

Text and Sequential Pattern Mining

Sequential Pattern Mining (1)

Solen Quiniou

`solen.quiniou@univ-nantes.fr`

IUT de Nantes

Master 2 DS – 2020-2021

(last update: December 13, 2020)



Outline

- 1 Introduction
- 2 Sequential Pattern Mining
- 3 References

Outline

- 1 Introduction
- 2 Sequential Pattern Mining
- 3 References

Introduction

● Sequential Pattern Mining

- ▶ Sequential pattern mining aims at discovering interesting, useful, and unexpected subsequences in a set of sequences coming from databases [FVLK⁺17].
- ▶ The ordering of elements in the sequences is taken into account.
- There are numerous applications in real-life because data are naturally encoded as sequences of symbols in many fields.

● Some Applications of Sequential Pattern Mining

- ▶ Text analysis: information extraction, stylistics
- ▶ Bioinformatics: DNA sequences and gene structures
- ▶ e-learning: user traces
- ▶ Market basket analysis: customer shopping sequences
- ▶ Natural disasters: *e.g.*, earthquakes
- ▶ Energy reduction in smartphones: telephone calling patterns
- ▶ ...

Outline

1 Introduction

2 Sequential Pattern Mining

- Definitions
- Sequential Pattern Mining Algorithms
- GSP: an Apriori-based Algorithm with Horizontal Data Format
- SPADE: an Apriori-based Algorithm with Vertical Data Format
- PrefixSPAN: a Pattern-Growth Algorithm
- Discussion

3 References

Outline

1 Introduction

2 Sequential Pattern Mining

- **Definitions**
- Sequential Pattern Mining Algorithms
- GSP: an Apriori-based Algorithm with Horizontal Data Format
- SPADE: an Apriori-based Algorithm with Vertical Data Format
- PrefixSPAN: a Pattern-Growth Algorithm
- Discussion

3 References

Sequences

- Definition of itemsets

- ▶ Let there be a set of items (symbols) $I = \{i_1, \dots, i_m\}$.
An **itemset** $X = (x_1 \dots x_n)$ is a set of items such that $x_i \in I$.
- We assume that there exists a **total order** \prec on items such as the lexicographic order (e.g., $a \prec b \prec c \prec d$).

- Definition of sequences

- ▶ A **sequence** $s = \langle X_1 \dots X_n \rangle$ is an ordered list of itemsets with $X_k \subseteq I$ ($1 \leq k \leq n$).

- Definition of sequence databases

- ▶ A **sequence database** $SDB = \langle s_1, \dots, s_p \rangle$ is a list of sequences with sequence identifiers (SIDs) $1, \dots, p$.

- Definition of subsequences

- ▶ A sequence $s_a = \langle A_1 \dots A_n \rangle$ is a **subsequence** of another sequence $s_b = \langle B_1 \dots B_m \rangle$ if and only if there exist integers $1 \leq i_1 < i_2 < \dots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \dots, A_n \subseteq B_{i_n}$. This is denoted $s_a \sqsubseteq s_b$.
- Moreover, s_b is called a **supersequence** of s_a .

Example: sequences and database

Sequence database

SID	Sequence
1	$\langle (ab) (c) (fg) (g) (e) \rangle$
2	$\langle (ad) (c) (b) (abef) \rangle$
3	$\langle (a) (b) (fg) (e) \rangle$
4	$\langle (b) (fg) \rangle$

- The database contains **4 sequences**: s_1 , s_2 , s_3 , and s_4 .
- The first sequence contains **5 itemsets**: (ab) , (c) , (fg) , (g) , and (e) .
- The itemset $(abef)$ contains **4 items**.
- $\langle (b) (f, g) \rangle$ is **a subsequence** of the first sequence $\langle (ab) (c) (fg) (g) (e) \rangle$.
- $\langle (b) (g) (f) \rangle$ is **not a subsequence** of $\langle (ab) (c) (fg) (g) (e) \rangle$.

Sequential Pattern Mining

- Definition of support

- ▶ The **support** (absolute support) of a sequence s_a in a sequence database SDB is denoted by $sup(s_a)$ and is defined as the number of sequences that contains s_a . In other words, $sup(s_a) = |\{s | s_a \sqsubseteq s \wedge s \in SDB\}|$.
- The number of occurrences of the sequence s_a in SDB is not used but only the number of sequences of SDB in which s_a appears.

- Definition of relative support

- ▶ The **relative support** of a sequence s_a in a sequence database SDB is denoted by $relSup(s_a)$ and is defined by $relSup(s_a) = \frac{sup(s_a)}{|SDB|}$.

- Definition of frequent sequences [SA96]

- ▶ A sequence s is said to be a **frequent sequence** (or a **sequential pattern**) if and only if $sup(s) \geq minsup$, where $minsup$ is a threshold set by the user.

- Definition of sequential pattern mining

- ▶ **Sequential pattern mining** is defined as the task of finding the set FS of all the frequent subsequences in a sequence database SDB .
 FS is thus defined by $FS = \{s | sup(s) \geq minsup \wedge s \in SDB\}$.

Example: frequent sequences

Sequence database

SID	Sequence
1	$\langle (ab) (c) (fg) (g) (e) \rangle$
2	$\langle (ad) (c) (b) (abef) \rangle$
3	$\langle (a) (b) (fg) (e) \rangle$
4	$\langle (b) (fg) \rangle$

- The **support** of $\langle (b) (fg) \rangle$ is 3 because the subsequence is contained in sequences 1, 3, and 4.
- The **relative support** of $\langle (b) (fg) \rangle$ is then $3/4 = 0.75$.
- Given the threshold $minsup = 3$, the **frequent subsequences** are:

$$FS = \{ \langle (a) \rangle, \langle (b) \rangle, \langle (e) \rangle, \langle (f) \rangle, \langle (g) \rangle, \langle (fg) \rangle, \langle (a) (e) \rangle, \langle (a) (f) \rangle, \langle (b) (e) \rangle, \langle (b) (f) \rangle, \langle (b) (g) \rangle, \langle (b) (f, g) \rangle \}$$

Exercise: extraction of frequent sequences

Sequence database

SID	Sequence
1	$\langle (ab) (ab) (bde) \rangle$
2	$\langle (abcde) (be) \rangle$
3	$\langle (bce) \rangle$
4	$\langle (ac) (bce) \rangle$
5	$\langle (c) (c) (d) (bce) \rangle$

- 1 With a threshold $minsup = 3$, what is the set FS of all the frequent subsequences of the database?

$$FS = \{ \langle (a) \rangle, \langle (b) \rangle, \langle (c) \rangle, \langle (d) \rangle, \langle (e) \rangle, \\ \langle (a) (b) \rangle, \langle (a) (e) \rangle, \langle (c) (b) \rangle, \langle (c) (e) \rangle, \langle (bc) \rangle, \langle (be) \rangle, \langle (ce) \rangle, \\ \langle (a) (be) \rangle, \langle (c) (be) \rangle, \langle (bce) \rangle \}$$

Outline

1 Introduction

2 Sequential Pattern Mining

- Definitions
- **Sequential Pattern Mining Algorithms**
- GSP: an Apriori-based Algorithm with Horizontal Data Format
- SPADE: an Apriori-based Algorithm with Vertical Data Format
- PrefixSPAN: a Pattern-Growth Algorithm
- Discussion

3 References

Sequential Pattern Mining

- The task of sequential pattern mining is an **enumeration problem**.
- Thus, there is always a **single correct answer** to a sequential pattern mining problem.
- Discovering sequential patterns is a **hard problem**.
- The **naive approach** consists in:
 - 1 generating all the possible subsequences of a sequence database;
 - 2 computing the support of the subsequences;
 - 3 outputting only those meeting the minimum support constraint.
- This approach is inefficient because the number of subsequences can be very large.

For example, if the set of items contains 265 items, the total number of generated subsequences is 2^{265} : this is greater than the number of atoms in the universe (10^{79})!
- It is necessary to design efficient algorithms to avoid exploring the whole search space of possible subsequences.

Sequential Pattern Mining Algorithms

- To efficiently explore the search space, first algorithms adopted a **generate and prune approach** (also called *Apriori paradigm*):
 - 1 generate candidate patterns;
 - 2 compute the support of these candidate patterns;
 - 3 prune non-frequent candidate patterns;
 - 4 generate new candidate patterns, using frequent subsequences from the candidate patterns of step 2 and going back to step 3 until no new candidate patterns are created.
- The **generation step** is performed thanks to 2 extension operations.
 - ▶ A **s-extension** is a **sequence extension**. A sequence s_b is a **s-extension** of a sequence $s_a = \langle X_1 \dots X_h \rangle$ with an item x if $s_b = \langle X_1 \dots X_h (x) \rangle$.
 - ▶ An **i-extension** is an **itemset extension**. A sequence s_c is an **i-extension** of a sequence $s_a = \langle X_1 \dots X_h \rangle$ with an item x if $s_c = \langle X_1 \dots (X_h \cup \{x\}) \rangle$.
- The **pruning step** is based on the **Apriori property** (or downward closure property or anti-monotonicity property).
 - ▶ The **Apriori property** states that, for any sequences s_a and s_b , if s_a is a subsequence of s_b ($s_a \sqsubseteq s_b$), then s_b must have a support lower or equal to the support of s_a .
 - Then, if a sequence is non-frequent, its extensions are also non-frequent.

Example: sequence extension

Sequence database

SID	Sequence
1	$\langle (ab) (c) (fg) (g) (e) \rangle$
2	$\langle (ad) (c) (b) (abef) \rangle$
3	$\langle (a) (b) (fg) (e) \rangle$
4	$\langle (b) (fg) \rangle$

- $\langle (a) (a) \rangle$, $\langle (a) (b) \rangle$, and $\langle (a) (c) \rangle$ are **s-extensions** of $\langle (a) \rangle$
- $\langle (ab) \rangle$, and $\langle (ac) \rangle$ are **i-extensions** of $\langle (a) \rangle$
- $\langle (c) (g) \rangle$ is **non-frequent** (with $minsup = 3$) and so are its extensions such as $\langle (c) (g) (e) \rangle$ or $\langle (c) (fg) \rangle$

Sequential Pattern Mining Algorithm Categories (1)

Sequential pattern mining algorithms return the same set of sequential patterns but differ in:

- ① the type of **database representation** they use internally or externally (i.e., a horizontal or vertical data format);
- ② the **search space strategy**;
 - ▶ a **breadth-first algorithm** first scans the database for frequent 1-sequences (sequences with 1 item), then generates 2-sequences by performing s-extensions and i-extensions of 1-sequences, then generates 3-sequences using 2-sequences and so on until no new sequences can be generated;
 - ▶ a **depth-first algorithm** starts from the 1-sequences and recursively performs s-extensions and i-extensions with one of these sequences to generate larger sequences. Then, when the pattern can no longer be extended, the algorithm backtracks to generate other patterns using the other sequences.
- ③ the **generation step**, i.e. how they generate or determine the next patterns to be explored in the search space;
- ④ the **pruning step**, i.e. how they count the support of patterns to determine if they satisfy the minimum support constraint.

Sequential Pattern Mining Algorithm Categories (2)

Sequential pattern mining algorithms can be divided into 3 categories [MR13]:

❶ **Apriori-based algorithms with a horizontal data format**

- ▶ GSP, PSP, SPIRIT

❷ **Apriori-based algorithms with a vertical data format**

- ▶ SPADE, SPAM, LAPIN

❸ **Pattern-growth algorithms**

- ▶ PrefixSpan, FreeSpan

Outline

1 Introduction

2 Sequential Pattern Mining

- Definitions
- Sequential Pattern Mining Algorithms
- **GSP: an Apriori-based Algorithm with Horizontal Data Format**
- SPADE: an Apriori-based Algorithm with Vertical Data Format
- PrefixSPAN: a Pattern-Growth Algorithm
- Discussion

3 References

GSP algorithm [SA96]

- Characteristics of the GSP algorithm (*Generalized Sequential Patterns*)
 - ▶ It performs a **breadth-first search** to discover the sequential patterns.
 - ▶ It uses a **standard database representation** (i.e., a horizontal database).
 - ▶ It **generates candidate patterns of length $k + 1$** by combining pairs of patterns of length k that share all but one item (the generation process starts with frequent 1-sequences).
 - ▶ It **prunes candidates** in 2 steps:
 - 1 first, it checks if a candidate pattern s_a , of length $k + 1$, has all its subsequences of length k that are frequent; if not, s_a is discarded from the candidates;
 - 2 then, if s_a has not been discarded, it computes the support of s_a in the database; if $sup(s_a) \geq minsup$, then the candidate is kept among the sequential patterns of length $k + 1$.
- Limitations of the GSP algorithm
 - ▶ Multiple database scans;
 - ▶ Non-existent candidates (in the database);
 - ▶ Maintaining candidates in memory.

Example using the GSP algorithm

Sequence database

SID	Sequence
1	$\langle (a\ b)\ (a\ b)\ (b\ d\ e) \rangle$
2	$\langle (a\ b\ c\ d\ e)\ (b\ e) \rangle$
3	$\langle (b\ c\ e) \rangle$
4	$\langle (a\ c)\ (b\ c\ e) \rangle$
5	$\langle (c)\ (c)\ (d)\ (b\ c\ e) \rangle$

- 1 With a threshold $minsup = 3$, what is the set FS of all the frequent sequences of the database, using the GSP algorithm? Give all the patterns generated and kept at each step of the algorithm.

Outline

1 Introduction

2 Sequential Pattern Mining

- Definitions
- Sequential Pattern Mining Algorithms
- GSP: an Apriori-based Algorithm with Horizontal Data Format
- **SPADE: an Apriori-based Algorithm with Vertical Data Format**
- PrefixSPAN: a Pattern-Growth Algorithm
- Discussion

3 References

SPADE algorithm [Zak01]

- Characteristics of the SPADE algorithm

- ▶ It performs a **depth-first search** or a **breadth-first search** to discover the sequential patterns.
- ▶ It uses a **vertical database representation**.
- ▶ It **generates candidate patterns of length $k + 1$** the same way as in the GSP algorithm but the vertical database representation makes it faster to combine pairs of patterns of length k .
- ▶ It **prunes candidates** the same way as in the GSP algorithm but the vertical database representation makes it faster to compute the support of candidate patterns.

- Limitations of the SPADE algorithm

- ▶ Non-existent candidates;
- ▶ Maintaining candidates in memory.

Example: the SPADE algorithm

Sequence database

SID	Sequence
1	$\langle (a\ b)\ (a\ b)\ (b\ d\ e) \rangle$
2	$\langle (a\ b\ c\ d\ e)\ (b\ e) \rangle$
3	$\langle (b\ c\ e) \rangle$
4	$\langle (a\ c)\ (b\ c\ e) \rangle$
5	$\langle (c)\ (c)\ (d)\ (b\ c\ e) \rangle$

- 1 With a threshold $minsup = 3$, what is the set FS of all the frequent sequences of the database, using the SPADE algorithm? Give all the patterns generated and kept at each step of the algorithm.

Outline

1 Introduction

2 Sequential Pattern Mining

- Definitions
- Sequential Pattern Mining Algorithms
- GSP: an Apriori-based Algorithm with Horizontal Data Format
- SPADE: an Apriori-based Algorithm with Vertical Data Format
- **PrefixSPAN: a Pattern-Growth Algorithm**
- Discussion

3 References

PrefixSPAN algorithm [PHMA⁺04]

● Characteristics of the PrefixSPAN algorithm

- ▶ It performs a **depth-first search** to discover the sequential patterns.
- ▶ It uses a **standard database representation** (i.e., a horizontal database).
- ▶ It **only generates patterns of length $k + 1$** , by starting with sequential patterns containing a single item and by recursively scanning the database to find larger patterns. For a given sequential pattern s_a of length k , the algorithm proceeds as follows:
 - 1 First, it creates the projected database of the pattern s_a , i.e. the postfix sequences whose prefix is s_a ;
 - 2 Then, it scans the projected database of s_a to count the support of items in order to find items that can be appended to s_a by s-extension or i-extension, to form sequential patterns of length $k + 1$.
- ▶ **No pruning step** is needed here as only frequent patterns are generated at each step.

● Limitations of the PrefixSPAN algorithm

- ▶ Multiple database scans;
 - ▶ Maintaining database projections in memory.
- **Pseudo-projections** can be used to reduce the memory cost: it consists of implementing a projected database as a set of pointers to the original database.

Example: the PrefixSPAN algorithm

Sequence database

SID	Sequence
1	$\langle (a\ b)\ (a\ b)\ (b\ d\ e) \rangle$
2	$\langle (a\ b\ c\ d\ e)\ (b\ e) \rangle$
3	$\langle (b\ c\ e) \rangle$
4	$\langle (a\ c)\ (b\ c\ e) \rangle$
5	$\langle (c)\ (c)\ (d)\ (b\ c\ e) \rangle$

- 1 With a threshold $minsup = 3$, what is the set FS of all the frequent sequences of the database, using the PrefixSPAN algorithm? Give all the patterns generated and kept at each step of the algorithm.

Outline

1 Introduction

2 Sequential Pattern Mining

- Definitions
- Sequential Pattern Mining Algorithms
- GSP: an Apriori-based Algorithm with Horizontal Data Format
- SPADE: an Apriori-based Algorithm with Vertical Data Format
- PrefixSPAN: a Pattern-Growth Algorithm
- Discussion

3 References






Discussion

- The **time complexity** of sequential pattern mining algorithms depends on
 - ▶ the number of patterns in the search space;
 - ▶ the cost of the operations for generating and processing each sequence.
- The **number of patterns** in the search space depends on
 - ▶ how the *minsup* threshold is set by the user;
 - ▶ how similar the sequences are in the sequence database.

Outline

- 1 Introduction
- 2 Sequential Pattern Mining
- 3 References**

References I

-  Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas, *A Survey of Sequential Pattern Mining*, Data Science and Pattern Recognition **1** (2017), no. 1, 54–77.
-  C.H. Mooney and J.F. Roddick, *Sequential Pattern Mining – Approaches and Algorithms*, ACM Computing Surveys (CSUR) **15** (2013), no. 2, 19.
-  J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. C. Hsu, *Mining Sequential Patterns by pattern-growth: the prefixspan approach*, IEEE Transactions on Knowledge and Data Engineering **16** (2004), no. 11, 1424–1440.
-  R. Srikant and R. Agrawal, *Mining sequential patterns: generalizations and performance improvements*, Proc. of the International Conference on Extending Database Technology, 1996, pp. 1–17.
-  M. J. Zaki, *SPADE: an efficient algorithm for mining frequent sequences*, Machine Learning **42** (2001), no. 1-1, 31–60.