Personalized Search

► Personalized Web Search

Personalized Web

► Data Integration in Web Data Extraction System

Personalized Web Search

Ji-Rong Wen[1], Zhicheng Dou[2], Ruihua Song[1]
[1]Microsoft Research Asia, Beijing, China
[2]Nankai University, Tianjin, China

Synonyms

Personalized search

Definition

For a given query, a personalized Web search can provide different search results for different users or organize search results differently for each user, based upon their interests, preferences, and information needs. Personalized web search differs from generic web search, which returns identical research results to all users for identical queries, regardless of varied user interests and information needs.

Historical Background

Web search engines have made enormous contributions to the web and society. They make finding information on the web quick and easy. However, they are far from optimal. A major deficiency of generic search engines is that they follow the "one size fits all" model and are not adaptable to individual users. This is typically shown in cases such as these:

1. Different users have different backgrounds and interests. They may have completely different information needs and goals when providing exactly the same query. For example, a biologist may issue "mouse" to get information about rodents, while programmers may use the same query to find information about computer peripherals. When such a query is issued, generic search engines will return a list of documents on different topics. It takes time for a user to choose which information he/she really wants, and this makes the user feel less satisfied. Queries like "mouse" are usually called ambiguous queries. Statistics has shown that the vast majority of queries are short and ambiguous. Generic web search usually fails to provide optimal results for ambiguous queries.
2. Users are not static. User information needs may change over time. Indeed, users will have different needs at different times based on current circumstances. For example, a user may use "mouse" to find information about rodents when the user is viewing television news about a plague, but would want to find information about computer mouse products when purchasing a new computer. Generic search engines are unable to distinguish between such cases.

Personalized web search is considered a promising solution to address these problems, since it can provide different search results based upon the preferences and information needs of users. It exploits user information and search context in learning to which sense a query refers. Consider the query "mouse" mentioned above: Personalized web search can disambiguate the query by gathering the following user information:

1. The user is a computer programmer, not a biologist.
2. The user has just input a query "keyboard," but not "biology" or "genome." Before entering this query, the user had just viewed a web page with many words related to computer mouse, such as "computing," "input device," and "keyboard."

Foundations

User Profiling

To provide personalized search results to users, personalized web search maintains a user profile for each individual. A user profile stores approximations of user tastes, interests and preferences. It is generated and updated by exploiting user-related information. Such information may include:

1. Demographic and geographical information, including age, gender, education, language, country, address, interest areas, and other information;
2. Search history, including previous queries and clicked documents. User browsing behavior when

viewing a page, such as dwelling time, mouse click, mouse movement, scrolling, printing, and bookmarking, is another important element of user interest.

3. Other user documents, such as bookmarks, favorite web sites, visited pages, and emails. Teevan et al. [15] and Chirita et al. [1] demonstrate that external user data stored in a user client is useful to personalize individual search results.

User information can be specified by the user (explicitly collecting) or can be automatically learnt from a user's historical activities (implicitly collecting). As the vast majority of users are reluctant to provide any explicit feedback on search results and their interests, many works on personalized web search focus on how to automatically learn user preferences without involving any direct user efforts [6,8,9,10,13]. Collected user information is processed and organized as a user profile in a certain structure, depending on the need of personalization algorithm. This can be completed by creating vectors of URLs/domains, keywords, topic categories, tensors, or the like.

A user profile can usually aggregate a user's history information and represent the user's *long-term* interests (information needs). Some work has investigated whether such a long-term user profile is ineffective in some cases. Consider the second case that was described in the historical background section: a user will have different needs at different times based on circumstances. In such situations, personalization based on a user's long-term interests may not provide a satisfying performance, because similar results could be returned. Some work [10] has considered the use of a user's active context to represent *short-term* information needs. Search context is incorporated into the user profile, or is constructed as a separate short-term user model/profile and is used in helping infer a user's information needs.

Personalized Search Based on Content Analysis

Personalized web search can be achieved by checking content similarity between web pages and user profiles.

Some work has represented user interests with topical categories. User's topical interests are either explicitly specified by users themselves, or can be automatically learned by classifying implicit user data. Search results are filtered or re-ranked by checking the similarity of topics between search results and user profiles. In some work [2,8], a user profile is structured as a concept/topic hierarchy. User-issued queries and user-selected snippets/documents are categorized into concept hierarchies that are accumulated to generate a user profile. When the user issues a query, each returned snippet/document is also classified. The documents are re-ranked based upon how well the document categories match user interest profiles. Chirita et al. [2] use the ODP (Open Directory Project, http://www.dmoz.org/) hierarchy to implement personalized search. User favorite topics nodes are manually specified in the ODP hierarchy. Each document is categorized into one or several topic nodes in the same ODP hierarchy. The distances between the user topic nodes and the document topic nodes are then used to re-rank search results.

Some other work uses lists of keywords (bags of words) to represent user interests. In [13], a user profile is built as a vector of distinct terms and is constructed by aggregating past user click history. The cosine similarity between the user profile vector and the feature vector of returned web pages are used to re-rank results. Shen et al. [10] first use language modeling to mine immediate search contextual and implicit feedback information. The approach selects appropriate terms from related preceding queries and corresponding search results to expand the current query. In a query session, the viewed document summaries are used to immediately re-rank documents that have not yet been seen by the user. Teevan et al. [15] and Chirita et al. [1] exploit rich models of user interests, built from both search-related information, and other information about the user. This includes documents and emails the user has read and created. In [6], keywords are associated with categories and thus user profiles are represented by a hierarchical category tree based on keywords categories.

Personalized Web Search Based on Hyperlink Analysis

Most generic web search approaches rank importance of documents based on the linkage structure of the web. An intuitive approach of personalized web search is to adapt these algorithms to compute personalized importance of documents. A large group of these works focuses on personalized PageRank. PageRank, proposed by Page and Brin [7], is a popular link analysis algorithm used in web search. The fundamental motivation underlying PageRank is the recursive notion that important pages are those linked-to by many important pages. This recursive notion can be formalized by the "random surfer" model [7] on

the directed web graph G. A directed edge $<p, q>$ exists in G if page p has a hyperlink to page q. Let $O(p)$ be the outdegree of web page p in G. $O(p)$ is equivalent to number of web pages that linked by page p. Let **A** be the matrix corresponding to the web graph G, where $A_{ij} = 1/O(j)$ if page j links to page i, and $A_{ij} = 0$ otherwise. In the random surfer model, when a surfer visits page p, he/she keeps clicking outlinks at random with probability (1-c), and jumps to a random web page with probability c. c is called teleportation constraint or damping factor. The PageRank of a page p is defined as the probability that the surfer visited page p. Iterative computation of PageRank is done as the following equation:

$$\boldsymbol{v}^{k+1} = (1 - c)\ \boldsymbol{A}\boldsymbol{v}^{k} + c\boldsymbol{u} \qquad (1)$$

Here, $\boldsymbol{u}$ is defined as a preference vector, where $|\boldsymbol{u}| = 1$ and $u(i)$ denotes the amount of preference for page i when the surfer jumps to a random web page i. The global PageRank vector is computed when there is no particular preference on any pages, i.e., $\boldsymbol{u} = [1/n,...,1/n]^{T}$. By setting variant preference to web pages, a PageRank vector with personalized views of web page importance is generated. It recursively favors pages with high preference, and pages linked by high-preference page. This PageRank vector is called a *personalized PageRank vector* (*PPV*). To accomplish personalized web search, a personalized PageRank is computed for each user based upon the user's preference. For example, web pages in the user's bookmarks are set higher preferences in $\boldsymbol{u}$. Rankings of the user's search results can be biased according to the user's Personalized PageRank vector instead of the global PageRank.

Unfortunately, computing a PageRank vector usually requires multiple scans of the web graph [7], which makes it impossible to carry out online in response to a user query. Furthermore, when a large number of users employ a search engine, it is impossible to compute and store so many personalized PageRank vectors offline. Many later works [4,5] make efforts to reduce the computation and storage cost of personalized PageRank vectors. Jeh and Widom [5] support the concept that a user's preference set is a sub-set of a set of hub pages H, selected as those of greater interest for personalization. For each hub page p in H, setting the preference to 1 for page p and 0 for other pages, the corresponding personalized PageRank vector is called a basis hub vector. The authors decompose each basis hub vector in two parts: hub skeleton vector and partial vector. Hub skeleton vector represents common interrelationships between hub vectors, and is computed offline. Each partial vector for a hub page p represents the part of p's hub vector unique to itself. Partial vector can be computed at construction-time efficiently. Finally, a personalized PageRank vector can be expressed as a linear combination of a set of basis hub vectors, and is computed at query time efficiently. Experiments show that the approach is feasible when size of hub set $> 10^4$.

Haveliwala [4] use personalized PageRank to enable "topic-sensitive" web search. The approach precomputes k personalized PageRank vectors using k topics, e.g., the 16 top level topics of the Open Directory. For each topic i, a preference vector $\boldsymbol{u}_i$ is generated. $(u_i)_j$ represents the confidence that web page j is classified into topic i. A PPV is computed base upon preference vector $\boldsymbol{u}_i$. The k personalized PageRank vectors are combined at query time, using the context of the query to compute the appropriate topic weights. The experiments concluded that the use of personalized PageRank scores can improve web search, but the number of personalized PageRank vectors used was limited due to the computational requirements. In fact, this approach modulates the rankings based on the topic of the query and query context, rather than for truly "personalizing" the rankings to a specific individual. Qiu and Cho [9] develop a method to automatically estimate a user's topic preferences based on Topic-Sensitive PageRank scores of the user's past clicked pages. The topic preferences are then used to bias future search results.

Community-based Personalized Web Search

In most of the above personalized search strategies, each user has a distinct profile and the profile is used to personalize search results for the user. There are also some approaches that personalize search results for the preferences of a community of like-minded users. These approaches are called community-based personalized web search or collaborative web search. In a community-based personalized web search, when a user issues a query, search histories of users who have similar interests to the user are used to filter or re-rank search results. For example, documents that have been selected for the target query or similar queries by the community are re-ranked higher in the results

list. Sugiyama et al. [13] use a modified collaborative filtering algorithm to constructed user profiles to accomplish personalized search. Sun et al. [14] proposed a novel method named CubeSVD to apply personalized web search by analyzing correlations among users, queries, and web pages in clickthrough data. Smyth et al. [12] show that collaborative web search can be efficient in many search scenarios when natural communities of searchers can be identified.

Server-Side and Client-Side Implement

Personalized web search can be implemented on either server side (in the search engine) or client side (in the user's computer or a personalization agent).

For server-side personalization, user profiles are built, updated, and stored on the search engine side. User information is directly incorporated into the ranking process, or is used to help process initial search results. The advantage of this architecture is that the search engine can use all of its resources, for example link structure of the whole web, in its personalization algorithm. Also, the personalization algorithm can be easily adapted without any client efforts. This architecture is adopted by some general search engines such as Google Personalized Search. The disadvantage of this architecture is that it brings high storage and computation costs when millions of users are using the search engine, and it also raises privacy concerns when information about users is stored on the server.

For client-side personalization, user information is collected and stored on the client side (in the user's computer or a personalization agent), usually by installing a client software or plug-in on a user's computer. In client side, not only the user's search behavior but also his contextual activities (e.g., web pages viewed before) and personal information (e.g., emails, documents, and bookmarks) could be incorporated into the user profile. This allows the construction of a much richer user model for personalization. Privacy concerns are also reduced since the user profile is strictly stored and used on the client side. Another benefit is that the overhead in computation and storage for personalization can be distributed among the clients. A main drawback of personalization on the client side is that the personalization algorithm cannot use some knowledge that is only available on the server side (e.g., PageRank score of a result document). Furthermore, due to the limits of network bandwidth, the client can usually only process limited top results.

Challenges of Personalized Search

Despite the attractiveness of personalized search, there is no large-scale use of personalized search services currently. Personalized web search faces several challenges that retard its real-world large-scale applications:

1. Privacy is an issue. Personalized web search, especially server-side implement, requires collecting and aggregating a lot of user information including query and clickthrough history. A user profile can reveal a large amount of private user information, such as hobbies, vocation, income level, and political inclination, which is clearly a serious concern for users [11]. This could make many people nervous and feel afraid to use personalized search engines. A personalized web search will be not well-received until it handles the privacy problem well.
2. It is really hard to infer user information needs accurately. Users are not static. They may randomly search for something which they are not interested in. They even search for other people sometimes. User search histories inevitably contain noise that is irrelevant or even harmful to current search. This may make personalization strategies unstable.
3. Queries should not be handled in the same manner with regard to personalization. Personalized search may have little effect on some queries. Some work [1,2,3] investigates whether current web search ranking might be sufficient for clear/unambiguous queries and thus personalization is unnecessary. Dou et al. [3] reveal that personalized search has little effect on queries with high user selection consistency. A specific personalized search also has different effectiveness for different queries. It even hurts search accuracy under some situations. For example, topical interest-based personalization, which leads to better performance for the query "mouse," is ineffective for the query "free mp3 download." Actually, relevant documents for query "free mp3 download" are mostly classified into the same topic categories and topical interest-based personalization has no way to filter out desired documents. Dou et al. [3] also reveal that topical interest-based personalized search methods are difficult to deploy in a real world search engine. They improve search performance for some queries, but they may hurt search performance for additional queries.

Key Applications

Personalized web search is considered a promising solution to improve the performance of generic web search. Currently, Google and other web search engines are trying to do personalized search.

Experimental Results

Experimental results have shown that personalized web search can indeed improve performance of web search. Detailed experimental results can be found in the corresponding reference for each presented method. Dou et al. [3] propose a personalized web search evaluation framework based upon large-scale query logs.

Cross-references

- ▶ Information Retrieval
- ▶ Privacy
- ▶ Relevance Feedback
- ▶ WEB Information Retrieval Models
- ▶ Web Search Relevance Feedback

Recommended Reading

1. Chirita P.A., Firan C., and Nejdl W. Summarizing local context to personalize global web search. In Proc. Int. Conf. on Information and Knowledge Management, 2006.
2. Chirita P.A., Nejdl W., Paiu R., and Kohlschütter C. Using ODP metadata to personalize search. In Proc. 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005, pp. 178–185.
3. Dou Z., Song R., and Wen J. A large-scale evaluation and analysis of personalized search strategies. In Proc. 33rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2007.
4. Haveliwala T.H. Topic-sensitive pagerank. In Proc. 11th Int. World Wide Web Conference, 2002.
5. Jeh G. and Widom J. Scaling personalized web search. In Proc. 12th Int. World Wide Web Conference, 2003, pp. 271–279.
6. Liu F., Yu C., and Meng W. Personalized web search by mapping user queries to categories. In Proc. Int. Conf. on Information and Knowledge Management, 2002, pp. 558–565.
7. Page L., Brin S., Motwani R., and Winograd T. The pagerank citation ranking: bringing order to the web. Technical report, Computer Science Department, Stanford University, 1998.
8. Pretschner A. and Gauch S. Ontology based personalized search. In Proc. 11th IEEE Int. Conf. on Tools with Artificial Intelligence, 1999, pp. 391–398.
9. Qiu F. and Cho J. Automatic identification of user interest for personalized search. In Proc. 15th Int. World Wide Web Conference, 2006, pp. 727–736.
10. Shen X., Tan B., and Zhai C. Implicit user modeling for personalized search. In Proc. Int. Conf. on Information and Knowledge Management, 2005, pp. 824–831.
11. Shen X., Tan B., and Zhai C. Privacy protection in personalized search. SIGIR Forum, 41(1):4–17, 2007.
12. Smyth B., Coyle M., Boydell O., Briggs P., Balfe E., Freyne J., and Bradley K. A live-user evaluation of collaborative web search. In Proc. 19th Int. Joint Conf. on AI, 2005.
13. Sugiyama K., Hatano K., and Yoshikawa M. Adaptive web search based on user profile constructed without any effort from users. In Proc. 12th Int. World Wide Web Conference, 2004, pp. 675–684.
14. Sun J.-T., Zeng H.-J., Liu H., Lu Y., and Chen Z. CubeSVD: a novel approach to personalized web search. In Proc. 14th Int. World Wide Web Conference, 2005, pp. 382–390.
15. Teevan J., Dumais S.T., and Horvitz E. Personalizing search via automated analysis of interests and activities. In Proc. 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005, pp. 449–456.

Personally Identifiable Data

▶ Individually Identifiable Data

Perturbation Techniques

▶ Randomization Methods to Ensure Data Privacy

Perusal

▶ Browsing

Pessimistic Scheduler

▶ Two-Phase Locking

Petri Nets

W. M. P. VAN DER AALST
Eindhoven University of Technology, Eindhoven, The Netherlands

Synonyms

Place transition nets; Condition event nets; Colored nets