



ISSN NO. 2320-5407

Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/1262
DOI URL: <http://dx.doi.org/10.21474/IJAR01/1262>



INTERNATIONAL JOURNAL OF
ADVANCED RESEARCH (IJAR)
ISSN 2320-5407
Journal homepage: <http://www.journalijar.com>
Journal DOI: 10.21474/IJAR01

RESEARCH ARTICLE

ACCESS ENABLED DATABASE QUERY RECOMMENDATION SYSTEM.

*Shiny Nair¹ and Varghese S Chooralil².

1. M. Tech Student, Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, Kochi, India.
2. Assistant Professor, Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, Kochi, India.

Manuscript Info

Manuscript History

Received: 11 June 2016
Final Accepted: 22 July 2016
Published: August 2016

Key words:-

Interactive database exploration, SQL query recommendation, query fragments, access control.

Abstract

Extensive exploration of the database is essential for an effective information mining process. Massive volumes of complex data from various sources or platforms that are integrated and available in datamarts have to be analyzed and studied for any form of knowledge discovery. For example, users like scientists need to query large databases for scientific data analysis and exploration. However, all users may not possess the competencies in Structured Query Language that is in general required, to query relevant data from relational database. Inexperienced users will seldom be proficient with the structural details of the schema and databases. This knowledge is required for all database related activities like generating reports for ad-hoc business requests. A database query recommendation system is a system that assists users by presenting personalized database query recommendations. The framework attempts to identify similarities with previous users' information needs and recommend queries to the current user. An access enabled recommendation engine is an extension to the query recommendation system that supports the capability of recommending queries based on the user's privileges to access a particular database object. For example, the system will refrain from suggesting a query that refers to a database table with confidential data, to a user with insufficient privileges, regardless of whether the query qualifies with highest similarity for recommendation.

Copy Right, IJAR, 2016.. All rights reserved.

Introduction:-

Recent times have witnessed significant growth in data that is available for analysis and mining. Data analysts and business users have excessive information that has to be analyzed exhaustively and methodically to arrive at crucial strategic decisions. There are also other users from the scientific world like scientists who need to conduct research and development activities over large volumes of data. This data is stored mostly in relational databases and datamarts. Scientists like astronomers and biologists use web-enabled applications or interfaces to access, study and analyze this data. For example, genome and astronomy related data are available online for students, research scholars and scientists to conduct various research related activities. These applications require the knowledge of

Corresponding Author:- Shiny Nair

Address:- M. Tech Student, Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, Kochi, India.

structured query language to query related data. Therefore, it is crucial that the users are competent in the query languages to leverage the vast information available. Users who are not adept in this skill may face challenges to query and understand the data. There are some users who may be able to formulate complex queries, however they may not be aware of structure of the schema, relations between tables, constraints in the data etc. Lack of documentation or annotation increases the complexity of the task. Consequently, users may not be successful in acquiring the required knowledge or relevant information from the available raw data. Generally for data analytics, inexperienced users are requested for data list-pulls that require thorough understanding of the database schema. In all such scenarios, users can be assisted by providing them with database query recommendations. These queries can be considered as templates that can be used as it is, or modified as per the requirement to query the database.

Database recommendation systems can make use of past users session data and query logs to come up with meaningful recommendations to the current user. Recommending queries in a collaborative environment poses additional challenges like maintaining privacy of user database queries, thereby necessitating strict access and security controls in the system.

Background:-

Recommendation systems are very popular in the web context. However, database query recommendations are not explored to the desired extent. Database query recommendation systems are generally considered as content based that use keyword or attribute specific search to generate recommendations. Recommendation frameworks on a high level can be categorized as non-personalized, content based, collaborative or hybrid. In Amazon.com [6], an item to item collaborative filtering approach is adopted to suggest recommendations to the users who visit the online store. As opposed to the user-based or item-based collaborative filtering that focuses on similar users/items, this approach generates recommendations based on the items that users tend to purchase or rate together. In keyword based search interfaces over web databases, the generated recommendations could be very open or too specific [7]. Including preferences considering binary relations or quantitative scores for database tuples can provide personalized or customized answers. FlexPref [8] provides another framework to include the preference functionality inside the DBMS. Context is also considered for preferences that are evaluated using functions that reside within the database engine. The plug and play paradigm is more efficient compared to the conventional layered or built-in approach in preference evaluation. A collaborative query management system [9] outlines the architecture for query recommendations in large scale shared data environments. Data mining techniques with components like query profiler, meta-query executor including administrative tasks like access control, can be used to leverage past users session data for meaningful recommendations. FlexRecs [10] implements a framework above the database query engine to define recommendation workflows. New SQL operators are introduced so that users can conveniently parameterize the workflow to provide customized recommendations. SnipSuggest [11] is a system that provides context aware SQL query completion. The system does not provide complete query suggestions; rather it assists in auto-completion like completing certain clauses of the SQL query. The QueRIE [1]-[5] system considers database query suggestion as a recommendation problem and implements a tuple-based, fragment-based and latent-factor model recommender system.

Database Query Recommendation System:-

Database query recommendation system recommends queries based on past user's session data (or session summary) and a predicted summary that captures the user's interest in the queries [1]-[5]. Based on the signature denoted by the predicted summary, the most similar queries can be suggested as recommendations to the present user. The formation of the predicted summary and consequently the recommendation engine can be based on neighborhood-based collaborative filtering or latent factor model.

Tuple-based – In the Tuple-based recommendation engine [1], [2], the user-based approach is adopted and database tuples form the framework for calculating the predicted summary. Database queries that provide similar result sets are considered as most similar and presented as recommendations. This approach is however not scalable as the predicted summary grows in size with the database. Compressed summaries may lead to loss in accuracy as well.

Fragment-based – In this recommendation engine, an item to item collaborative filtering [3], [4] is introduced wherein queries are suggested based on which query fragments are likely to co-exist together. Fragments of sql queries here refer to syntactical structure of the queries. This approach is clearly scalable compared to the previous one with accepted loss of precision.

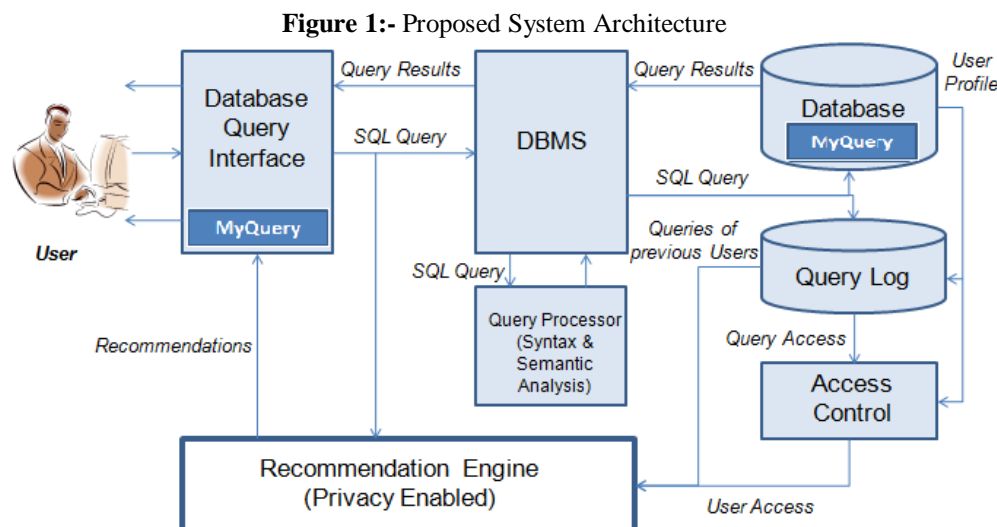
Latent Factor based – A user-item based collaborative filtering is adopted and the recommendations are generated using the matrix factorization techniques [5]. Latent or dormant factors [12] are considered to represent session summaries in terms of the weight of each query fragment in a user session. This approach is more scalable than the previous ones and is feasible for generating recommendations over large volumes of data.

Proposed System:-

The proposed system architecture is outlined in Figure 1. The proposed system enhances the database query recommendation system by enabling access control mechanism for the privacy of users. This is typically useful in a shared database environment where a plethora of users execute various SQL queries for research and analysis or merely for understanding the nature of data. Privileged users get access to a personal space where users can create, alter tables and in general own the space. However, users can still benefit from query recommendations of other users provided others have agreed to share their tables for recommendations. The Recommendation engine (Fragment-based) [3], [4] is based on SQL query fragments. Query fragments and session summaries of past users form the basis for recommending the top-n most similar queries to the current user. The personalized recommendation system also takes into consideration, the queries in the current session of the user.

The access enabled recommendation engine will provide a space that will be accessible only for privileged users of the application. A role based and programmatic control [13] of user access is implemented. Specific privileges are packaged in named roles, and each user is assigned to one or multiple roles. This serves as a simpler and effective approach for assigning and enforcing different sets of privileges and is crucial for access control management in complex applications. In programmatic control, the matrix corresponding to individual database privileges is maintained in a database table and is implemented programmatically to enforce access control rules and decisions. Programmatic controls can be exceedingly fine-grained and can incorporate arbitrarily complex logic for implementing access control decisions within the application. A privileged user will have access to a personal space to explore and experiment with the datasets. The user will also be entitled to hide or share their tables with other users as recommendations.

If a query statement q_i fired by a particular user u_i refers to multiple tables and if u_i chooses not to share at least one of them (that is owned by u_i), then q_i will not be recommended to other users regardless of the similarity calculated. If the user has privilege to access all the database objects referred by a query, then that query will qualify as a candidate for recommendation. Therefore, data objects can be manipulated, altered or dropped only by its owners and can only be shared or granted read privilege to other users in the form of recommended queries.



Experimental Study:-

The system is implemented incorporating primary modules of query pre-processing, building the fragment-based recommendation engine and plugging the privacy preserving feature. Efficient storage methods like Hash maps for session summaries and fragments similarity are used to increase performance. Database built-in indexes are used to leverage the benefits of quick access to data. Efficient algorithms are implemented to ensure that there is no

duplication or redundant data calculation in the recommendation engine. Role based and programmatic access control techniques are used to implement the complex logic of preserving privacy in recommending queries. The real-time SkyServer data can be accessed through the application.

The framework was evaluated using the traces of SkyServer database [14], [15] with traces of 2015 query logs. The query set was divided into training and test set and the precision and recall values calculated for the recommendations. The recommended queries were compared with the unseen query in the test set. The precision, recall and f-score were calculated taking into account the fragments of the recommended and unseen queries.

The precision, recall and f-score for a sample query are shown in Figure 2 and Figure 3. A point in the graph (a, b) means that a percentage of user sessions have prediction accuracy $\geq b$. High precision and recall is achieved for more than one third of the test sessions and acceptable precision for around one half of the test sessions for the coarse grained approach. The recommendation time takes 6 seconds on an average which can be considered acceptable for an interactive database exploration. The prototype for the access enabled recommendations is depicted in Figure 4.

Figure 2:- Evaluation - Precision and Recall for sample query

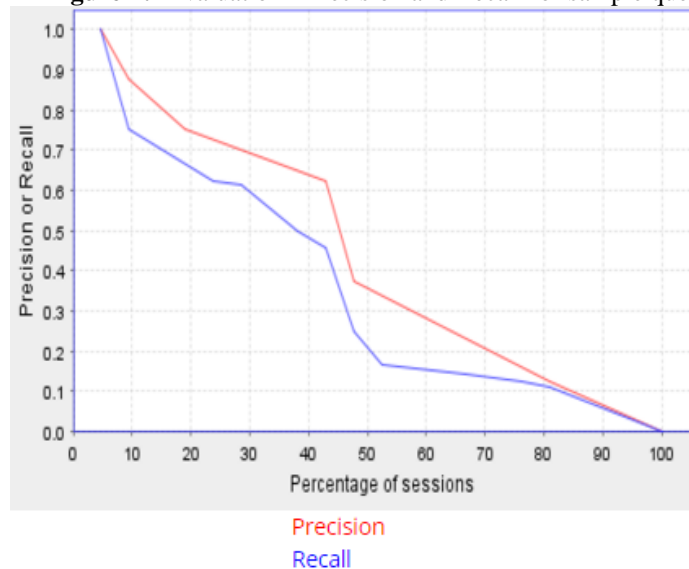
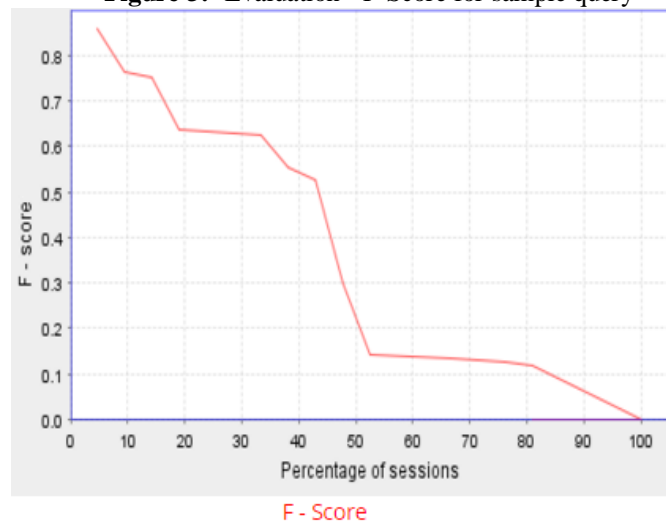
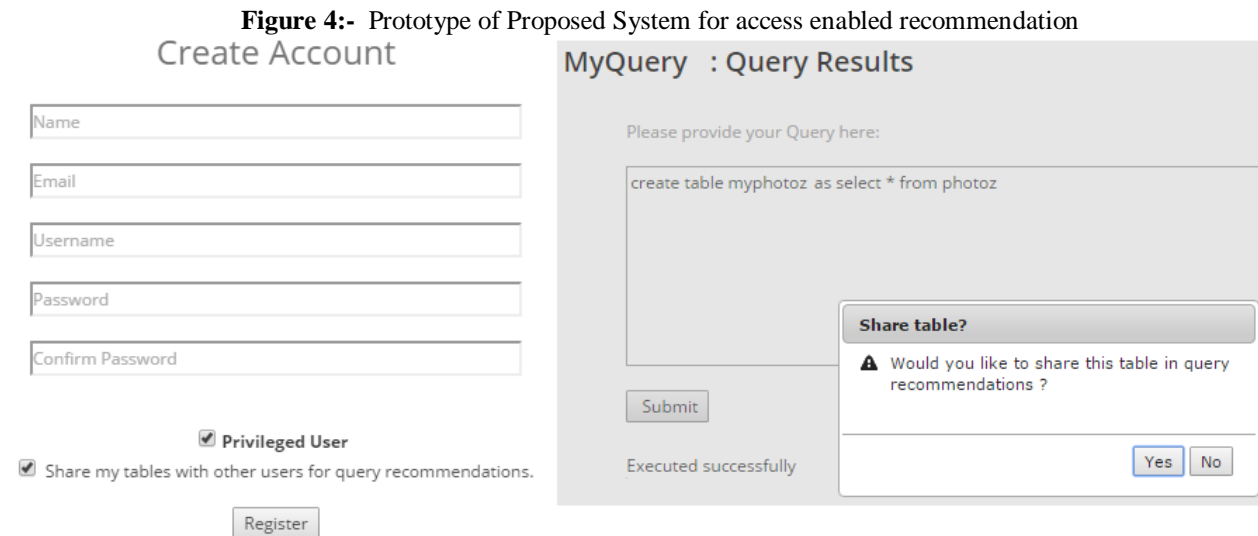


Figure 3:- Evaluation - F-Score for sample query





Conclusion and Future Work:-

The recommendation framework in the context of databases intends to generate SQL query recommendations that are personalized and benefits the relational database users. In the item (or fragment) based method user sessions are summarized using syntactical fragments or structural components of SQL query statements [3], [4]. As the statements from the query logs used are pre-processed and generalized (therefore coarse-grained), a certain level of prediction accuracy loss is expected, however this method is scalable with acceptable levels of accuracy and can be implemented efficiently. Access enabling functionality preserves privacy of users so that the users can control the queries that access tables owned by them to be presented as recommendation to other users. This is useful particularly in shared databases with a number of users having access both to shared data as well as a personal space for research and analysis.

There are many challenges in implementing the security related issues in applications that are web-enabled and have open and shared access to all thereby presenting interesting directions for future enhancements. One is the concept of hidden fields or attributes in table and maintaining access controls at attribute level in recommended queries. Augmenting queries with annotations effectively can also be useful to users in understanding the recommended queries.

References:-

1. G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "Collaborative filtering for interactive database exploration," Proc. 21st Int. Conf. SSDBM, New Orleans, LA, USA, pp. 318, 2009.
2. S. Mittal, J. S. V. Varman, G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "QueRIE: A recommender system supporting interactive database exploration," Proc. IEEE ICDM, Sydney, NSW, Australia, 2010.
3. J. Akbarnejad et al., "SQL QueRIE recommendations," PVLDB, vol. 3, no. 2, pp. 1597-1600, 2010.
4. Magdalini Eirinaki, Suju Abraham, Neoklis Polyzotis, and Naushin Shaikh, "QueRIE: Collaborative Database Exploration", IEEE Transactions on Knowledge And Data Engineering, vol. 26, no. 7, pp.1778-1790, July 2014.
5. M. Eirinaki and S. Patel, "QueRIE reloaded: Using matrix factorization to improve database query recommendations", IEEE International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, pp.1500-1508, Oct./Nov. 2015.
6. G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Comput., vol. 7, no. 1, pp. 7680, Jan./Feb. 2003.
7. G. Koutrika, "Personalized DBMS: An elephant in disguise or a chameleon?," IEEE Data Eng. Bull., vol. 34, no. 2, pp. 2734, Jun.2011.
8. J. Levandoski, M. Mokbel, and M. E. Khalefa, "FlexPref: A framework for extensible preference evaluation in database systems," Proc. IEEE ICDE, Long Beach, CA, USA, 2010.
9. N. Khoussainova, M. Balazinska, W. Gatterbauer, Y. Kwon, and D. Suciu, "A case for a collaborative query management system," Proc. 4th Biennial CIDR, Asilomar, CA, USA, 2009

10. G. Koutrika, B. Bercovitz, and H. Garcia-Molina, "FlexRecs: Expressing and combining exhible recommendations," Proc.SIGMOD Conf., New York, NY, USA, pp. 745757, Jun. 2009.
11. N. Khoussainova, Y. Kwon, M. Balazinska, and D. Suciu, "SnipSuggest: Context-aware autocompletion for SQL," PVLDB, vol. 4, no. 1, pp. 2233, 2011.
12. Koren, Y., Bell, R., Volinsky, C., "Matrix Factorization Techniques for Recommender Systems," IEEE Computer, Vol. 42, No. 8, pp. 30-37, August 2009.
13. D. Stuttard and M. Pinto, The web application hacker's handbook. Indianapolis: Wiley, 2011.
14. Skyserver.sdss.org, "SDSS SkyServer DR12", 2016. [Online]. Available: <http://skyserver.sdss.org/>. [Accessed: 16-Jun-2016].
15. Sciserver.org, "Sciserver - Collaborative Data-Driven Science", 2016. [Online]. Available: <http://www.sciserver.org/>. [Accessed: 16-Jun-2016].