Chapter 3
A Practical Comparative Study Of Data Mining Query Languages

Hendrik Blockeel, Toon Calders, Élisa Fromont, Bart Goethals, Adriana Prado, and Céline Robardet

Abstract An important motivation for the development of inductive databases and query languages for data mining is that such an approach will increase the flexibility with which data mining can be performed. By integrating data mining more closely into a database querying framework, separate steps such as data preprocessing, data mining, and postprocessing of the results, can all be handled using one query language. In this chapter, we compare six existing data mining query languages, all extensions of the standard relational query language SQL, from this point of view: how flexible are they with respect to the tasks they can be used for, and how easily can those tasks be performed? We verify whether and how these languages can be used to perform four prototypical data mining tasks in the domain of itemset and association rule mining, and summarize their stronger and weaker points. Besides offering a comparative evaluation of different data mining query languages, this chapter also provides a motivation for a following chapter, where a deeper integration of data mining into databases is proposed, one that does not rely on the development of a new query language, but where the structure of the database itself is extended.

Hendrik Blockeel
Katholieke Universiteit Leuven, Belgium and Leiden Institute of Advanced Computer Science, Universiteit Leiden, The Netherlands e-mail: hendrik.blockeel@cs.kuleuven.be

Toon Calders
Technische Universiteit Eindhoven, The Netherlands e-mail: t.calders@tue.nl

Élisa Fromont · Adriana Prado
Université de Lyon (Université Jean Monnet), CNRS, Laboratoire Hubert Curien, UMR5516, F-42023 Saint-Etienne, France e-mail: {elisa.fromont,adriana.bechara.prado}@univ-st-etienne.fr

Bart Goethals
Universiteit Antwerpen, Belgium e-mail: bart.goethals@ua.ac.be

Céline Robardet
Université de Lyon, INSA-Lyon, CNRS, LIRIS, UMR5205, F-69621, France e-mail: celine.robardet@insa-lyon.fr

3.1 Introduction

An important motivation for the development of inductive databases and query languages for data mining is that such an approach will increase the flexibility with which data mining can be performed. By integrating data mining more closely into a database querying framework, separate steps such as data preprocessing, data mining, and postprocessing of the results, can all be handled using one query language. It is usually assumed that standard query languages such as SQL will not suffice for this; and indeed, SQL offers no functionality for, for instance, the discovery of frequent itemsets. Therefore, multiple researchers have proposed to develop new query languages, or extend existing languages, so that they offer true data mining facilities. Several concrete proposals have been implemented and evaluated.

In this chapter, we consider four prototypical data mining tasks, and six existing data mining query languages, and we evaluate how easily the tasks can be performed using these languages. The six languages we evaluate are the following: MSQL [8], MINE RULE operator [11], SIQL [17], SPQL [2], and DMX [16]. All six are based on extending SQL and have special constructs to deal with itemsets and/or association rules.

The four tasks with which the expressivity of the languages will be tested can all be situated in the association rule mining domain. The tasks are “typical” data mining tasks, in the sense that they are natural tasks in certain contexts, and that they have not been chosen with a particular data mining query language in mind. The four tasks are: discretizing a numerical attribute, mining itemsets with a specific area constraint, and two association rule mining tasks in which different constraints are imposed on the rules to be discovered. It turns out that the existing languages have significant limitations with respect to the tasks considered.

Many of the shortcomings of the six languages are not of a fundamental nature and can easily be overcome by adding additional elements to the query languages. Yet, when extending a query language, however, there is always the question of how much it should be extended. One can identify particular data mining problems and then extend the language so that these problems can be handled; but whenever a new type of data mining task is identified, a further extension may be necessary, unless one can somehow guarantee that a language is expressive enough to handle any kind of data mining problem.

While this chapter offers a comparative evaluation of different data mining query languages, this comparison is not the main goal; it is meant mostly as an illustration of the limitations that current languages have, and as a motivation for Chapter 11, where the idea of creating a special-purpose query language for data mining is abandoned, and the inductive database principle is implemented by changing the structure of the database itself, adding “virtual data mining views” to it (which can be queried using standard SQL), rather than by extending the query language.

We dedicate the next section to the description of the chosen data mining tasks. In Section 3.3, we introduce the data mining query languages and describe how they can be used for performing these tasks. Next, in Section 3.4, we summarize the