Frequent Queries Computation in Arbitrary Relational Database

Keywords: database, datamining, query, pattern, level-wise algorithm, knowledge discovery.

Abstract

The problem of mining frequent queries in a database has motivated many research efforts during the last two decades. This is so because many interesting patterns, such as association rules, exact or approximative functional dependencies and exact or approximative conditional functional dependencies can be easily retrieved, which is not possible using standard techniques. However, the problem mining frequent queries in a relational database is not easy even when we deal with one table, on the one hand, because the size of the search space is huge (because encompassing all possible queries that can be addressed to a given database), and on the other hand, testing whether two queries are equivalent (which entails redundant support computations) is NP-Complete. In [5, 6, 4], an algorithm for mining Projection-Selection-Join (PSJ) queries has been proposed for database defined over a star schema. In this thesis, we focus on Projection-Selection-Join (PSJ) queries, assuming that the database is defined over an arbitrary relational database not only on star schema. In this setting, one goal of this thesis is to define a pre-ordering ($q \leq q'$) that satisfies the following basic properties:

1. The support measure is anti-monotonic with respect to $\leq$, and
2. Defining $q \equiv q'$ if and only if $q \leq q'$ and $q' \leq q$

all equivalent queries have the same support. This two properties will allow us to use algorithm like Apriori [1, 2]. The main contributions of this thesis will be, on the one hand to formally study properties of the pre-ordering and the equivalence relation mentioned above, and on the other hand, to propose a level-wise, Apriori like algorithm for the computation of all frequent queries in a relational database defined over an arbitrary schema. Moreover, the implementation of an algorithm for mining frequent queries in any relational schema.

References


