An Introduction to Episode Mining

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

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.

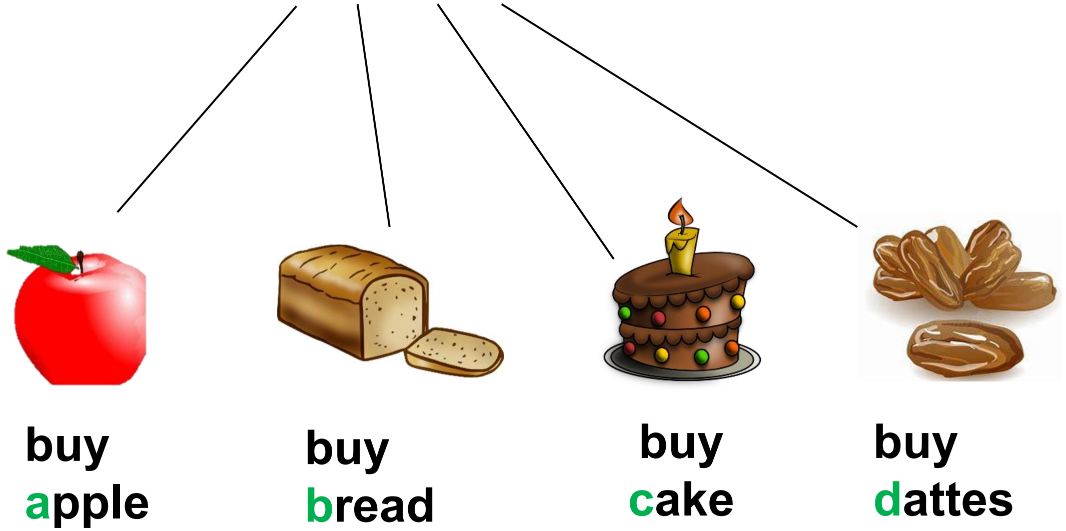
- In this presentation, we focus on **episode mining**, that is how to find **interesting patterns** in a **single**, **long sequence of events.**

Event types

- We have a set of different **event types**

$$E = \{i_1, i_2, \ldots, i_m\}$$

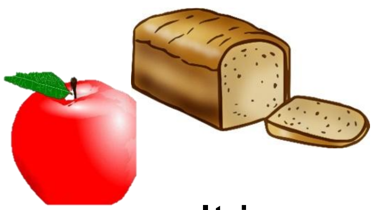
- For example: $E = \{a, b, c, d\}$



Event set

- An **event set** X is a set of events that have occurred at the same time. Formally, $X \subseteq E$.

- **Example 1**: **{a,b}** is an event set indicating that someone has bought **apple** and **bread** at the same time.



It is an event set of size 2

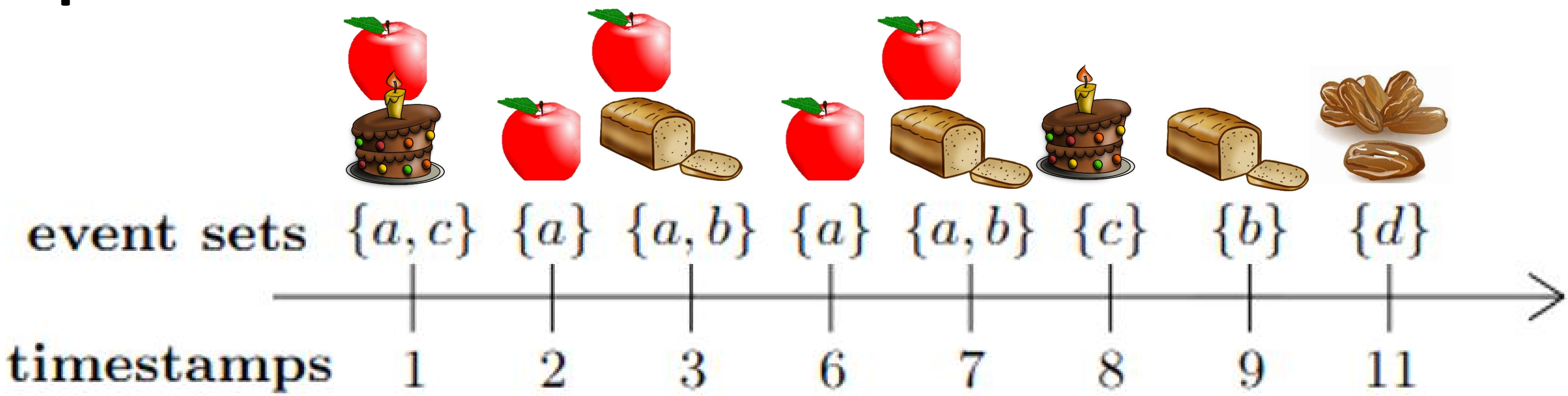
- **Example 1**: **{b, c, d}** is an event set indicating that someone has bought **bread**, **cake** and **dates** at the same time.

It is an event set of size 3

Event sequence

An event sequence is an ordered list of event pairs $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ where for any i, $SEt_i \subseteq E$ is the set of events observed at time t_i.

Example 1:



$$s = \langle (\{a, c\}, 1),\ \{a\}, 2), (\{a, b\}, 3), (\{a\}, 6), (\{a, b\}, 7),$$
$$(\{c\}, 8), (\{b\}, 9), (\{d\}, 11) \rangle$$

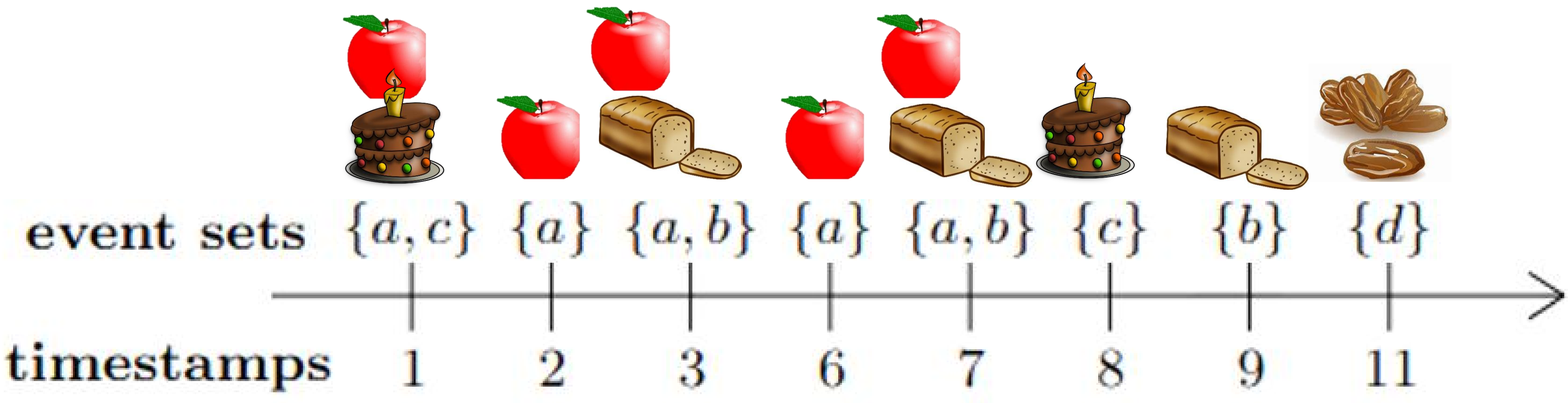
Event sequence

Event sequences can model various types of data such as:

- alarm sequences,
- cloud data,
- network data,
- stock data,
- malicious attacks,
- movements, and customer transactions.

The goal of Episode Mining

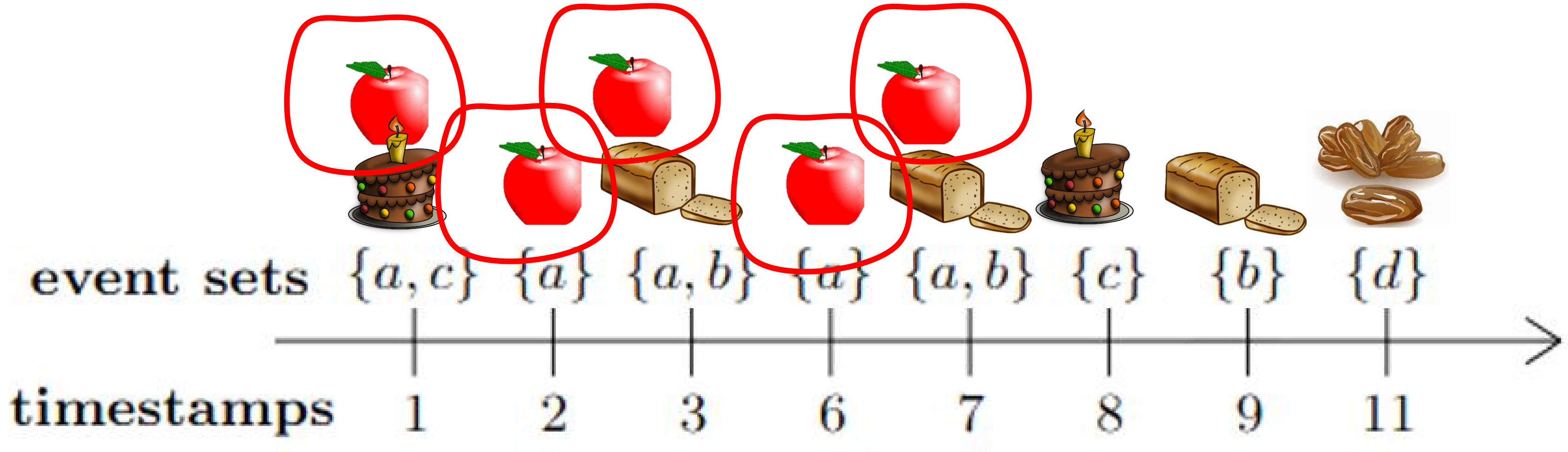
Given a sequence of events,



we want to discover subsequences of events that appear frequently (i.e. **frequent episodes**)

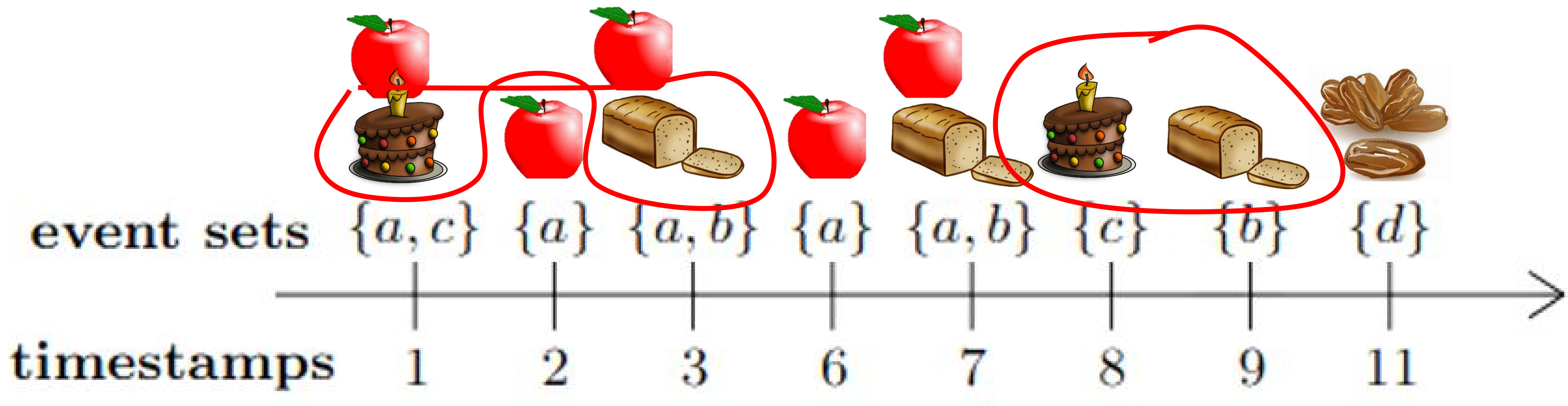
The goal of Episode Mining

For example, we may find that **apple** is bought many times



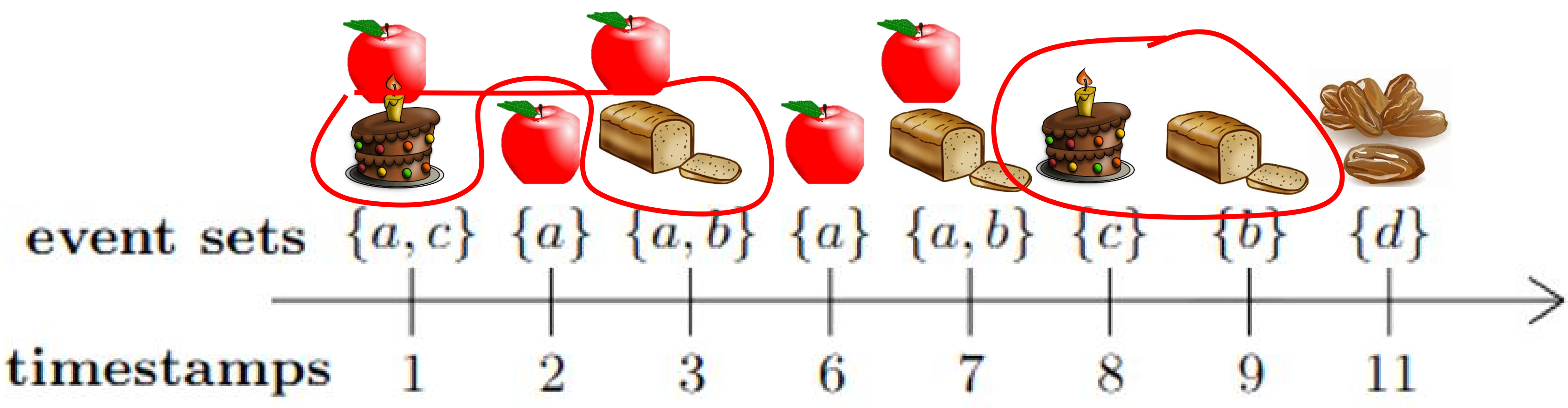
The goal of Episode Mining

Or that **cake** is frequently bought shortly before buying **bread**



The goal of Episode Mining

Or that **cake** is frequently bought shortly before buying **bread**



To give a clearer definition, we need to define:

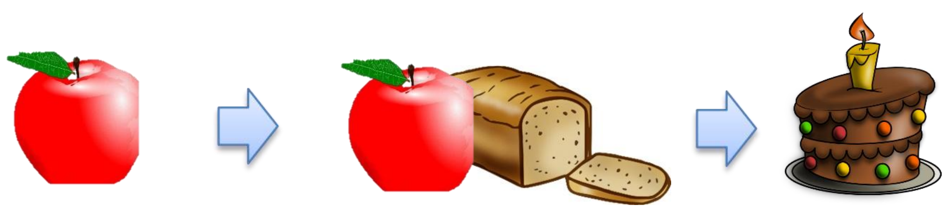
- what is an **episode**?
- how do we count the **support** of an episode (how many times it appears in an event sequence)?

Episode

(the general case)

A **(composite) episode** $\propto \; = \langle X_1, X_2, \ldots X_p \rangle$ is a list of event sets ordered by time, that is for any integers $1 \leq i < j \leq p$, X_i appeared before X_j .

Example: $\propto = \langle \{a\}, \quad \{a, b\}, \quad \{c\} \rangle$



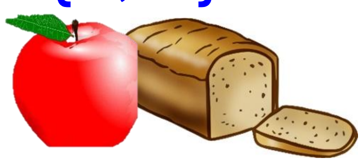
Apple was purchased. Then, **apple** and **bread** were bought at the same time, and then **cake** was purchased.

Parallel episode

(all events appeared at the same time)

A **parallel episode** $\propto = \langle X \rangle$ is an episode that contains a single event set ($X \subseteq E$).
Thus all events have appeared simultaneously
It can be written as $\propto = X$.

Example: $X = \{a, b\}$

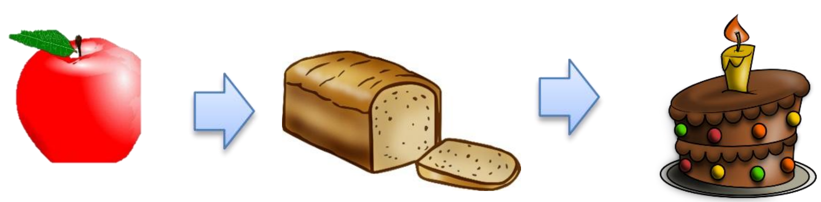


Apple and **bread** were bought together (at the same time)

Serial episode

A **serial episode** $\propto \; = \; \langle X_1, X_2, \ldots X_p \rangle$ is a list of event sets where each event set contains a single event.

Example: $\propto = \langle \{a\},\{b\}, \{c\} \rangle$



Apple was purchased. Then, **bread** was bought, and then **cake** was purchased.

How to count episodes?

- There are different ways (**functions**) for counting the *support* of episodes:
  - windows-based frequency
  - head support (head frequency),
  - total frequency,
  - non interleaved frequency,
  - minimal-occurrences based frequency
  - ...
- All these ways of counting may give different results.

I will explain the **head support -->**

Occurrence of an episode

An **occurrence** of an **episode** $\propto = \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ldots < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,} \ldots, X_p \subseteq SE_{zw}$

Occurrence of an episode

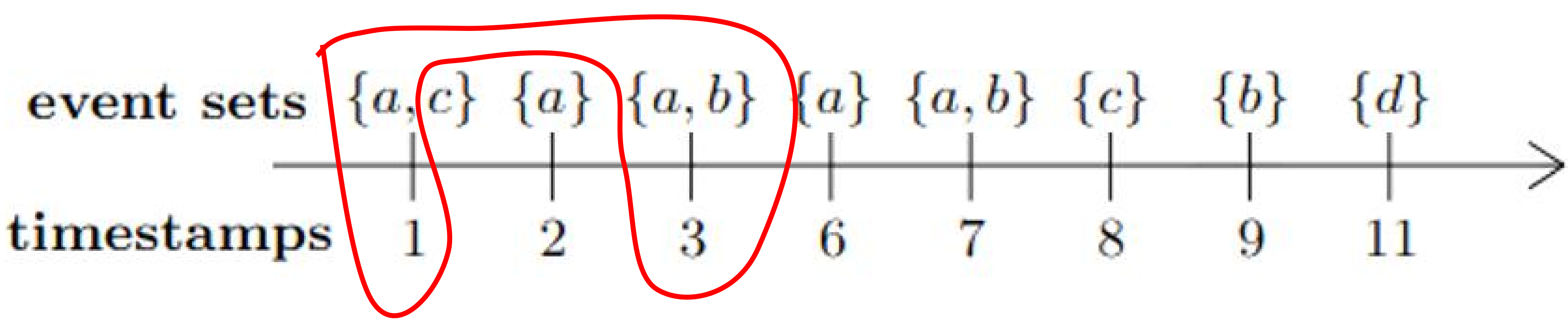
An **occurrence** of an **episode** $\propto = \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ldots < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,} \ldots, X_p \subseteq SE_{zw}$

Example: The episode $\propto = \langle \{a\}, \{a, b\} \rangle$ has an occurrence in time interval $[1,3]$ of sequence:



Occurrence of an episode

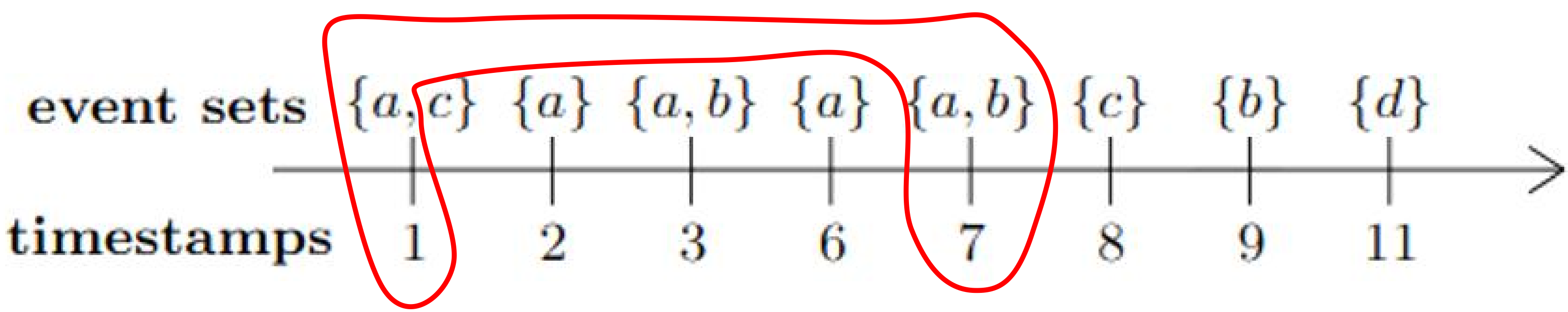
An **occurrence** of an **episode** $\propto = \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ldots < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,} \ldots, X_p \subseteq SE_{zw}$

Example: The episode $\propto = \langle \{a\}, \{a, b\} \rangle$ has an occurrence in time interval $[1,7]$ of sequence:



Occurrence of an episode

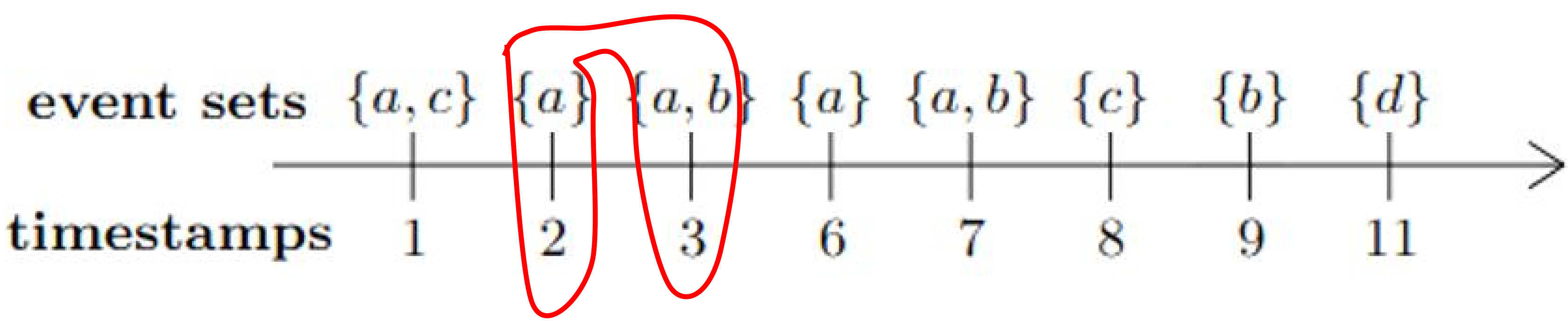
An **occurrence** of an **episode** $\propto = \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ldots < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,} \ldots, X_p \subseteq SE_{zw}$

Example: The episode $\propto = \langle \{a\}, \{a, b\} \rangle$ has an occurrence in time interval [2,3] of sequence:



Occurrence of an episode

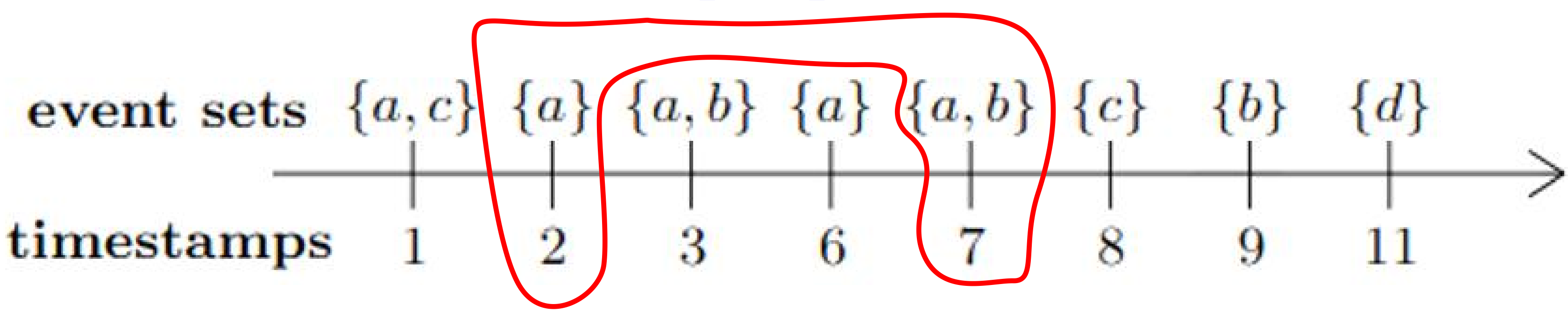
An **occurrence** of an **episode** $\propto = \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ldots < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,} \ldots, X_p \subseteq SE_{zw}$

Example: The episode $\propto = \langle \{a\}, \{a, b\} \rangle$ has an occurrence in time interval [2,7] of sequence:



Occurrence of an episode

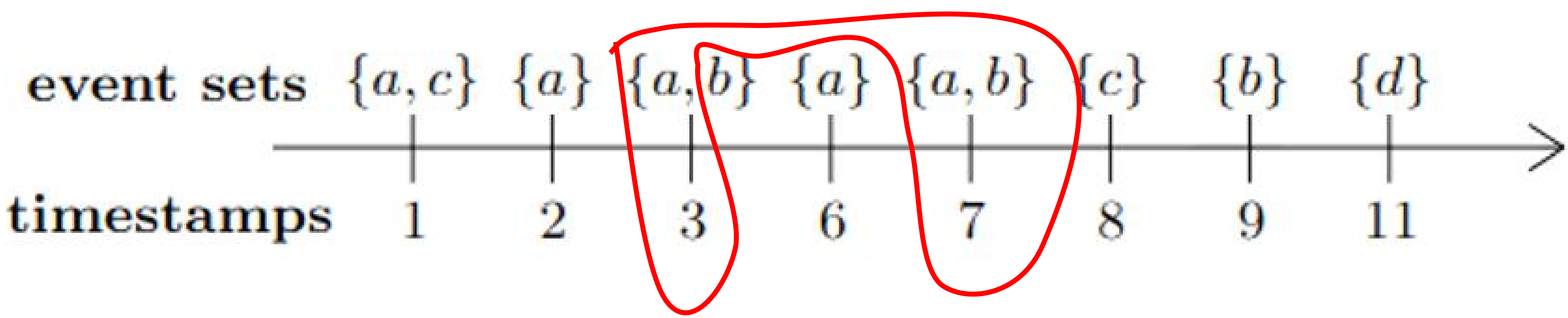
An **occurrence** of an **episode** $\propto\ =\ \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S\ =\langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ \ldots \ < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,}\ \ldots, X_p \subseteq SE_{zw}$

Example: The episode $\propto\ =\ \langle \{a\}, \{a, b\} \rangle$ has an occurrence in time interval $[3,7]$ of sequence:



Occurrence of an episode

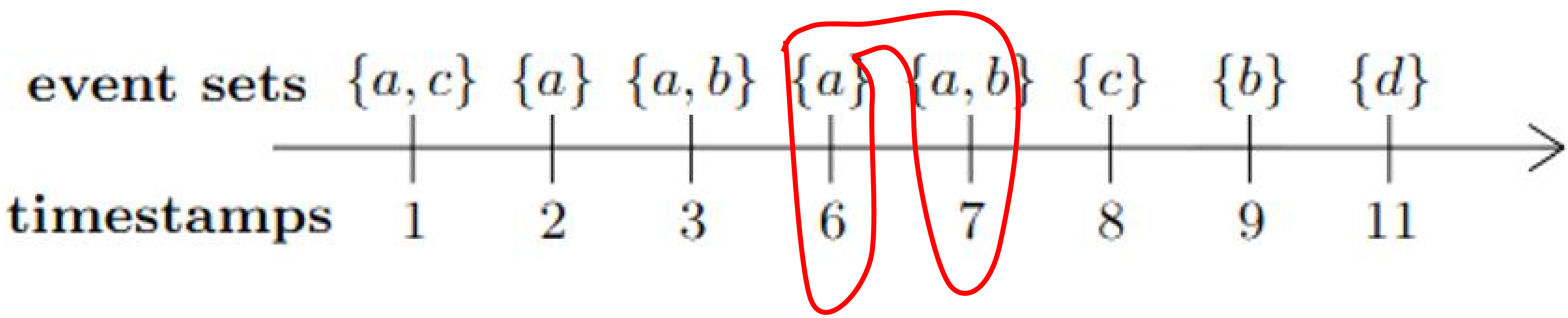
An **occurrence** of an **episode** $\propto = \langle X_1, X_2, \ldots X_p \rangle$ in a sequence $S = \langle (SEt_1, t_1), (SE_{t2}, t_2), \ldots, (SEt_n, t_n) \rangle$ is a time interval $[t_s, t_e]$ in which the episode appears.

Formally, it means that there exists integers

$$t_s = z_1 < z_2 < \ldots < z_w = t_e$$

such that $X_1 \subseteq SE_{z1,}\ X_2 \subseteq SE_{z2,} \ldots, X_p \subseteq SE_{zw}$

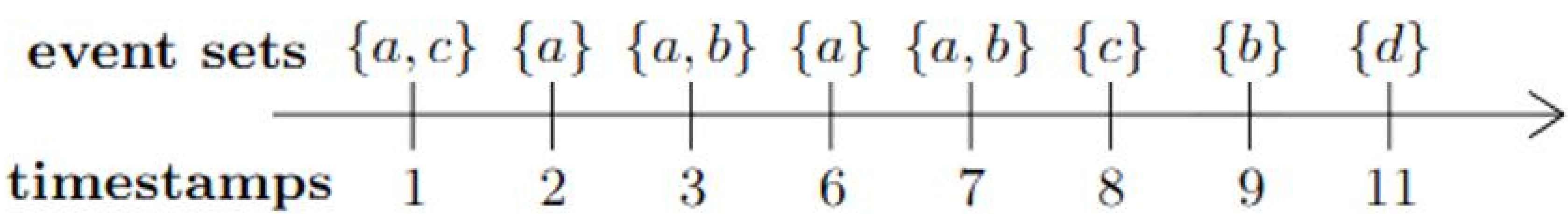
Example: The episode $\propto = \langle \{a\}, \{a, b\} \rangle$ has an occurrence in time interval $[6,7]$ of sequence:



All occurrences of an episode

The set of all occurrences of an episode $\propto$ in a sequence is denoted as $\text{occSet}(\propto)$.

Example: The set of all occurrences of episode $\propto = \langle\{a\}, \{a, b\}\rangle$ is $\text{occSet}(\propto) = \{[1,3], [1,7], [2,3], [2,7], [3,7], [6,7]\}$.



Head support

The **(head) support** an episode $\propto$ in a sequence is the number of distinct start times for its occurrences.

i.e. $sup(\propto) = |\{t_s | [t_s, t_e] \in occSet(\propto)\}|$

.

Example: The set of all occurrences of episode $\propto = \langle\{a\}, \{a, b\}\rangle$ is $occSet(\propto) = \{[1,3], [1,7], [2,3], [2,7], [3,7], [6,7]\}$.

Thus, $sup(\propto) = |\{1, 2, 3, 6\}| = 4$

Head support with window

- To avoid counting occurrences that span a very long period of times, we can introduce a user-defined parameter $\mathbf{winlen > 0}$.
- Then, we cound only occurrences that have a duration smaller than $\mathbf{winlen}$ time.

Example: Consider the episode $\propto = \langle\{\mathrm{a}\}, \{a, b\}\rangle$

If $\mathbf{winlen} = 6$, then

$\mathrm{occSet}(\propto) = \{[1,3], [1,7], [2,3], [2,7], [3,7], [6,7]\}.$

Thus, $\mathrm{sup}(\propto) = |\{1, 2, 3, 6\}| = 4$

Head support with window

- To avoid counting occurrences that span a very long period of times, we can introduce a user-defined parameter **winlen > 0**.
- Then, we cound only occurrences that have a duration smaller than **winlen** time.

Example: Consider the episode $\propto = \langle\{\mathrm{a}\}, \{a, b\}\rangle$

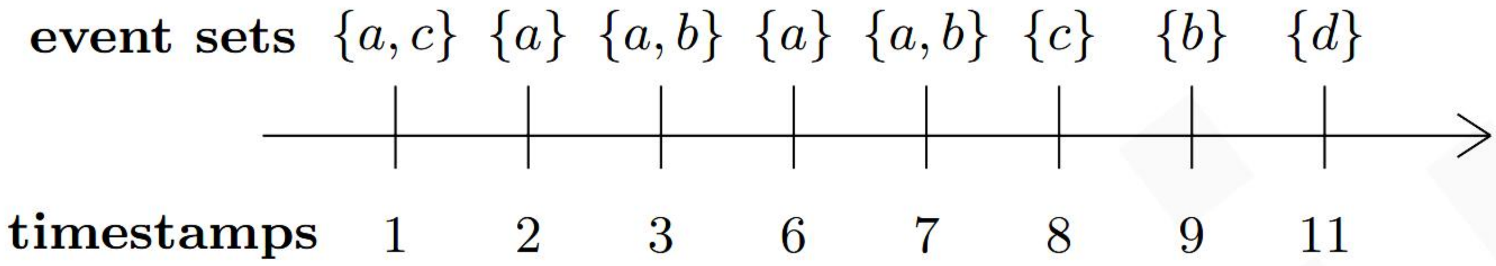
If $\mathbf{winlen} = \mathbf{2}$, then

$$\mathrm{occSet}(\propto) = \{[1{,}3], [1{,}7], [2{,}3], [2{,}7], [3{,}7], [6{,}7]\}.$$

Thus, $\mathrm{sup}(\propto) = |\{2, 6\}| = 2$

Frequent episode mining

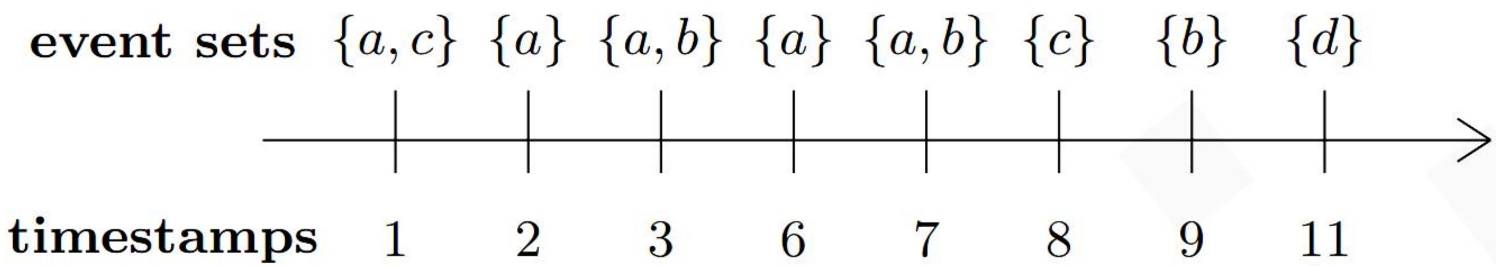
Input: **An event sequence**



Two parameters: $winlen = 2,\ minsup = 2$

Frequent episode mining

Input: **An event sequence**



Two parameters : $winlen = 2,\ minsup = 2$

Output: **All the frequent episodes** (**support** $\geq$ *minsup*)

Episode	Support
$\langle\{a,b\}\rangle$	2
$\langle\{a\},\{b\}\rangle$	2
$\langle\{a\},\{a,b\}\rangle$	2
$\langle\{a\},\{a\}\rangle$	3

Episode	Support
$\langle\{a\}\rangle$	5
$\langle\{b\}\rangle$	3
$\langle\{c\}\rangle$	2

How to find frequent episodes?

- There is a very large number of possible episodes.
- For only four items (a, b, c, d):

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

…

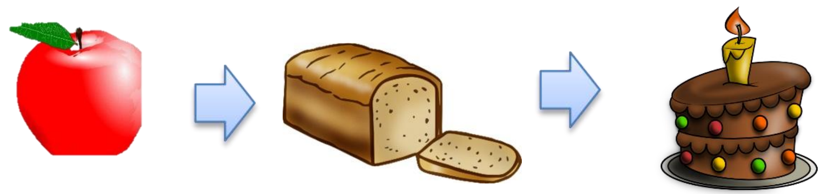
- Generally, if a sequence has **n** events, there could be up to $\mathbf{2^n - 1}$ distinct episodes.
- Thus, we need efficient algorithms that will not explore the whole search space to find the solution (the frequent episodes that we want to discover).

Algorithms

- Many algorithms such as:
  - **WINEPI (1995):** breadth-first search, window-based support
  - **MINEPI (1995)**: breadth-first search, minimal occurrences-based frequency
  - **EMMA** and **MINEPI+ (2008)**: depth-first search, head support
  - **TKE (2019)**: find the top-k most frequent episodes
  - **AFEM, MaxFEM (2022):** improved version of EMMA, can find the maximal episodes...
- They use different definitions of support, various data structures and search strategies

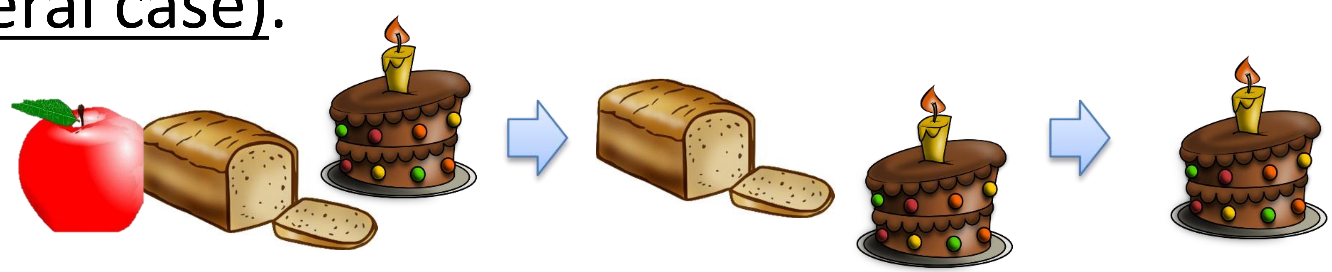
Algorithms

- Some algorithms can **only analyze <u>simple sequences</u>** (a sequence without simultaneous events).



$$\langle\{a\}, \quad \{\mathrm{b}\}, \quad \{\mathrm{c}\}\rangle$$

- Some algorithms can analyze <u>**complex sequences** (the general case)</u>.



$$\langle\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}, \quad \{\mathrm{b}, \mathrm{c}\}, \quad \{\mathrm{c}\}\rangle$$

THE EMMA ALGORITHM

Kuo-Yu Huang,Chia-Hui Chang (2008). **Efficient mining of frequent episodes from complex sequences**.Inf. Syst.33(1):96-114

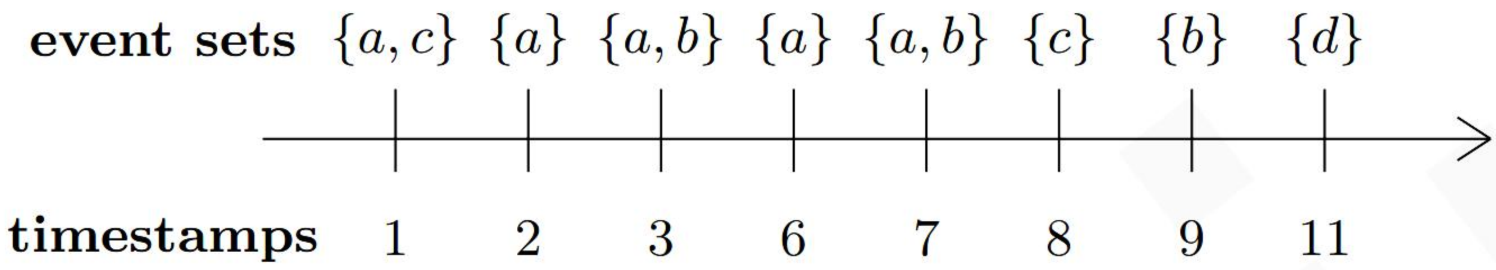
The EMMA algorithm

- Proposed Huang et al. (2008)
- The first algorithm to use the **head support**.
- An efficient algorithm
- Performs a depth-first search to find the frequent episodes.
- Uses a vertical data structure.

- We will look at how it works with an example.

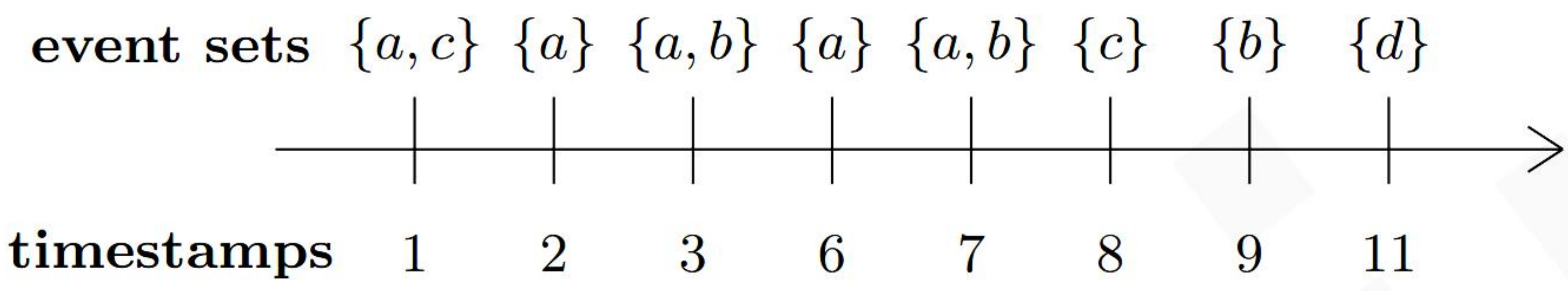
Example

Input: **An event sequence**



The parameters: $winlen = 2,\ minsup = 2$

Step 1: Scan the sequence fo count the support of each event

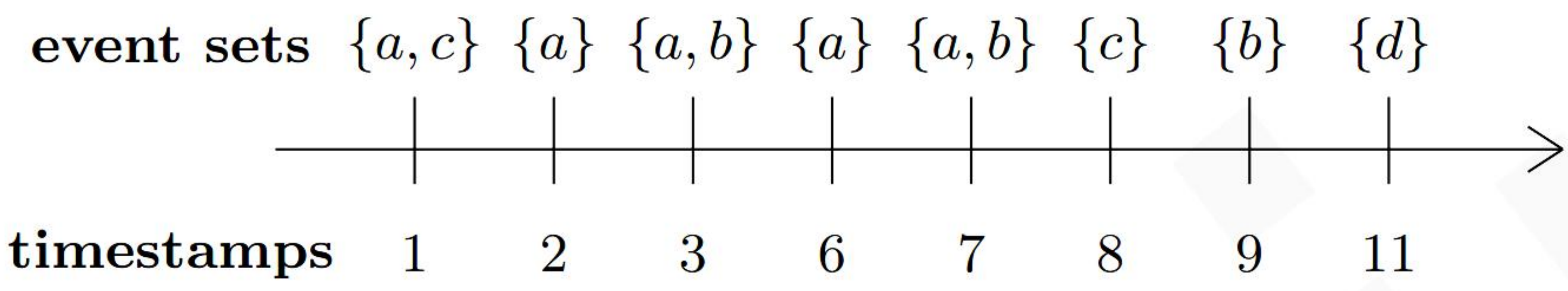


Events

Episode	Support
$\langle\{a\}\rangle$	5
$\langle\{b\}\rangle$	3
$\langle\{c\}\rangle$	2
$\langle\{\mathrm{d}\}\rangle$	1

Step 2: Keep only the frequent events

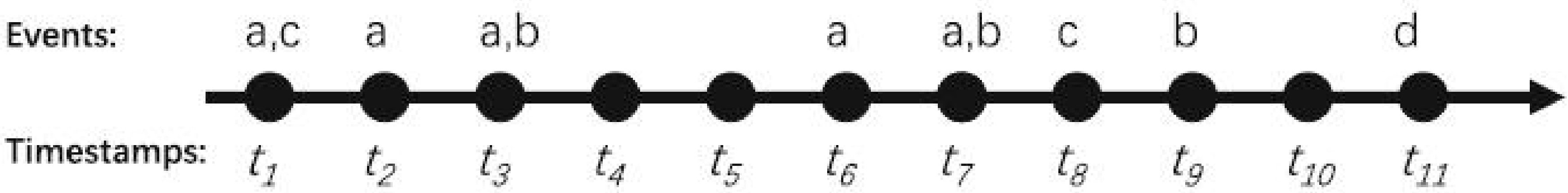
(events with a support $\geq$ minsup = 2)



Frequent events

Episode	Support
$\langle\{a\}\rangle$	5
$\langle\{b\}\rangle$	3
$\langle\{c\}\rangle$	2
~~$\langle\{d\}\rangle$~~	~~1~~

Step 3: Create the Location List of each frequent event

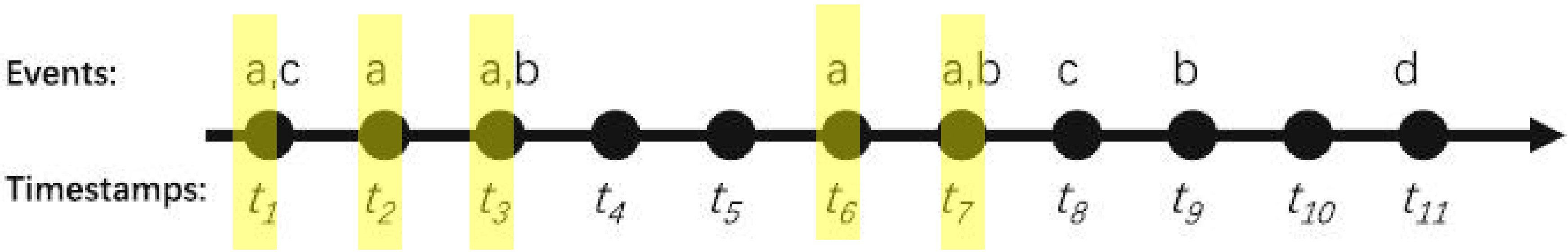


Create a *location list* for each frequent event

Episode	Support	location list
$\langle\{a\}\rangle$	5	$locList(a) = \{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$locList(b) = \{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$locList(c) = \{1, 8\}$

Note: for any episode α, we have $|locList(\alpha)| = \sup(\alpha)$

Step 3: Create the Location List of each frequent event

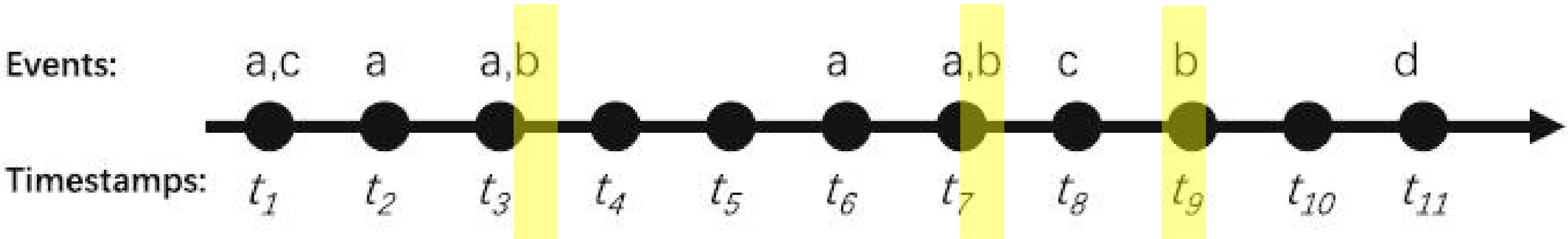


Create a *location list* for each frequent event

Episode	Support	location list
$\langle\{a\}\rangle$	5	$locList(a) = \{\mathbf{1},\ \mathbf{2},\ \mathbf{3},\ \mathbf{6},\ \mathbf{7}\}$
$\langle\{b\}\rangle$	3	$locList(b) = \{3,\ 7,\ 9\}$
$\langle\{c\}\rangle$	2	$locList(c) = \{1,\ 8\}$

Note: for any episode α, we have $|locList(\alpha)| = \sup(\alpha)$

Step 3: Create the Location List of each frequent event

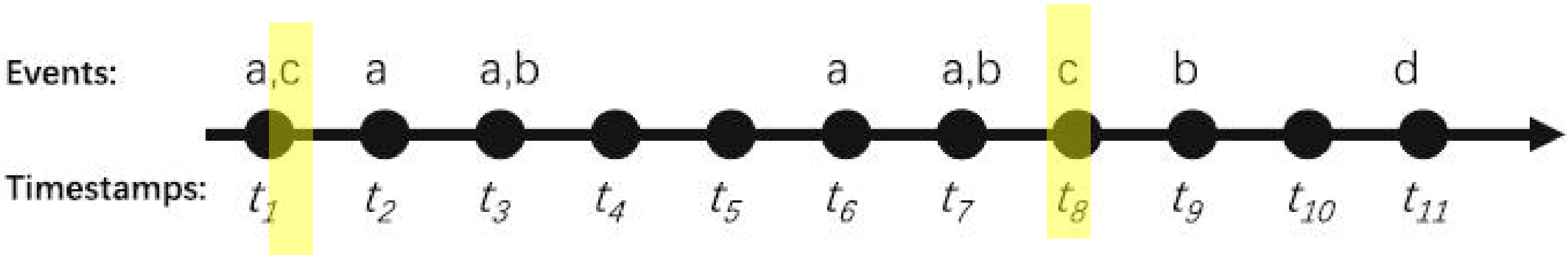


Create a *location list* for each frequent event

Episode	Support	location list
$\langle\{a\}\rangle$	5	$locList(a) = \{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$locList(b) = \{\mathbf{3}, \mathbf{7}, \mathbf{9}\}$
$\langle\{c\}\rangle$	2	$locList(c) = \{1, 8\}$

Note: for any episode α, we have $|locList(\alpha)| = \sup(\alpha)$

Step 3: Create the Location List of each frequent event



Create a *location list* for each frequent event

Episode	Support	location list
$\langle\{a\}\rangle$	5	$locList(a) = \{\mathbf{1, 2, 3, 6, 7}\}$
$\langle\{b\}\rangle$	3	$locList(b) = \{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$locList(c) = \{\mathbf{1, 8}\}$

Note: for any episode α, we have $|locList(\alpha)| = \sup(\alpha)$

Step 4: Find the Frequent Parallel Episodes

- Recursively combine frequent events to create **parallel episodes** with their locations lists.
- Keep only the parallel episodes that are frequent

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

Step 4: Find the Frequent Parallel Episodes

First, all the frequent events are frequent parallel episodes.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

→

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$

Step 4: Find the Frequent Parallel Episodes

Next, the algorithm combines frequent parallel episodes with frequent events to create more parallel episodes, and keep only the frequent episodes.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	{3, 7, 9}
$\langle\{c\}\rangle$	{1, 8}

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	3	{3, 7, 9}
$\langle\{c\}\rangle$	2	{1, 8}

- $\langle\{a\}\rangle$ and $\langle\{b\}\rangle$ are combined to get $\langle\{a, b\}\rangle$

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{\mathbf{3}, \mathbf{7}, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

$\cap$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, b\}\rangle$		

- The location list of $\langle\{a, b\}\rangle$ is the intersection of the locations lists of $\langle\{a\}\rangle$ and $\langle\{b\}\rangle$.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{\mathbf{3}, \mathbf{7}, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

$\cap$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, b\}\rangle$		$\{\mathbf{3}, \mathbf{7}\}$

- The support of $\langle\{\mathrm{a, b}\}\rangle$ is the number of elements in its location list. It is 2.
- Because $2 \geq minsup$, $\langle\{\mathrm{a, b}\}\rangle$ is frequent and it is kept.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{\mathbf{3}, \mathbf{7}, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

$\cap$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{\mathrm{a, b}\}\rangle$	**2**	$\{\mathbf{3,7}\}$

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{\mathrm{a}, \mathrm{b}\}\rangle$	2	$\{3,7\}$

- The algorithm continue combining frequent events with frequent parallel episodes to make more parallel episodes.
- $\langle\{a\}\rangle$ and $\langle\{c\}\rangle$ are combined to obtain $\langle\{a, c\}\rangle$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	3	{3, 7, 9}
$\langle\{c\}\rangle$	2	{1, 8}
$\langle\{a, b\}\rangle$	2	{3,7}
$\langle\{a, c\}\rangle$		

Frequent events

Episode	location list
$\langle\{a\}\rangle$	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	{3, 7, 9}
$\langle\{c\}\rangle$	{1, 8}

$\cap$

- The location list of $\langle\{a, c\}\rangle$ is the intersection of the locations lists of $\langle\{a\}\rangle$ and $\langle\{c\}\rangle$.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	{3, 7, 9}
$\langle\{c\}\rangle$	{**1**, 8}

∩

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	{**1**, 2, 3, 6, 7}
$\langle\{b\}\rangle$	3	{3, 7, 9}
$\langle\{c\}\rangle$	2	{1, 8}
$\langle\{a, b\}\rangle$	2	{3,7}
$\langle\{a, c\}\rangle$		{1}

- The support of $\langle\{a, c\}\rangle$ is the number of elements in its location list. It is 1.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{\mathbf{1}, 8\}$

$\cap$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{\mathbf{1}, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, b\}\rangle$	2	$\{3,7\}$
$\langle\{a, c\}\rangle$	1	$\{\mathbf{1}\}$

- The support of $\langle\{a, c\}\rangle$ is the number of elements in its location list. It is 1.
- Because $1 < minsup$, $\langle\{a, c\}\rangle$ is infrequent and it is discarded.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{\mathbf{1}, 8\}$

$\cap$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{\mathbf{1}, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, b\}\rangle$	2	$\{3,7\}$
~~$\langle\{a, c\}\rangle$~~	~~1~~	~~$\{\mathbf{1}\}$~~

- This process is repeated until no more parallel episodes can be generated

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{\mathbf{1}, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{\mathbf{1}, 8\}$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, \mathrm{b}\}\rangle$	2	{3,7}

- This process is repeated until no more parallel episodes can be generated
- Next $\langle\{b, c\}\rangle$ is created.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	{3, 7, 9}
$\langle\{c\}\rangle$	{1, 8}

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	3	{3, 7, 9}
$\langle\{c\}\rangle$	2	{1, 8}
$\langle\{a, b\}\rangle$	2	{3,7}
$\langle\{b, c\}\rangle$	0	{}

- This process is repeated until no more parallel episodes can be generated
- Next $\langle\{b, c\}\rangle$ is created.
- But it is infrequent.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

$\cap$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, b\}\rangle$	2	$\{3,7\}$
~~$\langle\{b, c\}\rangle$~~	~~0~~	~~$\{\}$~~

- Next $\langle\{a, b, c\}\rangle$ is created.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	$\{1, 8\}$

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, b\}\rangle$	2	$\{3,7\}$

- Next $\langle\{a, b, c\}\rangle$ is created.
- But it is infrequent.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	{**1**, 2, 3, 6, 7}
$\langle\{b\}\rangle$	{3, 7, 9}
$\langle\{c\}\rangle$	{**1**, 8}

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	3	{3, 7, 9}
$\langle\{c\}\rangle$	2	{1, 8}
$\langle\{a, b\}\rangle$	2	{3,7}
$\langle\{a, b, c\}\rangle$	0	{}

$\cap$

- This process is repeated until no more parallel episodes can be generated
- Next $\langle\{b, c\}\rangle$ is created.
- But it is infrequent.

Frequent events

Episode	location list
$\langle\{a\}\rangle$	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	{3, 7, 9}
$\langle\{c\}\rangle$	{1, 8}

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	{1, 2, 3, 6, 7}
$\langle\{b\}\rangle$	3	{3, 7, 9}
$\langle\{c\}\rangle$	2	{1, 8}
$\langle\{a, b\}\rangle$	2	{3,7}
~~$\langle\{a, b, c\}\rangle$~~	~~0~~	~~{}~~

It is the end of this step!

Frequent parallel episodes

Episode	Support	location list
$\langle\{a\}\rangle$	5	$\{1, 2, 3, 6, 7\}$
$\langle\{b\}\rangle$	3	$\{3, 7, 9\}$
$\langle\{c\}\rangle$	2	$\{1, 8\}$
$\langle\{a, \mathrm{b}\}\rangle$	2	{3,7}

Step 5: A unique identifier is given to each parallel episode

Frequent parallel episodes

Episode	Support	ID
$\langle\{a\}\rangle$	5	#1
$\langle\{b\}\rangle$	3	#2
$\langle\{c\}\rangle$	2	#3
$\langle\{a, \mathrm{b}\}\rangle$	2	#4

Then, the input sequence is re-encoded using these identifiers:

Frequent parallel episodes

Episode	Support	ID
$\langle\{a\}\rangle$	5	#1
$\langle\{b\}\rangle$	3	#2
$\langle\{c\}\rangle$	2	#3
$\langle\{a, \mathrm{b}\}\rangle$	2	#4

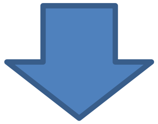
$$\mathrm{s} = \langle(\{a, c\}, 1), \{a\}, 2), (\{a, b\}, 3), (\{a\}, 6), (\{a, b\}, 7), (\{c\}, 8), (\{b\}, 9), (\{d\}, 11)\rangle$$

Then, the input sequence is re-encoded using these identifiers:

Frequent parallel episodes

Episode	Support	ID
$\langle\{a\}\rangle$	5	#1
$\langle\{b\}\rangle$	3	#2
$\langle\{c\}\rangle$	2	#3
$\langle\{a, \mathrm{b}\}\rangle$	2	#4

$$\mathrm{s} = \langle(\{a, c\}, 1),\ \{a\}, 2), (\{a, b\}, 3), (\{a\}, 6), (\{a, b\}, 7), (\{c\}, 8), (\{b\}, 9), (\{d\}, 11)\rangle$$



$$\mathrm{S} = \langle(\{\#1\#3\}, 1), (\{\#1\}, 2), (\{\#1, \#2, \#4\}, 3), (\{\#1\}, 6), (\{\#1, \#2, \#4\}, 7), (\{\#3\}, 8), (\{\#2\}, 9)\rangle$$

Then, the input sequence is re-encoded using these identifiers:

Frequent parallel episodes

Episode	Support	ID
$\langle\{a\}\rangle$	5	#1
$\langle\{b\}\rangle$	3	#2
$\langle\{c\}\rangle$	2	#3
$\langle\{a, \mathrm{b}\}\rangle$	2	#4

$$s = \langle(\{a,c\},1), \{a\},2), (\{a,b\},3), (\{a\},6), (\{a,b\},7), (\{c\},8), (\{b\},9), \sout{(\{d\},11)}\rangle$$

Note: By this process, infrequent events are ignored

$$S = \langle(\{\#1\#3\},1), (\{\#1\},2), (\{\#1,\#2,\#4\},3), (\{\#1\},6), (\{\#1,\#2,\#4\},7), (\{\#3\},8), (\{\#2\},9)\rangle$$

At the same time, a «**bound-list**» structure is created for each parallel episode:

Frequent parallel episodes

Episode	Support	ID	Bound-list
$\langle\{a\}\rangle$	5	#1	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	#2	
$\langle\{c\}\rangle$	2	#3	
$\langle\{a, b\}\rangle$	2	#4	

The bound-list of episode $\langle\{\mathbf{a}\}\rangle$ indicates a list of time intervals where $\langle\{\mathbf{a}\}\rangle$ appears in the input sequence

$$S = \langle(\{\#1\#3\}, 1), (\{\#1\}, 2), (\{\#1, \#2, \#4\}, 3), (\{\#1\}, 6), (\{\#1, \#2, \#4\}, 7), (\{\#3\}, 8), (\{\#2\}, 9)\rangle$$

At the same time, a «**bound-list**» structure is created for each parallel episode:

Frequent parallel episodes

Episode	Support	ID	Bound-list
$\langle\{a\}\rangle$	5	#1	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	#2	[3,3], [7,7], [9,9]
$\langle\{c\}\ \rangle$	2	#3	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	#4	[3,3], [7,7]

$$\mathrm{S} \ = \ \langle(\{\#1\#3\}, 1), (\{\#1\}, 2), (\{\#1, \#2, \#4\}, 3), (\{\#1\}, 6), (\{\#1, \#2, \#4\}, 7), (\{\#3\}, 8), (\{\#2\}, 9)\rangle$$

Step 6: Find Frequent Composite episodes

The frequent parallel episodes that we have until now are also composite episodes:

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

The algorithm recursively appends a parallel episode to a composite episode to create larger composite episode.
This process is called **serial extension** -->

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\ \rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\ \rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{a\}.\{\mathrm{a}\}\rangle$		

$\cap$

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{a\}.\{\mathrm{a}\}\rangle$		[1,2], [2,3], [6,7]

$\cap$

The bound list of $\langle\{a\}.\{\mathrm{a}\}\rangle$ is created by intersecting that of $\langle\{a\}\rangle$ and $\langle\{a\}\rangle$.

Note: Because $\mathrm{winlen} = 2$, some intervals are not considered like [1,3] and [1,6]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{a\}.\{\mathrm{a}\}\rangle$	**3**	[1,2], [2,3], [6,7]

The size of the bound list of $\langle\{a\}.\{\mathrm{a}\}\rangle$ is 3. Thus, its support is 3 and it is frequent!

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$		

$\cap$

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	**2**	[2,3],[6,7]

$\cap$

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]
$\langle\{a\}.\{\mathrm{c}\}\rangle$	**1**	[7,8]

$\cap$

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]
~~$\langle\{a\}.\{\mathrm{c}\}\rangle$~~	~~1~~	~~[7,8]~~

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]
$\langle\{a\}.\{\mathrm{a}, \mathrm{b}\}\rangle$	**2**	[2,3],[6,7]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]
$\langle\{a\}.\{\mathrm{a}, \mathrm{b}\}\rangle$	**2**	[2,3],[6,7]

Parallel episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]

Composite episodes

Episode	Support	Bound-list
$\langle\{a\}\rangle$	5	[1,1], [2,2], [3,3], [6,6], [7,7]
$\langle\{b\}\rangle$	3	[3,3], [7,7], [9,9]
$\langle\{c\}\rangle$	2	[1,1], [8,8]
$\langle\{a, \mathrm{b}\}\rangle$	2	[3,3], [7,7]
$\langle\{\mathrm{a}\}.\{\mathrm{a}\}\rangle$	3	[1,2], [2,3], [6,7]
$\langle\{a\}.\{\mathrm{b}\}\rangle$	2	[2,3],[6,7]
$\langle\{a\}.\{\mathrm{a}, \mathrm{b}\}\rangle$	**2**	[2,3],[6,7]

Then, this process continue recursively to try:

$\langle\{b\}\rangle$

$\langle\{b\}, \{\mathrm{a}[\rangle$

$\langle\{b\}, \{\mathrm{b}[\rangle$

$\langle\{b\}, \{\mathrm{c}[\rangle$

$\langle\{b\}, \{\mathrm{a}, \mathrm{b}]\rangle$

$\langle\{\mathrm{a}\}, \{\mathrm{a}\}, \{\mathrm{a}\}\rangle$

....

Final result

The result is this set of **frequent (composite) episodes:**

Episode	Support
$\langle\{a\}\rangle$	5
$\langle\{b\}\rangle$	3
$\langle\{c\}\ \rangle$	2
$\langle\{a, \mathrm{b}\}\rangle$	2
$\langle\{a, b\}\rangle$	2
$\langle\{a\}, \{b\}\rangle$	2
$\langle\{a\}, \{a, b\}\rangle$	2
$\langle\{a\}, \{a\}\rangle$	3

Observations

- EMMA first finds parallel episodes and then combines them to make composite episodes.
- EMMA reduces the search space by not extending the infrequent episodes.
- Generally, EMMA is a quite fast algorithm.
- An improved version is called AFEM.

DISCOVERING MAXIMAL EPISODES

(THE MAXFEM ALGORITHM)

Fournier-Viger, P., Nawaz, M. S., He, Y., Wu, Y., Nouioua, F., Yun, U. (2022). **MaxFEM: Mining Maximal Frequent Episodes in Complex Event Sequences.** Proc. of the 15th Multi-disciplinary International Conference on Artificial Intelligence (MIWAI 2022), pp. 86-98, Springer LNAI.

Limitation of FEM

- FEM algorithms can find **millions of episodes**!
- For each frequent episode, all the sub-episodes are often also frequent.

milk → bread → orange,

milk → bread,

milk → orange

bread → orange

milk

bread

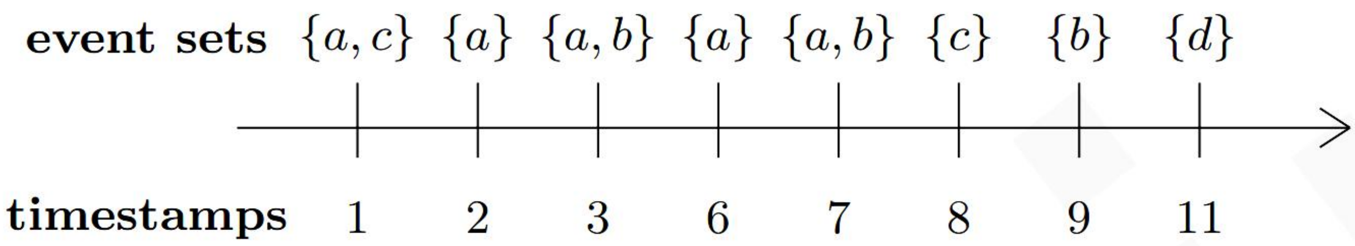
orange

A solution

- Discover only the **maximal episodes**.
- A frequent episode α is **maximal** if it is not a subsequence of another frequent episode β.
- **Benefit**: much less episodes and most of the information is preserved.
- How to deal with the more general case of finding maximal episodes in a complex sequence?

Example

Event sequence



Parameters

$winlen = 2$

$minsup = 2$

Frequent episodes

Episode	Support	Maximal?
$\langle\{a, b\}\rangle$	2	No
$\langle\{a\}, \{b\}\rangle$	2	No
$\langle\{a\}, \{a, b\}\rangle$	2	**Yes**
$\langle\{a\}, \{a\}\rangle$	3	No
$\langle\{a\}\rangle$	5	No
$\langle\{b\}\rangle$	3	No
$\langle\{c\}\rangle$	2	**Yes**

The MaxFEM algorithm

- An algorithm: **MaxFEM** (**Maximal Frequent Episode Mining**)
  - To find the *maximal* frequent episodes
  - Extends the EMMA algorithm
  - Applies techniques to keep only maximal episodes and some optimizations

The process is similar to EMMA

- **Step 1**: Count the support of each event
- **Step 2**: Keep only the frequent events
- **Step 3**: Create the location list of each frequent event
- **Step 4**: Find frequent parallel episodes
- **Step 5**: Re-encode the input sequence and create bound-lists
- **Step 6**: Find composite episodes (this step is modified)

Step 6: Find Frequent Composite episodes

- During the search, to find the maximal episodes:
  - A set W stores the episodes that are currently maximal.
  - When a new episode α is found:
    - **Sub-episode checking:**
      If α is included in an episode β already in W,
      then α is not added to W.
    - **Super-episode checking:**
      If an episode β from W is included in α,
      then β is removed from W

Step 6: Finding Frequent Composite episodes

Result:

Maximal frequent episodes

Episode	Support
$\langle \{c\} \rangle$	2
$\langle \{a\}, \{a, b\} \rangle$	2

Optimization 1
EFE: Efficient Filtering of Non-maximal episodes

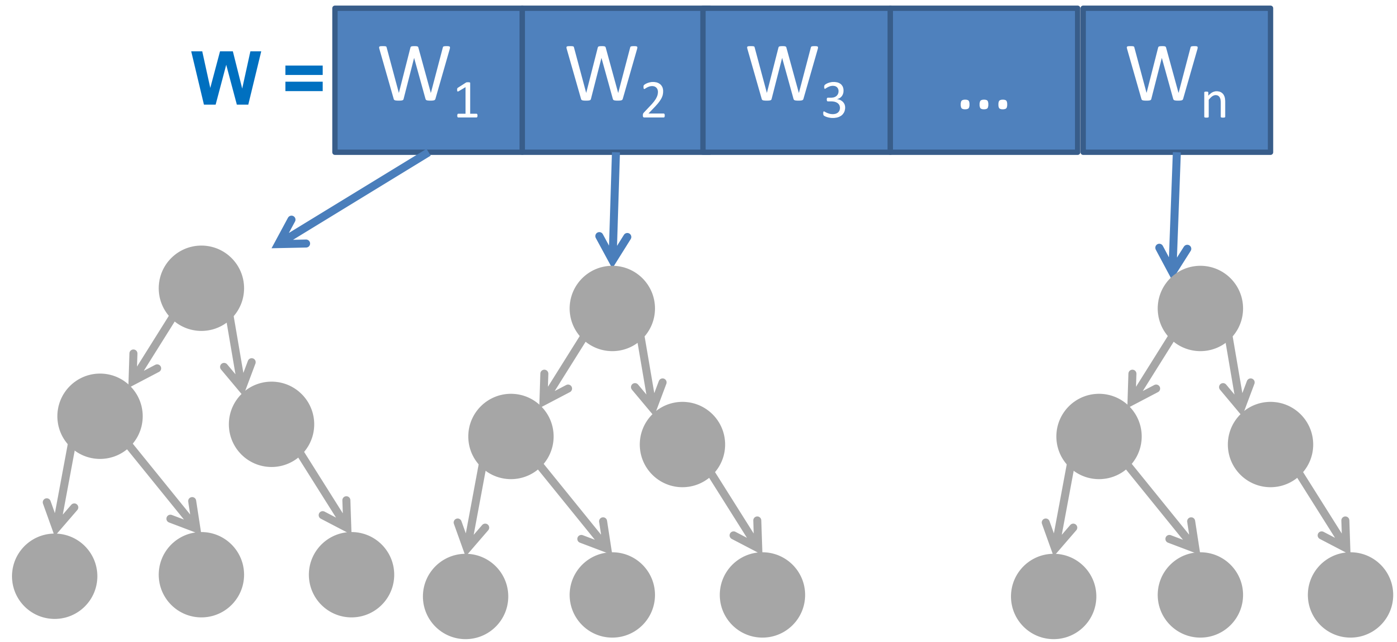
MaxFEM implements **W** as a List of heaps

W =	W_1	W_2	W_3	…	W_n

The **k-th** list entry contains episodes of size **k**

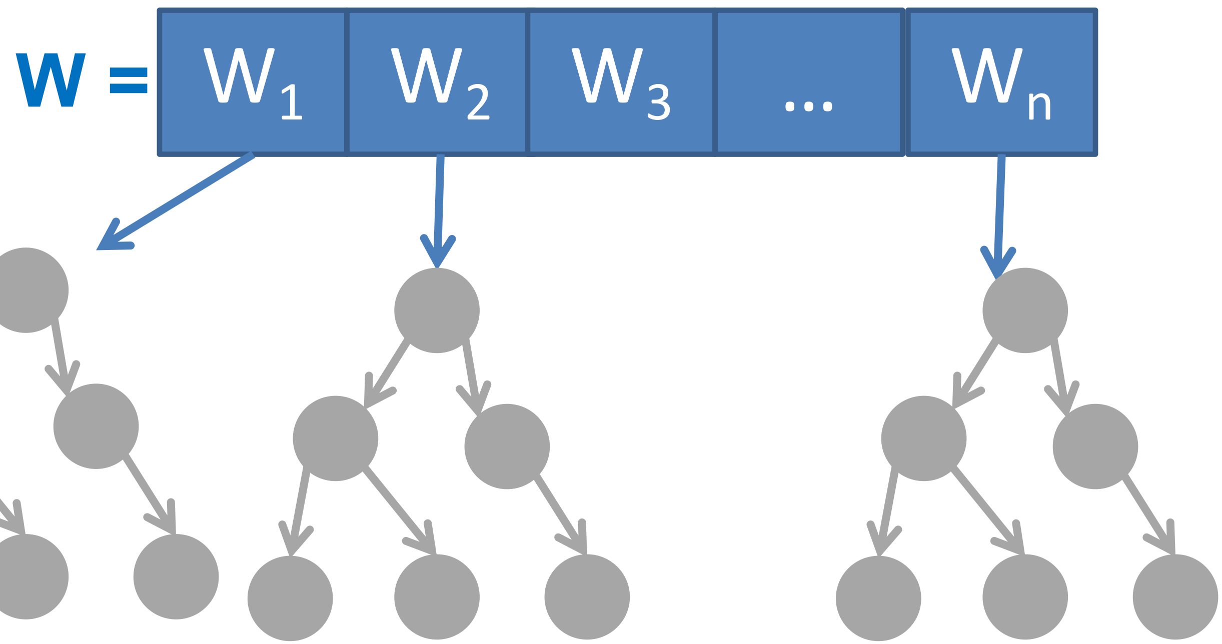
This allows to perform super-episode checking and sub-episode checking only with smaller and larger patterns

Optimization 1
EFE: Efficient Filtering of Non-maximal episodes



- The **sum of events** in each pattern is calculated.
- Each **heap** orders patterns by decreasing sum of events.
- For each pattern S_a found and pattern S_b in Z_k, if $sum(S_a) < sum(S_a)$ we don't need to perform super-episode checking with W_b and any following patterns in W_k.
- Similar for sub-episode-checking

Optimization 1
EFE: Efficient Filtering of Non-maximal episodes



- **Support check optimization**:
  - A pattern cannot be contained in another pattern if its support is smaller.
  - A pattern cannot contain another pattern if its support is larger.

Two more optimizations

- **Strategy 2. Skip Extension checking (SEC)**
  - If a frequent episode ***ep*** is extended by serial extension to form another frequent episode, then it is unnecessary to do super-episode and sub-episode checking for ***ep*** because it is **not maximal.**

- **Strategy 3. Temporal pruning (TP).**
  - When creating a bound-list, if at any point the number of remaining elements is not enough to satisfy *minsup*, the construction of the bound-list is stopped.

Experiments

- **Two benchmark datasets:**

Dataset	Avg. Sequ. Len.	#Events	#Sequences	Density(%)
Kosarak	8.1	41,270	990,000	0.02
Retail	10.3	16,470	88,162	0.06

- **Compared algorithms:**
  - MaxFEM
  - EMMA
- **Setup:**
  - Java, Windows 11, laptop with Core i7-8565U processor, 16GB RAM
  - **Experiment**: *Winlen* $\in \{5, 10, 15\}$ and *minsup* is varied
  - A 300 second time limit

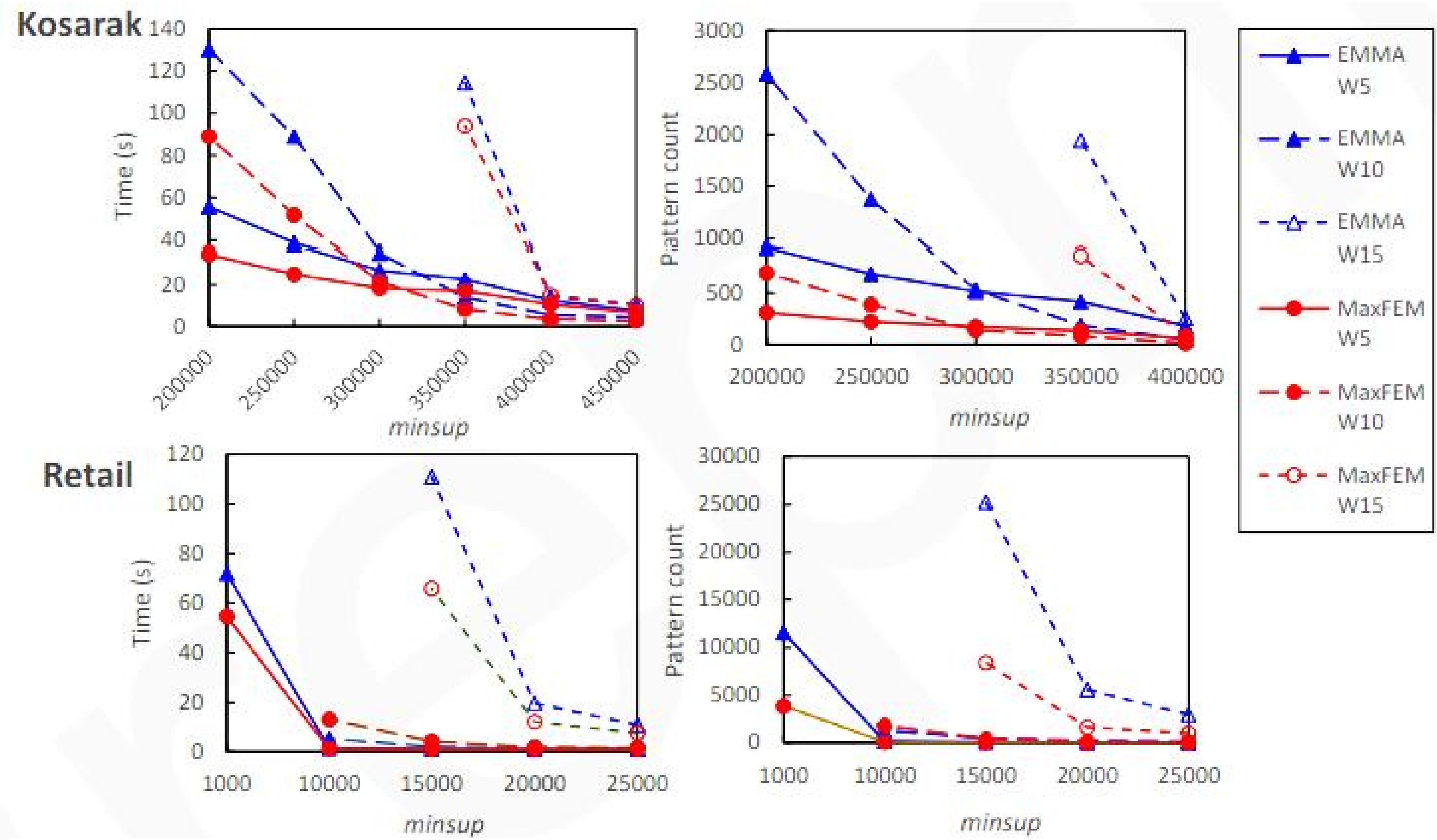


Fig. 2: Comparison of runtime and pattern count

Conclusion on maximal episodes

- Finding maximal episodes can reduce the number of episodes presented to the user
- **MaxFEM** is an algorithm for **maximal episode mining** for the general case of a **complex event sequence** and with the **head frequency support** function
- A version of MaxFEM to find all frequent episodes is called **AFEM.**
- There also exists other algorithms to find other compact representations of episodes such as closed episodes.

EPISODE RULE MINING

Mannila, H., Toivonen, H., Verkamo, A.I.**: Discovering frequent episodes in sequences**. In: Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining

Ao, X., Luo, P., Wang, J., Zhuang, F., He, Q.: **Mining precise-positioning episode rules from event sequences**. IEEE Transactions on Knowledge and Data Engineer_x0002_ing 30(3), 530–543 (2017)

Fahed, L., Brun, A., Boyer, A.: **Deer: Distant and essential episode rules for early prediction**. Expert Systems with Applications 93, 283–298 (2018)

Fournier-Viger, P., Chen, Y., Nouioua, F., Lin, J. C.-W. (2021). **Mining Partially-Ordered Episode Rules in an Event Sequence.** Proc. of the 13th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2021), Springer LNAI, pp 3-15

Ouarem, O., Nouioua, F., Fournier-Viger, P. (2021). **Mining Episode Rules From Event Sequences Under Non-Overlapping Frequency**. Proc. 34th Intern. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA AIE 2021), Springer LNAI, pp. 73-85

Chen, Y., Fournier-Viger, P., Nouioua, F., Wu, Y.. (2021). **Sequence Prediction using Partially-Ordered Episode Rules**. Proc. 4th International Workshop on Utility-Driven Mining (UDML 2021), in conjunction with the ICDM 2021 conference, IEEE ICDM workshop proceedings
....

Episode Rule Mining

- Applying an algorithm such as EMMA, TKE or MINEPI will find frequent episodes.
- These patterns may be interesting because they appear frequently in data.
- However, they may be of limited use to do prediction.
- **A solution**: Combine episodes to create rules, called **episode rules**.

Episode Rule Mining

- **Basic idea:** Take pairs of frequent episodes $\propto$ and β and try to combine them to generate a rule of the form:

  $\propto \rightarrow \beta$

- For example: $bread \rightarrow milk, noodles$

  $support = 100 \quad confidence = 75\%$

This rule means that someone buying **bread** will 75% of the time buy **milk** and **noodles** afterward.

DISCOVERING THE TOP-K MOST FREQUENT EPISODES

Fournier-Viger, P., Wang, Y., Yang, P., Lin, J. C.-W., Yun, U. (2020). **TKE: Mining Top-K Frequent Episodes**. Proc. 33rd Intern. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA AIE 2020), Springer LNCS , pp. 832-845.

Limitation of FEM

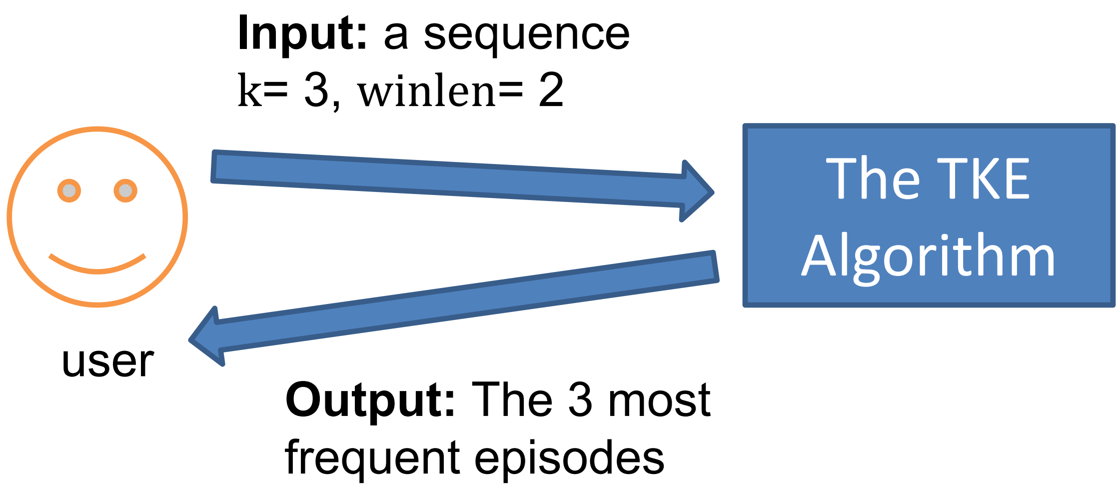
- To find frequent episodes, it is necessary to set a parameter called the minimum support threshold (minsup).

– This threshold is usually set by trial and error.

- Setting the threshold is unintuitive.
  - If the value is too **high**, no frequent episodes are found.
  - If the value is too **low**, millions of episodes may be found, and runtime and memory usage may greatly increase.

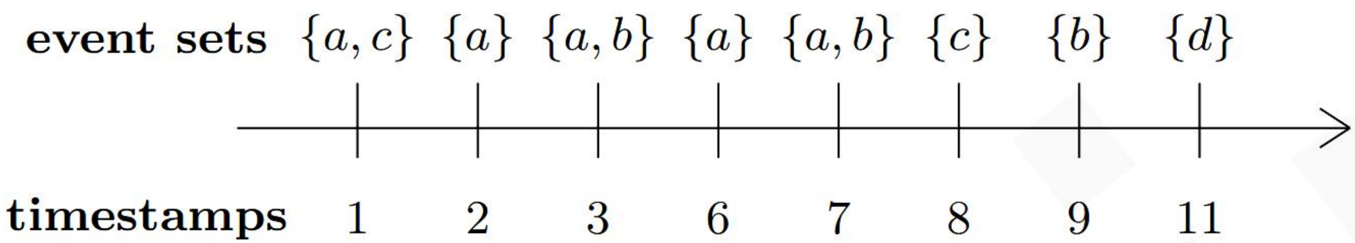
A solution

- The **TKE** algorithm to discover the **top-k most frequent episodes**.
- The user sets a parameter **k** instead of **minsup**.
- The algorithm directly returns the **top-k episodes**.



Example

Event sequence



Parameters

$winlen = 2$

$k = 3$

Top-k episodes

Episode	Support
$\langle\{a\},\{a\}\rangle$	3
$\langle\{a\}\rangle$	5
$\langle\{b\}\rangle$	3

The TKE algorithm

- **TKE (Top-K Episode mining)**
  - To find the top-k frequent episodes
  - Extends the EMMA algorithm
  - **Key idea**: start to search using an internal minsup value of 1, and then gradually increase the threshold when k episodes have been found.
  - Several optimizations

HIGH-UTILITY EPISODE MINING

Wu, C., Lin, Y., Yu, P.S., Tseng, V.S.: **Mining high utility episodes in complex event sequences**. In: Proc. 19th ACM SIGKDD Int. Conf. on Knowl. Discovery. pp. 536–544 (2013)

Guo, G., Zhang, L., Liu, Q., Chen, E., Zhu, F., Guan, C.: **High utility episode mining made practical and fast**. In: Proc. 10th Int. Conf. on Advanced Data Mining and Applications. pp. 71–84 (2014)

Rathore, S., Dawar, S., Goyal, V., Patel, D.: **Top-k high utility episode mining from a complex event sequence**. In: Proc. 21st Int. Conf. on Management of Data. pp. 56–63 (2016)

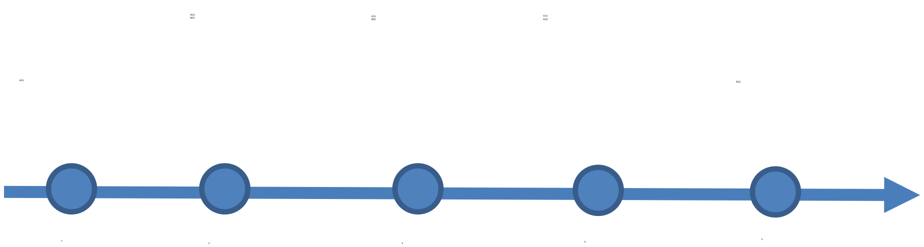
Fournier-Viger, P., Yang, P., Lin, J. C.-W., Yun, U. (2019). **HUE-SPAN: Fast High Utility Episode Mining**. Proc. 14th Intern. Conference on Advanced Data Mining and Applications (ADMA 2019) Springer LNAI, pp. 169-184.

... etc.

High Utility Episode Mining

Input:

A event sequence



A unit profit table

Event	A	B	C	D
Profit	2	1	3	2

Output:

High utility episodes (with utility $\geq \boldsymbol{minUtil}$ & duration $\leq \boldsymbol{maxDur}$)

If set $\boldsymbol{minUtil}$ = 15 and $\boldsymbol{maxDur}$ = 3, HUEs are:

Episode	Minimal Occurrences	Utility
< (BC), (AC), (D) >	[3, 5]	15
<(B), (BC), (AC)>	[2, 4]	15
<(BD), (BC), (AC)>	[2, 4]	17
<(D), (BC), (AC)>	[2, 4]	15

CONCLUSION

Conclusion

- There are many algorithms for **episode mining** and several variations of this task.
- **Episode mining** and **episode rule mining** are taks sfor analyzing a single sequence of events with timestamps.
- This is different from **sequential pattern mining** and **sequential rule mining**, which focus on analyzing **multiple sequences** (and that typically do not have timestamps).

Source code and datasets available in the SPMF open-source data mining library
http://www.philippe-fournier-viger.com/spmf/