Mining temporal data – Frequent sequences

Algorithmic Data Analysis – Coding Assignment 2

Giulia Ortolani

April 3, 2024

Resources: No resources
Collaborations: No collaborations

The Generalized Sequential Pattern Mining (GSP) algorithm is an apriori-like algorithm for solving sequential pattern mining, here I try to implement it in Python and use it for mining frequent patterns from discrete sequences. The algorithm is then applied to the CRSW dataset, which contains two months worth of weather data in Kuopio (January–February 2019).

GSP algorithm implementation idea

The implemented algorithm takes a dataset (list of itemsets) and a minimum support threshold. It first identifies frequent items (items with support greater than or equal to the minimum support threshold). Then, starting from $k = 1$, it iteratively generates candidate sequences of length $k + 1$ by joining pairs of sequences from the frequent patterns of length k. The support of these candidates is calculated and only those with support greater than or equal to the minimum support threshold are retained. This process continues until no more frequent patterns can be found. The output of the code is a dictionary where the keys are the frequent patterns and the values are their corresponding support counts.
The chosen object type for the sequences of the dataset is OrderedSet[1]. *[I'm still working in the code trying to figure out where is the problem, probably is in the function for counting support]*

Application to the CRSW dataset

The weather time series analysis using the Generalized Sequential Pattern (GSP) mining algorithm has revealed interesting patterns in the occurrences of clouds (C), precipitation (R), sunshine (S), and wind (W).

Setting different minimum support threshold

Setting different values for max gap and max span

[1] `https://pypi.org/project/ordered-set/`