A Framework for Refining Similarity Queries Using Learning Techniques

Yiming Ma, Qi Zhong, Sharad Mehrotra, Dawit Yimam Seid
Information and Computer Science
University of California, Irvine
{ maym, qzhong, sharad, dseid }@ics.uci.edu

ABSTRACT

In numerous applications that deal with similarity search, a user may not have an exact idea of his information need and/or may not be able to construct a query that exactly captures his notion of similarity. A promising approach to mitigate this problem is to enable the user to submit a rough approximation of the desired query and use the feedback on the relevance of the retrieved objects to refine the query. In this paper, we explore such a refinement strategy for a general class of SQL similarity queries. Our approach casts the refinement problem as that of learning concepts using examples. This is achieved by viewing the tuples on which a user provides feedback as a labeled training set for a learner. Under this setup, SQL query refinement consists of two learning tasks, namely learning the structure of the SQL query and learning the relative importance of the query components. The paper develops appropriate machine learning approaches suitable for these two learning tasks. The primary contribution of the paper is a general refinement framework that decides when each learner is invoked in order to quickly learn the user query. Experimental analyses over many real life datasets and queries show that our strategy outperforms the existing approaches significantly in terms of retrieval accuracy and query simplicity.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Relevance feedback, Query formulation, Retrieval models.

General Terms: Algorithms.

Keywords: Relevance feedback, Structured data, Refinement, Learning.

1. INTRODUCTION

With the proliferation of the web and emergence of applications requiring flexible search over diverse data types, effective support for personalized similarity search in database systems has emerged as an important research challenge. Similarity based retrieval systems are increasingly used for exploratory data analysis and retrieval where a user may not initially have a clear mental model of his exact information need [10].

A promising approach to overcome the "initial query" and "subjectivity" problems is that of automatic query refinement via user feedback. In such an approach, a user starts with an approximate initial query and communicates his preferences to the system by providing feedback (judgments on the relevance or quality of answers). The system then modifies the query internally (e.g., changes the levels of importance of the different search criterias, and adds/removes search criteria) to better focus on the distinguishing features of tuples deemed relevant. The modified query is re-evaluated and the cycle of refinement continues until the user is satisfied with the results.

Query refinement via feedback in the context of similarity search has been explored extensively in the context of text document retrieval [8, 10, 2, 1]. More recently, its effectiveness in feature-based image and multimedia similarity retrieval [6, 9, 3] as well as similarity search over metric spaces [11] has also been established. In work [5], we considered the problem of refining SQL queries from user interactions in an object relational database. This paper explored a limited set of SQL queries and illustrated that even simple techniques (i.e., mostly extensions of refinement approaches studied in the IR literature) can significantly enhance the users' search experience.

This paper considers the problem of refining a general class of SQL queries. In contrast to the ad-hoc approaches in [5], we postulate the SQL refinement problem as that of concept learning from examples to which many existing machine learning solutions can be applied. A direct (naive) strategy to modify the query is to view the records on which the user provides feedback as a labeled training set and use a classifier (e.g., decision tree [7]) to learn a new query representation that will replace the original query. Such a strategy to "learn" a query from feedback does not perform well in practice. Reasons for this include that classifiers do not work well when the training set is very small as is the case in our setting when a user may provide feedback on only a few records. Furthermore, the naive strategy largely ignores the initial query provided by the user; it may get trapped in a wrong hypothesis simply because the hypothesis fits the few examples that the user provided input on. In addition, it is very difficult to incorporate user defined types and similarity functions into these models. Consistency is also an important issue in the refinement task. Many existing classifiers are sensitive to the inputs. The models built from different iterations could differ substantially. This also makes these strategies less usable for refinement purposes.

In our approach, we consider the query refinement from feedback as consisting of two interrelated learning tasks – learning the query structure, and learning the query weights that capture the relative importance of different query components. The two learning tasks have different motivations

CIKM'04, November 8–13, 2004, Washington, DC, USA.
ACM 1-58113-874-1/04/0011.

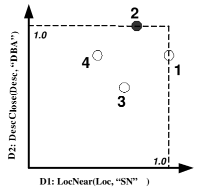

Figure 1: Example Conjunction Similarity Space with Conflicts

and serve different purposes. For instance, structural changes to the query are very useful when the initial user query is incomplete which may happen if either a user is not familiar with the database or finds it too laborious to postulate a proper query. In contrast, weight adjustment serves the purpose of customizing/tuning the ranking of results to reflect the subjective importance of the different query components to the user. A key issue in refining a query is to determine when to invoke the structure/weight learners. We develop a formal basis for making such a decision based on which an activation procedure is designed.

The major contributions of this paper are summarized as follows:

(1) We provide a powerful framework for refining a general class of SQL similarity queries that combines a multimodal refinement activation procedure that employees machine learning techniques to adjust both query predicate weights and query structures.

(2) We suggest and experimentally validate a novel query structure refinement technique that is based on a decision tree learner.

In the remainder of this paper, we briefly introduce our refinement framework and experiment setup and results. In the full version of this paper [4], we provide the detailed algorithms and experiment results.

2. REFINEMENT FRAMEWORK

Our general framework to control the refinement process that consists of two interrelated learning tasks – updating the weight template and modifying the structure of the query. The key aspect of the framework is an activation procedure that, given the user feedback, decides which learning task to invoke. Structural changes result in addition (deletion) of a new (old) conjunction to the DNF query or addition (deletion) of a new (old) predicate to a conjunction. In contrast to gradual weight adjustment that results in fine tuning of the rank order, structural changes can dramatically modify the return set by bringing objects deemed irrelevant to the original query to the top of the retrieved set. For this reason, structural changes to the query must be made conservatively since an incorrect change may lead a refinement along a wrong path causing the result set to become worse instead of better. Hence, the activation procedure invokes structural modifications only when feedback provided by the user requires that such modifications be made and the desired effect cannot be achieved by reweighting the different query components. Identifying situations in which structural modifications are needed requires us to explore the limitations of weight learning in the context of refinement.

Let $Q = C_1 \vee C_2 \vee C_n$ be a DNF query where $C_i = C_{i1} \wedge C_{i2} \ldots , C_{in}$ is a conjunction. Q is refined by refining each C_i. To refine a C_i, for a tuple on which feedback is given, the activation procedure first assigns the feedback to the conjunction only if the conjunction assigns the highest score to the tuple. This results a set of feedback for the conjunction. It then determines if there is a conflict in any pair of feedback tuples. If a conflict set is identified, structural modification to the query is invoked based on the conflicting set. For example, in Figure 1 (i.e., empty circle represents positive feedback), tuple 2 conflicts with tuple 3 and 4. To resolve the conflicts, the activation procedure invokes the structural learning using tuples 2, 3, and 4. Specifically, the structure learning algorithm will attempt to learn a predicate that when added to the conjunction will resolve the conflict.

3. EXPERIMENTS

In the full version of this paper [4], we present the results of our experiments to evaluate the effectiveness of our query refinement method. We first present our experimental setup, the 12 well-known real life datasets we used, and a query generator that embodies the user model and generates queries appropriately. Using total of 240 generated query paris, we then evaluate our new approach against four well-known algorithms. The goal of the evaluation is to establish our new algorithm as an accurate, and efficient relevance feedback refinement algorithm. For every query pair, we run each refinement algorithm four times. To measure the refinement accuracy, at each refinement iteration, an average precision measure is used. The new refinement algorithm outperforms the best alternative by an average of 24% after the first refinement cycle and remains to be the best algorithm for the rest of refinement cycles.

4. REFERENCES

[1] Ricardo Baeza-Yates and Ribeiro-Neto. *Modern Information Retrieval.* ACM Press Series/Addison Wesley, New York, May 1999.

[2] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, Valencia, Spain, 1992.

[3] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreader: Querying databases through multiple examples. In *Proc. 24th Int. Conf. on Very Large Data Bases VLDB '98*, pages 218–227, 1998.

[4] Y. Ma, Q. Zhong, S. Mehrotra, and D. Seid. A framework for refining similarity queries using learning techniques (tr-db-04-05). *http://www-db. ics. uci. edu/pages/publications/index. shtml*, 2004.

[5] Michael Ortega-Binderberger, Kaushik Chakrabarti, and Sharad Mehrotra. An Approach to Integrating Query Refinement in SQL. In *Proc. EDBT*, March 2002.

[6] Kriengkrai Porkaew, Sharad Mehrotra, Michael Ortega, and Kaushik Chakrabarti. Similarity search using multiple examples in mars. In *Proc. Visual'99*, June 1999.

[7] J. R. Quinlan. Induction of Decision Tree. *Machine Learning*, 1:81–106, 2002.

[8] J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[9] Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. Circuits and Systems for Video Technology*, 8(5):644–655, September 1998.

[10] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill Computer Science Series, 1983.

[11] L. Wu, C. Faloutsos, K. Sycara, and T. Payne. FALCON: Feedback adaptive loop for content-based retrieval. *Proceedings of VLDB Conference*, 2000.