MINING ASSOCIATION RULES USING THE
THEORY OF MONOTONE SYSTEMS

INNAR LIIV

Summary

The vast and ever-growing amount of information that different organizations store, needs to be analyzed and summarized into knowledge. A decade ago dealing with gigabytes in data processing and analysis seemed quite utopian – nowadays the largest companies have to analyse terabytes of data, some of them have already reached petabytes. As the amount of data is growing day by day, it is also constantly raising the computational and analytical complexity of mining knowledge from the data. Therefore, besides optimizing the present algorithms, new methods in problem solving need to be introduced as well.

One of the key areas of data mining is considered to be association rules, which is also known as market basket analysis and affinity analysis. The main goal of it is to find hidden associations between different objects or events. The aim of this thesis is to present a new algorithm monLIIV, which is based on monotone systems and is able to mine association rules directly from relational database systems.

In the first chapter, an overview of data mining is given. Besides, definitions and describing the interdisciplinarity of data mining, industry-recognized process guide of knowledge discovery in databases and practical applications in different areas (i.e. in insurance industry, banking, medicine, bioinformatics etc) with some example referenced case studies were described shortly. At the end of the first chapter, the position of association rule mining problem, among several other data mining activities, is given.

In the second chapter, the formal theoretical framework of association rule mining is presented and the semantic and technical aspects of the problem are being analyzed. The connection between support-confidence framework and observed frequencies in probability is also presented in order to be able to analyze different authors’ proposals for prejudgements how interesting the mined association rule apriori is. Also a problem decomposition, as
introduced in [1] is given and the description of different frequent sets – mining all, closed and maximal frequent sets, is given. Also a short description of several classical association rule mining problem algorithms and is presented.

The third chapter presents shortly the overview of monotone systems, its theoretical framework and focuses on using monotone systems in finding clustering. The overview of the classical and monotone systems methods for finding intersects is given and two very powerful pruning ideas introduced.

In the forth chapter, the algorithm monLIIV, is presented, which is based on monotone systems and is able to mine closed frequent sets and association rules directly from relational database systems. The results of algorithm observations are presented - properties, strengths and advantages of the algorithm are being analyzed – especially the compatibility of support-confidency framework. As the current algorithms confidency criteria is not in one-to-one accordance with support-confidency framework, a short discussion is given. Algorithm is given in C-like pseudocode and a short overview of the prototype implementation is also presented.

These are the main results of the thesis:

- The overview of the association rule mining problem and algorithms;
- Implementing the current monotone systems based algorithm into the context of association rule mining problem;
- A new algorithm – monLIIV – was presented, which is based on monotone systems and is able to mine association rules directly from relational database systems;
- A prototype implementation of monLIIV was developed.

Despite the fact ([76],[77]) that custom file-based implementations of different association rule algorithms show many times better results in mining speed, there are, according to [77], several advantages of a SQL implementation. For example one could make use of a database indexing capabilities, thereby leveraging on more than a decade of effort spent in making these systems robust, portable, scalable and concurrent. Also it is possible to exploit the underlying SQL parallelization. Thus, although the algorithm’s running time may be slower, we can reduce the time of preprocessing the data and concentrate only on the problem itself.