# Pattern Mining project

## Pre-requisite:

- Skill in coding languages (java or python).
- Basics in pattern mining

## Achieved skills :

- Develop some skills in data mining field.
- Improve skills in machine learning and high-performance computing
- Application on industry 4.0 (predictive maintenance)

## Context :

Sequential pattern mining is the discovery of subsequences that are frequent in a set of sequences. The process is similar to the frequent itemset mining[1] except that the input database is ordered. As output of a sequential pattern mining algorithm, it generates a set of frequent sequential patterns, which are sub-sequences that have a frequency in the database greater than or equal to the user-specified minimum support.

Let the data set shown in Table 1 where events are accompanied by instants of occurrence in each tuple.

Table 1: Data set of sequences (pairs of event/instant)

| Sequence Id | Events |
|---|---|
| 1 | (A,0), (B,5), (C,7) |
| 2 | (A,2), (B,3), (C,7) |

We can note that, for a fixed threshold equal to 1, the pattern < A, B,C > is considered as frequent because its **support** (the number of occurrence in the database) is equal to 2.

## Problematic and Goal:

Let us assume the example given in the Table 1. < A,B,C > is considered as frequent sequential pattern. It shows that events A, B and C occurred frequently in a sequence manner, but without providing any additional information about the gap between them. For instance, we do not know when B would happen, knowing that A already did. Therefore, we ask you to provide a richer pattern where time constraints are considered. In our data set example, we can deduce that A, B and C occur sequentially, and that B occurs after A at least after one instant and at most after 5 instants, while C occurs after B in the interval [2, 4] of instants. We

[1] reader may visit this link for further details :https://www.kdnuggets.com/2016/10/association-rule-learning-concise-technical-overview.html

represent our pattern as A[1,5]B and B[2,4]C. It is a direct graph where nodes are events and vertices are the instant intervals, denoted by time constraints as shown in Figure 1.

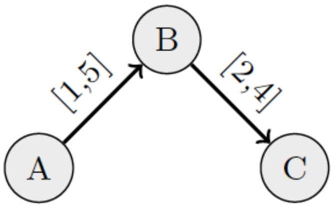

*Figure 1*

Formally,

**Definition (Event)** An event is a couple $(e, t)$ where $e \in \mathbb{E}$ is the type of the event and $t \in \mathbb{T}$ is its time.

A sequence contains timestamped events, which appear according to their time of occurrence.

**Definition (Sequence)** Let $\mathbb{E}$ be a set of event types, and $\mathbb{T}$ a time domain such that $\mathbb{T} \subseteq \mathbb{R}$. $\mathbb{E}$ is assumed totally ordered and is denoted $<_E$ . A sequence is a couple $< SID, < ( e1, t1), (e2\}, t2), \ldots, (en, tn) > >$ such that $SID$ is the index of the sequence and $< ( e1, t1), (e2\}, t2), \ldots, (en, tn) >$ is a sequence of events. For all $i, j \in [1, n]$, $i < j \Rightarrow ti < tj$. If $ti = tj$ then $ei <_E ej$.

To describe the interval time that separates two events, we introduce the notion of temporal constraint defined as follows:

**Definition (Temporal constraint)** A time constraint is a quadruplet $(e1, e2, t-, t+)$, denoted $e1[t-, t+]e2$, where $e1, e2 \in E$ and $e1 <_E e2$ , and $t-, t+ \in \mathbb{T}$.
A time constraint $e1[t-, t+]e2$ is said satisfied by a couple of events $((e, t), (e', t')), e <_E e'$ iff $e = e1, e' = e2$ and $t' - t \in [t-, t+]$.
We say that $e1[a, b]e2 \subseteq e'1[a', b']e'2$ iff $[a, b] \subseteq [a', b']$

**Definition (Sequential pattern)** A sequential pattern is a pair $C = (\mathcal{E}, \mathrm{T})$ such that:

1. $\mathcal{E} = \{e1, \ldots, en\}$, where $\forall i, ei \in \mathcal{E}$ and $ei <_E ei + 1$,
2. $\mathrm{T} = \{tij\}1 \leq i < j \leq |\mathcal{E}|$ is a set of temporal constraints on $\mathcal{E}$ such that for all pairs $(i, j)$ satisfying $i < j$, $tij$ is denoted by $ei[t - ij, t + ij]ej$ .

## 1. Task mandatory

In this part, you have to implement a code that generates these frequent sequential patterns (closed or maximal) that have a support greater than or equal to the threshold fixed by the user. To do so, you may generate as a first step frequent sequential patterns that have a frequency greater than or equal to the threshold fixed by the user.

## 2. Data engineering task

In this part, you have to use the mined sequential patterns to predict when the next event of a sequence. The approach should be studied and compared to state-of-the-art approaches using the quality measure discussed in the lecture.

## Evaluation

The project code, report (pdf format) as well as the presentation slides should be submitted through moodle. The report should detail the developed approach and an exhaustive experimentation section. The project defense is scheduled for the last lecture day. Be sure to submit you document at least 3 days before (April 12th 1PM).

Several criteria for the evaluation:

- The quality and the performance of the approach
- Optimization of the code to improve the performance.
- The quality of the presentation in the project defense.
- The quality of the report.
- Tasks share inside the group.