Knowledge-Based Systems 79 (2015) 68–79

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Mining closed partially ordered patterns, a new optimized algorithm

Mickaël Fabrègue [a,b,*], Agnès Braud [c], Sandra Bringay [d], Florence Le Ber [a], Maguelonne Teisseire [b]

[a] ICube, University of Strasbourg/ENGEES, CNRS, Illkirch, France
[b] TETIS, IRSTEA, Montpellier, France
[c] ICube, University of Strasbourg, CNRS, Illkirch, France
[d] LIRMM, Montpellier 3 University, CNRS, France

ARTICLE INFO

Article history:
Received 24 April 2014
Received in revised form 20 December 2014
Accepted 25 December 2014
Available online 17 January 2015

Keywords:
Data mining
Sequential patterns
Partially ordered patterns

ABSTRACT

Nowadays, sequence databases are available in several domains with increasing sizes. Exploring such databases with new pattern mining approaches involving new data structures is thus important. This paper investigates this data mining challenge by presenting *OrderSpan*, an algorithm that is able to extract a set of closed partially ordered patterns from a sequence database. It combines well-known properties of prefixes and suffixes. Furthermore, we extend *OrderSpan* by adapting efficient optimizations used in sequential pattern mining domain. Indeed, the proposed method is flexible and follows the sequential pattern paradigm. It is more efficient in the search space exploration, as it skips redundant branches. Experiments were performed on different real datasets to show (1) the effectiveness of the optimized approach and (2) the benefit of closed partially ordered patterns with respect to closed sequential patterns.

1. Introduction

Due to the exponential growth of temporal and spatiotemporal databases, sequential pattern mining has become a very active research area. Many studies have demonstrated the usefulness of such patterns for analysis [1], classification [2,3] or prediction [4]. These patterns were introduced in [5] and are an extension of association rules [6]. Several algorithms to mine such patterns have been proposed and are presented in [7,8]. They are used when information is totally ordered according to a specific criterion, which is usually temporal. For instance, let us take the well-know "market basket" problem. We consider a customer database where the pattern $\langle(Bread)(Chocolate)\rangle$ is found. This means that the product *Bread* is frequently purchased before the product *Chocolate*. Mining such related items according to temporal aspects is very useful for specialists in various domains such as marketing [9], software engineering [10] or medicine [11]. Despite their advantages, sequential patterns often generate little information since they only provide totally ordered information about data. For example, let us consider a second pattern, $\langle(Bread)(Milk)\rangle$, discovered in the same database. If these two patterns describe the same customers, their coexistence is not taken into account with sequential pattern approaches. However, they can be synthesized via partial ordering.

Fig. 1 presents a so-called partially ordered pattern that combines the two previous sequential patterns. This new pattern means that customers frequently purchase the product *Bread* before purchasing the two other products *Chocolate* and *Milk* which themselves are not ordered. Partially ordered patterns can be used in sequential databases and have many advantages: (1) they provide more information on order among elements; (2) they are represented as a directed acyclic graph, which facilitates the understanding; and (3) they summarize sequential pattern sets.

In a previous paper [12], we presented a method designed to directly extract closed partially ordered patterns in the general case of itemset sequences with item repetitions. In the present paper, we propose an improvement of our algorithm:

- Based on the property presented in [13], we present an optimized version of the so-called *OrderSpan* algorithm that explores the search space and outputs the complete set of closed partially ordered patterns.
- We use a new data structure to represent patterns in the algorithm. Properties of this new data structure lead to a different and generic way to remove the redundancy in patterns.
- We provide a complexity analysis of the approach and an upper bound on the number of extracted patterns given a minimum support.

* Corresponding author at: ICube, University of Strasbourg/ENGEES, CNRS, Illkirch, France.
E-mail addresses: mickael.fabregue@teledetection.fr (M. Fabrègue), agnes.braud@unistra.fr (A. Braud), sandra.bringay@lirmm.fr (S. Bringay), florence.leber@engees.unistra.fr (F. Le Ber), maguelonne.teisseire@teledetection.fr (M. Teisseire).

http://dx.doi.org/10.1016/j.knosys.2014.12.027
0950-7051/

The method proposed in [12] is close to the non-optimized algorithm presented in this paper. As we will see, the main difference is the use of an expanded data-structure that easily allows the addition of effective optimizations during the process. We thus integrated optimizations from [13].

Based on sequential pattern mining work, *OrderSpan* extracts partially ordered patterns based on the prefix and suffix properties of sequences. We opted to extract closed partially ordered patterns because they provide a compact representation of all partially ordered patterns. Thus, the output result set is smaller and it is possible to retrieve the complete set of all partially ordered patterns. There is no information loss. Our approach follows the *Pattern-Growth* paradigm on sequences, thus it is related to other approaches in sequential pattern mining. Some of these methods [13,14] are optimized to explore the search space of closed sequential patterns in a very efficient way. These optimizations are performed according to some properties that help to prune the search space to reduce its exploration. Thus, we analyzed closed sequential pattern properties that can be applied to the problem of mining closed partially ordered patterns. We adapted the optimization based on the equivalence databases proposed in *CloSpan* [13]. This property efficiently prunes the search space in the case of sequential pattern mining. We generalized it to the sub-search space that corresponds to a closed partially ordered pattern.

This paper is organized as follows. Section 2 gives some preliminary definitions on sequences and partially ordered patterns. Section 3 describes existing studies on partially ordered pattern mining. Section 4 introduces the *OrderSpan* algorithm including an optimization step and complexity analysis. Experimental results are presented in Section 5. Firstly, we compare the non-optimized and the optimized algorithm on a set of examples. Secondly, we compare the optimized version of *OrderSpan* with the algorithm proposed in [15]. Finally, we study the semantic aspects of closed partially ordered patterns.

2. Problem definition

Before presenting the partially ordered pattern concept, we provide some important definitions relative to closed sequential pattern mining. As we will see later, a partially ordered pattern is a more complex structure composed of closed sequential patterns. Let us first define a sequence (Definition 1), sub-sequence (Definition 2), a sequential pattern (Definition 3) and a closed sequential pattern (Definition 4).

Definition 1 (*Sequence*)
Let $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ be a set of **items**. An **itemset** IS is a non empty, unordered, set of **items** denoted $(I_{j_1} \ldots I_{j_k})$ where $I_{j_i} \in \mathcal{I}$. Let $\mathcal{IS}$ be the set of all **itemsets** built from I. A **sequence** S is a non-empty ordered list of **itemsets** denoted $\langle IS_1 IS_2 \ldots IS_p \rangle$ where $IS_j \in \mathcal{IS}$.

Definition 2 (*Sub-sequence*)
A **sequence** $S_\alpha = \langle IS_1 IS_2 \ldots IS_p \rangle$ is a **sub-sequence** of another **sequence** $S_\beta = \langle IS'_1 IS'_2 \ldots IS'_m \rangle$, denoted $S_\alpha \preceq_s S_\beta$, if $p \leqslant m$ and if there are integers $j_1 < j_2 < \ldots < j_k < \ldots < j_p$ such that $IS_1 \subseteq IS'_{j_1}$, $IS_2 \subseteq IS'_{j_2}, \ldots, IS_p \subseteq IS'_{j_p}$.

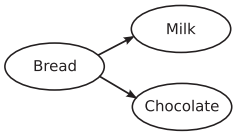

Fig. 1. Example of partially ordered pattern for the "market basket" problem.

Definition 3 (*Sequential pattern*)
Let S_P be a **sequence** and $\mathcal{DB}$ a sequence database. Let $\mathcal{S}' \subseteq \mathcal{DB}$ be the maximal set of **sequences** such that $\forall S_i \in \mathcal{S}', S_P \preceq_s S_i$. $|\mathcal{S}'|$ is called the support of S_P. S_P is called a **sequential pattern**, denoted **seq-pattern**, when $Support(S_P) \geqslant \theta$ where θ is a given value (minimum support).

Definition 4 (*Closed sequential pattern*)
Let S_P be a **sequential pattern**, S_P is a **closed sequential pattern** if there is no other **sequential pattern** S'_P such that $S_P \preceq_s S'_P$ and $Support(S_P) = Support(S'_P)$.

In the following, the database in Table 1 is used to illustrate these definitions. This database contains three sequences of itemsets based on the alphabet $\Sigma = \{a, c, d, e, f, g\}$. Given this database, the sub-sequences $\langle (g) \rangle, \langle (d) \rangle$ and $\langle (g)(d) \rangle$ are supported by sequences S_1, S_2 and S_3, and their support is equal to 3. Thus with a minimum support $\theta = 2$, these sub-sequences are seq-patterns because $3 \geqslant \theta$. But sequences $\langle (g) \rangle$ and $\langle (d) \rangle$ are not closed seq-patterns since the sequence $\langle (g)(d) \rangle$ is such that $\langle (g) \rangle \preceq_s \langle (g)(d) \rangle$ and $\langle (d) \rangle \preceq_s \langle (g)(d) \rangle$ with an equivalent support. Finally, the sequences $\langle (g)(d) \rangle, \langle (cd)(a) \rangle, \langle (cd)(g) \rangle, \langle (g)(d)(e) \rangle, \langle (g)(d)(f) \rangle$ and $\langle (g)(e)(f) \rangle$ in Table 2, give the complete set of closed seq-patterns for $\theta = 2$.

These are given with the associated set of supporting sequences. Some sets of closed seq-patterns are supported by the same set of sequences. For instance $\langle (g)(d)(e) \rangle, \langle (g)(d)(f) \rangle$ and $\langle (g)(e)(f) \rangle$ are supported by S_2 and S_3. A partial order can be used to obtain a synthetic representation of these closed seq-patterns relative to the sequence set $\{S_2, S_3\}$. We can define as many partial orders as there are corresponding sets of sequences. In our example, these sets are $\{S_1, S_2, S_3\}, \{S_1, S_2\}$ and $\{S_2, S_3\}$. Fig. 2a–c give the three partial orders corresponding to each set of sequences. We use two vertices labeled "$\langle$" and "$\rangle$", representing the beginning and end of the patterns.

These structures are used to represent partially ordered patterns which are defined in the following:

Definition 5 (*Partially ordered sequence*)
A **partially ordered sequence** is a set of itemsets with a partial order $(\mathcal{V}, <)$. It can be represented with a **labeled directed acyclic graph** $G = (\mathcal{V}, \mathcal{A}, \Sigma_\mathcal{V}, l_\mathcal{V})$ where:

- $\mathcal{V}$ is the set of **vertices** and $\mathcal{A}$ is the set of **arcs** where $\mathcal{A} = \{(u, v) \in <, with\ u, v \in \mathcal{V}\}$
- $\Sigma_\mathcal{V}$ is a finite alphabet representing possible vertex label values
- $l_\mathcal{V} : \mathcal{V} \rightarrow \Sigma_\mathcal{V}$ is a mapping giving the labeling on the **vertices**

In the graph, for all $u, v \in \mathcal{V}, u < v$ if there is a directed path from u to v. However, if there is no path from u to v or from v to u,

Table 1
An example of a sequence database.

Seq id	Sequence
S_1	$\langle (cd)(a)(g)(d) \rangle$
S_2	$\langle (g)(cde)(f)(aeg) \rangle$
S_3	$\langle (g)(d)(e)(f) \rangle$

Table 2
Set of closed sequential patterns related to Table 1 with the minimum support $\theta = 2$.

Supporting seq set	Sequence
$\{S_1, S_2, S_3\}$	$\langle (g)(d) \rangle$
$\{S_1, S_2\}$	$\langle (cd)(a) \rangle$
$\{S_1, S_2\}$	$\langle (cd)(g) \rangle$
$\{S_2, S_3\}$	$\langle (g)(d)(e) \rangle$
$\{S_2, S_3\}$	$\langle (g)(d)(f) \rangle$
$\{S_2, S_3\}$	$\langle (g)(e)(f) \rangle$