Submission for the Dutch-Belgian Database Day (DBDBD) 2006

Name: Adriana Bechara Prado.

Position/University: PhD student at the University of Antwerp, Belgium.

Supervisors: Bart Goethals and Toon Calders.

Title: Integrating Pattern Mining in Relational Databases.

Abstract

Almost a decade ago, Imielinski and Mannila introduced the concept of an *Inductive Database*, in which a *Knowledge and Data Discovery Management System* (KDDMS) manages KDD applications just as DBMSs successfully manage business applications. Generally speaking, besides allowing the user to query the data, the KDDMS should also give users the ability to query patterns and models extracted from these data. In this context, several researchers proposed extensions to the popular relational query language, SQL, as a natural way to express such mining queries.

In our work, we aim at extending the DBMS itself, not the query language. That is, we propose an approach in which the user can query the collection of all possible patterns as if these are stored in relational tables. The main challenge is how this storage can be implemented effectively. After all, the amount of all possible patterns can be extremely large, and impractical to store. For example, in the concrete case of itemsets, an exponential number of itemsets would need to be stored. To resolve this problem, we propose to keep these pattern tables virtual. That is, as far as the user is concerned, all possible patterns are stored, but on the physical layer, no such complete tables exist. Instead, whenever the user queries such a pattern table, or *virtual mining view*, an efficient data mining algorithm is triggered by the DBMS, which materializes at least those tuples needed to answer the query. Afterwards, the query can be executed as if the patterns had been there before. Of course, this assumes the user poses certain constraints in his query, asking for only a subset of all possible patterns, which should then be detected and exploited by the data mining algorithm. As a first step towards this goal, we propose such a constraint extraction procedure starting from a collection of simple constraints.

Notice that the user can now query mining results by using a standard relational query language, such as SQL. Furthermore, the user does not need to deal with the mining algorithms themselves as these are transparently triggered by the DBMS. We show how this approach can be implemented for the popular association rule and frequent set mining problems.