Collaborative Filtering with query logs in Search engines

Name: M.Suneetha(Assistant Prof)
College: University College For Women,Koti, Hyd
Dept: Dept of computer science and Informatics

Name: T.RamyaSri
College: Nagole Instite of Science and Tech.
Dept: M.Tech(Assistant Professor)

Abstract—In this paper we study a large query log of more than twenty million queries with the goal of extracting the semantic relations that are implicitly captured in the actions of users submitting queries and clicking answers. Previous query log analyses were mostly done with just the queries and not the actions that followed after them. We rst propose a novel way to represent queries in a vector space based on a graph derived from the query-click bipartite graph. We then an- alyze the graph produced by our query log, showing that it is less sparse than previous results suggested, and that almost all the measures of these graphs follow power laws, shedding some light on the searching user behavior as well as on the distribution of topics that people want in the Web. The representation we introduce allows to infer interesting semantic relationships between queries. Second, we provide an experimental analysis on the quality of these relations, showing that most of them are relevant. Finally we sketch an application that detects multitopical URLs.

Keywords: Graph mining, query logs analysis, knowledge extraction.

INTRODUCTION

The electiveness of information retrieval from the web largely depends on whether users can issue queries to search engines, which properly describe their information needs. Writing queries is never easy, because usually queries are short (one or two words on average) [19] and words are am- biguous [5]. To make the problem even more complicated, different search engines may respond differently to the same query. Therefore, there is no \standard" or \optimal" way to issue queries to search engines, and it is well recognized that query formulation is a bottleneck issue in the usability of search engines.

Typically, recommender systems are based on ¨ tive Filtering [14], [22], [25], [41], [46], [49], which is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items. The underlying assumption of collaborative filtering is that the active user will prefer those items which other similar users prefer [38]. Based on this simple but effective intuition, collaborative filtering has been widely employed in some large, well-known commercial systems, including product recommendation at Amazon,[1] movie recommendation at Netflix,[2] etc. Typical collaborative filtering algorithms require a user-item rating matrix which contains user-specific rating preferences to infer users' characteristics. However, in most of the cases, rating data are always unavailable since information on the Web is less structured and more diverse.

The first challenge is that it is not easy to recommend latent semantically relevant results to users. Take Query Suggestion as an example, there are several outstanding issues that can potentially degrade the quality of the recommendations, which merit investigation. The first one is the ambiguity which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithms which do not satisfy the information needs of users. Another consideration, as reported in [26] and [53], is that users tend to submit short queries consisting of only one or two terms under most circumstances, and short queries are more likely to be ambiguous. Through the analysis of a commercial search engine's query logs recorded over three months in 2006, we observe that 19.4 percent of Web queries are single term queries, and further 30.5 percent of Web queries contain only two terms. Third, in most cases, the reason why users perform a search is because they have little or even no knowledge about the topic they are searching for. In order to find satisfactory answers, users have to rephrase their queries constantly.

The second challenge is how to take into account the personalization feature. Personalization is desirable for many scenarios where different users have different information needs. As an example, Amazon.com has been the early adopter of personalization technology to recommend products to shoppers on its site, based upon their previous purchases. Amazon makes an extensive use of collaborative filtering in its personalization technology. The adoption of personalization will not only filter out irrelevant information to a person, but also provide more specific information that is increasingly relevant to a person's interests.

The last challenge is that it is time consuming and inefficient to design different recommendation algorithms for different recommendation tasks. Actually, most of these recommendation problems have some common features, where a general framework is needed to unify the recommendation tasks on the Web. Moreover, most of existing methods are complicated and require to tune a large number of parameters.

In this paper, aiming at solving the problems analyzed above, we propose a general framework for the recommendations on the Web. This framework is built upon the heat diffusion on both undirected graphs and directed graphs, and has several advantages.

1. It is a general method, which can be utilized to many recommendation tasks on the Web.
2. It can provide latent semantically relevant results to the original information need.
3. This model provides a natural treatment for personalized recommendations.
4. The designed recommendation algorithm is scalable to very large data sets.

The empirical analysis on several large scale data sets (AOL clickthrough data and Flickr image tags data) shows that our proposed framework is effective and efficient for generating high-quality recommendations.

The rest of the paper is organized as follows. We review related work in Section 2. Section 3 presents the diffusion models on both undirected graphs and directed graphs. In Section 4, we demonstrate the empirical analysis of our models and recommendation algorithms on several diversified data sources. Finally, conclusion is given in Section 5.

2 Related Work

Recommendation on the Web is a general term representing a specific type of information filtering technique that attempts to present information items (queries, movies, images, books, Web pages, etc.) that are likely of interest to the users. In this section, we review several work related to recommendation, including collaborative filtering, query suggestion techniques, image recommendation methods, and clickthrough data analysis.

2.1 Collaborative Filtering

Two types of collaborative filtering approaches are widely studied: neighborhood-based and model-based.

The neighborhood-based approaches are the most popular prediction methods and are widely adopted in commercial collaborative filtering systems [37], [47]. The most analyzed examples of neighborhood-based collaborative filtering include user-based approaches [7], [21] and item-based approaches [15], [37], [50]. User-based approaches predict the ratings of active users based on the ratings of their similar users, and item-based approaches predict the ratings of active users based on the computed information of items similar to those chosen by the active user. User-based and item-based approaches often use the Pearson Correlation Coefficient algorithm (PCC) [47] and the Vector Space Similarity algorithm (VSS) [7] as the similarity computation methods. PCC-based collaborative

2.2 Query Suggestion

In order to recommend relevant queries to Web users, a valuable technique, query suggestion, has been employed by some prominent commercial search engines, such as ,[3] Live Search,[4] Ask,[5] and Google.[6] However, due to commercial reasons, a few public papers have been released to reveal the methods they adopt.

The goal of query suggestion is similar to that of query expansion [11], [13], [56], [61], query substitution [31], and query refinement [35], [57], which all focus on understanding users' search intentions and improving the queries submitted by users. Query suggestion is closely related to query expansion or query substitution, which extends the original query with new search terms to narrow down the scope of the search. 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 [18]. Query refinement is another closely related notion, since the objective of query refinement is interactively recommending new queries related to a particular query.

In [61], local (i.e., query-dependent documents) and global (i.e., the whole corpus) documents are employed in query expansion by applying the measure of global analysis to the selection of query terms in local feedback. Although experimental results show that this method is generally

3. http://www.yahoo.com.
4. http://www.live.com.
5. http://www.ask.com.
6. http://www.google.com.

In Section 3, we introduced our graph diffusion models for recommendations. In this section, 1) we show how to convert different Web data sources into correct graphs in our models; and 2) we conduct several experiments on query suggestions, and image recommendations.[9]

3. Query Suggestion

Query Suggestion is a technique widely employed by commercial search engines to provide related queries to

3.1 Data Collection

Clickthrough data record the activities of Web users, which reflect their interests and the latent semantic relationships between users and queries as well as queries and clicked Web documents. As shown in Table 1, each line of clickthrough data contains the following information: a user ID (u), a query (q) issued by the user, a URL (l) on which the user clicked, the rank (r) of that URL, and the time (t) at which the query was submitted for search. Thus, the clickthrough data can be represented by a set of quintuples $\langle u; q; l; r; t \rangle$. From a statistical point of view, the query word set corresponding to a number of Web pages contains human knowledge on how the pages are related to their issued queries [55]. Thus, in this paper, we utilize the relationships of queries and Web pages for the construction of the bipartite graph containing two types of vertices $\langle q; l \rangle$. The information regarding user ID, rank and calendar time is ignored.

This data set is the raw data recorded by the search engine, and contains a lot of noise which will potentially affect the effectiveness of our query suggestion algorithm. Hence, we conduct a similar method employed in [59] to clean up the raw data. We filter the data by only keeping those frequent, well formatted, English queries (queries which only contain characters "a," "b," . . . , "z," and space). After cleaning and removing duplicates, we get totally 2,019,265 unique queries and 915,771 unique URLs in our data collection. After the construction of the query-URL bipartite graph using this data collection procedure, we observe that a total of 7,633,400 edges exist in the query-URL bipartite graph, which indicates that, on average, each query has 3.78 distinct clicks, and each URL is clicked by 8.34 distinct queries.

4.Graph Construction

For the query-URL bipartite graph, consider an undirected bipartite graph $B_{ql} \¼ ðV_{ql}; E_{ql}Þ$, where V_{ql} ; $q_2; \ldots ; q_n g$, and $L \¼ fl_1; l_2; \ldots ; l_p g$. $E_{ql} \¼ f ðq$ Þj there is an edge from q_i to $l_j g$ is the set of all edges. The edge $ðq$ Þ exists if and only if a user u_i clicked a URL l_k after issuing a query q

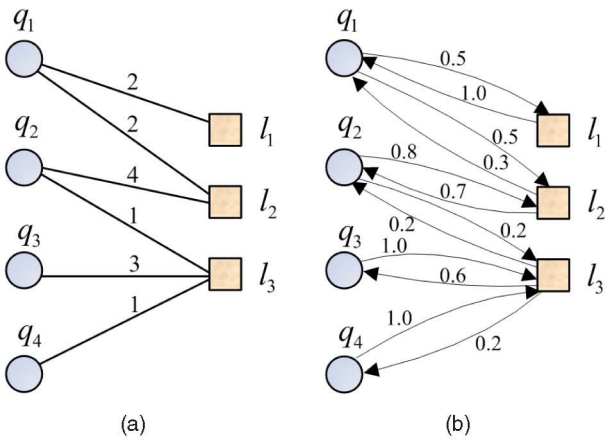

Fig. 2. Graph construction for query suggestion. (a) Query-URL bipartite graph. (b) Converted query-URL bipartite graph.

See Fig. 2a for an example. The values on the edges in Fig. 2a specify how many times a query is clicked on a URL.

We cannot simply employ the bipartite graph extracted from the clickthrough data into the diffusion processes since this bipartite graph is an undirected graph, and cannot accurately interpret the relationships between queries and URLs. Hence, we convert this bipartite graph into Fig. 2b. In this converted graph, every undirected edge in the original bipartite graph is converted into two directed edges. The weight on a directed query-URL edge is normalized by the number of times that the query is issued, while the weight on a directed URL-query edge is normalized by the number of times that the URL is clicked.

4.1 Query Suggestion Algorithm

After the conversion of the graph, we can easily design the query suggestion algorithm in Algorithm 1.

Algorithm 1. Query Suggestion Algorithm

1: A converted bipartite graph $G \¼ ðV Þ [V ; E Þ$ consists of query set $V Þ$ and URL set V . The two directed edges are weighted using the method introduced in previous section.
2: Given a query q in $V Þ$, 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 ¼ 1, and without loss of generality, set the initial heat value of query q $f_q ð0Þ \¼ 1$ (the choice of initial heat value will not affect the suggestion results). Start the diffusion process using
$$f ð1Þ \¼ e \R f ð0Þ.$$
4: Output the Top-K queries with the largest values in vector $f ð1Þ$ as the suggestions.

5. CONDUCTING QUERY SUGGESTIONS

In this section, we ¯rst introduce how to derive session data from a search log. We then develop a novel structure, concept sequence su±x tree, to organize the patterns mined from session data. Finally, we present the query suggestion method based on the patterns mined.

5.1 Query Sessions

As explained in Section 2, the context of a user query consists of the immediately preceding queries issued by the same user. To learn a context-aware query suggestion model, we need to collect query contexts from user query sessions. We construct session data in three steps. First, we extract each individual user's behavior data from the whole search log as a separate stream of query/click events. Second, we segment each user's stream into sessions based on a widely- used rule [20]: two consecutive events (either query or click) are segmented into two sessions if the time interval between them exceeds 30 minutes. Finally, we discard the click events and only keep the sequence of queries in each session.Query sessions can be used as training data for query suggestion. For example, Table 2 shows some real sessions as well as the relationship between the queries in the sesssion.

5.2 Image Recommendation

Finding effective and efficient methods to search and retrieve images on the Web has been a prevalent line of research for a long time [58]. The situation is even tougher in the research of Image Recommendation. In this section, we present how to recommend related images to the given images using data set.

We use Flickr, a popular image hosting Web site and online community for users to share personal photographs, tag

The data will be popup which was retrieved in the last ranked out along with top ranked results. After these top ranked results last visible data will be come up in stack. Based on input which was given by users as per user requirements and interests auto updates will be generate or send to users. The data will be retrieve which was shown up at last using click-through and session data. Unlike previous methods, our approach considers not only the current query but also the recent queries in the same session to provide more meaningful suggestions. Moreover, we group similar queries into concepts and provide suggestions based on the concepts. The experimental results on a large-scale data containing billions of queries and URLs clearly show our approach outperforms two

6.CONCLUSION

In this paper, 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.

7.FUTURE WORK

Search Results Improvement

The way we are geting is to be When a user submits a query q, our approach first captures the context of q which is represented by a short sequence of queries issued by the same user immediately before q. We then check the historical data and what queries many users often ask after q in the same context. Those queries become the candidate suggestions.

if a particular user viewing a particular webisite more number of times ,later any changes or updates occured in that website in the sense the updates will be sent to that valuable user who used to hit that server regularly than normal users.

The Top-5 Web sites given the queries “sony,” “camera,” “microsoft,” and “chocolate” are shown in Table 4. For example, the ranking order for “sony” is different from all of the results retrieved by those four commercial search engines (which we do not list here due to the space limitation). If this order is incorporated into the original results, the search results can be greatly improved since they are the representations of the implicit votes of all the search users. In the future, we plan to compare this ranking method with other previous Web search results ranking approaches.

Our results are really promising if we consider that the query log was actually small and over a short period of time. This implies that we can neither follow patterns over time nor consider the number of di erent users involved in the clicks (more users, more wisdom). So an immediate exten- sion is to use larger logs and include information on unique users to classify the quality of results.Our results are really promising if we consider that the query log was actually small and over a short period of time. This implies that we can neither follow patterns over time nor consider the number of di erent users involved in the clicks (more users, more wisdom). So an immediate exten- sion is to use larger logs and include information on unique users to classify the quality of results.

REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais, “Improving Web Search Ranking by Incorporating User Behavior Information,” SIGIR ’07: Proc. 29th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 19-26, 2006.

[2] E. Auchard, “Flickr to Map the World’s Latest Photo Hotspots,” Proc. Reuters, 2007.

[3] R. TiberiBaeza-Yates and A. Tiberi, “Extracting Semantic Relations from Query Logs,” KDD ’07: Proc. 13th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 76-85, 2007.

[4] R.A. Baeza-Yates, C.A. Hurtado, and M. Mendoza, “Query Recommendation Using Query Logs in Search Engines,” Proc. Current Trends in Database Technology (EDBT) Workshops, pp. 588-596, 2004.

[5] D. Beeferman and A. Berger, “Agglomerative Clustering of a Search Engine Query Log,” KDD ’00: Proc. Sixth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 407-416, 2000.

[6] M. Belkin and P. Niyogi, “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation,” Neural Computation, vol. 15, no. 6, pp. 1373-1396, 2003.

[7] J.S. Breese, D. Heckerman, and C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), 1998.

[8] S. Brin and L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” Computer Networks and ISDN Systems, vol. 30, nos. 1-7, pp. 107-117, 1998.

[9] J. Canny, “Collaborative Filtering with Privacy via Factor Analysis,” SIGIR ’07: Proc. 25th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 238-245, 2002.

[10] 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.

[11] P.A. Chirita, C.S. Firan, and W. Nejdl, “Personalized Query