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

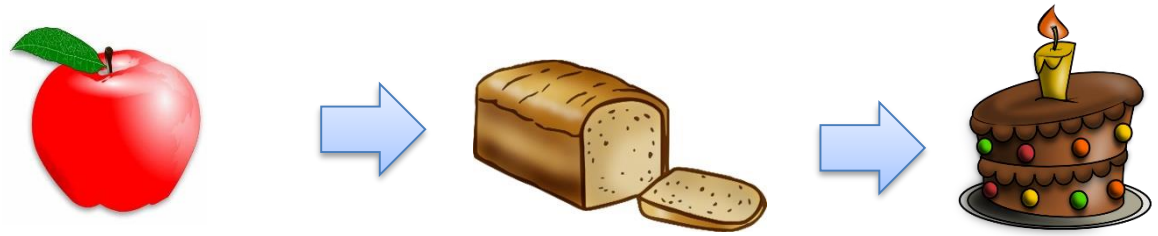
- **Data Mining**: the goal is to discover or extract useful knowledge from data.
- Many types of data can be analyzed: graphs, relational databases, time series, sequences, etc.
- In this presentation, we focus on analyzing a common type of data called **discrete sequences** to find interesting patterns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

Computer Monitor Router



What is a discrete sequence?

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

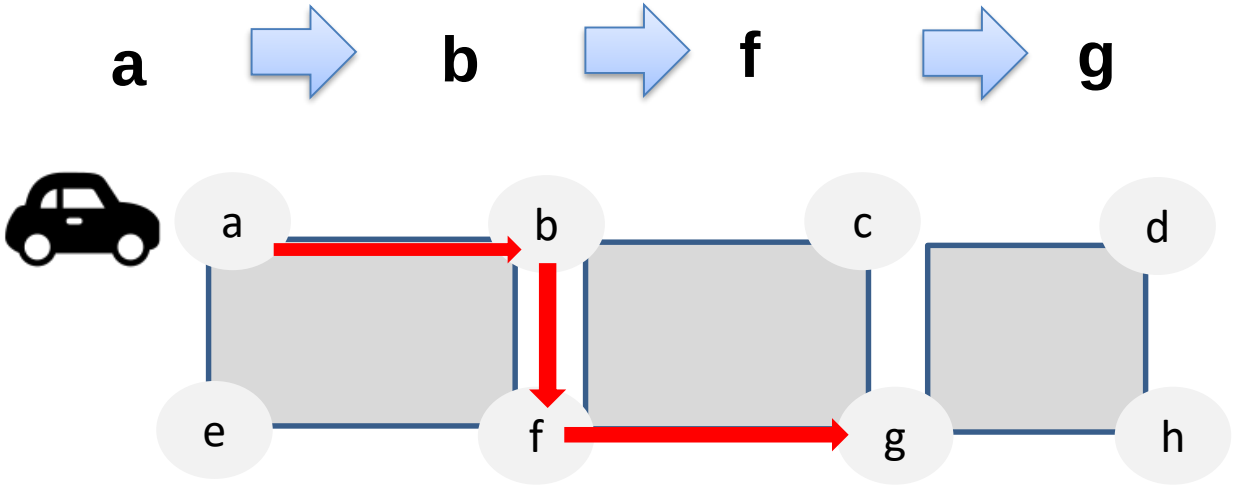
Example 2: a sequence can be the list of words in a sentence:

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.

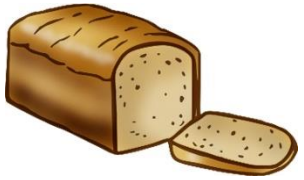
Definition: Items

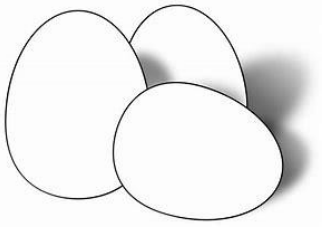
Let there be a **set** of **items** (symbols) called I.

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

a = apple 

d = dattes 

b = bread 

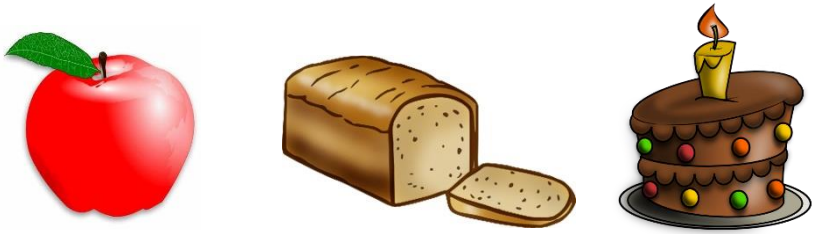
e = eggs 

c = cake 

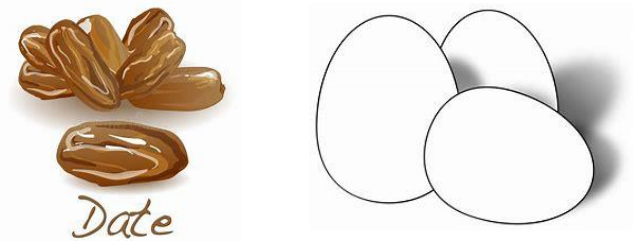
Definition: Itemset

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

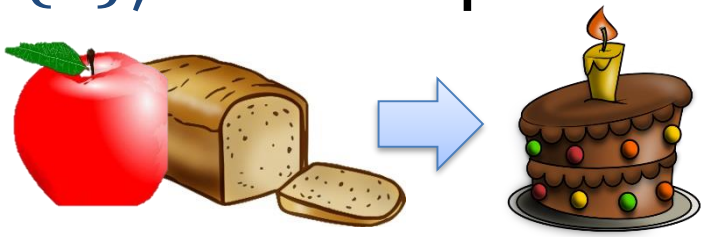


- Note: an itemset cannot contain a same item twice.
- An itemset having k items is called a *k-itemset*.

Definition: Sequence

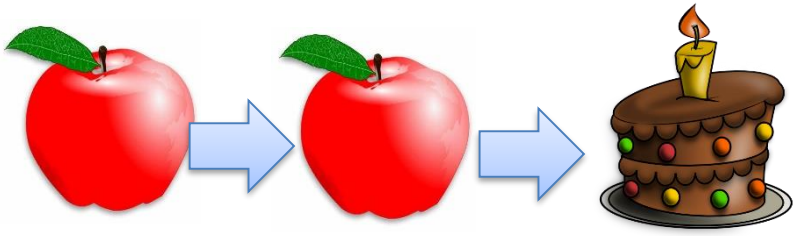
A **discrete sequence** S is a an ordered list of itemsets $S = \langle X_1, X_2, \ldots, X_n \rangle$ where $X_j \subseteq I$ for any $j \in \{1,2..n\}$

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:
$S_A = \langle A_1, A_2, \ldots, A_r \rangle$ and $S_B = \langle B_1, B_2, \ldots, B_t \rangle$.
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 $A_1 \subseteq B_{i1}, A_2 \subseteq B_{i2}, \ldots A_r \subseteq B_{ir}$.

This is denoted as $S_A \sqsubseteq S_B$

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:

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

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:

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

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

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

$minsup = 3$

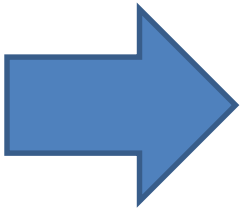
Example 1

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

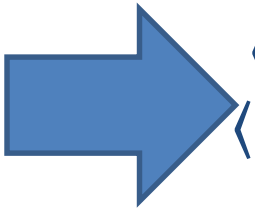
Example 2

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

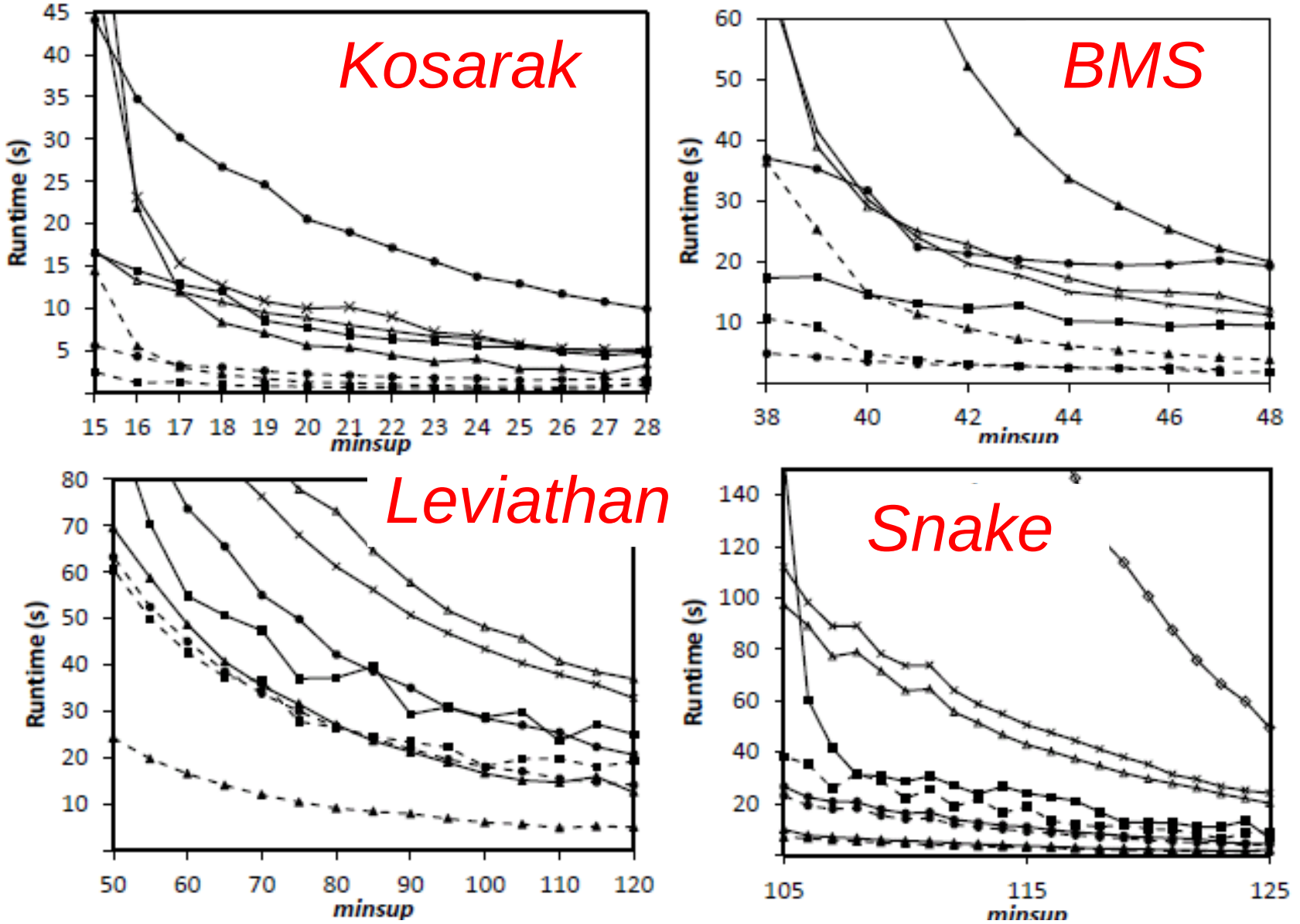
- **GSP**: R. Agrawal, and R. Srikant, Mining sequential patterns, ICDE 1995, pp. 3–14, 1995.
- **SPAM:** Ayres, J. Flannick, J. Gehrke, and T. Yiu, Sequential pattern mining using a bitmap representation, KDD 2002, pp. 429–435, 2002.
- **SPADE**: M. J. Zaki, SPADE: An efficient algorithm for mining 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 $\mathbf{X} \sqsubseteq \mathbf{Y}$, 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

THE PREFIXSPAN ALGORITHM

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.

The PrefixSpan algorithm

- Proposed by Jian Pei et al (2001)
- This algorithm is designed to only consider patterns that *exist* in the database.
- This algorithm uses a concept of *database projection* and a *depth-first search*.
- This is not the most efficient algorithm, but it is simple and easy to extend, so it is popular.
- I will explain with an example.

Example

This is the input:

Sequence database

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

$minsup = 3$

Step 1

PrefixSpan first counts the support of each item by scanning the database:

Sequence database

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

$minsup = 3$

Step 1

PrefixSpan first counts the support of each item by scanning the database:

Sequence database

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

Result:

$\langle\{a\}\rangle$ support : 3
$\langle\{b\}\rangle$ support : 4
$\langle\{c\}\rangle$ support : 4
$\langle\{d\}\rangle$ support : 1

$minsup = 3$

Step 2

PrefixSpan eliminates infrequent items:

Sequence database

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

Result:

$\langle\{a\}\rangle$ support : 3

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

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

~~$\langle\{d\}\rangle$ support : 1~~

$minsup = 3$

Step 2

PrefixSpan eliminates infrequent items:

Sequence database

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

Result:

$\langle\{a\}\rangle$ support : 3

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

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

$minsup = 3$

Step 2

PrefixSpan eliminates infrequent items:

Sequence database

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

$minsup = 3$

Result:

$\langle\{a\}\rangle$ support : 3
$\langle\{b\}\rangle$ support : 4
$\langle\{c\}\rangle$ support : 4

Those are the sequential patterns containing one item!

Step 2

PrefixSpan eliminates infrequent items:

Sequence database

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

$minsup = 3$

Result:

$\langle\{a\}\rangle$ support : 3
$\langle\{b\}\rangle$ support : 4
$\langle\{c\}\rangle$ support : 4

Those are the sequential patterns containing one item!

Prefixspan then extends each item recursively…
Lets start with $\langle\{a\}\rangle$ →

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

Sequence database

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

$minsup = 3$

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

PrefixSpan does a database projection with $\langle\{a\}\rangle$:

Sequence database

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

$minsup = 3$

What is a database projection?
It means to keep only the sequences containing $\langle\{a\}\rangle$.

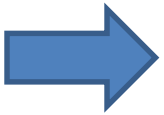
Moreover, for these sequences, we delete the first occurrence of $\langle\{a\}\rangle$ and everything that appears before.

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

PrefixSpan does a database projection with $\langle\{a\}\rangle$:

Sequence database

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



Projected database of $\langle\{a\}\rangle$

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

$minsup = 3$

What is a database projection?
It means to keep only the sequences containing $\langle\{a\}\rangle$.

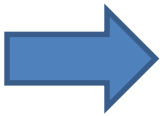
Moreover, for these sequences, we delete the first occurrence of $\langle\{a\}\rangle$ and everything that appears before.

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

PrefixSpan does a database projection with $\langle\{a\}\rangle$:

Sequence database

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



Projected database of $\langle\{a\}\rangle$

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

$minsup = 3$

What is a database projection?
It means to keep only the sequences containing $\langle\{a\}\rangle$.

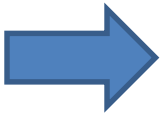
Moreover, for these sequences, we delete the first occurrence of $\langle\{a\}\rangle$ and everything that appears before.

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

PrefixSpan does a database projection with $\langle\{a\}\rangle$:

Sequence database

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



Projected database of $\langle\{a\}\rangle$

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

$minsup = 3$

What is a database projection?
It means to keep only the sequences containing $\langle\{a\}\rangle$.

Moreover, for these sequences, we delete the first occurrence of $\langle\{a\}\rangle$ and everything that appears before.

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a\}\rangle$ that has one more item:

Projected database of $\langle\{a\}\rangle$

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

$minsup = 3$

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a\}\rangle$ that has one more item:

Projected database of $\langle\{a\}\rangle$

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

$minsup = 3$

Result:

$\langle\{a\},\{a\}\rangle$ support : 1
$\langle\{a\},\{b\}\rangle$ support : 1
$\langle\{a\},\{c\}\rangle$ support: 3
$\langle\{a,b\}\rangle$ support : 3

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

Then, infrequent patterns are removed:

$minsup = 3$

Projected database of $\langle\{a\}\rangle$

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

Result:

$\langle\{a\},\{a\}\rangle$ support : 1
$\langle\{a\},\{b\}\rangle$ support : 1
$\langle\{a\},\{c\}\rangle$ support: 3
$\langle\{a,b\}\rangle$ support : 3

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

Then, infrequent patterns are removed:

Projected database of $\langle\{a\}\rangle$

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

Result:

$minsup = 3$

$\langle\{a\},\{c\}\rangle$ support: 3
$\langle\{a,b\}\rangle$ support : 3

Step 3 – Find patterns starting with $\langle\{a\}\rangle$

Then, infrequent patterns are removed:

Projected database of $\langle\{a\}\rangle$

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

Result:

$minsup = 3$

$\langle\{a\},\{c\}\rangle$ support: 3
$\langle\{a,b\}\rangle$ support : 3

Prefixspan then extends each pattern recursively…
Lets start with $\langle\{a\},\{c\}\rangle$ →

Step 4 – Find patterns starting with $\langle\{a\},\{c\}\rangle$

PrefixSpan does a database projection with $\langle\{a\},\{c\}\rangle$:

Projected database of $\langle\{a\}\rangle$

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

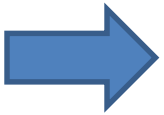
$minsup = 3$

Step 4 – Find patterns starting with $\langle\{a\},\{c\}\rangle$

PrefixSpan does a database projection with $\langle\{a\},\{c\}\rangle$:

Projected database of $\langle\{a\}\rangle$

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



Projected database of $\langle\{a\},\{c\}\rangle$

~~$S_1 =$~~	~~$\langle\{\_b\},\{c\},\{a\}\rangle$~~
~~$S_2 =$~~	~~$\langle\{\_b\},\{b\},\{c\}\rangle$~~
~~$S_4 =$~~	~~$\langle\{\_b\},\{c\}\rangle$~~

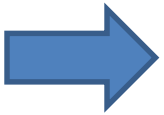
$minsup = 3$

Step 4 – Find patterns starting with $\langle\{a\},\{c\}\rangle$

PrefixSpan does a database projection with $\langle\{a\},\{c\}\rangle$:

Projected database of $\langle\{a\}\rangle$

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



Projected database of $\langle\{a\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Step 4 – Find patterns starting with $\langle\{a\},\{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a\}\rangle,\{c\}\rangle$ that has one more item:

Projected database of $\langle\{a\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Step 4 – Find patterns starting with $\langle\{a\},\{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a\}\rangle,\{c\}\rangle$ that has one more item:

Projected database of $\langle\{a\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

Result:

$\langle\{a\},\{c\},\{a\}\rangle$ support : 1

$minsup = 3$

Step 4 – Find patterns starting with $\langle\{a\},\{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a\}\rangle,\{c\}\rangle$ that has one more item:

Projected database of $\langle\{a\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

Result:

~~$\langle\{a\},\{c\},\{a\}\rangle$ support : 1~~

$minsup = 3$

This pattern is infrequent!

Then PrefixSpan try to find patterns starting with $\langle\{a,b\}\rangle$ →

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

PrefixSpan does a database projection with $\langle\{a, b\}\rangle$:

Projected database of $\langle\{a\}\rangle$

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

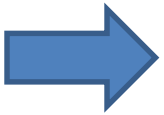
$minsup = 3$

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

PrefixSpan does a database projection with $\langle\{a, b\}\rangle$:

Projected database of $\langle\{a\}\rangle$

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



Projected database of $\langle\{a, b\}\rangle$

~~$S_1 =$~~	~~$\langle\{\_b\}, \{c\}, \{a\}\rangle$~~
~~$S_2 =$~~	~~$\langle\{\_b\}, \{b\}, \{c\}\rangle$~~
~~$S_4 =$~~	~~$\langle\{\_b\}, \{c\}\rangle$~~

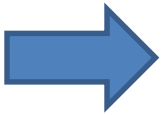
$minsup = 3$

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

PrefixSpan does a database projection with $\langle\{a, b\}\rangle$:

Projected database of $\langle\{a\}\rangle$

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



Projected database of $\langle\{a, b\}\rangle$

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

$minsup = 3$

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

PrefixSpan does a database projection with $\langle\{a, b\}\rangle$:

Projected database of $\langle\{a, b\}\rangle$

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

$minsup = 3$

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a, b\}\rangle$ that has one more item:

Projected database of $\langle\{a, b\}\rangle$

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

$minsup = 3$

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a, b\}\rangle$ that has one more item:

Projected database of $\langle\{a, b\}\rangle$

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

$minsup = 3$

Result:

$\langle\{a, b\}, \{c\}\rangle$ support : 3
$\langle\{a, b\}, \{a\}\rangle$ support : 1
$\langle\{a, b\}, \{b\}\rangle$ support : 1

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

Then, PrefixSpan removes infrequent patterns:

Projected database of $\langle\{a, b\}\rangle$

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

$minsup = 3$

Result:

$\langle\{a, b\}, \{c\}\rangle$ support : 3

~~$\langle\{a, b\}, \{a\}\rangle$ support : 1~~

~~$\langle\{a, b\}, \{b\}\rangle$ support : 1~~

Step 5 – Find patterns starting with $\langle\{a, b\}\rangle$

Then, PrefixSpan removes infrequent patterns:

Projected database of $\langle\{a, b\}\rangle$

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

Result:
$\langle\{a, b\}, \{c\}\rangle$ support : 3

$minsup = 3$

Then PrefixSpan try to find patterns starting with $\langle\{a, b\}, \{c\}\rangle$ →

Step 6 – Find patterns starting with $\langle\{a, b\}, \{c\}\rangle$

PrefixSpan does a database projection for $\langle\{a, b\}, \{c\}\rangle$:

Projected database of $\langle\{a, b\}\rangle$

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

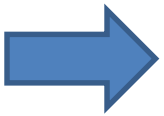
$minsup = 3$

Step 6 – Find patterns starting with $\langle\{a, b\}, \{c\}\rangle$

PrefixSpan does a database projection for $\langle\{a, b\}, \{c\}\rangle$:

Projected database of $\langle\{a, b\}\rangle$

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



Projected database of $\langle\{a, b\}, \{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Step 6 – Find patterns starting with $\langle\{a, b\}, \{c\}\rangle$

PrefixSpan does a database projection for $\langle\{a, b\}, \{c\}\rangle$:

Projected database of $\langle\{a, b\}, \{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Step 6 – Find patterns starting with $\langle\{a, b\}, \{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a, b\}, \{c\}\rangle$ that has one more item:

Projected database of $\langle\{a, b\}, \{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Step 6 – Find patterns starting with $\langle\{a, b\}, \{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a, b\}, \{c\}\rangle$ that has one more item:

Projected database of $\langle\{a, b\}, \{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Result:

$\langle\{a, b\}, \{c\}, \{a\}\rangle$ support : 1

Step 6 – Find patterns starting with $\langle\{a, b\}, \{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{a, b\}, \{c\}\rangle$ that has one more item:

Projected database of $\langle\{a, b\}, \{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$

$minsup = 3$

Result:

$\langle\{a, b\}, \{c\}, \{a\}\rangle$ support : 1

This pattern is infrequent!

Then, PrefixSpan tries to find patterns starting with $\langle\{b\}\rangle$ →

Step 7 – Find patterns starting with $\langle\{b\}\rangle$

Sequence database

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

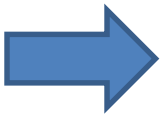
$minsup = 3$

Step 7 – Find patterns starting with $\langle\{b\}\rangle$

PrefixSpan does a database projection for $\langle\{b\}\rangle$:

Sequence database

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



Projected database of $\langle\{b\}\rangle$

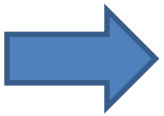
$minsup = 3$

Step 7 – Find patterns starting with $\langle\{b\}\rangle$

PrefixSpan does a database projection for $\langle\{b\}\rangle$:

Sequence database

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



Projected database of $\langle\{b\}\rangle$

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

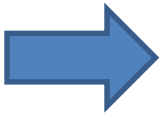
$minsup = 3$

Step 7 – Find patterns starting with $\langle\{b\}\rangle$

PrefixSpan does a database projection for $\langle\{b\}\rangle$:

Sequence database

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



Projected database of $\langle\{b\}\rangle$

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

$minsup = 3$

Step 7 – Find patterns starting with $\langle\{b\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with $\langle\{b\}\rangle$ that has one more item:

Projected database of $\langle\{b\}\rangle$

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

$minsup = 3$

Result:

$\langle\{b\},\{a\}\rangle$ support : 2
$\langle\{b\},\{b\}\rangle$ support : 2
$\langle\{b\},\{c\}\rangle$ support : 3
$\langle\{b\},\{d\}\rangle$ support : 1

Step 7 – Find patterns starting with $\langle\{b\}\rangle$

Then, PrefixSpan eliminates infrequent patterns:

Projected database of $\langle\{b\}\rangle$

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

$minsup = 3$

Result:

~~$\langle\{b\},\{a\}\rangle$ support : 2~~

~~$\langle\{b\},\{b\}\rangle$ support : 2~~

$\langle\{b\},\{c\}\rangle$ support : 3

~~$\langle\{b\},\{d\}\rangle$ support : 1~~

Then, PrefixSpan tries to find patterns starting with $\langle\{b\},\{c\}\rangle$ →

Step 8 – Find patterns starting with $\langle\{b\},\{c\}\rangle$

PrefixSpan does a database projection for $\langle\{b\},\{c\}\rangle$:

Projected database of $\langle\{b\}\rangle$

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

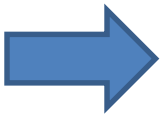
$minsup = 3$

Step 8 – Find patterns starting with $\langle\{b\},\{c\}\rangle$

PrefixSpan does a database projection for $\langle\{b\},\{c\}\rangle$:

Projected database of $\langle\{b\}\rangle$

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



Projected database of $\langle\{b\},\{c\}\rangle$

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

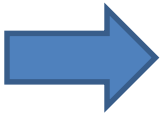
$minsup = 3$

Step 8 – Find patterns starting with $\langle\{b\},\{c\}\rangle$

PrefixSpan does a database projection for $\langle\{b\},\{c\}\rangle$:

Projected database of $\langle\{b\}\rangle$

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



Projected database of $\langle\{b\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$
$S_3 =$	$\langle\{d\}\rangle$

$minsup = 3$

Step 8 – Find patterns starting with $\langle\{b\},\{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with$\langle\{b\},\{c\}\rangle$ that has one more item:

Projected database of $\langle\{b\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$
$S_3 =$	$\langle\{d\}\rangle$

$minsup = 3$

Step 8 – Find patterns starting with $\langle\{b\},\{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with$\langle\{b\},\{c\}\rangle$ that has one more item:

Projected database of $\langle\{b\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$
$S_3 =$	$\langle\{d\}\rangle$

Result:

$\langle\{b\},\{c\},\{a\}\rangle$ support : 1

$\langle\{b\},\{c\},\{d\}\rangle$ support : 1

$minsup = 3$

Step 8 – Find patterns starting with $\langle\{b\},\{c\}\rangle$

Then, PrefixSpan counts the support of each sequential pattern starting with$\langle\{b\},\{c\}\rangle$ that has one more item:

Projected database of $\langle\{b\},\{c\}\rangle$

$S_1 =$	$\langle\{a\}\rangle$
$S_3 =$	$\langle\{d\}\rangle$

Result:

~~$\langle\{b\},\{c\},\{a\}\rangle$ support : 1~~

~~$\langle\{b\},\{c\},\{d\}\rangle$ support : 1~~

$minsup = 3$

All these patterns are infrequent!
PrefixSpan has finished its work.

Final result:

Those are the frequent 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 : 3
- $\langle\{a,b\},\{c\}\rangle$ support : 3
- $\langle\{b\},\{c\}\rangle$ support : 3

Observation

PrefixSpan performs a depth-first search:

⟨⟩

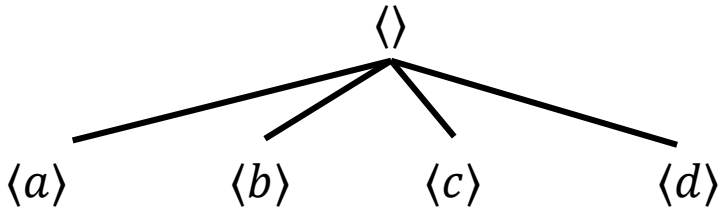
Notation:

Frequent sequential pattern

~~Infrequent sequential pattern~~

Observation

PrefixSpan performs a depth-first search:



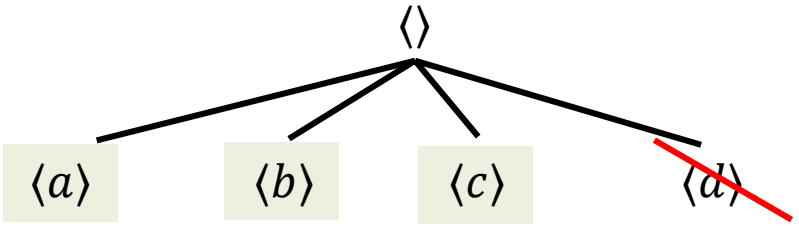
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



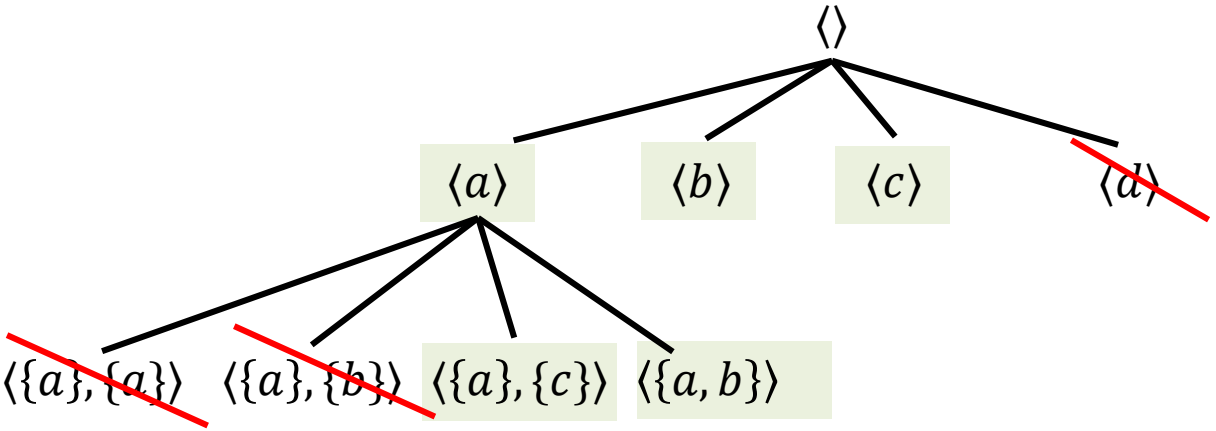
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



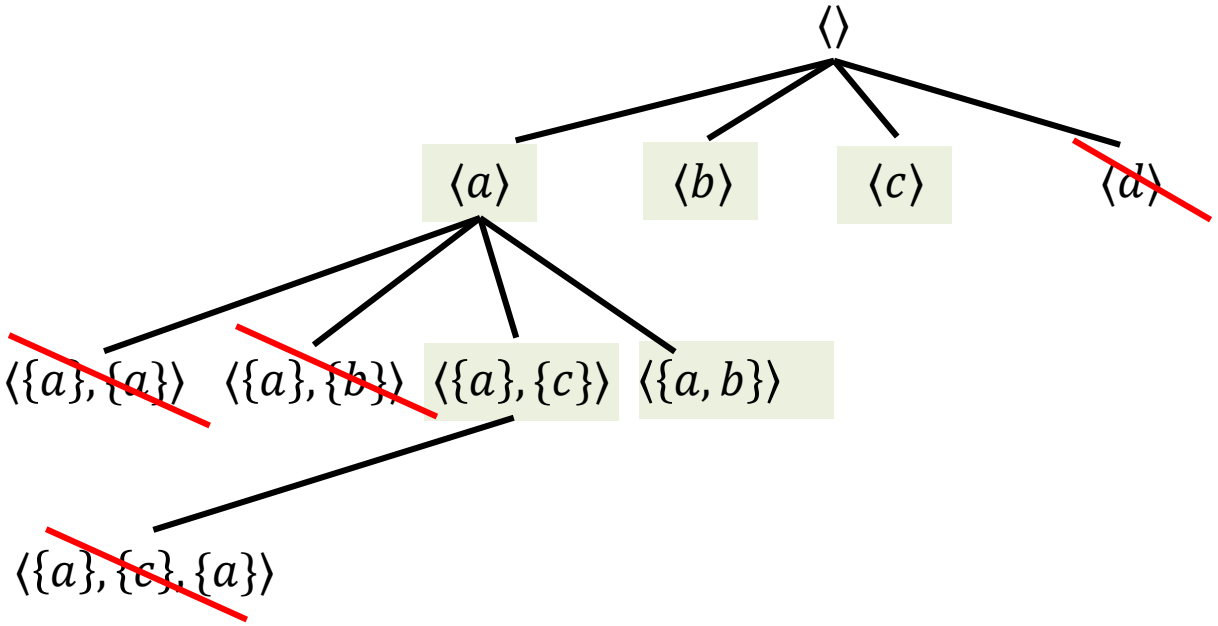
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



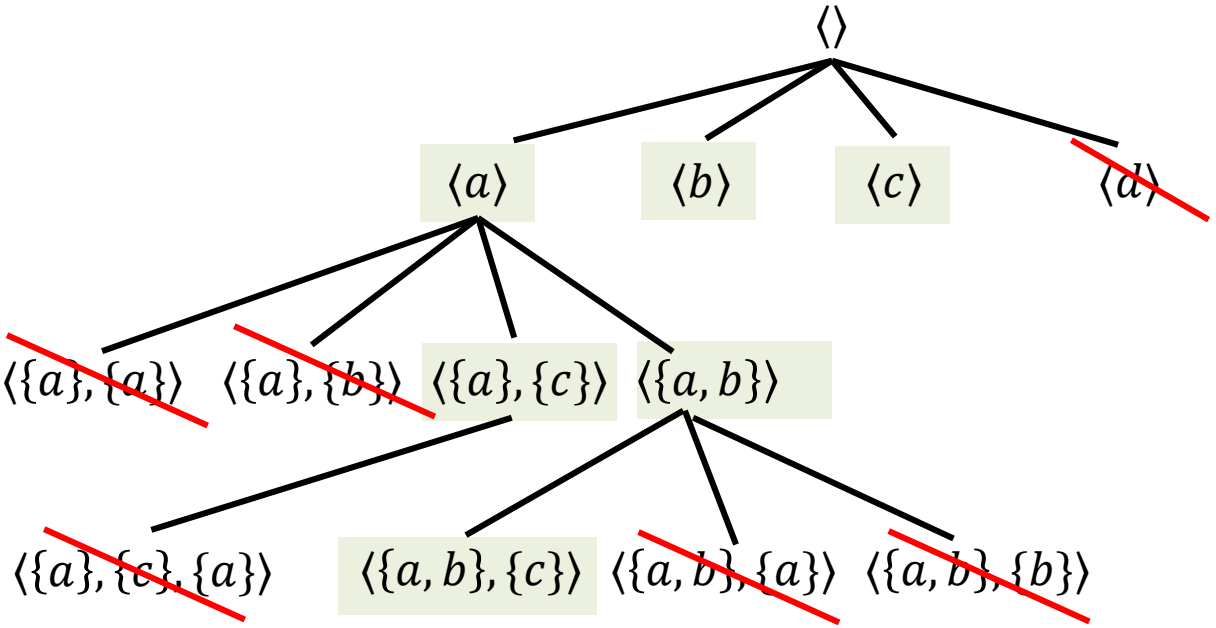
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



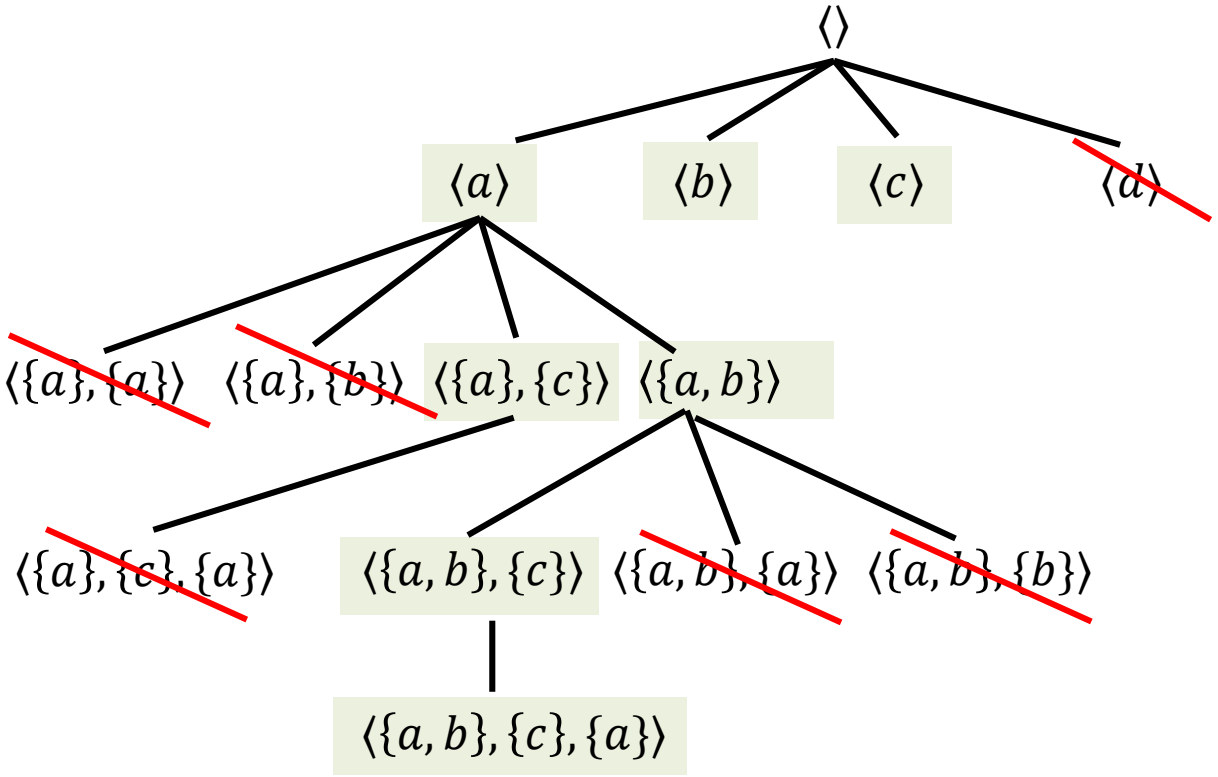
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



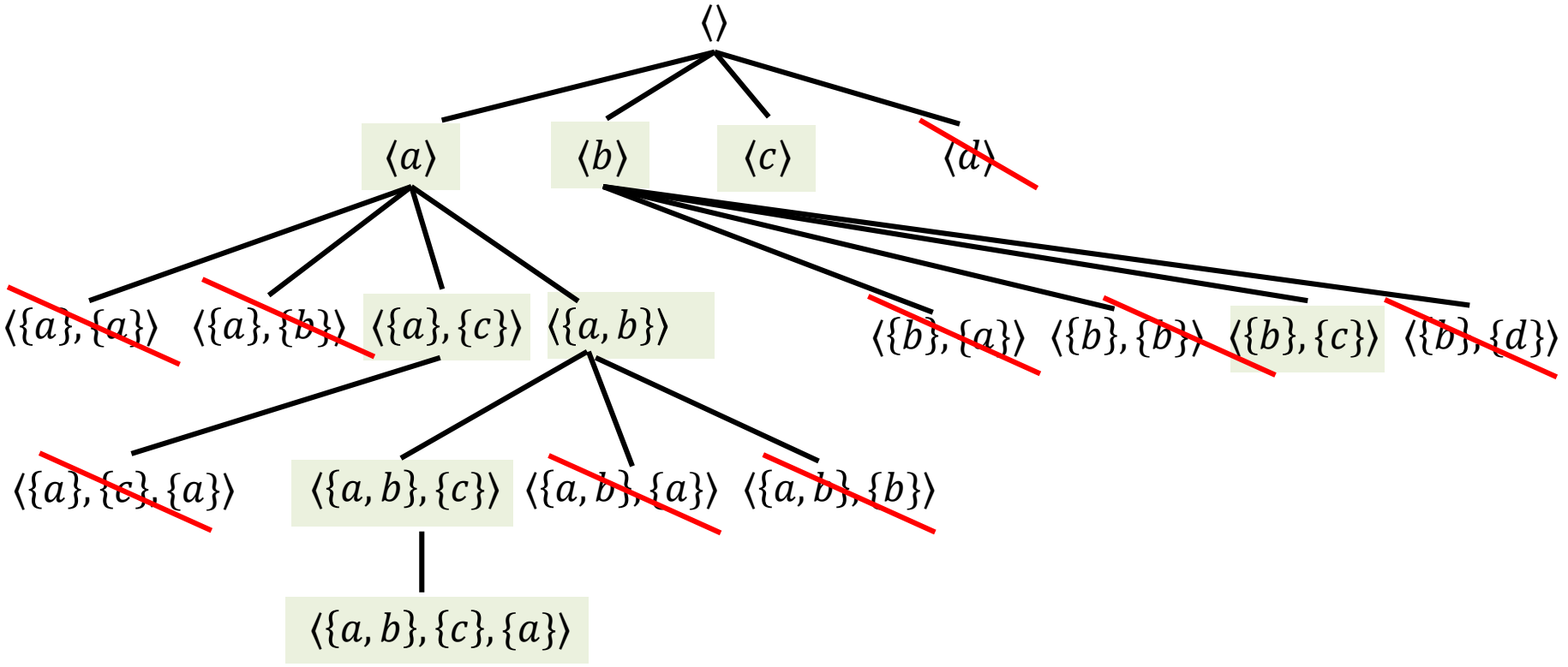
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



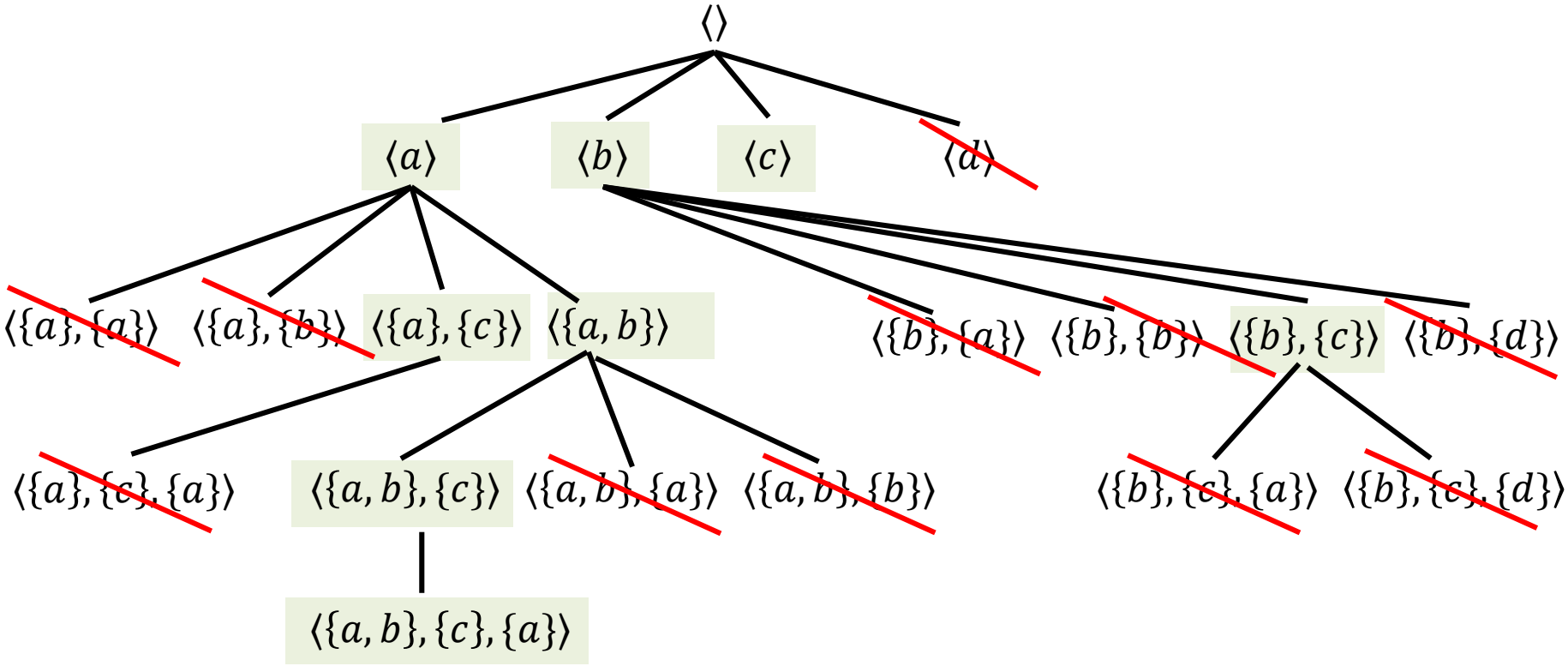
Notation:

Frequent sequential pattern

Infrequent sequential pattern

Observation

PrefixSpan performs a depth-first search:



Notation:

Frequent sequential pattern

Infrequent sequential pattern

Pseudocode of PrefixSpan (simple version)

PrefixSpan(a database D, a sequence S (initially empty $\langle\rangle$), $minsup$)

1. Scan *D* to find the support of each sequence starting with *S* that has one more item.
2. For each sequence R such that $\sup(R) \geq minsup$
3. Output R
4. Create the projected database D_R of R by doing a projection with D
5. Call **PrefixSpan**($D_R, R, minsup$)

Optimization 1

- **Observation**:
  - Making a copy of the database for each projection can spend a lot of time!
  - A projected database can also take a lot of memory.
- **Solution**:
  - do *pseudo-projections*
  - This means that we don't make a real copy. We use *pointers* on the original database instead.

Sequence database

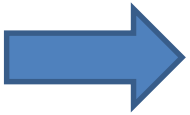
$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{d\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

Optimization 1

- **Observation**:
  - Making a copy of the database for each projection can spend a lot of time!
  - A projected database can also take a lot of memory.
- **Solution**:
  - do *pseudo-projections*
  - This means that we don't make a real copy. We use *pointers* on the original database instead.

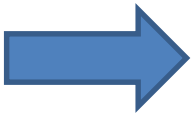
Sequence database

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



Projected database of $\langle\{a\}\rangle$

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



Pseudo-projected database of $\langle\{a\}\rangle$

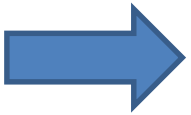
$S_1 =$	$\langle\{a\_b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a\_b\},\{b\},\{c\}\rangle$
$S_4 =$	$\langle\{b\},\{a\_b\},\{c\}\rangle$

Optimization 2

- **Observation**:
  - After reading the database to count the support of each item, PrefixSpan can remove all infrequent items from the database.
  - This will reduce the database size…
  - This could be done also when creating projected databases.

Sequence database

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



Sequence database

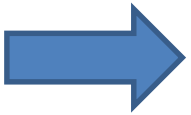
$S_1 =$	$\langle\{a,b\},\{c\},\{a\}\rangle$
$S_2 =$	$\langle\{a,b\},\{b\},\{c\}\rangle$
$S_3 =$	$\langle\{b\},\{c\},\{\boldsymbol{d}\}\rangle$
$S_4 =$	$\langle\{b\},\{a,b\},\{c\}\rangle$

Optimization 2

- **Observation**:
  - After reading the database to count the support of each item, PrefixSpan can remove all infrequent items from the database.
  - This will reduce the database size…
  - This could be done also when creating projected databases.

Sequence database

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



Sequence database

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

PrefixSpan is a good algorithm?

- Generally, very fast.
- For each frequent pattern, PrefixSpan scans the database once to count the support of patterns. This takes linear time w.r.t the database size.
- Creating a projected database is done in linear time
  - This can still consume a lot of time and memory.
  - But projected databases are always smaller than the original database.
- Unlike some other algorithms (e.g. GSP), PrefixSpan only considers patterns that exist in the database.
- PrefixSpan can be easily extended to add constraints (e.g. maximum length, maximum gap)

What influence the performance of PrefixSpan?

- The *minsup* threshold
- The database:
  - The number of sequences
  - The length of sequences
  - The sequences are similar?
  - The number of distinct items

Code, datasets and more…

- A fast Java implementation of **PrefixSpan** is available in the **SPMF data mining software** (http://www.philippe-fournier-viger.com/spmf/)

  - It can be used as a stand alone software, or as a library.
  - Several other sequential pattern mining algorithms are also provided.
  - Datasets are given
- A survey of sequential pattern mining:
  - 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.