Learning Effective and Robust Knowledge for Semantic Query Optimization*

Chun-Nan Hsu
Computer Science and Engineering Department
Arizona State University, Box 875406,Tempe, AZ 85287-5406
phone: (602) 965-1757, E-mail: chunnan@asu.edu
http://www.isi.edu/sims/chunnan

Semantic Query Optimization for Information Mediators

New applications of information systems, such as electronic commerce and healthcare information systems, need to integrate heterogeneous information sources. A promising solution to this problem is through the use of information mediators that can automatically generate a query plan to retrieve and combine data from heterogeneous information sources. Existing query optimization techniques can provide local optimization for a query plan by optimizing subqueries to individual databases. However, due to the complexity of queries and the heterogeneity of databases, it is difficult for these techniques to provide global optimization.

We have developed an extension of semantic query optimization [2] that can complement conventional techniques to overcome the heterogeneity and considerably reduce redundant data transmission. This approach takes a *query plan* generated by the information mediator as the input and modifies conditions specified in the query plan to optimize the query. Suppose we have a query as follows:

Query 1 *Retrieve the classes of active ships with container capability that can dock in the wharves with cranes at Long Beach seaport; list by ship class name and wharf id.*

The information required to answer this query is spread over two heterogeneous databases. The database `Assets` contains information about ships, ship class, aircrafts, etc. The other database `Geo` stores data about geographic locations, seaports, wharves, airports, and runways, etc. Receiving this query, the information mediator will construct a query plan to retrieve the requested information. The diagram on the left half of Figure 1 shows the diagram of this plan, in which Subquery 1 and 2 retrieve data from remote databases and Subquery 3 joins the intermediate data at the local information mediator.

*This research was supported in part by NSF Grant No. IRI-9313993, and in part by Rome Laboratory of the Air Force and DARPA Contract No. F30602-94-C-0210.

This query plan is optimal in terms of the number of subqueries, but could still be expensive because the system needs to retrieve and transmit lots of redundant ship class and wharf data that will eventually be discarded when executing Subquery 3.

Suppose in the beginning, the optimizer possesses a set of semantic knowledge about those databases.

- Ranges of attribute values that bind the variables in the query. In this case, we have
  `?depth 7 to 50,     ?draft 12 to 72,`
  `?length 325 to 950, ?wlength 580 to 2700.`
- Semantic rules about ship classes and ships:
  - if the maximum draft of a ship is less than 50 then its status is active.
  - if the ships in a ship class have container capability, then we can find some ships that belong to the ship class in the database.

Given the range information and semantic rules, the optimizer can reduce the intermediate data and optimize subqueries as well in the following optimization steps in our example.

- From the range information, inferring that `?wlength > ?length` is always true for this query. Therefore, this literal can be deleted from Subquery 3. Also, since they are not used as output, data about `?wlength` and `?length` are no longer necessary and can be deleted from the parameter list of Subquery 1 and 2.
- From the range information again, deriving a more restrictive range of `?draft`, which must be less than 50 to satisfy `?depth > ?draft`.
- Moving this newly derived literal `?draft < 50` to Subquery 2 because this literal can be evaluated more efficiently at the remote database system.
- Deriving that it is redundant to specify `"Active"` based on the semantic rules that a ship is always `"Active"` if `?draft < 50`.
- Removing the entire `ship` literal, because `"Active"` is redundant, and the semantic rule states that if a ship class with container capability (specified as `"Y"`), then there must exist some

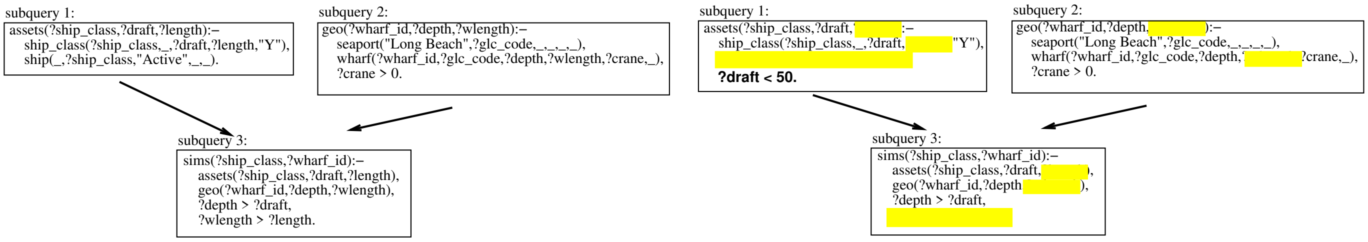

Figure 1: Original and optimized multidatabase query plans

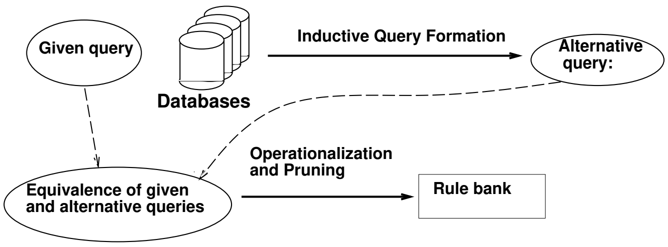

Figure 2: Organization of the learning approach for SQO

ships belong to that ship class in the database. That is, the join with `ship` is not necessary.

The resulting plan is shown on the right half of Figure 1, where terms in bold font are inserted new terms, and terms under shaded regions are deleted. This plan will retrieve the same answer as the original plan but is much less expensive than the original plan. In our experiment, it takes 3.17 seconds to execute the original plan and 1.78 seconds to execute the optimized plan. This amounts to a 43.8 percent of reduction. The execution time of the optimized plan includes 0.03 second of overhead optimization time. Therefore, the optimizer effectively optimizes this query and reduces much execution cost.

In addition to its effectiveness, our approach is more general and flexible than the state-of-the-art semantic query optimization techniques for stand-alone databases (e.g., Chakravarthy 90, Shekhar 88, Shenoy 89, Yu 89,94). Our query optimization approach can be implemented on top of existing query optimizers in a heterogeneous environment and hence supports the extensibility of an information mediator.

To perform this kind of optimization, however, requires the query processor possess sufficient high utility semantic knowledge. The focus of this research is to develop a general and efficient approach to learning high utility semantic rules required for SQO.

Learning for Semantic Query Optimization

Since the number of potential rules that can be derived from a database is enormous, an important issue is how to constrain learning. High utility semantic rules should maximize the net savings, that is they should be effective in producing cost reduction while require a minimal overhead to be used and maintained. Generally speaking, to maximize the cost reduction the semantic rules should match query patterns and reflect the regularities of the data. To minimize the maintenance cost, semantic rules must be *robust* against database changes and noise. Otherwise, a learned semantic rule may easily become invalid after data changes and increase the maintenance cost.

The notion of the *robustness of knowledge* [3, 2] is new and radically different from other confidence measures of knowledge used in inductive learning and data mining. Robustness of a rule is the likelihood that the rule remains consistent with a database after data insertions, deletions, and updates. Other confidence factors (e.g., predictive accuracy) measure the coverage of a rule over the space of possible data instances.

We observed that robustness of a rule may roughly interact with its effectiveness in cost reduction. In many cases, an effective rule is the one that "overfits" the data and is likely to become invalid after data changes, while a robust rule may be a trivial tautology and hardly contribute to the cost reduction.

We have developed a general learning approach that attempts to balance the tradeoff between effectiveness and robustness. As illustrated in Figure 2, the learning system consists of two components: an inductive learning component, and an operationalization component. The learner is triggered to learn a new set of rules when the query processor encounters an expensive query that can not be satisfactorily optimized with existing rules. Given such an expensive query, the inductive learning component inductively forms from

	Average savings	hand-coded rules
All	31.07%	25.84%
Multidb	41.05%	36.19%
# of Rules	111	112
opt time (s)	0.045	0.044

Table 1: Performance data of learned rules and hand-coded rules

data an alternative query, which is the desired reformulation of the given query. This inductive learning component can form a conjunctive alternative query with relational joins from a relational database. It uses knowledge about query execution cost to bias the induction so that the learned alternative query is much cheaper to be evaluated.

From this alternative query, the operationalization component derives operational semantic rules that enable the system to optimize expensive queries similar to the given query. However, they may not be robust and could be overly-specific. We use a rule pruner to prune the antecedents of those rules. This rule pruner balances the tradeoff between the length of a rule and its robustness by applying a beam search strategy to retain the top robust rules for each set of partially pruned rules with the same length. Then the pruner selects a subset of pruned rules with a good combination of length and robustness based on how often the database changes. The pruner uses a Bayesian estimation approach [3] to estimate the robustness of rules and guide the pruning. This estimation approach can bring to bear different forms of database information (schema, transaction log, etc.) and tolerate data noise.

Experimental Results

We have implemented the approach into a rule learning system called BASIL and a semantic query optimization system called PESTO [2] as components of the SIMS information mediator [1]. For the purpose of our experiments, SIMS was connected with two large remote ORACLE relational databases (about 10 MB) via the Internet. These databases originally were part of a real-world transportation logistic planning application. Together with the databases, there were 36 sample queries written by the users of the databases. Among these 36 queries, 18 were multidatabase queries that require access to multiple databases to retrieve the answer. To train the learner, BASIL, 23 queries were selected to server as the training queries The selection was based on the similarity of queries. This experiment applies a *k-fold cross validation* procedure so that each training query may have a chance to be used for training as well as testing. Prior to this experiment, we have hand-crafted a set of 112 semantic rules that were tailored to the sample queries to test the performance of the optimizer. We report the optimization performance produced by the hand-coded rules here for the purpose of comparison.

Table 1 shows the average of the savings for all queries and the average of savings for multidatabase queries. The data shows that the learned rules can produce a significant savings on the test queries, with a ten percent higher savings for multidatabase queries. The data also shows that the learned rules outperform hand-coded rules in all four tests. We note that some of our test queries are already very cost-effective, and there is not much room for optimization for those queries. The savings may reach up to 70% to 90% for expensive queries that required more than one minute to complete.

We also conducted an experiment to evaluate the accuracy of robustness estimation in order to establish the claim that using the robustness estimation allows a learner to minimize the cost of dealing with database changes. The result showed that our estimation approach correctly identified 92.70% of rules that would become inconsistent after a sequence of data modification transactions. As a result, using the estimated robustness to guide the learning will generate high robust rules that require low maintenance cost.

Conclusions

This paper outlines a general learning approach to the knowledge acquisition problem of semantic query optimization. The results shows that the learned rules outperformed hand-coded rules and provided significant savings. For more detailed description of the approaches discussed in this paper, please refer to the author's doctoral dissertation [2]. Other references are available upon request.

References

[1] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu, and Craig A. Knoblock. Retrieving and integrating data from multiple information sources. *International Journal on Intelligent and Cooperative Information Systems*, 2(2):127–159, 1993.

[2] Chun-Nan Hsu. *Learning Effective and Robust Knowledge for Semantic Query Optimization.* PhD thesis, Department of Computer Science, University of Southern California, 1996.

[3] Chun-Nan Hsu and Craig A. Knoblock. Discovering robust knowledge from dynamic closed-world data. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, Oregon, 1996. AAAI Press.