An Introduction to Sequential Pattern Mining

Philippe Fournier-Viger
http://www.philippe-Fournier-viger.com

Fournier-Viger, P., Lin, J. C.-W., Kiran, R. U., Koh, Y. S., Thomas, R. (2017). A Survey of Sequential Pattern Mining. Data Science and Pattern Recognition (DSPR), vol. 1(1), pp. 54-77.

Source code and datasets available in the SPMF library

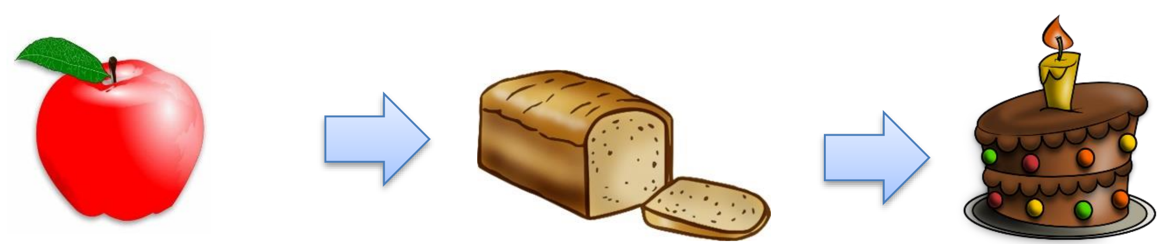
Introduction

- **Data Mining**: the goal is to discover or extract useful knowledge from data.
- Many types of data can be analyzed: graphs, relational databases, time series, sequences, etc.
- In this presentation, we focus on analyzing a common type of data called **discrete sequences** to find interesting patterns in it.

What is a discrete sequence?

A **sequence** is an ordered list of symbols.

Example 1: a sequence can be the items that are purchased by a customer over time:



What is a discrete sequence?

A **sequence** is an ordered list of symbols.

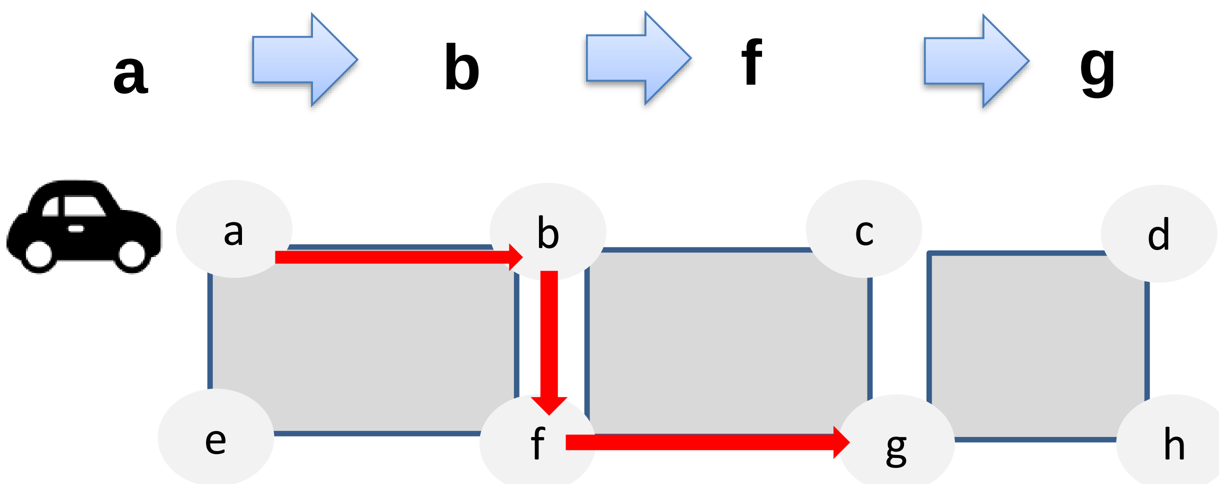
Example 2: a sequence can be the list of words in a sentence:

I → **go** → **back** → **home**

What is a discrete sequence?

A **sequence** is an ordered list of symbols.

Example 3: a sequence can be the list of locations visited by a car in a city



Sequential Pattern Mining

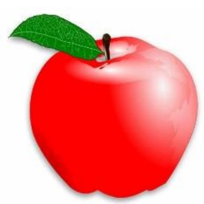
- It is a popular data mining task, introduced in 1994 by Agrawal & Srikant.
- The goal is to find all subsequences that appear frequently in a set of discrete sequences.
- **For example:**
  - find sequences of items purchased by many customers over time,
  - find sequences of locations frequently visited by tourists in a city,
  - Find sequences of words that appear frequently in a text.

Definition: Items

Let there be a **set** of **items** (symbols) called I.

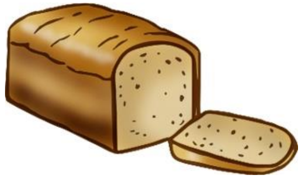
Example: $I = \{a, b, c, d, e\}$

a = apple

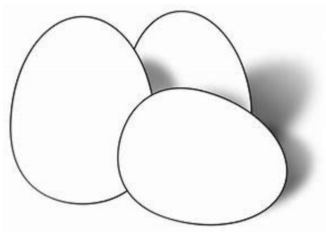


d = dattes

b = bread



e = eggs

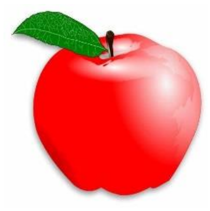
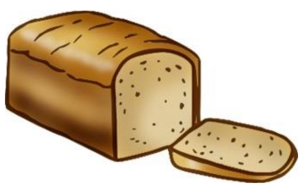


c = cake

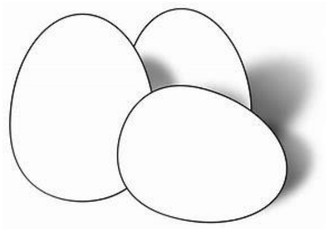
Definition: Itemset

An itemset is a set of **items** that is a subset of I.

Example: $\{a, b, c\}$ is an itemset containing 3 items

$\{d, e\}$ is an itemset containing 2 items

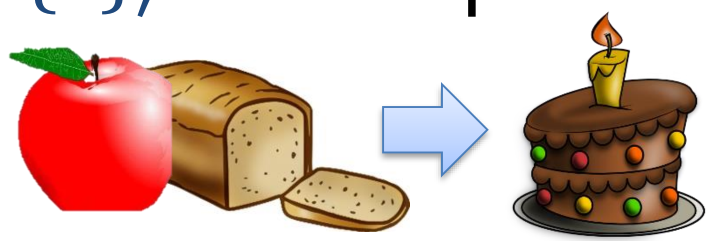


- Note: an itemset cannot contain a same item twice.
- An itemset having k items is called a *k-itemset*.

Definition: Sequence

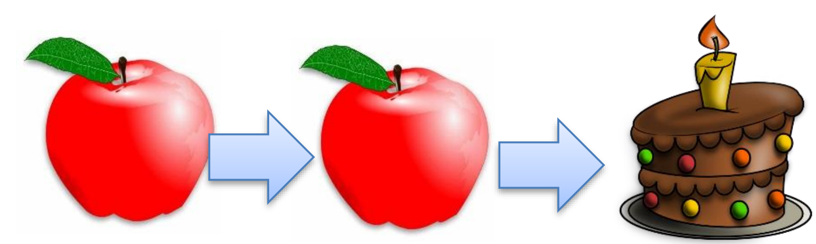
A **discrete sequence** S is a an ordered list of itemsets $S = \langle X_1, X_2, \ldots, X_n \rangle$ where $X_j \subseteq I$ for any $j \in \{1,2..n\}$

Example 1: $\langle \{a,b\}, \{c\} \rangle$ is a sequence containing two itemsets.



It means that a customer purchased $apple$ and $bread$ at the same time and then purchased $cake$.

Example 2: $\langle \{a\}, \{a\}, \{c\} \rangle$



Definition: Subsequence ($\sqsubseteq$)

Let there be two sequences:
$S_A = \langle A_1, A_2, \dots, A_r \rangle$ and $S_B = \langle B_1, B_2, \dots, B_t \rangle$.
The sequence S_A **is a subsequence** of S_B if and only if there exists r integers $1 \leq i1 < i2 < \cdots < ir \leq t$ such that $A_1 \subseteq B_{i1}, A_2 \subseteq B_{i2}$, … $A_r \subseteq B_{ir}$.

This is denoted as $\mathrm{S_A} \sqsubseteq S_B$

Examples:

$\langle \{a, c\} \rangle \sqsubseteq \langle \{a, b, c\} \rangle$
$\langle \{a, c\} \rangle \not\sqsubseteq \langle \{a\}, \{c\} \rangle$
$\langle \{a\}, \{c\} \rangle \sqsubseteq \langle \{a, b\}, \{d\}, \{b, c\} \rangle$
$\langle \{a\}, \{c\} \rangle \not\sqsubseteq \langle \{a, c\}, \{d\} \rangle$

Definition: Sequence database

A **sequence database** D is a set of discrete sequences $D = \{S_1, S_2, \ldots S_m\}$ where each sequence $S_j \in D$ has a unique identifier j.

Example 1: This is a sequence database with four sequences $D = \{S_1, S_2, S_3, S_4\}$:

Sequence database

$S_1 =$	$\langle\{a, b\}, \{c\}, \{a\}\rangle$
$S_2 =$	$\langle\{a\}, \{b\}, \{c\}\rangle$
$S_3 =$	$\langle\{b\}, \{c\}, \{d\}\rangle$
$S_4 =$	$\langle\{b\}, \{a, b\}, \{c\}\rangle$

Definition: Support of a sequence

The number of sequences in a **sequence database** D that contain a sequence S_A is called the support of S_A. It is defined as:

$$sup(S_A) = |\{S \mid S \in D \text{ and } S_A \sqsubseteq S\}|$$

Example 1:

Sequence database

$S_1 =$	$\langle\{a, b\}, \{c\}, \{a\}\rangle$
$S_2 =$	$\langle\{a\}, \{b\}, \{c\}\rangle$
$S_3 =$	$\langle\{b\}, \{c\}, \{d\}\rangle$
$S_4 =$	$\langle\{b\}, \{a, b\}, \{c\}\rangle$

$sup(\langle\{a\}\rangle) = 3$

Definition: Support of a sequence

The number of sequences in a **sequence database** D that contain a sequence S_A is called the support of S_A. It is defined as:

$$sup(S_A) = |\{S \mid S \in D \text{ } and \text{ } S_A \sqsubseteq S\}|$$

Example 2:

Sequence database

$S_1 =$	$\langle\{a, b\}, \{c\}, \{a\}\rangle$
$S_2 =$	$\langle\{a\}, \{b\}, \{c\}\rangle$
$S_3 =$	$\langle\{b\}, \{c\}, \{d\}\rangle$
$S_4 =$	$\langle\{b\}, \{a, b\}, \{c\}\rangle$

$sup(\langle\{b\}\rangle) = 4$

Definition: Support of a sequence

The number of sequences in a **sequence database** D that contain a sequence S_A is called the support of S_A. It is defined as:

$$sup(S_A) = |\{S \mid S \in D \ and \ S_A \sqsubseteq S\}|$$

Example 3:

Sequence database

$S_1 =$	$\langle\{a, b\}, \{c\}, \{a\}\rangle$
$S_2 =$	$\langle\{a\}, \{b\}, \{c\}\rangle$
$S_3 =$	$\langle\{b\}, \{c\}, \{d\}\rangle$
$S_4 =$	$\langle\{b\}, \{a, b\}, \{c\}\rangle$

$$sup(\langle\{a\}, \{b\}\rangle = 1$$

Definition: Support of a sequence

The number of sequences in a **sequence database** D that contain a sequence S_A is called the support of S_A. It is defined as:

$$sup(S_A) = |\{S \mid S \in D \text{ and } S_A \sqsubseteq S\}|$$

Example 4:

Sequence database

$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

$sup(\langle\{a,b\}\rangle) = 2$

Definition: Sequential pattern mining

- **Input**: A sequence database D and a minimum support threshold $minsup > 0$.
- **Output**: All sequential patterns.
  A sequential pattern is a sequence S where $\sup(S) \geq minsup$.

Example 1

INPUT:

OUTPUT:

Sequence database

$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

$$minsup = 3$$

Example 1

INPUT:

Sequence database

$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

$minsup = 3$

OUTPUT:

all sequential patterns:

$\langle\{a\}\rangle$ support = 3
$\langle\{b\}\rangle$ support = 4
$\langle\{c\}\rangle$ support = 4
$\langle\{a\},\{c\}\rangle$ support = 3
$\langle\{a,b\}\rangle$ support = 2
$\langle\{b\},\{c\}\rangle$ support = 4
$\langle\{a,b\},\{c\}\rangle$ support = 3

What will happen if we change the threshold? →

Example 2

INPUT:

OUTPUT:

Sequence database

$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

$minsup = 4$

Observation: If we increase the *minsup* threshold, less patterns may be found

Example 2

INPUT:

Sequence database

$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

$minsup = 4$

OUTPUT:

all sequential patterns:

$\langle\{b\}\rangle$ support = 4

$\langle\{c\}\rangle$ support = 4

$\langle\{b\},\{c\}\rangle$ support = 4

Observation: If we increase the *minsup* threshold, less patterns may be found

It is a difficult problem!

- **A naïve algorithm** would read the database and count the support (frequency) of **all possible patterns.**
- **Inefficient** because there can be **a very large number of sequential patterns.**
- **For example:**

$\langle\{a\}\rangle, \langle\{b\}\rangle, \langle\{c\}\rangle$ ....

....

$\langle\{a,b\}\rangle, \langle\{a,c\}\rangle, \langle\{a,d\}\rangle$ ...

...

$\langle\{a\},\{a\}\rangle, \langle\{a\},\{a\},\{a\}, \langle\{a\},\{a\},\{a\},\{a\}\rangle\rangle$ .... . $\langle\{a,b\}\{a\}\rangle$,....
$\langle\{a\},\{b\}\{a\}\rangle$,....

....

- **An efficient algorithm** must find the frequent sequential patterns, without checking all possibilities.

Some popular algorithms

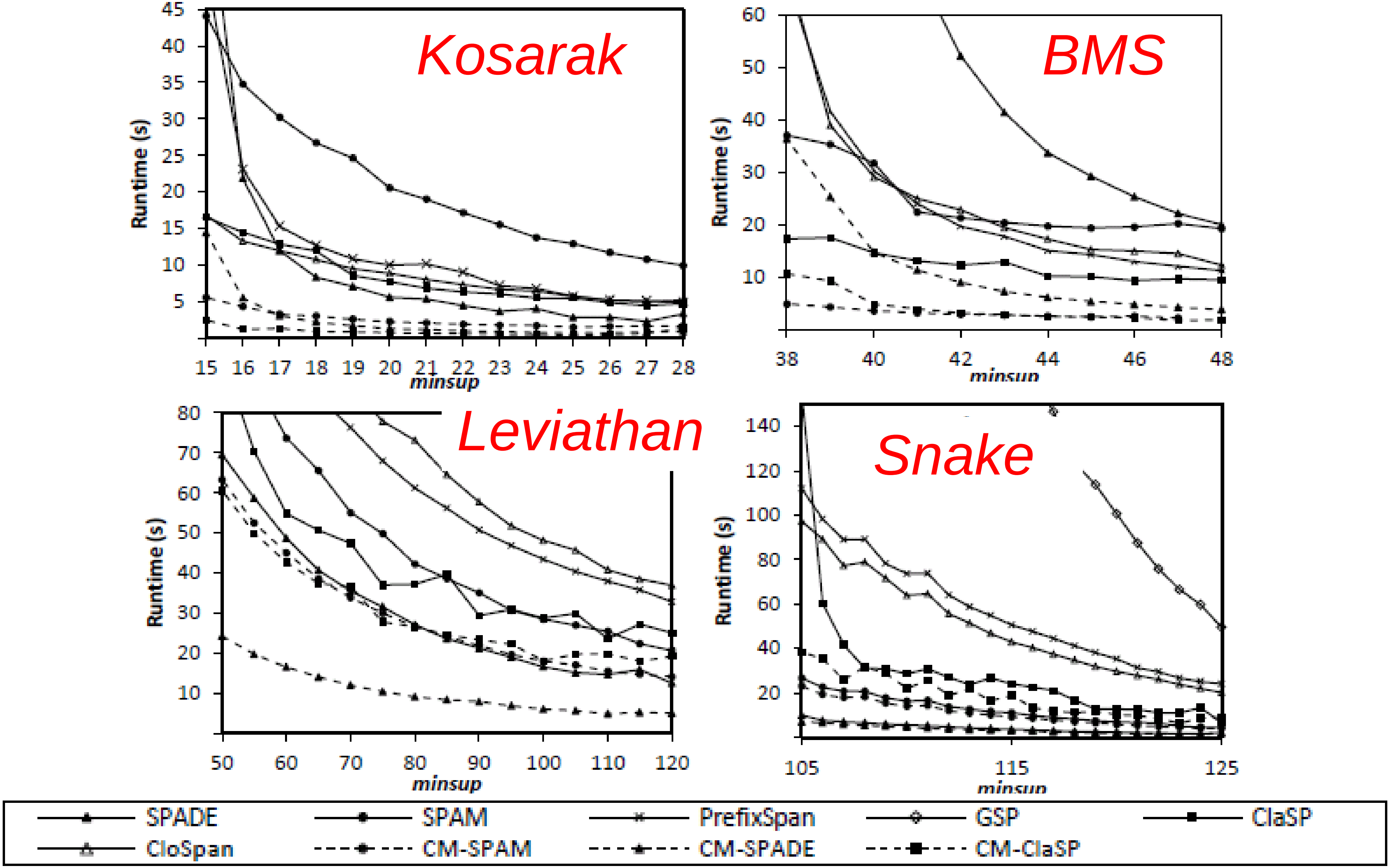
- **GSP**: R. Agrawal, and R. Srikant, Mining sequential patterns, ICDE 1995, pp. 3–14, 1995.
- **SPAM:** Ayres, J. Flannick, J. Gehrke, and T. Yiu, Sequential pattern mining using a bitmap representation, KDD 2002, pp. 429–435, 2002.
- **SPADE**: M. J. Zaki, SPADE: An efficient algorithm for mining frequent sequences, Machine learning, vol. 42(1-2), pp. 31–60, 2001.
- **PrefixSpan**: J. Pei, et al. Mining sequential patterns by pattern-growth: The prefixspan approach, IEEE Transactions on knowledge and data engineering, vol. 16(11), pp. 1424–1440, 2004.
- **CM-SPAM** and **CM-SPADE**: P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information, PAKDD 2014, pp. 40–52, 2014.

They all have the same input and output.
The difference is performance due to optimizations, search strategies and data structures!

Fast implementations available in the SPMF library

A performance comparison

Four benchmark datasets are used



The "Apriori" property

Property (anti-monotonicity).

Let be two subsequences **X** and **Y.** If **X** $\sqsubseteq$ **Y**, then the support of Y is less than or equal to the support of **X**.

Example

Sequence database

$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

The support of $\langle\{b\}\rangle$ **is** 4
The support of $\langle\{b\},\{c\}\rangle$ **is 4**
The support of $\langle\{b\},\{c\},\{d\}\rangle$ **is 1**