Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.217 – 222

RESEARCH ARTICLE

Clustering For Collaborative Filtering Application in Web Recommendations

M. Kanimozhi [1], K. Logeshwari [2], K. Saranya [3], D. Sowbha [4], Mr. R. Velumani [5]
[1,2,3,4]Department Of Computer Science and Engineering, Anna University Chennai, India
[5]Assistant Professor, Department Of Computer Science and Engineering,
K.S.R. College Of Engineering, Tiruchengode, India

[1]kanimozhi2101@gmail.com; [2]logeshwari53@gmail.com;
[3]saranksrce@gmail.com; [4]sowbhacs105@gmail.com

Abstract— This paper mainly presents about the explosion of various contents generated on the web, Recommendation techniques have become necessary. Many types of data sources are used for recommendations and these data sources can be modeled in the form of graph. Aiming at providing the general limit on mining web graphs, we illustrate the generalization of different recommendation problems into our graph diffusion framework which can be utilized in many recommendation techniques on the World Wide Web including query suggestion and image recommendation.

Key Terms: - Recommendation; Query suggestion; Generalization; Diffusion and Image recommendation

I. INTRODUCTION

In modern world the user generated information is more free style and less structured, which increases the difficulties in mining useful information needs of web users. To improve the user experience in many web applications, Recommender systems have been developed which mainly based on Collaborative filtering. It is a technique that automatically predicts the interest of active users by collecting rating information from other similar users or items. The Under lying assumption of the collaborative filtering the active users will prefer the those item that the other similar users prefer. Based on this effective intuition, this collaborative filtering has been employed in some large, well-known commercial systems. The typical collaborative filtering algorithms require a user-item rating preferences to infer the users characteristics. The rating datas are always unavailable since the information on the web is less structured and more diverse. Many types of data sources are used for recommendations in most of the cases these data sources can be modeled in the form of graph. If we design a general graph recommendation algorithm, we can solve many recommendation problems on the web.

In Query suggestion there are several outstanding issues that can degrade the quality of the recommendations. The ambiguity is one which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithm which do not satisfy the information needs of the users and also the short queries as reported in [4] and [5] which are more likely to be ambiguous may confuse the algorithm where the users have to rephrase their queries constantly. Based on the survey taken over three months in the commercial search engine query log's it is observed that 19.4 percent of Web queries are single term and 30.5 percent web queries consists of only two terms. Most of the existing methods are complicated and needed to unify the recommendation tasks on the web. Personalization is desirable for many scenarios where the different users have different information needs. The designed recommendation algorithm is scalable to very large data sets.

© 2013, IJCSMC All Rights Reserved

II. BACKGROUND

The last challenge is that it is time consuming and inefficient to design different recommendation algorithms for different recommendation tasks. Moreover, most of existing methods are complicated and require large number of parameters. In this paper, aiming at solving these problems, we propose a general framework for the recommendations on the web as discussed below. All the pre-defined method is for only small amount of data and our concept is to achieve for larger set of data.

III. LITERATURE REVIEW

G.Jen and J.Widom, in their paper " SimRank[1]: A Measure of Structural-Context Similarity," checks whether the two URLs are clicked as a result of several similar queries and then iteratively update the similarities until they come closer.-H. Yang, P.-T. Wu, C.-W.Lee, K.-H. Lin, W.H.Hsu in their paper "Context Seer[2]: Context Search and Recommendation at Query Time for Shared Consumer Photos," by employing the Flickr data set,Yang et al. proposed a context-based image search and recommendation method to improve the image search quality and recommend related images and tags in which the complexity is very high and it's not possible to very large scale data set. We proposed D_{Rec} algorithm which means recommendation by diffusion which increases the recommendation accuracy for about 19.81 percent comparing with SimRank algorithm.

IV. COLLABORATIVE FILTERING

Collaborative filtering is a technique that automatically predicts the interest of active users by collecting rating information from other similar users or items. Collaborative filtering approaches in which neighborhood based approach is most widely used. Neighborhood collaborative filtering includes two type of approach.

1. User-based approach
2. Item-based approach

User-based approaches predict the rating of active users based on the ratings of their similar users and Item-based approaches predict the rating of active users based on the computed information of items similar to those chosen by the active user. Both the User based and item-based approaches often use the Pearson Correlation Coefficient algorithm (PCC) as the similarity computation methods. This PCC based collaborative filtering generally can achieve higher performance than any other popular algorithm because this considers the differences of the user rating style. In web the rating data are always unavailable since the information on web is less structured and more diverse. There are different methods which all focus the user-item rating matrix using low-rank approximations which can be used to make further prediction. He premise behind these low dimensional factor models is that there are only a small number of factors influencing preferences, and that a user preference vector is determined by how each factor applies to that user. Here the query suggestion algorithms cannot be applied directly to most of the recommendation tasks on the web like query suggestion and image recommendation.

V. QUERY SUGGESTION

Query suggestion is closely related to query expansion or query substitution, which extends the original query with new search terms. In order to recommend similar queries to Web users, a valuable technique, query suggestion, has been employed by some commercial search engines. But different from query expansion, query suggestion aims to suggest full queries that have been formulated by previous users so that query integrity and coherence are preserved in the suggested queries. In the local documents such as query –dependent documents and the global documents such as the whole corpus are all employed in query expansion by applying the measure of global analysis to the selection of query terms in local feedback.

Although most of the experimental results show that this method is more effective than global analysis, it performs worse than the query expansion method based on user interactions recorded in user logs. These methods employ different kinds of data sources like documents, anchor texts, query logs, etc. for such suggested queries. Comparing the anchor text and Web queries is another approach, in which both the anchor text and the web queries are all highly similar. Since this method is only designed for query suggestion, the extendibility of this method is very limited as in [6]. Cao et al[7] developed the context- aware query suggestion method by mining click through and suggestion data. The query sessions are employed to build a concept sequence suffix tree for query suggestion. Recently, Mei et al. proposed a general query suggestion method using hitting time on the query-click bipartite graph. This method can generate semantically relevant queries to users' information needs. The main advantage of this work is that it can suggest some long tail queries to users.

VI. QUERY SUGGESTION ALGORITHM

1. A converted bipartite graph $G = (V+ \cup V *,E)$ consists of query set $V+$ and URL set $V *$. The two directed edges are weighted.

2: Given a query q in $V+$, a subgraph is constructed by using depth-first search in G. The search stops when the number of queries is larger than a predefined number.

3: As analyzed above, set $\alpha = 1$, and without loss of generality, set the initial heat value of query q $fq(0) = 1$ (the choice of initial heat value will not affect the suggestion results). Start the diffusion process using f(1) = $e\alpha$Rf(0).

4: Output the Top-K queries with the largest values in vector f(1) as the suggestions.

VII. QUERY SUGGESTION RESULTS

The proposed algorithm for the query suggestion result is called as $D_{Rec,}$ which means the recommendations by Diffusion and comparing it with SimRank[1], Forward Random Walk[2] and Backward Random walk[3]. An example for our query suggestion result is that if our required is a technique like java the recommendations such as virtual machine and the sun micro system are produced and then later the details of the company name are suggested. In such a way the recommendations for our related test queries like name, the delivery of the product and the details of an estate are all produced.

It is clear that the query suggestion result generated by our method is as good as those generated by a commercial search engine. It is observed that our recommendation algorithm not only suggest the queries that are similar to the test queries but also provides latent semantically relevant result for our query . Since the data set that we use is different from the data sets that these commercial search engines employ. It is difficult to quantitatively evaluate our result with those from the commercial search engines.

The D_{Rec} method is compared with other approaches like SimRank[1], Forward Random Walk and Backward Random Walk. In the method of SimRank the query-URL bipartite graph is used to measure the similarities between the queries, then the top-5 similar queries are recommended to the users based on their similarities. Here the similarities between the URLs is calculated and then compute the same queries based On the similarities of URLs. In the Forward and Backward Random Walk the top ranked queries are used as suggestions. In this paper a manual evaluation is conducted by three human experts and also by ODP[10] databases. In the manual evaluation three PhD students are evaluates without any overlaps with the author and those three experts may not know each other and the result produced by them for the D_{Rec} algorithm is 19.81,13.0,7.5 percent more accurate than SimRank, Forward Random Walk and Backward Random Walk. When the user types the query in ODP besides site matches we can also find categories matches in the form of paths between the directories. It is shown that how the D_{REC} algorithm is efficient than the other three methods like simrank, Forward Random Walk and Backward Random Walk.

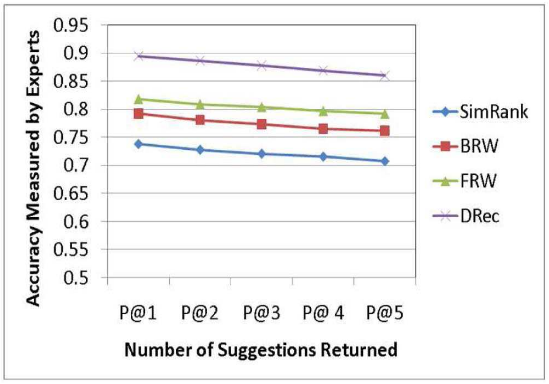

Fig . 1 Accuracy comparison measured by experts

ODP is also known as dmoz which is one of the largest human-edited directories of the web. Here the quality of suggested queries is used for evaluation, the similarity between the two queries is measured by the most similar categories of two queries among the top-5 queries and the accuracy is about 22.45, 11.9 and 7.5.

© 2013, IJCSMC All Rights Reserved

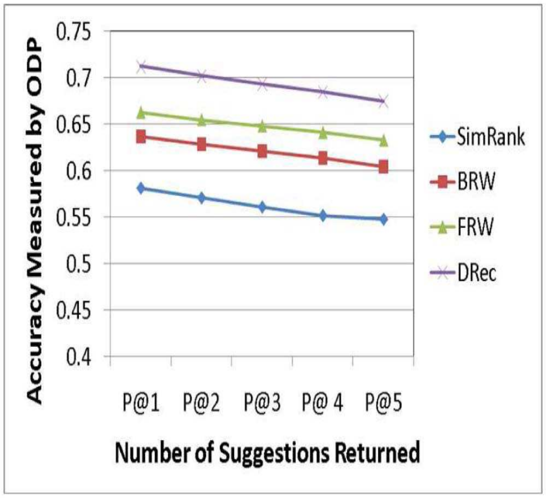

Fig. 2 Accuracy measured by ODP

VIII. Impact of Subgraph

We will perform our algorithm on a subgraph extracted from the original graph. Hence, it is necessary to evaluate how the size of this subgraph affects the recommendation accuracy as shown in fig.

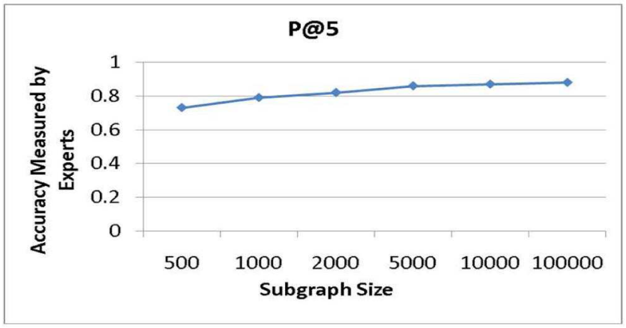

Fig. 4 Impact of the size of sub graph

This shows the performance changes with different subgraph sizes. We observe that when the size of the graph is very small, like 500, the performance of our algorithm is not very good since this subgraph must ignore some very relevant nodes. When the size of subgraph is increasing, the performance also increases. We also notice that the performance on subgraph with size 5,000 is very close to the performance with size 100,000. This indicates that the nodes that are far away from the query node are normally not relevant with the query node.

IX. Image Recommendation

Another interesting recommendation application on the Web is image recommendation which focus on recommending interesting images to Web users based on user’s preference. Normally, these systems first ask users to rate some images as they like or dislike, and then recommend images to the users based on the taste of the users.

However, the quality of recommendations can be evaluated along a number of dimensions, and trusting on the accuracy of recommendations alone may not be enough to find the most relevant items for each User, one of the goals of recommender system is to provide a user with highly personalized items, and more different recommendations result in more opportunities for users to get recommended items. In personalized image recommendation we can set all the images submitted by specified user as the source so that the recommended image is obtained.

An example for image recommendation for a picture which is taken from Grand Canyon, a national park in United States.

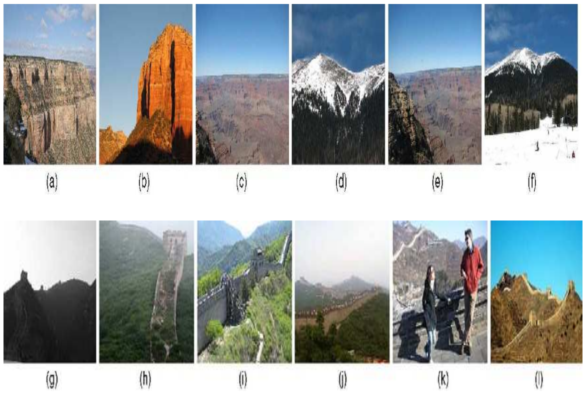

Fig. 4 Relevant Image Recommendations

In the above figure fig (a) shows the required picture of the user and the following fig (b)-(f) shows the corresponding recommendations following the fig (a). We can observe that these recommendations are all latent semantically related to the original picture. This shows the effectiveness of our work. Fig (g) is also an another example the images from fig (h)-fig(l) are the corresponding recommendations for fig (g).

With this motivation, some studies proposed new recommendation methods that can increase the diversity of recommendation sets for a given individual user. So the user can give the feedback of such items on their own. The personalized image recommendation is more important in many applications since it is the best way to understand all information needs from different users. The quality of the personalized image recommendation is evaluated in which the 10 groups are created. Group 1 means all the user only submitted 1 images. Then randomly select 50 users from the user list for each group, hence totally we have 500 users. After generating the results, three experts are asked to rate these recommendations. We again define a 6-point scale (0, 0.2, 0.4, 0.6, 0.8, and 1) to measure the relevance between the testing images and the suggested images, in which 0 means "totally irrelevant" while 1 indicates "entirely relevant". From this method it is observed that if the number of images increases, the recommendation quality will also be increased.

X. Methodology

We are using $D_{Rec\ algorithm}$ in which the accuracy of our search has been increased. First the user registration has to be done, and then the user can login into the system by using their username and password. The overall concept has been implemented in online shopping where the users can purchase their required product, and also they can give feedback about the product based on own suggestion. Based on the feedback given by users the product is ranked and listed depends on their ranks.

URL search and image search is also implemented, if the user is searching any URL the top n URL will be displayed, from that the user can click any link then the homepage of that link will be opened. So while searching, the inefficient data will get reduced. The admin can also login into our system in which the admin can upload the images and he can add, update and remove the product as needed. The admin can view all the purchase details and also users' profile. Bar chart is an additional application in our project from which an unknown user can know the status of an entire website.

XI. Conclusion

We present a novel framework for recommendations on large scale Web graphs using heat diffusion. This is a general framework which can basically be adapted to most of the web graphs for the recommendation tasks, such as query suggestions, image recommendations, personalized recommendations, etc. The generated suggestions are semantically related to the inputs. The experimental analysis on several large scale Web data sources shows the promising future of this approach.

References

[1] G. Jeh and J. Widom, "Simrank: A Measure of Structural-Context Similarity," KDD '02: Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 538-543, 2002.

[2] Y.-H. Yang, P.-T. Wu, C.-W. Lee, K.-H. Lin, W.H. Hsu, and H. Chen, "Context Seer: Context Search and Recommendation at Query Time for Shared Consumer Photos," Proc. 16th ACM Int' Conf. Multimedia, pp. 199-208, 2008.

[3] N. Craswell and M. Szummer, "Random Walks on the Click Graph," SIGIR '07: Proc. 30th Ann. Int'l ACM SIGIR

Conf. Research and Development in Information Retrieval, pp. 239-246, 2007.

[4] B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic, “Real Life Information Retrieval: A Study of User Queries on the Web,” ACM SIGIR Forum, vol. 32, no. 1, pp. 5-17, 1998.

[5] C.Silverstein, M.R.Henzinger, H.Marais,and M.Moricz. “Analysis of the Very Large Web Search Engine Query Log”, ACM SIGIR Forum, vol 32, no.1, pp. 6-12, 1999.

[6] R. Kraft and J. Zien, “Mining Anchor Text for Query Refinement, ” WWW ’04: Proc 13th Int’l Conf. World Wide Web, pp. 666-674, 2004.

[7] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, “Context-Aware Query Suggestion by Mining Click-Through and Session Data,” KDD ’08: Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 875-883, 2008.