Chapter 1

Automatic Categorization of Web Database Query Results

Xiangfu Meng
Liaoning Technical University, China

Li Yan
Northeastern University, China

Z. M. Ma
Northeastern University, China

ABSTRACT

Web database queries are often exploratory. The users often find that their queries return too many answers and many of them may be irrelevant. Based on different kinds of user preferences, this chapter proposes a novel categorization approach which consists of two steps. The first step analyzes query history of all users in the system offline and generates a set of clusters over the tuples, where each cluster represents one type of user preference. When a user issues a query, the second step presents to the user a category tree over the clusters generated in the first step such that the user can easily select the subset of query results matching his needs. The problem of constructing a category tree is a cost optimization problem and heuristic algorithms were developed to compute the min-cost categorization. The efficiency and effectiveness of our approach are demonstrated by experimental results.

INTRODUCTION

As internet becomes ubiquitous, many people are searching their favorite cars, houses, stocks, etc. over the Web databases. However, Web database queries are often exploratory. The users often find that their queries return too many answers, which are commonly referred to as “information overload”. For
example, when a user submits a query to MSN House&Home Web site to search for a house located in Seattle with a price between $200,000 and $300,000, 1,256 tuples are returned. Information overload makes it hard for the user to separate the interesting items from the uninteresting ones, and thereby lead to a huge wastage of user’s time and effort. In such a situation, the user would pose a broad query in the beginning to avoid exclusion of potentially interesting results, and then iteratively refine their queries until a few answers matching their preferences are returned. However, this iterative procedure is time-consuming and many users will give up before they reach the final stage.

In order to resolve the problem of “information overload”, two types of solutions have been proposed. The first type categorizes the query results into a category tree (Chakrabarti, Chaudhuri & Hwang, 2004; Chen & Li, 2007), and second type ranks the results (Agrawal, Chaudhuri, Das & Gionis, 2003; Agrawal, Rantzau & Terzi, 2006; Bruno, Gravano & Marian, 2002; Chaudhuri, Das, Hristidis & Weikum, 2004; Das, Hristidis, Kapoor & Sudarshan, 2006). The success of both approaches depends on the utilization of user preferences. But these approaches always assume that all users have the same user preferences, but in real life different users often have different preferences. Let us look at the following example.

**Example 1.** Consider a real estate searching Web site. Figure 1 and Figure 2 respectively show a fraction of category trees generated by using the methods of Greedy (Chakrabarti, Chaudhuri & Hwang, 2004) and C4.5-Categorization (Chen & Li, 2007) over 214 houses returned by a query with the condition “Price between 250000 and 350000 ∧ City = Seattle”. Each of tree nodes specifies the range or equality conditions on an attribute, and the number in the parentless is the number of tuples satisfying all conditions from the root to the current node. Users can use this tree to select the houses they are interested in.

Consider three users $U_1$, $U_2$, and $U_3$. Assume that $U_1$ prefers houses with large square, $U_2$ prefers houses with water views, and $U_3$ prefers both water views and Burien living area. The Greedy method assumed that all users have the same preferences. As a result, attributes “Livingarea” and “Schooldistrict” are placed at the first two levels of the tree because more users are concerned with “Livingarea” and “Schooldistrict” than other attributes. However, there may be some users (such as $U_2$ and $U_3$) who want to first visit the large square and water view houses. Then they have to visit many nodes if they go along with the tree built in Figure 1. Considering the diversity of user preferences and the cost of both visiting intermediate nodes and leaf nodes, the C4.5-Categorization method took advantage of C4.5 algorithm.
Related Content

Semi-Automatic Composition of Situational Methods
[www.igi-global.com/article/semi-automatic-composition-situational-methods/61339?camid=4v1a](www.igi-global.com/article/semi-automatic-composition-situational-methods/61339?camid=4v1a)

Information Mediation Using Metamodels: An Approach Using XML and Common Warehouse Metamodel
[www.igi-global.com/article/information-mediation-using-metamodels/3375?camid=4v1a](www.igi-global.com/article/information-mediation-using-metamodels/3375?camid=4v1a)

Comparing Metamodels for ER, ORM and UML Data Models
Terry Halpin (2004). *Advanced Topics in Database Research, Volume 3* (pp. 23-44).
[www.igi-global.com/chapter/comparing-metamodels-orm-uml-data/4352?camid=4v1a](www.igi-global.com/chapter/comparing-metamodels-orm-uml-data/4352?camid=4v1a)

Applying UML for Modeling the Physical Design of Data Warehouses
[www.igi-global.com/chapter/applying-uml-modeling-physical-design/7021?camid=4v1a](www.igi-global.com/chapter/applying-uml-modeling-physical-design/7021?camid=4v1a)