International Conference on Information Engineering, Management and Security
2015 [ICIEMS 2015]

ISBN	978-81-929742-7-9
Website	www.iciems.in
Received	10 - July - 2015
Article ID	ICIEMS019

VOL	01
eMail	iciems@asdf.res.in
Accepted	31- July - 2015
eAID	ICIEMS.2015.019

Top K Sequential Pattern Mining Algorithm

Karishma B Hathi[1], Jatin R Ambasana[2]
[1]Student, M.E.(CSE), Gardi Vidyapith, Gujarat, India
[2]Assistant Professor, CSE Department, Gardi Vidyapith, Gujarat, India

ABSTRACT: *Sequential pattern mining is a very chief mining technique with wide applications. Still, tune up the minsup parameter of sequential pattern mining algorithms to produce enough patterns is complex and time-consuming. To solve this problem, the assignment of top-k sequential pattern mining has been described, here k is the number of sequential patterns to be discovered, and is set by the user. In this paper, we present proposed approach for improving parameters in TKS Algorithm.*

***KEYWORDS*:** Sequential Patterns, Top K, Sequence Database, Pattern mining.

I. INTRODUCTION

The sequential pattern mining is a very important concept of data mining, a further extension to the concept of association rule mining [1]. That has a huge range of real-life application. This mining algorithm solves the problem of discovering the presence of frequent sequences in the given database [2]. Sequential Pattern Mining finds interesting sequential patterns among the huge database. It discovers frequent subsequences as patterns from a given sequence database. It is a well-understood data mining problem with broad applications such as the analysis of web clickstreams, program executions, medical data, biological data and e-learning data [1, 5]. Although many studies have been done on constructing sequential pattern mining algorithms [1, 2, 3, 4], the main problem is how the user should choose the *minsup* threshold to produce a desired amount of patterns. This problem is important because in practice, users have limited resources (time and storage space) for discovering the results and thus are often only interested in analyzing a certain amount of patterns, and fine-tuning the *minsup* parameter is very time-consuming. Depending on the choice of the *minsup* threshold, algorithms can become very slow and produce an extremely huge amount of results or generate none or too few results, getting valuable information. To address this difficulty, it was proposed to redefine the problem of mining sequential patterns as the problem of mining the top-*k* sequential patterns, where k is the number of sequential patterns to be discovered and is set by the user.

II. RELATED WORK

The problem of sequential pattern mining was proposed by Agrawal and Srikant [2] and is defined as follows. A *sequence database SDB* is a set of sequences $S = \{s1, s2\ldots, ss\}$ and a set of items $I = \{i1, i2, \ldots, im\}$ happening in these sequences. An *item* is a symbolic value. An *itemset* $I = \{i1, i2, \ldots, im\}$ is an unordered set of different items. For example, the itemset $\{a, b, c\}$ shows the sets of items a, b and c. A *sequence* is an ordered list of itemsets S=< $I1, I2, I3, \ldots, In,$ > such that $Ik \subseteq I$ for all 1≤k≤n .For example having a sequence the sequence database *SDB* depicted in Figure 1. It contains mainly four sequences having accordingly the *sequences ids* (SIDs) 1, 2, 3 and 4. In this example, each solo letter represents an item. Items between curly brackets describes an itemset. For in-stance, the first

This paper is prepared exclusively for **International Conference on Information Engineering, Management and Security** 2015 [ICIEMS] which is published by ASDF International, Registered in London, United Kingdom. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honoured. For all other uses, contact the owner/author(s). Copyright Holder can be reached at copy@asdf.international for distribution.
2015 © Reserved by ASDF.international

Cite this article as: Karishma B Hathi , Jatin R Ambasana. "Top K Sequential Pattern Mining Algorithm." *International Conference on Information Engineering, Management and Security (2015)*: 115-120. Print.

sequence $\langle\{a, b\}, \{c\}, \{f\}, \{g\}, \{e\}\rangle$ shows that items *a* and *b* happened at the same time, were followed successively by *c, f, g* and lastly *e*. A sequence $sa=< A1, A2, A3, \ldots, An>$ is called to be included in another sequence $sb=< B1, B2, B3, \ldots, Bm>$ if and only if there exists integers $1\leq I1< I2 <\ldots\ldots< In\leq Im$ such that $A1\subseteq Bi1$, $A2\subseteq Bi2$, $A3\subseteq Bi3, \ldots, An\subseteq Bin$. The support of a subsequence sa in a sequence database SDB is described as the number of sequences s ∈ S such that $sa \sqsubseteq$ s and is denoted by sup(sa). The problem of mining sequential patterns in a sequence database SDB is to locate all frequent sequential patterns, i.e. each subsequence sa such that sup(sa)≥minsup for a threshold minsup set by the user. For example, Figure 2 displays five of the 29 sequential patterns found in the database of table Figure 1 for minsup = 2. Many algorithms have been proposed for the problem sequential pattern mining such as PrefixSpan [3], SPAM [4], GSP and SPADE [6].

SID	Sequences
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}⟩

Figure 1: Sequence Database

ID	Pattern	Support
P1	⟨{a},{f}⟩	3
p2	⟨{a},{c}{f}⟩	2
p3	⟨{b},{f,g}⟩	2
p4	⟨{g},{e}⟩	1
p5	⟨{b}⟩	4

Figure 2: Some Sequential Patterns

To address the problem of setting minsup, the problem of sequential pattern mining was reconsidered as the problem of top-k sequential pattern mining [7]. The current state-of-the-art algorithm for top-k sequential pattern mining is TSP [7]. There are two versions of TSP have been proposed for correspondingly mining (1) top-k sequential patterns and (2) top-k closed sequential patterns. Here we are addressing the first case. Extending algorithm to the second case will be considered in future work. The TSP algorithm is based on PrefixSpan [3]. TSP first generates frequent sequential patterns holding a single item. Then it recursively extends each pattern s by (1) it projecting the database by s, (2) it scanning the resulting projected database to identify items that appear more than minsup times after s, and (3) it append these items to s. The main benefit of this projection-based approach is that it only considers patterns appearing in the database unlike "generate-and-test" algorithms [2, 7]. However, the drawback of this approach is that projecting/scanning databases repeatedly is costly, and that cost becomes huge for dense databases where multiples projections have to be performed. Given this limitation, a chief research challenge is to define an algorithm that would be more efficient than TSP and that would perform well on dense datasets.

III. THE BASIC TKS ALGORITHM

TKS, an algorithm to find the top-k sequential patterns having the highest support, where k is set by the user. TKS employs the vertical database representation and basic candidate-generation procedure of SPAM [8]. Furthermore, it also includes various efficient strategies to find top-k sequential pattern efficient Fine-tuning the minsup parameter of sequential pattern mining algorithms to generate enough patterns is hard and time-consuming. To address this problem, the task of top-k sequential pattern mining has been defined, where k is the number of sequential patterns to be found, and is set by the user. So here an efficient algorithm for this problem named TKS (Top-K Sequential pattern mining) is present. TKS utilizes a vertical bitmap database representation, a new data structure named PMAP (Precedence Map) and various efficient strategies to prune the search space. The experimental study on real datasets shows that TKS outperforms TSP, the current state-of-the-art algorithm for top-k sequential pattern mining by more than an order of magnitude in execution time and memory.

TKS Algorithm [9]

It takes as parameters a sequence database SDB and k.

1) It first scans SDB once to construct V(SDB).

2) Let $Sinit$be the list of items in V(SDB)
3) Then, for each item s∈$Sinit$, if s is frequent according to bv(s) it calls the procedure "SAVE".
4) R=R∪{s, $Sinit$,items from $Sinit$ that are lexically larger than s}
5) WHILE ∃<r, S1,S2>∈ R AND sup(r)≥ minsup DO
6) Select the tuple <r,S1,S2> having the pattern r with the highest support in R
7) Then calls "SEARCH" find tuple.
8) Finally calls "REMOVE" and delete infrequent patterns from database.

IV. THE PROPOSED ALGORITHM

Limitation of basic TKS algorithm is number of database scan are higher so execution time grows higher due to this limitation. For improving the efficiency of TKS algorithm and overcome the drawbacks of TKS we propose an efficient approach for mining top k sequential patterns. We can improve the efficiency of TKS algorithm by using tree structure in TKS. By using this we can improve the efficiency of TKS algorithm in the terms of execution time.

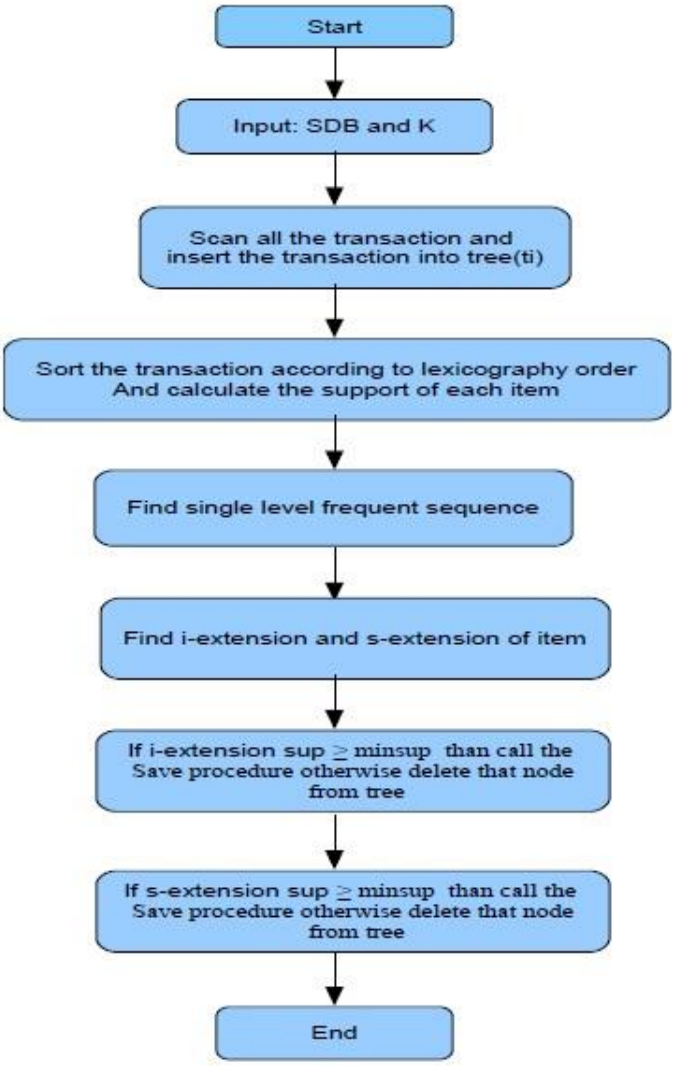

Figure 3: Propose Flow

Proposed Algorithm
Input: SDB, K
Output: Top K Sequential Patterns
Let $qtemp$ be the list of items in tree
For each q∈$qtemp$
Save(q,L,k,minsup)

Cite this article as: Karishma B Hathi , Jatin R Ambasana. "Top K Sequential Pattern Mining Algorithm." *International Conference on Information Engineering, Management and Security (2015)*: 115-120. Print.

☐ If i-extension sup(q)≥$minsup$
☐ *Save(q, all the items in* $qtemp$ that are lexically larger than q,L,k)
☐ And
☐ If s-extension sup(q)≥$minsup$
☐ *Save(q, all the items in* $qtemp$ that are lexically larger than q,L,k)
☐ End if
☐ Remove q∈$qtemp$ *when* sup(q)<$minsup$

End for
Return L

V. TOOL STUDY

Java Technology:

JAVA is an object oriented platform independent and middle level language. It contains JVM (Java Virtual Machine) which is able to execute any program more efficiently. The feature of Platform Independence makes it different from the other Technologies available today.

Eclipse Tool:

Eclipse is an integrated development environment (IDE).It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and thus can be used to develop applications. A vendor-neutral open-source workbench for multi-language development. An extensible platform for tool integration. Plug-in based framework to create, integrate and utilize software tools.

Sequential Pattern Mining Framework:

SPMF is an open-source data mining library written in Java, specialized in pattern mining. The source code of each algorithm can be integrated in other Java software. Moreover, SPMF can be used as a standalone program with a simple user interface or from the command line. The current version is **v0.96r16** and was released the **28th April 2015.**

VI. EXPERIMENTAL RESULTS

Various parameters are used for sequential pattern mining. Here this proposed approach conclude parameter execution time. The execution time is depend on the number of patterns are found and the number of passes that requires for database scan. We compared the performance of Proposed Algorithm with TKS Algorithm. All algorithms were implemented in Java. Experiments were carried on three real-life datasets having varied characteristics and representing three different types of data (web click stream, text from books, sign language utterances). Those datasets are accordingly FIFA, Bible and Sign. The comparison between TKS and Proposed algorithm is shown in following figures.

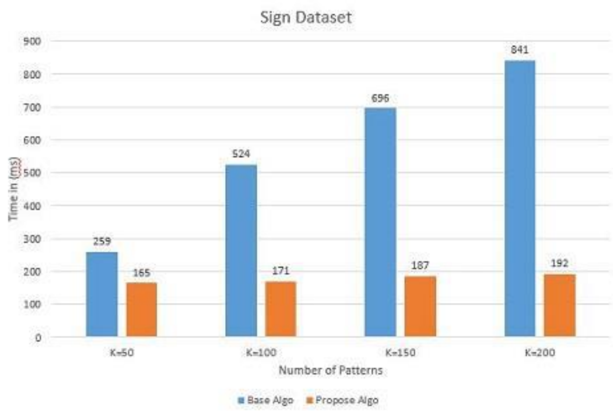

Figure 4: Sign Dataset

Cite this article as: Karishma B Hathi , Jatin R Ambasana. "Top K Sequential Pattern Mining Algorithm." *International Conference on Information Engineering, Management and Security (2015)*: 115-120. Print.

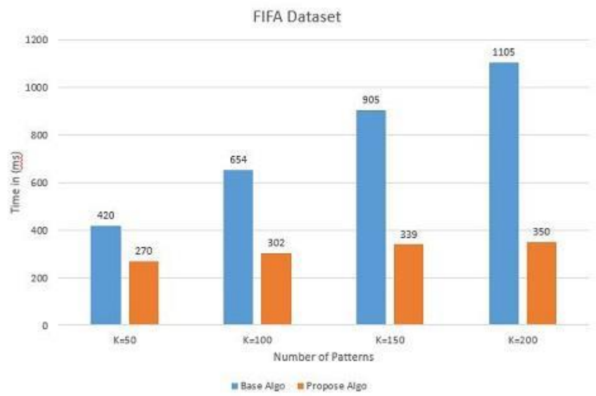

Figure 5: FIFA Dataset

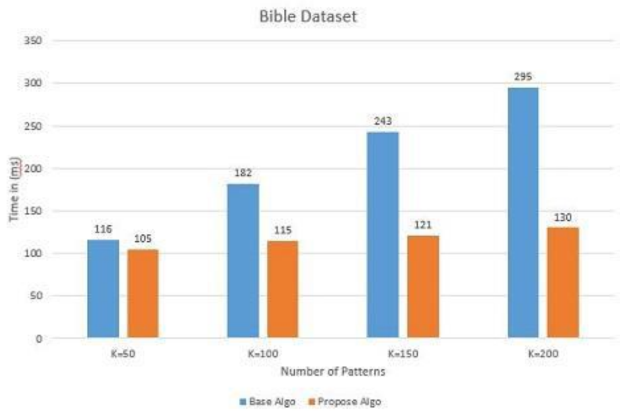

Figure 6: Bible Dataset

VII. CONCLUSION AND FUTURE WORK

The Proposed System has improves the performance of TKS algorithm by using tree Structure. From the result analysis it is clearly seen that the execution time of proposed algorithm reduced. There is less number of database scan requiring so the efficiency of the TKS algorithm improves. Thus, we conclude that the proposed system has better performance than TKS algorithm. Extending TKS algorithm for finding Top K Closed Sequential Pattern can be considered as future work.

ACKNOWLEDGEMENT

We are deeply indebted & would like to express gratitude to our thesis guide Prof. Jatin Ambasana , B. H. Gardi College of Engineering & Technology for his great efforts and instructive comments in the dissertation work.
We would also like to extend our gratitude to Prof.Hemal Rajyaguru, Head of the Computer Science & Engineering Department, B. H. Gardi College of Engineering & Technology for his continuous encouragement and motivation.
We would also like to extend our gratitude to Prof. Vaseem Ghada, PG Coordinator, B. H. Gardi College of Engineering & Technology for his continuous support and cooperation.
We should express our thanks to our dear friends & our classmates for their help in this research; for their company during the research, for their help in developing the simulation environment.
We would like to express our special thanks to our family for their endless love and support throughout our life. Without them, life would not be that easy and beautiful.

REFERENCES

1] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufman publishers, 2001.
[2] R. Agrawal and R. Srikant, "Mining Sequential Patterns", In Proceedings of the 11th International Conference on Data Engineering, pp. 3-14, Taipei, Taiwan, 1995.
[3] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Mining Se-quential Patterns by Pattern-Growth: The PrefixSpan Approach, IEEE Trans. Knowledge and Data Engineering, vol. 16, no. 10, pp. 1-17 (2001)

Cite this article as: Karishma B Hathi , Jatin R Ambasana. "Top K Sequential Pattern Mining Algorithm." *International Conference on Information Engineering, Management and Security (2015)*: 115-120. Print.

[4] Ayres, J., Flannick, J., Gehrke, J. and Yiu, T.: Sequential PAttern mining using a bitmap representation, Proc. 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2002), July 23-26, 2002, Edmonton, Alberta, pp. 429-435 (2002)
[5] Mabroukeh, N. R. and Ezeife, C. I.: A taxonomy of sequential pattern mining algorithms, ACM Computing Surveys, vol. 43, no. 1, pp. 1-41 (2010)
[6] Cormen, T. H., Leiserson, C. E., Rivest, R. and Stein, C.: Introduction to Algorithms, 3rd ed., Cambridge:MIT Press (2009)
[7] Tzvetkov, P. Yan, X. and Han, J.: TSP: Mining Top-k Closed Sequential Patterns, Knowledge and Information Systems, vol. 7, no. 4, pp. 438-457 (2005)
[8] Jian Pei, "Mining Sequential Patterns by Pattern-Growth:The PrefixSpan Approach", IEEE Transactions on Knowledge and Data Engineering, VOL. 16, NO. 10, OCTOBER 2004.
[9] Philippe Fournier-Viger,"TKS: Efficient Mining of Top-K Sequential Patterns", Springer Advanced Data Mining and Application, vol. 8346, pp. 109-120, 2013.
[10] Vishal S. Motegaonkar, Prof. Madhav V. Vaidya,"A survey on sequential pattern mining algorithms",International Journal of Computer Science and Information Technologies (IJCSIT),Vol.5 (2),2014.
[11] C.-C. Yu and Y.-L. Chen, "Mining Sequential Patterns from Multi-Dimensional Sequence Data", IEEE Trans. Knowledge and Data Eng., Vol. 17, No. 1, pp. 136-140, Jan. 2005.
[12] Karishma B Hathi, Jalpa A Varsur, Sonali P Desai, Sagar R Manvar," A Performance Analysis of Sequential Pattern Mining Algorithms', ournal of Emerging Technologies and Innovative Research (JETIR), February 2015, Volume 2, Issue 2.
[13] Jian Pei, Jiawei Han and Wei Wang, "Constraint-based sequential pattern mining: the pattern-growth methods", Journal of Intelligent Information Systems,
Vol:28, No: 2 ,pp:133-160, 2007.
[14] Jian Pei, Jiawei Han and Wei Wang, "Constraint-based sequential pattern mining: the pattern-growth methods", Journal of Intelligent Information Systems, Vol:28, No: 2 ,pp:133-160, 2007.

Cite this article as: Karishma B Hathi , Jatin R Ambasana. "Top K Sequential Pattern Mining Algorithm." *International Conference on Information Engineering, Management and Security (2015)*: 115-120. Print.