ISSN 2395-1621

Study of Interactive Database Exploration

[#1]Aashish Bhagwat, [#2]Mahesh Langote, [#3]Vishal Shelke, [#4]Mohini Phajge
[#5]Prof. Vinod Wadne

[1]aashishbhagwat4u@gmail.com
[2]maheshlangote1@gmail.com
[3]princesuraj94@gmail.com
[4]ph.mohini@gmail.com

[#1234]Department of Computer Engineering
[#5]Department of Computer Engineering
JSPM's
Imperial College Of Engineering & Research,
Wagholi, Pune – 412207

ABSTRACT

Relational database systems are becoming increasingly popular in the scientific community to support the interactive exploration of large volumes of data. In this scenario, users employ a query interface (typically, a web-based client) to issue a series of SQL queries that aim to analyze the data and mine it for interesting information. First-time users, however, may not have the necessary knowledge to know where to start their exploration. Other times, users may simply overlook queries that retrieve important information. To assist users in this context, we draw inspiration from Web recommender systems and propose the use of personalized query recommendations. The idea is to track the querying behavior of each user, identify which parts of the database may be of interest for the corresponding data analysis task, and recommend queries that retrieve relevant data. We discuss the main challenges in this novel application of recommendation systems, and outline a possible solution based on collaborative filtering. Preliminary experimental results on real user traces demonstrate that our framework can generate effective query recommendations.

Keywords:

ARTICLE INFO

Article History

Received: 26th October 2015

Received in revised form :

28st October 2015

Accepted:28th October, 2015

Published online :

28th October 2015

I. INTRODUCTION

Relational database systems are becoming increasingly popular in the scientific community to manage large volumes of experimental data. Examples include the Genome browser that provides access to a genomic database, and Sky Server that stores large volumes of astronomical measurements. The main advantage of a relational database system is that it supports the efficient execution of complex queries, thus enabling users to interactively explore the data and retrieve interesting information. It should be noted that the aforementioned systems employ web-based query interfaces in order to be accessible to a broad user base.

Even though a database system offers the means to run complex queries over large data sets, the discovery of useful information remains a big challenge. Users who are not familiar with the database may overlook queries that retrieve interesting data, or they may not know what parts of the database provide useful information. This issue clearly hinders data exploration, and thus reduces the benefits of using a database system. To address this important problem, we draw inspiration from the successful application of recommender systems in the exploration of Web data. In particular, we focus on approaches based on user-based collaborative filtering. The premise is simple: If a user A has similar querying behavior to user B, then they are likely interested in the same data. Hence, the queries of user B can serve as a guide for user A. The transfer of this paradigm entails several challenging problems. In web collaborative filtering systems, a user-item matrix approach is used to generate recommendations. More specifically, each user is represented as an item vector, where the values of the vector elements correspond to the user's preferences for each item (such as movie ratings, purchased products, read articles, etc.) The similarities between users in this representation can be easily computed using vector similarity metrics. Given the most similar users and their preferences, the collaborative filtering system can subsequently predict what items will interest each user, and generate item recommendations. The aforementioned methodology cannot be directly applied to

the context of a relational database for several reasons. First, we observe that in the case of a database the "items" of interest are database records, and the users access these items indirectly by posing SQL queries. Thus, even though the users' behavior is identified by the set of queries they send to the database, their interest lies on the database tuples they retrieve. Given that SQL is a declarative language, however, the same data can be retrieved in more than one way. This complicates the evaluation of similarity among users based on their queries alone, since it is no longer obvious whether they are interested in the same "items". This raises a second important issue that needs some consideration. The similarity between users can be expressed as the similarity between the fragments of their queries or, alternatively, the data that they retrieve. This is not as straightforward, since a query fragment or a tuple might have different levels of importance in different user sessions. Thus, we must be able to create implicit user profiles that model those levels of importance, in order to effectively compare the users. Finally, contrary to the user-based collaborative filtering approach, the recommendation to the users has to be in the form of SQL queries, since those actually describe what the retrieved data represent. Thus, we need to "close the loop" by first decomposing the user queries into lower-level components in order to compute similarities and make predictions, and then re-construct them back to SQL queries in order to recommend them. Moreover, these SQL queries must be meaningful and intuitive, so that users can parse them and understand their intent and usefulness.

II. PROPOSED WORK

Our work considers the interactive exploration of a relational database.

Database and Querying Model

We consider a relational database comprising N relations denoted as R1,...,RN . We use Q to denote a Select-Project-Join (SPJ) query over the database, and ans(Q) for its result set. We focus on the class of SPJ queries because they are common in interactive database exploration, particularly among the group of novice users which is the focus of our work.

Interactive Data Exploration

Users typically explore a relational database through a sequence of SQL queries. The goal of the exploration is to discover interesting information or verify a particular hypothesis. The queries are formulated based on this goal and reflect the user's overall information need. As a consequence, the queries posted by a user during one "visit" (commonly called session) to the database are typically correlated in that the user formulates the next query in the sequence after having inspected the results of previous queries.

Personalized Query Recommendations

The problem of personalized query recommendations can be formulated as follows: Given a user that is currently exploring the database, recommend queries that might be of interest to him/her. To generate such recommendations, the system will rely on information gathered from the querying behaviour of past users, as well as the queries posed by the current user so far.

The information flow of the QueRIE framework is shown in Figure1. The active user's queries are forwarded to both the DBMS and the Recommendation Engine. The DBMS processes each query and returns a set of results. At the same time, the query is stored in the Query Log. The Recommendation Engine combines the current user's input with information gathered from the database interactions of past users, as recorded in the Query Log, and generates a set of query recommendations that are returned to the user.

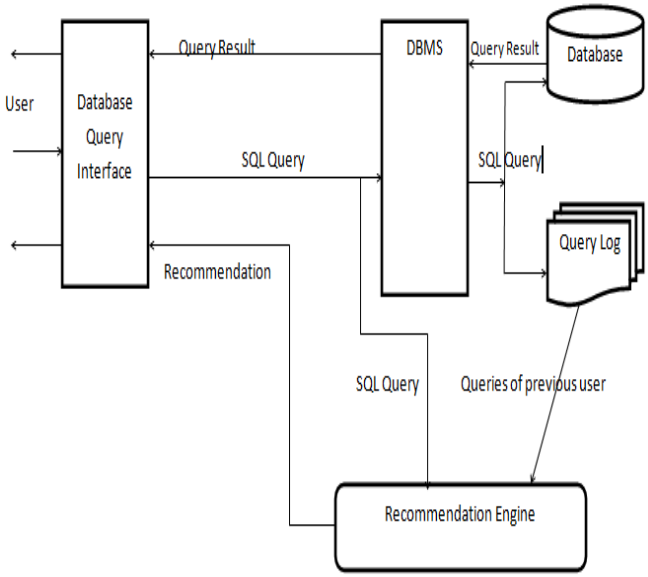

Fig 1. System Architecture

The information flow of the QueRIE framework is shown in Figure1. The active user's queries are forwarded to both the DBMS and the Recommendation Engine. The DBMS processes each query and returns a set of results. At the same time, the query is stored in the Query Log. The Recommendation Engine combines the current user's input with information gathered from the database interactions of past users, as recorded in the Query Log, and generates a set of query recommendations that are returned to the user.

III. CONCLUSION

In this paper, we present a query recommendation framework supporting the interactive exploration of relational databases and an instantiation of this framework based on user-based collaborative filtering. The experimental evaluation demonstrates the potential of the proposed approach. We should stress that this is a first-cut solution to the very interesting problem of personalized query recommendations. There are many open issues that need to be addressed. For instance, an interesting problem is

that of identifying "similar" queries in terms of their structure and not the tuples they retrieve. Two queries might be semantically similar but retrieve different results due to some filtering conditions. Such queries need to be considered in the recommendation process. We are currently working on extending our framework to cover such query similarities. Another interesting direction is to apply item-based collaborative filtering instead of the user-based approach of the current framework. We also intend to explore other approaches for instantiating the proposed conceptual framework.

REFERENCE

[1] Koutrika, G., Ioannidis, Y.: Personalized queries under a generalized preference model. In: ICDE 2005: Proceedings of the 21st International Conference on Data Engineering, pp. 841–852 (2005)

[2] Adomavicius, G., Kwon, Y.: New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems 22(3), 48–55 (2007) 18 G. Chatzopoulou, M. Eirinaki, and N. Polyzotis.

[3] Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17(6), 739–749 (2005).

[4] Bell, R., Koren, Y., Volinsky, C.: Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: KDD 2007: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 95–104 (2007).

[5] Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. 22(1), 143–177 (2004) .

[6] Greco, G., Greco, S., Zumpano, E.: Collaborative filtering supporting web site navigation. AI Commun. 17(3), 155–166 (2004) .

[7] Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004) .

[8] Lee, H.J., Kim, J.W., Park, S.J.: Understanding collaborative filtering parameters for personalized recommendations in e-commerce. Electronic Commerce Research 7(3-4) (2007) .

[9] Mohan, B.K., Keller, B.J., Ramakrishnan, N.: Scouts, promoters, and connectors: the roles of ratings in nearest neighbor collaborative filtering. In: EC 2006: Proc. of 7th ACM Conference on Electronic Commerce, pp. 250–259 (2006) .

[10] Park, S., Pennock, D.M.: Applying collaborative filtering techniques to movie search for better ranking and browsing. In: KDD 2007: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 550–559 (2007).

