IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received June 2, 2020, accepted June 16, 2020, date of publication June 24, 2020, date of current version July 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004528

An Efficient Method for Mining Top-*K* Closed Sequential Patterns

THI-THIET PHAM[1], TUNG DO[2], ANH NGUYEN[3], BAY VO[4], AND TZUNG-PEI HONG[5,6], (Senior Member, IEEE)

[1]Faculty of Information Technology , Industrial University of Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam
[2]Faculty of Basic Science, Van Lang University, Ho Chi Minh City 700000, Vietnam
[3]Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
[4]Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh City 700000, Vietnam
[5]Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[6]Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan

Corresponding author: Bay Vo (vd.bay@hutech.edu.vn)

This work was supported by the Industrial University of Ho Chi Minh City under Grant 20/1.6CNTT01.

ABSTRACT The problem of exploiting Closed Sequential Patterns (CSPs) is an essential task in data mining, with many different applications. It is used to resolve the situations of huge databases or low minimum support (*minsup*) thresholds in mining sequential patterns. However, it is challenging and needs a lot of time to customize the *minsup* values for generating appropriate numbers of CSPs desired by users. To conquer this issue, the TSP algorithm for mining top-*k* CSPs was previously proposed, with *k* being a given parameter. The algorithm would return the *k* CSPs which have the highest support values in a database. However, its execution time and memory usage were high. In this paper, an algorithm named TKCS (Top-*K* Closed Sequences) is proposed to mine the top-*k* CSPs efficiently. To improve the execution time and memory usage, it uses a vertical bitmap database to represent data. Besides, it adopts some useful strategies in the process of exploiting the top-*k* CSPs such as: always choosing the sequential patterns with the greatest support values for generating candidate patterns and storing top-*k* CSPs in an ascending order of the support values to increase the *minsup* value more quickly. The empirical results show that TKCS has better performance than TSP for discovering the top-*k* CSPs in terms of both runtime and memory usage.

INDEX TERMS Closed sequential pattern, data mining, sequential pattern, top-k sequential patterns.

I. INTRODUCTION

In the domain of data mining from a sequence database, exploiting sequential patterns is an essential task that has been extensively examined [1], [3], [4], [8]–[11], [14], [17], [23], [27], [35]. AprioriAll [1] was the first algorithm designed to solve the sequential pattern mining problem. It was proposed by Agrawal *et al.* in 1995 and is also the basis for later algorithms such as GSP [27], SPADE [35], SPAM [3], FREESPAN [12], PREFIXSPAN [23], PRISM [11], and MCM-SPADE [14]. These algorithms have shown good performance with respect to sequence databases that have short frequent sequences. However, when exploiting sequential patterns in a sequence database with lengthy frequent sequences or when employing very low *minsup* values to mine sequential patterns, the process will produce a huge amount of frequent sub-sequences, such that it needs a lot of time to execute and a lot of space to store. Therefore, the performance of these algorithms is often falling dramatically. However, the issue of exploiting CSPs has been suggested to resolve this problem, with several recent studies pursuing this idea [13], [15], [20], [25], [26], [28], [31], [32].

The algorithms for exploiting sequential patterns or CSPs from a sequence database mentioned above always require a minimum support threshold by the user. However, in practical applications, it is difficult for users to choose an appropriate *minsup* value to generate the expected number of sequential patterns. As such, the algorithm can produce too few meaningful patterns or too many meaningless patterns. To resolve this problem, researchers have recently proposed algorithms to find top-*k* sequential patterns with *k* being the highest number of sequential patterns set by the user. This solution is not only for mining the top-*k* sequential patterns [7], [18] and top-*k* CSPs [29], but is also effective in many other areas

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.