CS 4: Reduction of computation time in Big Data analysis by GUHA method

Addressed Problems: The GUHA method is a descriptive method to data mine large data matrices. A computer implementation LISpMiner, developed and maintained at the University of Economics in Prague, can handle matrices of typically tens of thousands of columns and tens of thousands of rows, whose cells can contain all kinds of symbols.

GUHA is a particular logic based data mining method. Thus, if a user is able to ask relevant question related to the data, then the question can be written in a symbolic way by GUHA language, and LISpMiner will find all answers, that are supported by the data, i.e. are TRUE.

In practice, LISpMiner goes through several constancy tables, usually even millions, and outputs the true ones. So far, so good. However, as the data matrix size grows, problems arise as computation time by LISpMiner increases. This raises the question of how to reduce the computation time.

Existing Solution: Until now, the problem has been partially solved by two means.

Firstly, since GUHA has a firm logic foundations, and there are plenty of possible statements (whose truth value should be tested), which are logically dependent, we can omit those statements that are true by pure logic reasons (e.g. if A implies B is true, then also A implies B or C is true, so there is no need to test the last statement). Such matters are taking into consideration all the time when the software LISpMiner has been and is written.

Another solution was a grid system invented at TUT some years ago. Since computation can be done in parallel computing, the LISpMiner software was recoded such that more than 400 desktop computers were connected to the network called Techila at TUT. This reduced the LISpMiner calculation times by as much as to 1% of the original. The solution is universal. i.e. also implementable in other environments, and done among others at Prague university of Economics. However, even the Techila solution is not always enough.

Proposed Solution: During the spring and summer 2018, we consulted Professor Timo Hämäläinen, a software specialist at TUT, and it turned out that there are tools for automated acceleration of computing. However, they require at least a partial opening of the LISpMiner software code, which at the time being is not possible.

Practical Scenarios: While we have solved the problem, at least in principle, the project is currently frozen (mainly due to lack of suitable staff). We believe that we can end the project positively within a year.