IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. V (Nov – Dec. 2014), PP 29-34
www.iosrjournals.org

A Survey Report on the Novel Approach on Use of Recommendation Outline in Query Recommender System

Pooja K. Akulwar[1], Disha Deotale[2]

[1](*M. E. Research Scholar – Department of Computer Engineering, G. H. Raisoni Institute of Engineering & Technology, Pune, India)*
[2](*Assistant Professor – Department of Computer Engineering, G. H. Raisoni Institute of Engineering & Technology, Pune, India)*

Abstract: *The DBMS applications are becoming increasingly popular in the scientific community to support the interactive exploration of huge data. A result of heavy usage has also lead to a lot of tavles generated in the warehouse. This has tremendously increased the need for recommendation system & various tools for the user as user is not capable to explore such huge data by using Structured Query language. This motivates to turn towards the idea of query recommendation system which will assist the technical & non-technical users to generate top-N query recommendations as per their needs. In this survey paper, the focus is given on opportunities that exist in domain of query recommendation, brief description of recommendation outline and techniques. Along with this, the terminology to identify the similarity and comparison of fragment based approach & matrix factorization approach is also explored.*

Keywords: *Database Management System, Query Recommendation, Recommendation techniques.*

I. Introduction And Motivation

The Database Management Systems are used to organize, maintain and retrieve the huge amount of data. To acquire the essential data the user need to interact with the database by taking the support of declarative Structured Query Language (SQL). Most of times, it is observed that user faces the difficulties in interacting with the database by using SQL query language as the database have hundreds of schema's and thousands of attributes. This problem has been solved with the support of Database Management Applications such as SAP, Easy Query, and Microsoft Access. These applications involve the user interaction and manual editing of the SQL query components but when the user is unacquainted about the underlying database structure then it becomes difficult for the user to formulate the query. To formulate the query, the research work has been done [1] and that results into the query assistant tool which helps the user to either formulate the query or to recommend the relevant query items based on formulated query. This work is the reflection of the web recommendation techniques those are used in gaining the on-line marketing intelligence. The web recommendation techniques and algorithms have their strong relation with the Information Retrieval Science (IR). In IR, the recommendation techniques and algorithms are used to retrieve the most relevant top K documents as per the formulated query.

Most of times the Business Verticals have their own off-line database and the user will be interested in exploration of the database and faces the difficulties. Therefore, it may also possible to extend the idea specified by the authors [1] to provide the Query Recommendation Assistant Tool that enable the user to formulate and refine the SQL query. This take off to cataloged view on how to make the imperative use of IR and the web recommendation techniques to generate and recommend the top K relevant query objects by identifying the similarities among the SQL query fragments. This becomes the novel approach that handles crucial responsibility in proposing the assistant tool to retrieve data over the database.

The rest of the paper is organized as follows: section-2 describes the related work and the section-3 concentrates on recommendation outline. The section-4 describes the recommendation techniques and similarity metrics those have been used in finding the similarity among the query fragments. The section-5 focuses on terminology used to calculate weights of items required for finding similarity. The section-6 states the comparison between the fragment based approach and matrix factorization approach.

II. Related Work

The problem of generating recommendations has been broadly addressed in the web context & only miniature work has been done in the database context.

QueRIE framework [1] -The authors have used the recommendation techniques and algorithms to generate Query Recommendation Framework which acts as an assistant tool for the technical and non technical users to form & redefine the query where user need to retrieve scientific data from Sky Server Database. This framework generates top N personalized query item by using item to item recommendation algorithms.

Survey of collaborative filtering techniques [2]: The authors have introduced the collaborative filtering techniques, its characteristics, challenges and categories of collaborative filtering. The different approaches of finding similarities to generate the top N recommendation by using collaborative filtering are described.

More Like This: Query Recommendation for SQL [3] – The author has described the use of TF-IDF method in finding the similarity between two queries and its associated features to recommend the top N query objects. The author has specified that the TF-IDF is also used in maximizing the diversifying function to generate the recommendations to explore the unseen portion of the database.

Matrix factorization Techniques For Recommender Systems: A Survey [4] - The authors have described matrix factorization method which became leading methodology within collaborative filtering recommendation techniques and it is superior to classical nearest neighbor techniques. This method also offer a compact memory efficient model and useful in collecting feedback in multiple forms.

III. Recommendation Outline

The goal of a recommender system is to provide lists of top N recommended object that is as per user requirement which is evaluated based on predictions. Sometimes the recommendation scores are assigned to the object those are unknown to the user and objects with the highest recommendation scores will be recommended to the active users. The following figure no.3 describes the approaches of the recommendation systems:

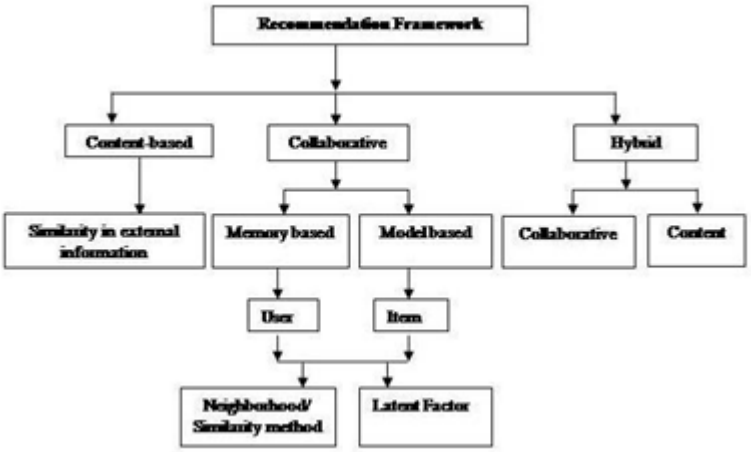

Fig.1. Types of recommendation system

3.1 Content-based filtering–Content-based filtering approach utilizes a series of discrete characteristics of an item in order to recommend additional items with similar properties. In this method, User profile is used which consists of the user preferences that user has given to item in the past. The recommendations are generated by finding similarity between active user and the preferences made to the item that was rated earlier by this active user only. The TF – IDF (Term Frequency – Inverse Document Frequency) method is used to compute the weight of contents.

3.2. Collaborative filtering- Instead of finding similarity between new item and past preferred items, the recommended objects are selected on the basis of past evaluation of large groups of users. This method collects and analyze large amount of information related to user behaviors, activities or preferences. The prediction of users rating is done based on their similarity with ratings of other users. The Collaborative filtering approaches often suffer from three problems: cold start, scalability, and scarcity. Collaborative filtering method is grouped into two general classes: Memory based collaborative filtering and model based collaborative filtering.

3.3. Memory based collaborative filtering – This mechanism uses user rating data to compute similarity between users or items. In this, the algorithm calculates the similarity between two users or items and produces a prediction for the user, taking the weighted average of all the ratings. Similarity computation between items or

users is done by using neighborhood-based algorithm. Multiple mechanisms such as Pearson correlation and vector cosine based similarity are used for this. The Memory-based algorithms are used to overcome the problems faced by collaborative filtering by using the entire database.

3.4 Model based collaborative filtering-Model-based recommendation systems involve building a model, based on the dataset of ratings. In other words, we extract some information from the dataset, and use that as a "model" to make recommendations without having to use the complete dataset every time. This approach potentially offers the benefits of both speed and scalability. In model-based approach, the similarities between users and/or items can be calculated and then stored as a model, and then we can use the stored similarity values to predict ratings. These models can also be built using similarities between items rather than users and in fact, sometimes it is more desirable to do so.

3.5 Hybrid Recommendation - This approach is the combination of the features of collaborative filtering and content-based filtering methods. A good recommender algorithm faces difficulties to address the diverse needs of heterogeneous users and this will be overcome with the support of content or collaborative recommendation methods. Hybrid recommendation method is used to overcome the cold-start problem faced by the content and collaborative filtering methods. This method is used by implementing a collaborative and a content-based method separately and then a prediction of each method is combined to recommend the objects. Sometimes, it may be possible to incorporate the content-based characteristics with collaborative approach to generate recommendation list & vice versa.

IV. Recommendation Techniques

To compute the similarity between two items, the numbers of different mathematical formulas are used. Each formula consists of the terms that are summed over the set of common users U. The different approaches are described as follows:

4.1.1 Cosine-based similarity - The metric measures the similarity between two n-dimensional vectors based on angle between them. The similarity between two items i & j is viewed as corresponding vectors and is formally defined as follows:

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}$$ ---------- (1)

4.1.2 Correlation-based similarity -Correlation-based similarity is also known as Pearson based correlation which is used to compute the similarity between users based upon ratings. This similarity measure is based on how much the rating by common users for a pair of items deviate from average ratings for those items. The Pearson correlation between two user's u and v is shown by the following:

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_j)^2}}$$ ---------- (2)

4.1.3 Adjusted cosine similarity–The basic cosine measure does not take the differences in the average rating behavior of the users into account. This problem is solved by using Adjusted Cosine similarity measure, which subtracts the user average from the ratings. The values for Adjusted Cosine measure correspondingly range from -1 to +1.Let U be the set of users that rated both items i & j. The Adjusted cosine similarity is calculated as:

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_u)^2}}$$ ---------- (3)

4.1.4.Jaccard's similarity coefficient: It is a statistic used for comparing the similarity and diversity of sample sets.

The Jaccard's coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

$$J(A,B) = \frac{|A\cap B|}{|A\cup B|}.$$ ---------- (4)

(If A and B are both empty, we define J (A, B) = 1.) Clearly, $0 \leq J(A,B) \leq 1.$

V. Terminology

In order to identify the similarities among the users, the weighting scheme will be applied. The similarities will be identified by computing the weights of SQL Fragments. There are two types of the weight scheme: 1) binary weight scheme 2) Result Based Weight Scheme. In binary weighting scheme, the weight is assigned in terms "1" or "0". If the SQL fragment exists then a weight of "1" is assigned and if a SQL fragment does not exit then a weight of "0" is assigned. With this method, it may not be possible to compute the accurate weight and to overcome this limitation, the result based weighting scheme will be used. In this method, the weight will be computed by using the TF-IDF method.

Weighted Scheme TF-IDF – TF-IDF is called as the Term Frequency – Inverse Document Frequency (TF-IDF) which is a classical method and used in Information Retrieval (IR) and Text Mining. The TF-IDF weight is statistical measure which is used to evaluate the importance of a word in document collection. The variations of TF-IDF weighting schemes are often used by search engines in computing the scoring and creating the rank to a document relevant to a given user query.

Term Frequency Scheme (TF) - It measures how frequently a term occurs in a document. In IR, the weight of a term ti in a document dj is the number of times the term ti appears in dj and denoted by Fij.

TF(t) = Number of times term t appears in a document / Total number of Terms in the document. ------- (1)

Inverse Document Frequency Scheme (IDF) –This scheme is used to measure the importance of the term "t" in a document and it is computed by using the following way.

IDF(t) = Number of Documents / Number of Documents with term t in it. -------- (2)

Weight of a Term - The weight of term is computed by using following formula:

Weight(t) = TF(t) * IDF(t) ---------- (3)

Example – Suppose there are two documents containing the certain terms, for example suppose it contains the word like this, is, another, example and a, etc. To compute the TF let us consider the following term frequency tables: Table No.2, Table No.3 that contains the terms collected from two documents which are listed on the right and the left it contains the term counts.

Document 1		Document 2	
Term	**Term Count**	Term	**Term Count**
This	1	this	1
Is	1	is	1
A	2	another	2
Sample	1	example	3

TF, in its basic form, is just the frequency of the term "this" appears in the respective document. Therefore the TF-IDF value for the term "this" is defined as TF (this, D1) = 1.

IDF is a bit more involved:

$$\mathrm{idf}(\mathsf{this}, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

---------- (4)

The numerator of the fraction is the number of documents and it is two. The number of documents in which "this" appears is also two, giving the IDF of term "this" is shown in equation number -5. This equation shows that, the TF-IDF value of a term "this" is zero and it indicates that this term is the most similar term

across all the documents.

$$\mathrm{idf}(\mathsf{this}, D) = \log \frac{2}{2} = 0$$

---------- (5)

The term count for the word "example" is three but it occurs only in the Document Number 2, therefore the TF–IDF of this term is given below:

$$\mathrm{tf}(\mathsf{example}, d_2) = 3 \qquad \mathrm{idf}(\mathsf{example}, D) = \log \frac{2}{1} \approx 0.3010$$

$$\mathrm{tfidf}(\mathsf{example}, d_2) = \mathrm{tf}(\mathsf{example}, d_2) \times \mathrm{idf}(\mathsf{example}, D) = 3 \log 2 \approx 0.9030$$

VI. Comparison Between Fragment Based Approach And Matrix Factorization

6.1. Fragment based approach**:** In query recommendation, the items are considered as fragments of SQL statements. These fragments are used to find out the similarities between the queries posed by the different users [1]. This approach is based on the pair-wise similarity among the items involved in the recorded user sessions. Items that co-appear in many sessions are considered similar to each other and these similarities are used in order to generate recommendations for an active session. This technique allows the calculation of all similarities offline, thus accelerating the real-time calculations and enabling fast recommendations' generation. These similarities are subsequently used to predict, in real time, the "ranking" (i.e. importance) of each fragment with regards to the current user session. In turn, the highest ranked query fragments are selected and used to retrieve queries that include them, which are used as recommendations. The most relevant queries are recommended as a top N query objects.

6.2. Matrix Factorization: The neighborhood method focuses on either finding the relationship between users or between the items. There must be some latent features that determine how a user rates an item. Example: 2 users give high ratings to certain movie if they both like actors/actresses of movie or if movie is an action movie. Hence if these latent features are discovered then it becomes possible to predict a ratings with respect to certain user and certain items as features associated with user should match features associated with them. Therefore, the latent factor model is another approach that finds the ratings by characterizing both items and users. Matrix Factorization is based on latent factor model which finds out the hidden ratings under the data. This is more effective as it allows discovering the latent features underlying the interactions between users and items. Matrix Factorization factorizes a matrix such that two more matrices are generated and when combined then gives original matrix.

VII. Conclusion

With this survey it is observed that with the support of well established Web Recommendation Algorithms and Techniques it is possible to design the Recommendation System to explore the Database interactively by using the Structured Query Language (SQL). Most of the existing Database Management Applications provide GUI interface to formulate queries to retrieve data from database. However, the design of required queries depends on user's manual editing of the query characteristics and if user is not familiar with the database schemas then multiple underlying attributes will confuse them. Even if the user is technically sound and has database knowledge then it is also a big challenge for them to issue complex queries over large datasets to discover the required knowledge. As a result, different ways are provided to interact with the database by formulating queries that result into top N recommended list of correlated queries through the supportive framework called as Query Recommendations. The one of the frequent use of Query Recommendation system is found to interact with scientific database, since such database explored by technical as well as non technical users to retrieve the information. Therefore, query recommendation applications motivate to extend this idea in business environment where most of times user faces difficulties in exploring the data through use of SQL queries. This motivates to design and build the assistant tool in a simple way by using well known web recommendation ways that enables user to explore the database interactively
to get top N personalized query object recommendations.

References

[1]. M. Eririnaki, S. Abraham, N. Polyzotis, and N. Shaikh,"QueRIE: Collaborative Database Exploration", IEEE Transaction on Knowledge and Data Engineering, 2013.

[2]. X. Su, and T. M. Khoshgoftarr, Review Article-"A Survey of Collaborative Filtering Techniques", in Advances in Artificial Intelligence", vol. 2009, Article Id 421425, Aug 3, 2009.

[3]. Christopher Miles, "More like This: Query Recommendation for SQL".

[4]. Y. Koren, Yahoo Research, and R. Bell, C. Volinsky, "Matrix factorization techniques for recommender system", in IEEE Computer Society, AT & T Labs, 2009.

[5]. L. Lu, M. Medo, C.H. Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and T. Zhou,"Recommender System", Feb 7,

[6]. 2012.

[7]. G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", IEEE Internet Computing, vol. 7, no.1, pp. 76–80, Jan/Feb 2003.

[8]. L. Tang, T. Li, Y. Jiang, and Z. Chen,"Dynamic Query Forms for Database Queries", IEEE Transaction on Knowledge and Data Engineering, 2013.

[9]. Z. Chen, T. Li, and Y. Sun, "A Learning Approach to SQL Query Results Ranking Using Skyline and User's Current Navigational Behavior", IEEE Transactions on Knowledge And Data Engineering, vol.25, December 2013.

[10]. J. Fan, G. Li, and L. Zhou, "Interactive SQL Query Suggestions: Making Database User-Friendly", Department of Computer Science and Engineering, Tsinghua University.