

Efficient Prediction of Difficult Keyword Queries using Collaborative Filtering Recommendation Framework

¹SK.Asif ali & ²S.Naga Lakshmi

¹M.Tech (CSE), Priyadarshini Institute of Technology & Science

² Assistant Professor (Dept.of CSE), Priyadarshini Institute of Technology & Science

Abstract: -

Keyword queries on databases provide easy access to data, but often suffer from low ranking quality, i.e., low precision and/or recall, as shown in recent benchmarks. Collaborative filtering (CF) is significant and admired technology for recommender systems. Recommender frameworks have been turned out to be significant means for web online clients to adapt to the information overload and have ended up a standout amongst the most effective and prevalent tools in electronic commerce. Recommending and personalization are critical ways to deal with combating information overload. Machine Learning is an imperative piece of frameworks for these assignments. Collaborative filtering has issues, substance based routines address these issues integrating both is best.

Keywords: Collaborative Filtering; Content-based Recommender System; Neighbor Selection

1 INTRODUCTION

Keyword query interfaces (KQIs) for databases have attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Since any entity in a data set that contains the query keywords is a potential answer, keyword queries typically have many possible answers. KQIs must identify the information needs behind keyword queries and rank the answers so that the desired answers appear at the top of the list. Unless otherwise noted, it refers to keyword query as query in the remainder of this project. Databases contain entities, and entities contain attributes that take attribute values. Some of the difficulties of answering a query are as follows: First, unlike queries in languages like SQL, users do not normally specify the desired schema element(s) for each query term. For instance, query Q1: Godfather on the IMDB database (<http://www.imdb.com>) does not specify if the user is interested in movies whose title is Godfather or movies distributed by the Godfather Company. Thus, a KQI must find the desired attributes associated with each term in the query. Second, the schema of the output is not specified, i.e., users do not give enough information to single out exactly their

desired entities. For example, Q1 may return movies or actors or producers. It is important for a KQI to recognize such queries and warn the user or employ alternative techniques like query reformulation or query suggestions. It may also use techniques such as query results diversification. To the best of our knowledge, there has not been any work on predicting or analyzing the difficulties of queries over databases. Researchers have proposed some methods to detect difficult queries over plain text document collections. However, these techniques are not applicable to our problem since they ignore the structure of the database. In particular, as mentioned earlier, a KQI must assign each query term to a schema element(s) in the database. It must also distinguish the desired result type(s). Recommendation systems found their application in the field of e-commerce and internet where items suggest to a group of user on the basis of their requirement based on their area of interest. A recommendation system is an information filtering system that built a model from the characteristic of an item according to the rating or prediction, given by a user to an item. Recommendation system has an important component in social media sites (such as Amazon, IMDB, Movie Lens), social sites giants such as Amazon have been greatly gained from the



capability of their recommenders in accurately delivering the correct item to the correct user [1]. Collaborative filtering (CF) is an important and popular technology for recommender system. CF methods are classified into user based CF and item-based CF. The basic idea of user-based CF approach is to find out a set of users who have similar favor patterns or interest to a given user and the basic idea of item-based CF approach is to find out a set of items having highest correlation with the given item. In reality, people may like to group items into categories, and for each category there is a corresponding group of people who like items in the category. Cognitive psychologists find that objects (items) have different typicality degrees in categories in real life. But these collaborative filtering methods have facing some problems.

II. RELATED WORK:

Prediction of query performance has long been of interest in information retrieval. It is invested under a different names query difficulty, query ambiguity and sometimes hard query. Keyword Searching and Browsing in Databases using BANKS [4] describe techniques for keyword searching and browsing on databases that we have developed as part of the BANKS system (BANKS is an acronym for Browsing AND Keyword Searching). The BANKS system enables data and schema browsing together with keyword-based search for relational databases. BANKS enables a user to get information by typing a few keywords, following hyperlinks, and interacting with controls on the displayed results; absolutely no query language or programming is required. The greatest value of BANKS lies in near zero-effort web publishing of relational data which would otherwise remain invisible to the web. BANKS may be used to publish organizational data, bibliographic data, and electronic catalogs. Search facilities for such applications can be hand crafted: many web sites provide forms to carry out limited types of queries on their backend databases. For example, a university web site may provide form interface to search for faculty and students. Searching for departments would require

yet another form, as would search for courses offered. Creating an interface for each such task is laborious, and is also confusing to users since they must first expend effort finding which form to use. Efficient IR-Style Keyword Search over Relational Databases [2] A key contribution of this work is the incorporation of IR-style relevance ranking of tuple trees into our query processing framework. In particular, our scheme fully exploits single-attribute relevance-ranking results if the RDBMS of choice has text-indexing capabilities (e.g., as is the case for Oracle 9.1, as discussed above). By leveraging state-of-the-art IR relevance-ranking functionality already present in modern RDBMSs, we are able to produce high quality results for free-form keyword queries. For example, a query [disk crash on a net vista] would still match the comments attribute of the first Complaints tuple above with a high relevance score, after word stemming (so that “crash” matches “crashed”) and stop-word elimination (so that the absence of “a” is not weighed too highly).

III. SYSTEM ANALYSIS:

3.1. Presented System:

Existing Work As per our recent literature reviews, there has not been any work on predicting or analyzing the difficulties of queries over databases. Researchers have proposed some methods to detect difficult queries over plain text document collections recently. But techniques are not applicable to our problem since they ignore the structure of the database. There are two categories of existing methods, pre-retrieval and post-retrieval for predicting the difficulties of query. But below are limitations of this method: - Pre-retrieval methods are having less prediction accuracies. - Post-retrieval methods are having better prediction accuracies but one requires domain knowledge about the data sets to extend idea of clarity score for queries over databases. - Each topic in a database contains the entities that are about a similar subject. - Some Post-retrieval methods success only depends on the amount and quality of their available training data

3.2. Recommender Systems

Frameworks for recommending items (e.g. books, films, CD's, site pages, newsgroup messages) to clients taking into account cases of their inclinations. Numerous on-line stores give suggestions (e.g. Amazon, CDNow). Recommenders have been appeared to considerably increase deals at on-line stores. There are two essential ways to deal with recommending: Collaborative Filtering (a.k.a. social filtering), Substance based

Personalization

- Recommenders are occurrence of personalization software.
- Personalization distresses get used to to the being needs, interests, and favorite of each user.
- Includes:
 - Recommending
 - Filtering
 - Predicting (e.g. form or calendar appt. completion)

From a business perspective, it is viewed as part of Customer Relationship Management (CRM).

Machine Learning and Personalization

- Machine Learning can allow learning a user model or profile of a particular user based on:
 - Sample interaction
 - Rated examples
- This model or profile can then be used to:
 - Recommend items
 - Filter information

Predict behavior

3.3. Collaborative Filtering

Maintain a database of numerous clients' ratings of an assortment of items. For a given client, find other comparative clients whose ratings unequivocally associate with the current user. Recommend items

appraised vary by these comparative clients, however not evaluated by the current user. Almost all existing commercial recommenders utilize this methodology (e.g. Amazon).

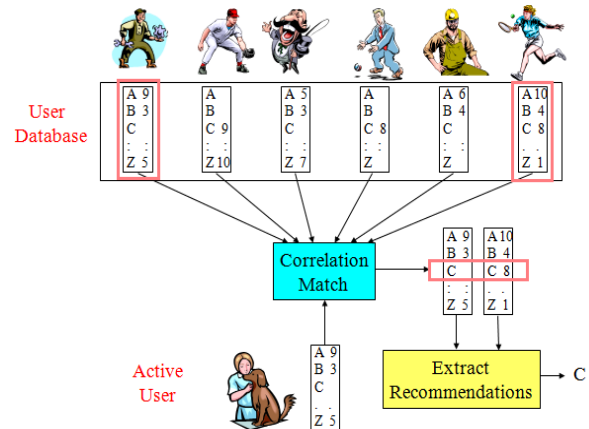


Fig 1. Collaborative Filtering

- Weight all clients as for comparability with the dynamic user. Select a subset of the clients (neighbors) to use as predictors. Normalize ratings and process an expectation from a weighted combination of the selected neighbors' ratings. Present items with most noteworthy anticipated ratings as proposals.
- Typically use Pearson correlation coefficient between ratings for active user, a , and another user, u .

$$c_{a,u} = \frac{\text{covar}(r_a, r_u)}{\sigma_{r_a} \sigma_{r_u}}$$

r_a and r_u are the ratings vectors for the m items rated by

both a and u

$r_{i,j}$ is user i 's rating for item j

Covariance and Standard Deviation

Covariance:

$$\text{covar}(r_a, r_u) = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sum_{i=1}^m 1} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

$$\bar{r}_x = \frac{\sum_{i=1}^m r_{x,i}}{m}$$

Standard Deviation:

$$\sigma_{r_x} = \sqrt{\frac{\sum_{i=1}^m (r_{x,i} - \bar{r}_x)^2}{m}}$$

3.4. Neighbor Selection

For a given dynamic client, a , select connected clients to serve as the wellspring of predictions. The standard methodology is to utilize the most comparative n clients, u , taking into account closeness weights, $w_{a,u}$. Exchange methodology is to include all clients whose similitude weight is over a given limit.

- For a given dynamic client, a , select related clients to serve as the wellspring of forecasts.
- Standard approach is to use the most similar n users, u , based on similarity weights, $w_{a,u}$.
- Alternate approach is to include all users whose similarity weight is above a given threshold.

Rating Prediction

- Predict a rating, $p_{a,i}$, for each item i , for active user, a , by using the n selected neighbor users, $u \in \{1, 2, \dots, n\}$.
- To represent clients diverse ratings levels, base expectations on contrasts from a client's normal rating.
- Weight clients' ratings commitment by their closeness to the active user.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n w_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u=1}^n w_{a,u}}$$

4. CONTENT-BOOSTED COLLABORATIVE FILTERING

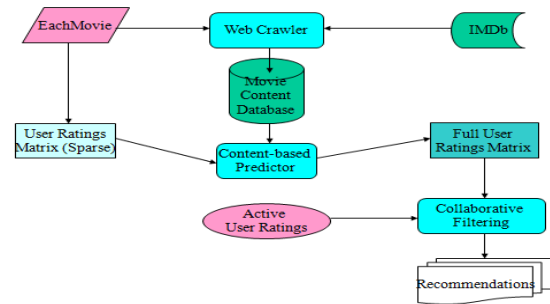


Fig 2. Content-boosted collaborative filtering architecture

Content-Boosted CF – I

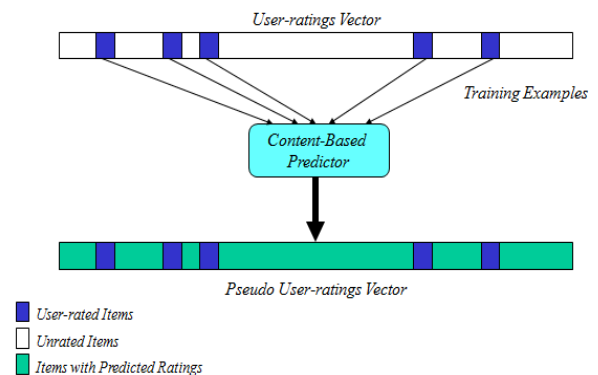


Fig 3. Content-Boosted CF – I

Problems with Collaborative Filtering

Cold Start: There should be sufficient different clients as of now in the framework to find a match.

Sparsity: If there are numerous items to be suggested, regardless of the fact that there are numerous clients, the client/ratings lattice is inadequate, and it is elusive clients that have appraised the same items.

First Rater: Can't suggest a thing that has not been already appraised.

New items

– Esoteric items

Popularity Bias: Can't prescribe items to somebody with special tastes.

Tends to recommend popular items.



4.1. Content-Based Recommending

Proposals depend on information on the substance of items as opposed to on other clients' opinions. Utilizes a machine learning calculation to induce a profile of the client's inclinations from samples taking into account a featural depiction of the substance.

Some previous applications:

Newsweeder (Lang, 1995), Syskill and Webert (Pazzani et al., 1996)

Advantages of Content-Based Approach

- No need for data on other users.
 - No cold-start or sparsity problems.
- Able to recommend to users with unique tastes.
- Able to recommend new and unpopular items
 - No first-rater problem.
- Can provide explanations of recommended items by listing content-features that caused an item to be recommended.

Disadvantages of Content-Based Method

Requires content that can be encoded as meaningful features. Users' tastes must spoke to as a learnable capacity of this substance features. Unable to adventure quality judgments of different users. Unless these are some way or another included in the substance highlights.

5. SYSTEM IMPLEMENTATION

Stop words removed from all bags. A book's title and author are added to its own related title and related author slots. All probabilities are smoothed using Laplace estimation to account for small sample size. Lisp implementation is quite efficient: Training: 20 exs in 0.4 secs, 840 exs in 11.5 secs ,Test: 200 books per second

Explanations of Profiles and Recommendations

- Feature strength of word w_k appearing in a slot s_j :

$$\text{strength}(w_k, s_j) = \log \frac{P(w_k | \text{positive}, s_j)}{P(w_k | \text{negative}, s_j)}$$

6. CONCLUSION

In this paper we investigate on Collaborative filtering (CF). Recommending and personalization are imperative ways to deal with combating information overload. Machine Learning is a vital piece of frameworks for these assignments. Community oriented filtering has issues. Substance based techniques address these issues (yet have issues of their own). Integrating both is best.

REFERENCES

- [1] Z. Huang, H. Chen, and D. Zeng, "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering," ACM Trans. Information Systems, vol. 22, no. 1, pp. 116-142, 2004.
- [2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.
- [3] K.M. Galotti, Cognitive Psychology In and Out of the Laboratory, third ed. Wadsworth, 2004.
- [4] G.L. Murphy, The Big Book of Concepts. MIT Press, 2002.
- [5] L.W. Barsalou, Cognitive Psychology: An Overview for Cognitive Scientists. Lawrence Erlbaum Assoc., 1992.
- [6] S. Schiffer and S. Steele, Cognition and Representation. Westview Press, 1988.
- [7] D.L. Medin and E.E. Smith, "Concepts and Concept Formation," Ann. Rev. of Psychology, vol. 35, pp. 113-138, 1984.



[8] W. Vanpaemel, G. Storms, and B. Ons, “A Varying Abstraction Model for Categorization,” Proc. Cognitive Science Conf. (CogSci’05), pp. 2277-2282, 2005.

[9] C. Hauff, V. Murdock, and R. Baeza-Yates, —Improved query difficulty prediction for the Web, in Proc. 17th CIKM, Napa Valley, CA, USA, 2008, pp. 439–448

[10] Y. Zhou and W. B. Croft, —Query performance prediction in websearch environments, in Proc. 30th Annu. Int. ACM SIGIR, NewYork, NY, USA, 2007, pp. 543–550.

[11] C. Hauff, L. Azzopardi, and D. Hiemstra, —The combination and evaluation of query performance prediction methods, in Proc. 31st ECIR, Toulouse, France, 2009, pp. 301–312. [12] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow, —Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval, in Proc. 28th Annu. Int. ACM SIGIR Conf. Research Development Information Retrieval, Salvador, Brazil, 2005, pp. 512–519.

[13] J. A. Aslam and V. Pavlu, —Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions, in Proc. 29th ECIR, Rome, Italy, 2007, pp. 198–209.

[14] A. Trotman and Q. Wang, —Overview of the INEX 2010 datacentric track, in 9th Int. Workshop INEX 2010, Vught, The Netherlands, pp. 1–32,

[15] T. Tran, P. Mika, H. Wang, and M. Grobelnik, —Semsearch ‘S10, in Proc. 3rd Int. WWW Conf., Raleigh, NC, USA, 2010.