Demian Lessa
April 10th, 2008
CSE 718 - Seminar on Database Interfaces

K-Relevance: A Spectrum of Relevance for Data Sources Impacting a Query

Overview

Huang and Naughton propose a parametric definition for the relevance of data sources in deriving the answer tuples of an input query. In particular, they propose that the relevance of a source depends on the number (k) of updates the user wishes to consider from the underlying sources. Intuitively, a data source S is k relevant to a query Q and the current database instance I if there is one updated relation from S (that is, a relation in I would be updated if data from source S is extracted and integrated back into I), and at most k-1 other updated relations such that they join (possibly with additional non-updated relations) to satisfy Q.

The notion of relevance has been previously studied in a rather restricted form through the concepts of lineage and provenance. In terms of k-relevance, lineage is closely related to 0-relevance, whereas the other kinds of relevance have no known counterpart anywhere else in the literature.

Three main algorithms are considered for computing k-relevance. The first algorithm computes query answers, and lineage information- that is, the set of 0-relevant sources. The second algorithm computes the ∞-relevant sources, that is, the sources whose updates can always modify the answers to the input query Q, independently of the database instance I. Since these sources only depend on the restrictions (selection and joins) observed in Q, its computation is usually very efficient. Finally, the last algorithm computes k-relevance for all values of k between 1 and m-1, where m is the maximum number of updated relations in Q. This is the more complex algorithm, since it needs to consider all possible combinations of k relations from all possible updated relations in Q. Hence, in the worst case, the running time of this algorithm is combinatorially expensive in $C_{m,k}$. In order to optimize this algorithm, the authors propose a few modifications to the base algorithm. A very naïve optimization involves the computation of k-relevance for alternating values of k (that is, k=1, m-1, 2, m-2, ...) and the use of the monotonicity of the sizes of the sets of k-relevant sources in order to avoid the computation for a range of indices. Other more involved optimizations were considered as well.

Besides considering the notion of k-relevance and providing algorithms for online computation of k-relevant sources, the authors consider the problem of maintaining materialized views for answering commonly asked queries. In particular, the idea is to materialize the set of relevant sources for the common queries once, and maintain them incrementally. Existing algorithms for materialized view maintenance are extended by considering relevant source information prior to actually updating the materialized views on disk.

Last, the authors present a set of experimental results obtained by running the algorithms on synthetic data. The main goal here was to show that the running times of the algorithms for particular values of k will generally scale with the running time of the input queries. In fact, the provided results shows that the computation of sets of k-relevant sources actually scales with the input queries, with the exception of some very particular cases in which no optimization is possible and the algorithm runs in time exponential to the size of the input query.

Detailed Comments

1. Describe the strengths and weaknesses of the paper.

The **main strength** of the paper is a new interpretation of the concept of source relevance in deriving answer tuples for an input query. This is a natural problem that had only been addressed with limited semantics of lineage and 1-relevant sources (previous work by the authors). The current paper extends these notions in a single framework. The **main weakness** lies in the implied assumption that the integration and extraction are perfect. The authors motivate the problem by considering the issue of debugging the set of query answers. According to their interpretation, the problems in the answers are caused by the data in the sources- data is incorrect, out-of-date, etc. It turns out that it is more natural to think about debugging the integration itself, and not the data sources. Another weakness of the paper is its rather futile attempt of exploring maintenance of materialized views. The contributions there are void.

2. How would you describe the technical depth of the paper?

The paper is well balanced in terms of depth. It provides a main definition, a number of lemmas and a theorem. It avoids most proofs, but provides enough hint to allow the reader to work them out. Further, it also provides the main sketches of the proposed algorithms, leaving out details and optimizations. This is a nice strategy, as it allows for immediate understanding of the main algorithm, while allowing the reader to “fill in the blanks” for the minute details, and optimizations.

3: Is the paper technically sound? Why?

The paper is technically sound. A main definition is provided, which is clear and minimal, and serves as the basis for deriving all subsequent results. Lemmas and theorems rely on the main definition and on previously presented lemmas. I could come up with proofs or sketches for most of the lemmas that were not proved in the paper. All algorithms are faithful to both their descriptions and theoretical foundations. Finally, the experimental results make sense in light of the algorithms presented. In short, no leaps of faith are necessary to fully understand and follow the ideas, algorithms, and results in the paper.

4. How does the paper compare with the related work?

This has already been discussed. The paper makes a solid contribution by providing a parametric definition of relevance. In other words, they define the relevance of a source based on the number of updates a user is willing to consider. Related work has not considered this problem. The closest work on relevance is in the area of lineage and/or provenance, which deals with 0-relevant sources. The authors have also published in a previous work a limited version of k-relevance, where k=1. In other words, they considered as relevant sources that contributed with one updated relation.