) or ({M},{V,K} are ignored. Similarly I-Step pruning technique prunes I-step children. For pruning at I-Step it applies similarly Apriori principle for item set i.e. Let for the item set sequences ({M,K}) and ({M,V}) if ({M,V}) is not frequent then ({M,K,V}) is also not frequent.

3.2 Bottleneck of Vertical bitmap Representation

To improve the performance of an algorithm pruning techniques are used i.e. we use generate candidate and test approach which can generate large amount of infrequent pattern .Here the main issue is is we have to design effective candidate pruning method for vertical mining algorithm to improve mining performance and this can be done if we design an effective candidate pruning mechanism which is effective at pruning candidate and have small runtime and memory cost and it is applicable to any vertical mining algorithm. To solve this issue we will study based on item co occurrence.

IV CM (Co occurrence Map) SPAM Algorithm

In CM SPAM First, to store item co-occurrence information, we introduce a new data structure named *Co-occurrence MAP* (CMAP). CMAP is a small and compact structure, which can be built with a single database scan. Second, we propose a generic candidate pruning mechanism for vertical sequential pattern mining algorithms based on the CMAP data structure. To understand how CM_SPAM algorithm work let take one example of sequence database.

SID	Sequence
1	<{a, b}, {c}, {f, g}, {g}, {e}>
2	<{a, d}, {c}, {b}, {a, b, e, f}>
3	<{a}, {b} ,{f}, {e} >
4	<{b} , {f ,g} >

Table 2 Sequence database

Each single letter represents the item and items within curly bracket represents item sets. From this sequence database we can find some sequential patterns.

ID	Pattern	Support
P1	<{a}, {f}>	3
P2	<{a} ,{c} ,{f}>	2
P3	< {b} , {f, g} >	2
P4	< {g} ,{e} >	2
P5	<{c}, {f} >	2
P6.....	<{b} >	4

Fig 5 Some sequential pattern found from table 2

Now we will represent the vertical data structure for sequence database of table 2.

a	
SID	**Item set**

1	1
2	1,4
3	1
4	

b

SID	Item set
1	1
2	3,4
3	2
4	1

c

SID	Item set
1	2
2	2
3	
4	

d

SID	Item set
1	
2	1
3	
4	

Fig 6 vertical representation of sequence database from table 2

e

SID	Item set
1	5
2	4
3	4
4	

F

SID	Item set
1	3
2	4
3	3
4	2

g

SID	Item set
1	3,4
2	
3	
4	2

The candidate generation process is same as that is used in SPAM algorithm.

4.1 Co-occurrence pruning

In this section, new approach is introduced , consisting of a data structure for storing co-occurrence information, and its properties for candidate pruning for vertical sequential pattern mining.

4.1.1 The co –occurrence Map

A *Co-occurrence MAP* (CMAP) is a structure mapping each item to a set of items succeeding it. We define two CMAPs named *CMAPi* and *CMAPs*. *CMAPi* maps each item to set of all items *succeeding sequence* by i-extension in no less than *minsup* sequences of *SDB*. *CMAPs* maps each item *to* the set of all items succeeding by s -extension in no less than *minsup* sequences of *SDB*. **Example.** The CMAP structures built for the sequence database of Table 2(left) are shown in Table 3, being *CMAPi* on the left part and *CMAPs* on the right part. Both tables have been created considering a *minsup* of two sequences. For instance, for the item *a*, we can see that it is associated with an item, *cmi*(*a*) = *{b}*, in *CMAPi*, because item a is associated with b in SID 1=<{ **a ,b**},{c}, {f, g}, {g}, {e}> and in SID 2= <{a, d}, {c}, {b}, {**a, b**, e, f}> so we will consider item b for i-extension of a. same now for *cmi*(*b*) is contain a null value because item b is associated with e in SID 2= <{a, d}, {c}, {b}, {a, b, e, f}> but its min_support will be less than 2 so we will not consider that item for i – extension.now for s-extension , *cms*(*a*) ={b, c, e, f} because from table 2 ,SID 2= <{**a**, d}, {c}, **{b},** {a, b, e, f}> and SID 3= **<{a}, {b}** ,{f}, {e} > so we will take b and SID 1=<{ **a** ,b},**{c},** {f, g}, {g}, {e}> and in SID 2= <{**a**, d}, {**c**}, {b}, {a, b, e, f}> so c al so consider and d is present only in SID 2 so it is not consider for S-extension of a and for e SID 1=<{ **a ,**b},{c}, {f, g}, {g}, **{e}**> and SID 3=<**{a}** , {b} ,{f} ,**{e}** > so e will be consider and so on…. So for cms (*a*) ={b, c, e, f} is taken.

CMAPi

Item	Is succeed by(I-extension)
A	{b}
B	Φ
C	Φ
E	Φ
F	{g}
G	Φ

CMAPs

Item	Is succeed by(S-extension)
A	{b, c, e, f}
B	{e, f, g}
C	{e, f}
E	Φ
F	{e, g}
G	Φ

Table 3 CMAPi and CMAPs for database of table 2 and Min_supp =2

4.1.2Size optimization

Let $n = |I|$ be the number of items in *SDB*. To implement a CMAP, a simple solution is to use an $n \times n$ matrix (two-dimensional array) M where each row (column) correspond to a distinct item and such that each entry $mj,k \in M$ represents the number of sequences where the item k succeed to the item i by i-extension or s-extension. The size of a CMAP would then be $O(n2)$. However, the size of CMAP can be reduced using the following strategy. It can be observed that each item is succeeded by only a small subset of all items for most datasets. Thus, few items succeed another one by extension, and thus, a CMAP may potentially waste large amount of memory for empty entries if we consider them by means of a $n \times n$ matrix. For this reason, in our implementations we instead implemented each CMAP as a hash table of hash sets, where an hash set corresponding to an item k only contains the items that succeed to k in at least *minsup* sequences.

5 Conclusions

Sequential pattern mining algorithms using the vertical format are very efficient because they can calculate the support of candidate patterns by avoiding costly database scans. However, the main performance bottleneck of vertical mining algorithms is

that they usually spend lot of time evaluating candidates that do not appear in the input database or are infrequent. To address this problem, we presented a novel data structure named CMAP for storing co-occurrence information. We have explained how CMAPs can be used for pruning candidates generated by vertical mining algorithms.

6 Future Work

In CMAP structure we will use n * n matrix or hash table which can cause the large number of null value .Thus CMAP may potentially waste large amount of memory for empty entries if we consider them by means of n*n matrix.. So to improve the overall performance of CM-SPAM algorithm we will generate the new structure that will eliminate the empty space so memory will be utilized and efficient searching is done using another structure.

7 References

1. Thabet Slimani, and Amor Lazzez," SEQUENTIAL MINING: PATTERNS AND ALGORITHMS ANALYSIS ", International Journal of Computer & Electronics Research,Vol 2,Issues 5,2013.
2. Jawahar. S," A COMPARATIVE STUDY OF SEQUENTIALPATTERN MINING ALGORITHMS", International Journal of Application or Innovation in Engineering & Management, Volume 4, Issue 12, 2015.
3 .Vishal S. Motegaonkar, Prof. Madhav V. Vaidya," A Survey on Sequential Pattern Mining Algorithms", International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014.
4 .Dr.S.Vijayarani and Ms.S.Deepa," AN EFFICIENT ALGORITHM FOR SEQUENCE GENERATION IN DATA MINING", International Journal on Cybernetics & Informatics (IJCI) Vol.3, No.1, 2014
5. Rashmi Mane," A Comparative Study of Spam and PrefixSpan Sequential Pattern Mining Algorithm for Protein Sequences", 2013.
6. Jiawei Han, Qiming Chen, Meichun Hsu," Free Span: Frequent pattern-projected sequential pattern mining", 2000.
7. Jay Ayres, Johannes Gehrke, Tomi Yiu, and Jason Flannick ," Sequential PAttern Mining using A Bitmap Representation",2002
8.Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen,Umeshwar Dayal," Mining Sequential Patterns by Pattern-Growth:The PrefixSpan Approach", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 10, OCTOBER 2004.
9. MOHAMMED J. ZAKI," SPADE: An Efficient Algorithm for Mining Frequent Sequences", 2001.
10. Zhenglu Yang," Effective Sequential Pattern Mining Algorithms ny Last Position Induction", 2005.
11. V. Purushothama Raju1 and G.P. Saradhi Varma," MINING CLOSED SEQUENTIAL PATTERNS IN LARGE SEQUENCE DATABASES", International Journal of Database Management Systems (IJDMS) Vol.7, February 2015
12 Philippe Fournier-Viger, Antonio Gomariz ,Manuel Campos, and Rincy Thomas," Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information", Springer International Publishing Switzerland ,2014