Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.8 No.S3, 2019, pp.50-53
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajcst-2019.8.S3.2088

An Efficient Closed Maximal Pattern Sequences Mining on High Dimensional Datasets

J. Krishna[1] and M. Haritha[2]
[1]Assistant Professor, [2]UG Student, [1&2]Department of Computer Science and Engineering,
Annamacharya Institute of Technology and Sciences, Andhra Pradesh, India
E-Mail: krishna.j.jk@gmail.com,harithareddym1998@gmail.com

(Received 2 April 2019; Revised 13 April 2019; Accepted 22 April 2019; Available online 29 April 2019)

***Abstract* - Previous methods have presented convincing arguments that mining complete set of patterns is huge for effective usage. A compact but high quality set of patterns, such as closed patterns and maximal patterns is needed. Most of the previously maximal pattern sequences mining algorithms on high dimensional sequence, such as biological data set, work under the same support. In this paper, an efficient algorithm Closed Maximal Pattern Sequences (CMPS-Mine) for mining closed maximal patterns based on multi-support is suggested. Careful exhibitions once Beta-globin gene sequences have exhibited that CMPS-Mine expends less memory utilization and runtime over Prefix Span. It generates compacted outcomes and two kinds of interesting patterns.**

***Keywords:* Multi Support, Sequential Pattern Mining, Maximal Pattern, High Dimensional Sequence**

I. INTRODUCTION

Sequential pattern mining discovers frequent subsequences as patterns in a sequence database. It is an important problem with broad applications, including the analysis of customer purchase behavior, web access patterns, DNA sequences, protein formation of a journal article in [1] and so on. Biological sequence pattern mining is a key technique in data mining, such as DNA sequence analysis and protein sequence analysis in Bioinformatics of a book in a series in [2][3][7]. Previous sequential pattern mining methods on high dimensional sequence, such as biological data set, are carried out from two aspects, one is in single sequence, and the other is in multiple sequences with same type. The problem is the method only using one support, it can’t find the patterns that occur frequently in each specific sequence, or patterns with enough total occurrence frequency in all sequences of a conference paper in [4]. Previous methods mining complete set of patterns, which is huge for effective usage. We need a compact but high quality set of patterns, such as closed patterns and maximal patterns of a book in [5].

In this paper, we propose a novel algorithm to mining maximal sequential patterns based on multi-support. There are two kinds of supports: support and local support a conference paper in [4]. Therefore, two kinds of patterns are mined. The first one is sequential pattern, which is a subsequence whose occurrence frequency in the set of sequences is no less than minimum support (min_sup). It corresponds to the support. The second one is local sequential pattern, which is a subsequence whose occurrence frequency in one specific sequence is no less than local minimum support (local_min_sup). It corresponds to the local support. The rest of this article is organized as follows. Section II reviews Prefix Span algorithm, and an example of mining complete biological sequential patterns is provided. In section III, some concepts are defined, and an improvement of Prefix Span algorithm: CMPS-Mine (Closed Maximal and Multi-support-based Pattern Sequences) is proposed. Section IV shows the results of sequential pattern mining and some interesting patterns. Finally, the conclusion is provided in Section V.

II. EXISTING PREFIX SPAN ALGORITHM

The key advantage of Prefix Span, an algorithm that examines the prefix subsequences and projects only their corresponding suffix subsequences into projected databases, is that it does not generate any candidates and only counts the frequency of local items. It utilizes a divide-and-conquer framework by creating subsets of sequential patterns that can be further divided when necessary. For example, suppose the biological database *S* is given in Table I and min_sup=75% (0.75), so the subsequences occurrence frequency in the set of sequences is no less than 3 (4*0.75). The set of items in the database is {P, Q, R, S}, and the sequence_id are {0, 1, 2, 3}. There are 7 transactions in sequence 0. Since all the sequences contain subsequence *x*=PSP, *x* is a sequential pattern of length-3 pattern, and its *support*(*x*) =4(100%).

TABLE I DNA SEQUENCES

Sequence id	Sequence
0	PSPPSPA
1	SPSQSQPRQQSPSP
2	SRQPSPPQSPS
3	SQQPSPRQQ

When min_sup is 0.75, Prefixes and the corresponding projected databases and patterns of database S are shown in table II. It is clear that, there are 10 patterns, 4 length-1 patterns, 4 length-2 patterns and 2 length-3 patterns. From the complete patterns, we can see that the patterns: P, PS, PSP in line 1 can be compressed as one pattern PSP. The reason is that PSP is super pattern of P and PS.

III. CLOSED MAXIMAL PATTERN SEQUENCES MINING BASED ON MULTI SUPPORT

Traditional algorithms Prefix Span, for the sequential pattern mining may generate lots of redundant patterns when dealing with the biological sequence. The Maximal Sequential Pattern is preferable to compress the function and structure of the biological sequence. At first, we give some definitions.

1. *Definition 1:* (Sequential Pattern) Sequential pattern is a subsequence whose occurrence frequency in the set of sequences is no less than minimum support (min_sup).
2. *Definition 2:* (Maximal Sequential Pattern) A pattern X is a maximal sequential pattern if there exists no super pattern Y such that X⊂Y and *Y* is sequential pattern.
3. *Definition 3:* (Local Sequential Pattern) Local sequential pattern is a subsequence whose occurrence frequency in one specific sequence is no less than local minimum support (local_min_sup).
4. *Definition 4:* (Support) The support of a subsequence X in a dataset *S* is the number of tuples in the dataset containing X, denoted as
   support(X)=|{<sequence_id,s>|(<sequence_id,s>**ϵ**S)∧ (X⊆S)}|.
5. *Definition 5:* (Local Support) the local support of a subsequence *X* in a dataset *S* is the number of tuples in a specific sequence Y containing X, denoted as
   local_support(X,Y)=|{<transaction_id,Y>|(Y**ϵ**S)∧(X⊆ Y)}|

TABLE II MAXIMAL SEQUENTIAL PATTERNS AND SUPPORTS

Prefix	Pattern	Maximal Pattern	Support
P	P, PS, PSP	PSP	4
Q	Q, QP	QP	3
R	R, RQ	RQ	3
S	S, SP, SPS	SPS	3

For example, given the DNA database S and min_sup in example 1. We get 4 maximal sequential patterns: PSP, QP, RQ, and SPS as shown in table 2. It is much less than the number of complete sequential patterns (that is 10). The support (PSP)=4 means that the PSP appears in 4 sequences. Specifically, it appears 2 times in sequence 0 and the transaction_id are 0 and 3(denoted as <sequence_id, transaction_id>, <0, {0,3}>); 1 time in sequence 1, <1, {11}>; 1 time in sequence 2, <2, {3}> and 1 time in sequence 3, <3,{3}>.

TABLE III LOCAL SEQUENTIAL PATTERNS

Sequence id	#local pattern	< local pattern, transaction id>
0	2	<PSP, {0,3}>, <SPS, {4}>
1	4	<PSP, {11}>, <QP, {5}>, <RQ, {7}>, <SPS, {0,10}>
2	4	<PSP, {3}>, <QP, {2}>, <RQ, {1}>, <SPS, {8}>
3	3	<PSP, {3}>, <QP, {2}>, <RQ, {6}>

The local patterns and the transaction locations of patterns are shown in table III. As shown in line 1, there are 2 local patterns in sequence 0. The patterns are PSP and SPS, while the local_support(PSP, 0)=2 and local_support(SPS, 0)=1. We can also get that local_support(SPS, 1)=2. Supposed local_min_sup is 2, we get two local sequential patterns: <PSP, 0> and <SPS, 1>. In this section, we propose a novel algorithm called CMPS-Mine, which is used to mine closed maximal sequential pattern based on multi-support. The algorithm of CMPS-Mine is presented as follow:

A. *Algorithm* (CMPS-Mine) Prefix projected maximal sequential pattern mining based on multi-support.
B. *Input:* A sequence database *S*, and the minimum support threshold min_sup, local_min_sup.
C. *Output:* The set of maximal sequential patterns and local sequential patterns
D. *Method:* Call *CMPS*-Mine (*α*, *l*, *S*|*α*).
E. The parameters are (1) *α* is a sequential pattern; (2) *l* is the length of *α*; and (3) *S*|*α* is the *α*-projected database if *α* ≠ <>; otherwise, it is the sequence database *S*, and *l*=0.
F. *Steps:*

```
1. Scan S|α once, find each frequent item, b.
2. For each frequent item b,
   a) If support (b) min_sup, append b to α to from a
      prefix α'. Goto step 3 and step 4.
   b)If it does not generate new prefix, then output
      maximal pattern α and local pattern<α, s>.
3. For each α' and each special sequence s (s S|α), if
   (α',s) local_min_sup, then records the tuples <α',s> (s
   S|α).
4. For each α', construct α' projected database S|α', and
   call CMPS-Mine (α', l+1, S|α').
```

IV. PERFORMANCE EVALUATION

In this section, we provide experimental results to compare the performance of Prefix Span with CMPS-Mine. In our performance study, we select 11 DNA sequences of beta-globin gene from (National Center for Biotechnology Information) NCBI. The sequence lengths of datasets are shown in table IV, and the average length is 1736.

TABLE IV DNA SEQUENCES

Id	Sequence	Length of sequence
0	Mouse	1926
1	Home sapiens	2128
2	Xenopus laevis	1989
3	Gallus gallus	2157
4	Peromuscus maniculatus	1225
5	Rat	1616
6	Pan troglodytes	1012
7	Danio frankei	1046
8	Bovine adult	2072
9	Capra hircus	1877
10	Ovis aries	2040

Fig. 1 shows the processing time of the two algorithms at different support thresholds. The min_sup are from 0.5 to 1. It is clear that the runtime of CMPS-Mine is lower than Prefix Span. When min_sup is 1, CMPS-Mine is about two times faster than Prefix Span. The memory usage of the two algorithms at different support thresholds is shown in fig.2.

We can conclude that the distance between the two algorithms is growing with the decreasing of support. Fig.3 shows the number of sequence patterns of the two algorithms at different support thresholds. It is clear that mining maximal sequential pattern compress the result of complete sequential patterns. The number of maximal sequential patterns is almost half of complete patterns.

From Fig.1 to 3, it is clear that the performance of CMPS-Mine, which is used to mine maximal sequential patterns, is better than Prefix Span, which mines the complete sequential patterns. Compared with the old one, the novel algorithm reduces the runtime and memory usage. While, it generates a more compress result than the old one.

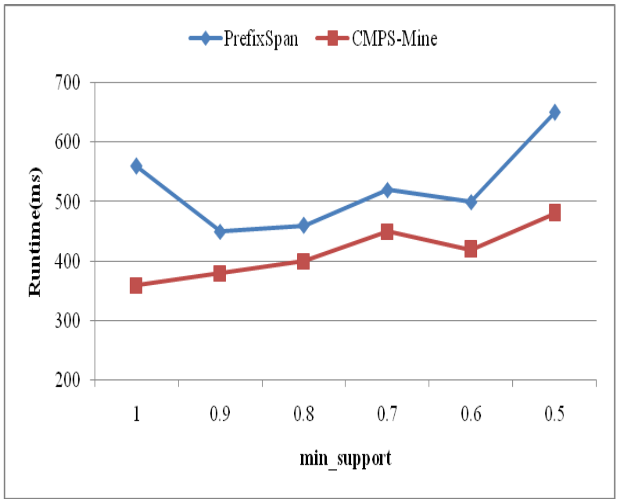

Fig. 1 Runtime of two algorithms on DNA sequences

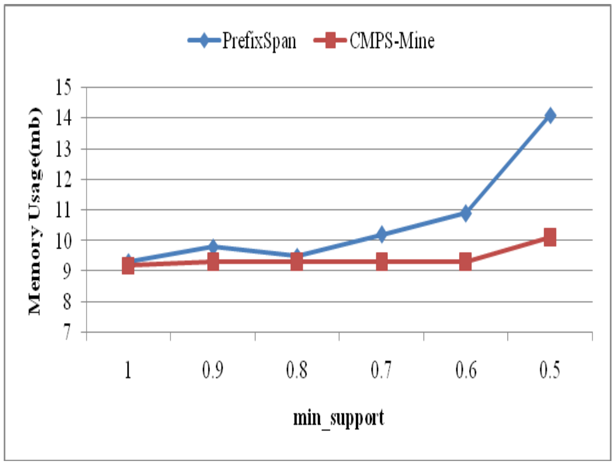

Fig. 2 Memory usage of two algorithms on DNA sequences

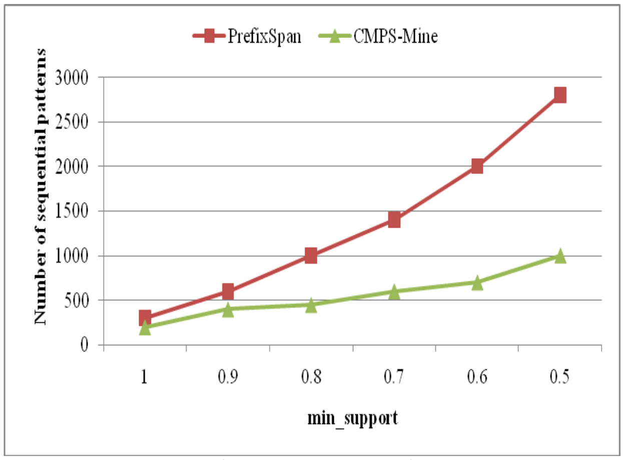

Fig. 3 Number of sequential patterns of two algorithms

V. CONCLUSION

This article presents a novel algorithm, called CMPS-Mine (Closed Maximal and Multi-support-based Pattern Sequences). It is an improvement of Prefix Span algorithm. The novel algorithm is used to mine maximal sequential patterns based on multi-support. Instead of mining complete sequential patterns, CMPS-Mine algorithm compresses the results through mining maximal sequential patterns.

There are two kinds of subsequence supports: support and local_support. The support (X) (X is a subsequence) is the number of tuples in the dataset containing *X*. If support(X) min_sup, then X is a sequential pattern. The local_support(X, Y) (Y is a sequence) is the number of tuples in sequence Y containing X. If local_support(X, Y), local_min_sup, then X is a local sequential pattern. CMPS-Mine algorithm generates sequential patterns and local sequential patterns. Besides, locations and times of local sequential patterns appeared in one sequence are provided by novel algorithm. The locations and times of one pattern appeared in some sequences are provided too.

In our experiments, it is clear that the performance of CMPS-Mine is better than Prefix Span. And the number of maximal sequential patterns is about half of the number of complete patterns.

There are many interesting issues that need to be studied, such as mining close DNA sequential patterns of a journal article in [8], mining sequential patterns in very long genome and protein sequence, and mining DNA or protein sequence pattern with constraints of journal articles in [9][10][11], and so on.

REFERENCES

[1] N.R. Mabroukeh, and C.I. Ezeife, “A Taxonomy of Sequential Pattern Mining Algorithms”, *Journal ACM Computing Surveys*, Vol. 43, No. 1, pp.1-41, 2010.

[2] J. Cohen, “Bioinformatics-an Introduction for computer scientists”, *ACM Computing Surveys (CSUR)*, Vol. 36, No. 2, pp.122-158, 2004.

[3] Z. Ezziane, “Applications of artificial intelligence in bioinformatics”, *A review Expert Systems with Applications*, Vol. 30, pp.2-10, 2006.

[4] Y. Xiong, and Y.Y. Zhu, "BioPM: an efficient algorithm for protein motif mining", *in Proceedings of the 1st International conference on Bioinformatics and Biomedical Engineering*, pp.394-397, 2007.

[5] J.W. Han, H. Cheng, D. Xin, and X.F. Yan "Frequent pattern mining: current status and future directions", *Data Mining and Knowledge Discovery*, Vol. 15, pp.55-86, 2007.

[6] J.W. Pei, and J.Y. Wang, *et al*. "Mining sequential patterns by pattern-growth: The prefixspan approach", *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, pp.1-17, 2004.

[7] R. Alves, D.S.R. Baena, and J.S.A. Ruiz, "Gene association analysis: a survey of frequent pattern mining from gene expression data", *Briefings in Bioinformatics*, pp.1-12, 2009.

[8] B. Lavanya, and A. Murugan, "A DNA based approach to find closed repetitive gapped subsequences from a sequence database", *International Journal of Computer Applications,* Vol.29, No.5, pp.45-49, 2011.

[9] P.G. Ferreira, and P.J. Azevedo, "Protein sequence pattern mining with constraints", *Knowledge Discovery in Databases*, Vol. 3721, pp.96-107, 2005.

[10] D. He, X.G. Zhu, X.D. Wu, "Mining approximate repeating patterns from sequence data with gap constraints", *Computational Intelligence*, Vol. 27, No. 3, pp.336-362, 2011.

[11] J. Krishna, P. Suryanarayana Babu, "DFP-MINER: Assessing the Accuracy of Correlated Sequence Patterns from High Dimensional Biological Datasets", *International Journal of Creative Research Thoughts*, Vol. 5, No. 4, pp. 1233-1241, November, 2017.