The Journal of Systems and Software 125 (2017) 170–182

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Discovering partial periodic-frequent patterns in a transactional database

R. Uday Kiran[a,b,*], J.N. Venkatesh[d], Masashi Toyoda[a], Masaru Kitsuregawa[a,c], P. Krishna Reddy[d]

[a] The University of Tokyo, Japan
[b] National Institute of Communication Technology, Japan
[c] National Institute of Informatics, Japan
[d] International Institute of Information Technology-Hyderabad, India

ARTICLE INFO

Article history:
Received 24 March 2016
Revised 27 September 2016
Accepted 23 November 2016
Available online 28 November 2016

Keywords:
Data mining
Knowledge discovery in databases
Pattern mining
Partial periodicity
Algorithms

ABSTRACT

Time and frequency are two important dimensions to determine the interestingness of a pattern in a database. Periodic-frequent patterns are an important class of regularities that exist in a database with respect to these two dimensions. Current studies on periodic-frequent pattern mining have focused on discovering full periodic-frequent patterns, i.e., finding all frequent patterns that have exhibited complete cyclic repetitions in a database. However, partial periodic-frequent patterns are more common due to the imperfect nature of real-world. This paper proposes a flexible and generic model to find partial periodic-frequent patterns. A new interesting measure, *periodic-ratio*, has been introduced to determine the periodic interestingness of a frequent pattern by taking into account its proportion of cyclic repetitions in a database. The proposed patterns do not satisfy the anti-monotonic property. A novel pruning technique has been introduced to reduce the search space effectively. A pattern-growth algorithm to find all partial periodic-frequent patterns has also been presented in this paper. Experimental results demonstrate that the proposed model can discover useful information, and the algorithm is efficient.

1. Introduction

Frequent patterns (or itemsets) are an important class of regularities that exist in a transactional database. These patterns play a key role in many knowledge discovery tasks such as association rule mining (Agrawal et al., 1993; Han et al., 2007), clustering (Zhang et al., 2010), classification (Dong and Li, 1999) and recommender systems (Adomavicius and Tuzhilin, 2001, 2005). The popular adoption and successful industrial application of finding these patterns has been hindered by a major obstacle: *frequent pattern mining often generates a huge number of patterns and majority of them may be found insignificant depending on application or user requirements.* When confronted with this problem in real-world applications, researchers have tried to reduce the desired set by finding user interest-based patterns such as maximal frequent patterns (Gouda and Zaki, 2001), demand driven patterns, utility patterns (Yao et al., 2004), constraint-based patterns (Pei et al., 2004), diverse-frequent patterns (Swamy et al., 2014), top-*k* patterns (Han et al., 2002) and periodic-frequent patterns (Tanbeer et al., 2009). This paper focuses on finding periodic-frequent patterns.

An important criterion to assess the interestingness of a frequent pattern is its temporal occurrences in a database. That is, whether a frequent pattern is occurring periodically, irregularly, or mostly at specific time intervals in a database. The class of frequent patterns that are occurring periodically within a database are known as periodic-frequent patterns. Finding these patterns is a significant task with many real-world applications. Examples include improving the performance of recommender systems (Stormer, 2007), intrusion detection in computer networks (Ma and Hellerstein, 2001) and discovering events in Twitter (Kiran et al., 2015). A classic application to illustrate the usefulness of these patterns is market-basket analysis. It analyzes how regularly the set of items are being purchased by the customers. An example of a periodic-frequent pattern is as follows:

$$\{Bat, Ball\} \quad [support = 5\%, \quad periodicity = 1\ hour].$$

[*] Corresponding author.
E-mail addresses: uday_rage@tkl.iis.u-tokyo.ac.jp, uday.rage@gmail.com (R.U. Kiran), jn.venkatesh@research.iiit.ac.in (J.N. Venkatesh), toyoda@tkl.iis.u-tokyo.ac.jp (M. Toyoda), kitsure@tkl.iis.u-tokyo.ac.jp (M. Kitsuregawa), pkreddy@iiit.ac.in (P.K. Reddy).
URL: http://researchweb.iiit.ac.in/~uday_rage/index.html (R.U. Kiran), http://www.tkl.iis.u-tokyo.ac.jp/~toyoda/index_e.html (M. Toyoda), http://www.tkl.iis.u-tokyo.ac.jp/Kilab/Members/memo/kitsure_e.html (M. Kitsuregawa), http://faculty.iiit.ac.in/~pkreddy/index.html (P.K. Reddy)

http://dx.doi.org/10.1016/j.jss.2016.11.035
0164-1212/© 2016 Published by Elsevier Inc.

The above pattern says that 5% of the customers have periodically purchased the items '*Bat*' and '*Ball*' at least once in every hour. This predictive behavior of the customers' purchases can facilitate the users in product recommendation and inventory management.

The problem of finding periodic-frequent patterns has been widely studied in the past (Tanbeer et al., 2009; Amphawan et al., 2009; Kiran and Reddy, 2010; Surana et al., 2011; Kiran and Kitsuregawa, 2014). However, most of these studies have focused on finding full periodic-frequent patterns, i.e., frequent patterns that have exhibited complete cyclic repetitions in a database. A useful related type of periodic-frequent patterns is partial periodic-frequent patterns, i.e., frequent patterns that have exhibited partial cyclic repetitions in a database. These patterns are a looser kind of full periodic-frequent patterns, and they exist ubiquitously in the real-world databases. The proposed patterns can find useful information in many real-life applications. Few examples are as follows:

- In the market-basket analysis, partial periodic-frequent patterns provide useful information pertaining to regularly purchased itemsets. This information can be useful for inventory management.
- Partial periodic-frequent pattern mining on a web-log data can find the sets of web pages that were not only visited heavily, but also regularly by the users. This information can be useful for the user for improved web site design or web administration.
- In an accident data set, partial periodic-frequent patterns can discover useful information pertaining to the periodicity of regularly occurring accidents. This information can be high useful for improving passenger safety.
- Partial periodic behavior of sets of hashtags has been exploited to discover minor events from Twitter data (Kiran et al., 2015).

The purpose of this paper is to discover partial periodic-frequent patterns efficiently.

Finding partial periodic-frequent patterns is a non-trivial and challenging task. The reasons are as follows:

- The problem of finding partial periodic patterns has been widely studied in time series data (Han et al., 1998; 1999; Yang et al., 2003; Aref et al., 2004). Unfortunately, these studies cannot be enhanced to find partial periodic-frequent patterns in a transactional database. The main reason is that these studies consider time series as a symbolic sequence, and therefore, do not take into account the actual temporal information of the items within a series.
- Existing full periodic-frequent pattern mining approaches assess the periodic interestingness of a frequent pattern by simply determining whether all of its *periods* (or inter-arrival times) are within the user-specified maximum *period* threshold value. As a result, there exists no interestingness measure to assess the partial periodic behavior of a frequent pattern in a transactional database.

With this motivation, this paper introduces a generic and flexible model to find partial periodic-frequent patterns. The proposed model is generic because it allows the user to find both full and partial periodically occurring frequent patterns. The model is flexible as it enables every pattern to satisfy a different minimum number of cyclic repetitions depending on its frequency. This flexible nature of our model facilitates us to capture the non-uniform frequencies of the items (or patterns) within a database effectively.

The contributions of this paper are as follows:

- A generic model of partial periodic-frequent patterns has been proposed in this paper. This model employs a new measure, *periodic-ratio*, to determine the partial periodic interestingness of a frequent pattern in a database. The proposed measure determines the interestingness of a pattern by taking into account its proportion of cyclic repetitions in a database.
- The patterns discovered by the proposed model do not satisfy the anti-monotonic property. That is, all non-empty subsets of a partial periodic-frequent pattern may not be partial periodic-frequent patterns. This increases the search space, which in turn increases the computational cost of finding these patterns. A novel pruning technique has been introduced to reduce the search space and the computational cost of these patterns effectively.
- A pattern-growth algorithm, called Generalized Periodic-Frequent pattern-growth (GPF-growth), has been proposed to discover the complete set of partial periodic-frequent patterns in a database.
- Experimental results demonstrate that the proposed model can discover useful information, and GPF-growth is efficient.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the existing model of periodic-frequent patterns. Section 4 introduces the proposed model of partial periodic-frequent patterns. Section 5 describes the GPF-growth algorithm. Section 6 reports on the experimental results. Section 7 concludes the paper with future research directions.

The problem of finding partial periodic-frequent patterns was first discussed in Kiran and Reddy (2011). In this paper, we introduce a novel pruning technique to tackle the predicament caused by the violation of anti-monotonic property by the *periodic-ratio* measure. We also provide theoretical correctness of GPF-growth, and discuss the usefulness of proposed model by conducting extensive experiments on both synthetic and real-world databases.

2. Related work

Time series is a collection of events obtained from sequential measurements over time. The problem of finding partial periodic patterns has received a great deal of attention in time series data (Han et al., 1998; 1999; Yang et al., 2003; Berberidis et al., 2002; Cao et al., 2004; Aref et al., 2004; Chen et al., 2011). The basic model used in all of these approaches, however, remains the same. It involves the following two steps (Han et al., 1998):

1. Partition the time series into distinct subsets (or periodic-segments) of a fixed length (or *period*).
2. Discover all partial periodic patterns that satisfy the user-specified minimum support (*minSup*). The *minSup* controls the minimum number of periodic-segments that a pattern must cover.

Example 1. Given the time series $TS = a\{bc\}baebace$ and the user-specified $period = 3,$ TS is divided into three periodic-segments: $TS_1 = a\{bc\}b$, $TS_2 = aeb$ and $TS_3 = ace$. Let $a\star b$ be a pattern, where '$\star$' denotes a wild character that can represent any single set of events. This pattern appears in the periodic-segments of TS_1 and TS_2. Therefore, its *support* count is 2. If the user-defined $minSup = 2,$ then $a\star b$ represents a partial periodic pattern in TS.

The major limitation of this model is that it considers time series as a symbolic sequence, and therefore, do not take into account the actual temporal information of the events within a sequence.

Ma and Hellerstein (2001) have proposed an alternative partial periodic pattern model by considering the temporal information of items within a series. This model considers time series as a time-based sequence, specifies *period* for each pattern using Fast Fourier Transformation (FFT), and discovers all patterns that satisfy the user-defined minimum support (*minSup*). The discovered patterns are known as **p-patterns**. Unfortunately, this model is