

Mining Association Rules of Simple Conjunctive Queries

Bart Goethals Wim Le Page*
ADReM research group
University of Antwerp, Belgium

The discovery of recurring patterns in databases is one of the main topics in data mining and many efficient solutions have been developed for relatively simple classes of patterns and data collections. Indeed, most frequent pattern mining or association rule mining algorithms work on so called transaction databases. Not only for itemsets, but also for more complex patterns such as trees, graphs, or arbitrary relational structures, databases consisting of a set of transactions are used. For example, in the tree case [2], every transaction in the database contains a tree, and the presented algorithm tries to find all frequent subtrees occurring within all such transactions. For all these pattern classes, specialized algorithms exist to discover them efficiently. The motivation for these works is the potentially high business value of the discovered patterns [1].

Unfortunately, many relational databases are not suited to be converted into a transactional format and even if this would be possible, a lot of information implicitly encoded in the relational model would be lost after conversion. In this talk we consider association rule mining on arbitrary relational databases by combining pairs of queries which could reveal interesting properties in the database. Intuitively, we pose two queries on the database such that the second query is more specific than the first query. Then, if the number of tuples in the output of both queries is almost the same, this could reveal a potentially interesting discovery.

To illustrate, consider the well known Internet Movie Database containing almost all possible information about movies, actors and everything related to that, and consider the following queries: first, we ask for all actors that have starred in a movie of the genre ‘drama’; then, we ask for all actors that have starred in a movie of the genre ‘drama’, but that also starred in a (possibly different) movie of the genre ‘comedy’. Now suppose the answer to the first query consists of 1000 actors, and the answer to the second query consists of 900 actors. Obviously, these answers do not necessarily reveal any significant insights on themselves, but when combined, it reveals the potentially interesting pattern that actors starring in ‘drama’ movies typically (with a probability of 90%) also star in a ‘comedy’ movie. Of course, this pattern could also have been found by first preprocessing the database, and creating a transaction for each actor containing the set of all genres of movies he or she appeared in. Similarly, a pattern like: 77% of the movies starring Ben Affleck, also star Matt Damon, could be found by posing the query asking for all movies starring Ben Affleck, and the query asking for all movies starring both Ben Affleck and Matt Damon. Again, this could also be found using frequent set mining methods, but this time, the database should have been differently preprocessed in order to find this pattern. Furthermore, it is even impossible to preprocess the database only once in such a way that the above two patterns would be found by frequent set mining as they are essentially counting a different type of transactions. Indeed, we are counting actors in the first example, and movies in the second example.

In general, we are looking for pairs of queries Q_1, Q_2 , such that Q_1 asks for a set of tuples satisfying a certain condition and Q_2 asks for those tuples satisfying a more specific condition. When it turns out that the size of the output Q_2 is close to the size of the output of Q_1 , we learned that most of the tuples in the output of Q_1 actually satisfy a more specific condition, as specified in Q_2 . Clearly, such findings could reveal interesting patterns in the given database.

Towards this goal, we consider a new pattern class consisting of conjunctive queries over relational databases, called *simple conjunctive queries* and define associations using the well known notion of query containment. We propose an completely novel algorithm, Conqueror, efficiently generating and pruning the search space of all simple conjunctive queries. We illustrate that next to many different kinds of interesting patterns, our algorithm is also able to discover *functional dependencies*, *inclusion dependencies*, but also their variants, such as the very recently studied *conditional functional dependencies*, which turn out to be very useful for data cleaning purposes.

References

- [1] L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery*, 3(1):7–36, 1999.
- [2] M.J. Zaki. Efficiently mining frequent trees in a forest. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 71–80. ACM Press, 2002.

*Contact Author and PhD Student at the University of Antwerp