

Implementation of the multi-temporal mining method

Richárd Károly, János Abonyi

September 29, 2014

Chapter 1

Introduction

This section gives a brief introduction of the multi-temporal mining algorithm proposed by Xiaoxiao Kong, Qiang Wei, and Gouqing Chen[1].

In real world, scenarios objects' states can be caused by many events. These events may result the same or different state of each object under various time periods. To generate these relationships between event sequences and states we must have a tool capable of handling the relations between them. These relations need to occur in a specified time window, in which we tell if two events correlate. Four different relations are interpreted: *before*, *during*, *overlaps*, *equals*.

The mining process consists of four steps. First of all we need to read the input database and separate each state and allocate the events in conjunction with them. Then we can tell if these states are frequent or not by counting their event base. Subsequently, we can generate the patterns consisting of $k+1$ elements, step by step. After an iteration in which we couldn't find any frequent pattern the mining process is stopped and the output file is written. The mining process is shown as Fig. 1.



Figure 1.1: The multi-temporal mining process.

Our implementation differentiates from the algorithm proposed by the authors. In order to reach the fastest mining our candidate generation is based on the apriori principle. We also changed the way the frequent states are stored under the mining process. Only their effective time interval is stored to reduce the memory requirement.

multiTemporal

Purpose

Generate all the frequent multi temporal patterns occurring in the given input temporal event database or log.

Synopsis

`multiTemporal(Input_file, Supp_threshold, Window, Output_file)`

Description

The **multiTemporal** function produces an output `.xls` file from the given input `.xls` file, **support threshold** and **window**. It generates all the frequent patterns contained in the input *temporal* database.

Algorithm

The **multiTemporal** function uses the algorithm proposed by Xiaoxiao Kong, Qiang Wei, and Gouqing Chen with the stated modifications.

Bibliography

- [1] Xiaoxiao Kong, Qiang Wei, and Gouqing Chen, *An approach to discovering multi-temporal patterns and it's application to financial databases*. Information Sciences 180, 873-885, 2010.