# Probabilistic Ranking of Database Query Results

Abhishek Jamloki

## Overview

The goal of this paper is to rank the answers to a database query when many tuples are returned. The probabilistic IR model is extended to structured data to rank the results. The proposed solution is domain independent and makes use of workload statistics and correlation.

- The first step is to divide the attributes in a table into specified (the ones specified by the query) and the unspecified attributes (the ones not specified by the query).
- Every tuple in the database is considered a document and correlation between different attributes in the tuple are found. Correlations are often ignored in high dimensional and sparsely populated data spaces in IR but there are strong correlations between the attribute values in relational data spaces.
- The authors make a limited independence assumption the specified (and unspecified) attributes within them are assumed to be independent but correlations between specified and unspecified attributes are allowed.
- Two kinds of scores are used to rank the documents. A global score which captures the global score of unspecified attributes and a conditional score that captures the strength of correlations between specified and unspecified attribute values.
- In the preprocessing phase of the computation two lists global list and the conditional list containing the global and conditional scores of the attributes for tuples are calculated. These are stored as auxiliary tables in the database.
- Instead of pre-computing the Top K results for all possible queries a ranked list of tuples for all atomic queries is calculated. Threshold Algorithm, a well known Top K algorithm is adapted to rank the queries using the computed scores at query time.
- The efficient adaptation of the algorithm is due to the limited independence assumption that is novel to this paper
- The processing involves two modules: (i) The index module (the preprocessing step), where the global and the conditional lists are constructed, and (ii) the List Merge Algorithm that is used to merge the lists associated with the attributes at query time.
- Extensive experiments are carried out on the internet movie database and the MSN home advisor database.
- The results are compared to the results from a rival query ranking method.

## Detailed Comments

The authors propose a fully automated query ranking method based on probabilistic IR and correlations. They give a detailed derivation to reach the expression for evaluating the score of a document. Detailed algorithmic implementation of the method is provided. Extensive experiments were carried out and efficiency of various algorithmic implementations were compared for different kinds of queries.

## Strengths:

- The paper successfully adapts the probabilistic IR model to rank query results in databases. Mathematical derivations are used to reach the expressions for calculating score.
- The authors also give an efficient implementation of the idea. Some well known algorithms are adapted to the problem and used for ranking query results. An optimization between pre processing and on the fly computation is made.
- The algorithm is trained using a work load that contains actual queries by the users. Hence user preferences are considered while ranking the results.
- The architecture allows users and domain experts to fine tune the global and conditional scores.

## Weakness:

- The authors do not mention the queries that they used during the experiment. Hence it is difficult to get an idea of how the algorithm will behave for different kinds of queries.
- Since the solution is defined for categorical values only data needs to be discreetized before proceeding with the solution. This may not be possible in all scenarios.

## Questions and Comments:

1) The authors have described the solution only for conjunctive point queries with categorical data. How will the method be extended to involve other type of queries and numerical data?
2) What method will the authors use to access global and conditional score tables in constant time?
3) What queries did the authors use to test the method?
4) The number of users in the survey for experiments is small.