Aggregating User-Centered Rankings to Improve Web Search

Lin Li ,† Kulwadee Somboonviwat † and Masaru Kitsuregawa†

This paper is to investigate rank aggregation based on multiple user-centered measures in the context of the web search. We introduce a set of techniques to combine ranking lists in order of user interests. To learn user interests, we build a user profile that contains a taxonomic hierarchy for the long-term model and a recently visited page-history buffer for the short-term model. Dynamic adaptation strategies are devised to capture the accumulation and degradation changes of user interests, and then adjust the content and structure of the user profile. Moreover, our user profile can include a variety of attributes of user interests. We mainly focus on the topics a user is interested in and the degrees of user interests in these topics. The primary goal of our work is to form a broadly acceptable ranking list, rather than that determined by an individual ranking measure. Experiment results on a real click-through data set show the effectiveness of our aggregation techniques to improve the web search.

1. Introduction

With the advent of the era of the information explosion, never before have so many information sources been availably indexed by search engines on the Internet. Ideally, users should be able to take advantage of the wide range of the valuable information while being able to find only those which are appealing to them. On the contrary, it becomes more difficult than ever to obtain desired results due to the ambiguity of user's needs. Moreover, present search engines generally handle search queries without considering user interests or contexts in which users submit their queries. For example, supposing a information retrieval researcher who wants to search information about Text Retrieval Conference and a engineer who is interested in taking advantage of the quantities of solar energy falling, they both input "TREC" on Google. Regardless of different intentions of the two users on the same query, the results turn out to be an official site of the Texas real estate commission, training resources for the environmental community, a site about educational research experiences, and so on. Current search engines are inadequate in making a difference among the various needs of users.

To address this problem, personalized search has recently become an active on-going research field. Some systems have required users to explicitly enter their contextual interests including interest topics, bookmarks, and so on. These interests are used to expand user queries or re-rank search results. Forcing users to submit their contextual interests would be a task that few users would be willing to do. Furthermore, it is very difficult for users to define their own contextual interests accurately. Much attention has been paid to learn user interests transparently without any extra effort from users[14),16),18),19)]. These studies have modelled user profiles or user representations to indicate user interests automatically. Speretta et al. [18)] have created user profiles by classifying some information into concepts from the ODP☆ taxonomic hierarchy and then re-ranked search results based on the conceptual similarity between the web page and the user profile. The authors, however, have not taken into account the hierarchical structure of the ODP when calculating the similarity.

† Institute of Industrial Science, The University of Tokyo

☆ http://dmoz.org

In this paper, we focus on learning user profiles and utilizing the learned user profiles to re-rank search results. Most studies have deemed user profiles to be static. A related problem occurs when user interests vary with time. For instance, if a user changes her vocation from a IT specialist to a lawyer, it is natural that her interests will shift with this change. It becomes important to keep the user profile up-to-date, and for a search engine to adapt accordingly. Furthermore, a user profile covers both short-term and long-term user interests, which may increase or reduce respectively and corelatedly over time. Using one model to represent two differently featured parts of the user profile will be far from perfect. Accordingly, suitable strategies are needed to capture the accumulation and degradation of changes of user interests, and then adapt the content and structure of the user profile to these changes. For re-ranking search results, our rank mechanism is similar to that proposed by Chirita et al.[2)] who introduced a semantic similarity measure for the web page rank with consideration to the hierarchy of the ODP structure. Meanwhile, Chirita et al.[2)] suffer from the problem of requiring users to select topics which best fit their interests from the ODP.

On the other hand, a user profile may contain a number of attributes which describe user interests from their respective viewpoints[10)]. In most cases, any individual attribute is deficient in representing user interests accurately. In order to leverage the different ranking lists produced by the different attributes, the rank aggregation should intend to form a single ranking list supported by a broad consensus among these attributes. Merging the values of the attributes in a simply linear combination[2),18)] may result in neglecting the respective characteristics of them. Moreover, it is important to observe that if the ranking measure is value-based, the ordering implied by the values makes more sense than the actual values themselves[5)]. Dwork et al.[5)] also developed the theoretical ground work for describing and evaluating rank aggregation methods. Their main work is to effectively combat "spam". We study the rank aggregation of the attributes of user interests learned from the click-through data to improve the web search.

Our contributions in this paper could be summarized as:

(1) We devise independent models for long-term and short-term user interests which contain two attributes, the topics a user is interested in and the degrees of user interests in these topics.

(2) Dynamic adaptation strategies are proposed for modelling user profiles automatically. Based on click-through data, these strategies consider the accumulation and degradation changes of user interests, thus modify our user profiles, not only in contents, but also in structures.

(3) Finally, we present a set of effective techniques to aggregate the attribute-based ranking lists plus the original ranking list of a search engine (i.e., Google we refer to).

The rest of this paper begins with a review of the related work. Then we describe two independent models and dynamic adaptation strategies for user profiles. In addition, user-centered ranking lists are addressed and methods for rank aggregation are presented. Finally, we report the experimental results which are followed by the conclusions and some directions for the future work.

2. Related Work

2.1 Context Search

Kraft et al.[8)] state that the context, in its general form, refers to any additional information associated with the query in the web search field, and also present three different algorithms to implement the contextual search instead of modelling user profiles. Generally speaking, if the context information is provided by an individual user in any form, whether automatically or manually, explicitly or implicitly, the search engine can use the context to custom-tailor results. The process is named as a personalized search.

In this way, such a personalized search could be either server-based or client-based. Ferragina et al.[6)] introduced an available server-based search engine that unifies a hierarchical web-snippet clustering system with a web interface for the personalized search. Google and Yahoo! also supply personalized search services. On a client-based personalized search, studies[4),16),19)] focus on capturing all the documents edited or viewed by users through computation-consuming procedures. Allowing for scalability, the client-based personalized search could learn user contexts more accurately than the server-based personalized search, while it is unavoidable that keeping track of user contexts has to be realized by middleware in the proxy server or the client. Users, however, may feel unsafe to install such software even if it is guaranteed to be non-invasive, and intend to enjoy the services provided by search engines instead. Moreover, at home if a user uses her private computer different from that of her office, keeping her contexts consistent becomes a problem. Therefore, our work is server-based.

2.2 User Profile

There have been various schemes of learning user profiles to figure user interests from text documents. We found that most of them model user profiles represented by bags of words without considering term correlations[1),9),17),20)]. To overcome the drawbacks of the bag of words, the taxonomic hierarchy, particularly constructed as a tree structure, has been widely accepted in many works[2),13),15)]. Schickel-ZuberF et al.[15)] scored user interests and concept similarity based on the structure of ontology. But their work needs users to express their interests by rating a given number of items.

Meanwhile, these studies omit that user interests could change with time. Some topics will become more interesting to the user, while the user will lose interests in other topics completely or to some extent. Studies[1),9),20)] suggested that relevance feedback and machine learning techniques have shown promise in adapting to changes of user interests and reducing user involvements, while still overseeing what users dislike and their interest degradation. Lam et al.[9)] proposed a two-level approach to learn user profiles for information filtering. While the lower level learns the stationary user interests, the higher level detects changes of user interests. Widyantoro et al.[20)] introduced a multiple three-descriptor representation learn changes in multiple interest categories, and they also needed positive and negative relevance feedback provided by users explicitly.

Our work, particularly our dynamic adaptation strategies for user profiles, are based on the idea that sufficient contextual information is already hidden in the web log with little overhead, and all the visited web pages can reflect user interests to various degrees because the users have accessed them. This contextual information motivates us to capture the accumulation and degradation changes of user interests implicitly, to learn user profiles automatically.

3. User Profile and Adaptation Strategies

Widyantoro et al.[20)] indicate, for user profiles, long-term user interests generally hold user interests and the degree of user interests accumulated by experiences over a long time period. Hence it is fairly stable. On the other hand, short-term user interests are unstable by nature. For instance, interests in current hot topics could vary on a day-to-day basis. It is crucial to design a temporal structure for shot-term user interests. Based on these features, We propose two novel models for long-term and short-term user interests respectively and discuss them together with the adaptation strategies for their close correlations.

3.1 Long-term Model of User Profile

The taxonomic hierarchy for our long-term model is the top four levels of topics in the Google Directory[☆] (the preliminary analysis to select the number of levels is discussed in our previous work[11)]). These topics (nodes) are linked as a tree structure, called a user topic tree (UTT) in this paper. We also use search results and web pages interchangeably when referring to the URLs returned from the web search engine on a specific query.

In the Google Directory, each web page is classified into a topic [☆☆]. The adaption strategies for long-term model include two operations, the

☆ http://directory.google.com

☆☆ We do not consider the symbolic links. If needed, readers may load all the symbolic links into memory or compute the shortest distance on the graph.

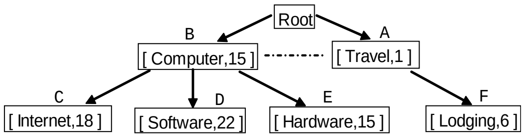

図 1 Schema of long-term user profile

"adding" and "deleting" operations. When dealing with the "adding" operation, topics associated with the clicked search results, but not all the search results, are added into the UTT click by click. In addition, each node in the UTT has a value of the number of times the node has been visited. This value is denoted by "$TopicCount$", representing the degree of user interests. The "deleting" operation is influenced by the changes of the short-term model. It will be addressed in the forthcoming discussion. Figure 1 illustrates the schema of the UTT. For example, node C is represented by the $[Internet, 18]$ which means one user has clicked a web page associated with the topic "$Internet$" and the user has visited this topic 18 times before this search.

3.2 Short-term Model of User Profile

We frame the Page-History Buffer (PHB) for the short-term model. The PHB caches the most recently clicked pages with a fixed size that is determined by the ability of the search engine. We now meet the same problem as the cache in the processor, and that is how to kick off the "old" pages in time to keep up with the changes of short-term user preferences. As it is known, in the cache management, there are popular cache replacement algorithms that are all designed for the processor, the web cache and the database disk buffering. No such research could be available in the personalized search, especially in the short-term model of the user profile. Our goal, keeping track of the most recent accesses of search results in the PHB, is basically similar to that in the cache management. As a result, the LFU (Least Frequently Used), one of these replacement algorithms, is adjusted to our scheme, which is named the Least Frequent Used Page Replacement (LFUPR). The details are shown in Table 3. The LFUPR reflects the changes of the short-term model, including how to add (line 3 $\sim$ line 6) and replace (line 10 $\sim$ line 12) web pages in the PHB.

表 1 LFUPR Algorithm

Input:	current short-term model, current long-term model, search results
Output:	updated user profile
	PageCount=Vector of the number of clicked times for pages in the PHB TopicCount=Vector of the number of clicked times for nodes in the UTT BufferPages=Vector of pages in the PHB Results=Vector of pages returned by a search engine UserTopics=Vector of nodes in the user topic tree
1.	For i=1 to Size(Results)
2.	Begin Loop
3.	If Result[i] is the nth page IN the PHB
4.	PageCount[n]++;
5.	Else If PHB is NOT FULL
6.	Add the clicked page into the end of the PHB;
7.	Else
8.	Begin
9.	For j=1 to Size (PHB)
10.	BufferPages[k] ← Find one page in the
11.	BufferPages with the Minimum PageCount[j];
12.	Replace the BufferPages[k] with the Results[i];
13.	TopicCount[m]- -; //BufferPages[k] is the mth
14.	node IN the UserTopics
15.	End
16.	End loop
17.	For t=1 to Size (UserTopics)
18.	If TopicCount[t] ==0
19.	Clear the UserTopics[t] out from the UTT

From Figure 1 and the LFUPR algorithm in Table 3, our dynamic adaptation strategies maintain user profiles such that the short-term model is updated by the LFUPR (line 1 $\sim$ line 16), while the degree of preferences in the long-term model could be degraded (line 13) when the page in the PHB is replaced, and could be accumulated when the user clicks the page ("adding" operation). On the other hand, if the user accesses the web page whose associated topic is not in the current user topic tree, the new node could be added into the tree ("adding" operation). From line 17 to line 19, if the "$TopicCount$" of one node becomes zero, the node would be deleted from the tree. This procedure is called the "deleting" operation. The "adding" and "deleting" operations dynamically adapt the structure of the long-term model to the user click behaviors. Although we design independent models for short-term and long-term user preferences, our strategies ensure that the inherent correlations between them are not ignored, and that the changes of the short-term model have an even influence on the long-term model. Here, the meaning of "even" is that we degrade the "$TopicCount$" not on an hour-to-hour or a day-to-day basis, only after a period of time during which the user has not accessed the *topic* in the whole search process (least frequently used one will be processed).

4. User-Centered Ranking Lists

4.1 Hierarchical Semantic Similarity

Li et al.[12] define the hierarchical semantic similarity as

$$HS(i,j) = e^{-\alpha \cdot l} \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}}\,,\ \alpha \geq 0\,, \beta > 0\,, \quad (1)$$

Their experiment results show that the optimal values of the two parameters are, α=0.2 and β=0.6. h means the depth of the subsumer (the deepest node common to two nodes), and l is the naïve distance (the number of edges between two nodes). Because one user profile includes a number of nodes, we further define the semantic similarity between one search result denoted by i and one user profile denoted by j as the maximum value among all the values computed by Equation (1). The re-ranked search results by our semantic similarity form a ranking list in order of one attribute of user interests (i.e., the topics a user is interested in).

4.2 Degree of User Interests

It is intuitional to think that the degree of the user interests ($TopicCount$) in a node of the user profile help improve the quality of the web search. The larger the value of $TopicCount$ is, the more interested the user is in one topic. Thus, the values of $TopicCount$ can also order the search results and produce a ranking list. To keep our rank aggregation from missing the high quality web pages in Google, we also consider the original rankling list of Google.

5. Methods for Rank Aggregation

5.1 Borda's Rule

The Borda's rule[21] is a single winner election method in which votes rank candidates in order of preference. The Borda's rule determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which she is ranked by each voter. Once all votes have been counted, the candidate with the most point is the winner. Because, from each voter, candidates receive a certain number of points, the Borda's rule is also classified as a positional voting system.

Let $A = a_1, a_2, \cdots, a_m$ be the set of positions in the ranking list, and let the attributes of user interests plus PageRank be named by elements of n. We shall assume for the present that every element of n can be expressed by a linear order in the position set A. We denote a linear order by a sequence $A_i = a_{i_1}, a_{i_2}, \cdots, a_{i_m}$ where for $j < k$, a_{i_j} is preferred to a_{i_k}. We apply a sort of modified Borda's rule here. The voter awards the first-ranked candidate with one point (i.e., 1). The second-ranked candidate receives half of the point (i.e., 1/2), the third-ranked candidate receives on third of the point (i.e., 1/3), etc. When all elements of n have been counted, and each A_i can be thought of a position vector, we sort the search results by the L_1 norm and the L_2 norm of these vectors, the median of the n points, and the geometric mean of the n points.

5.2 Spearman's Footrule

According to Diaconis et al.[3], the two measures which we consider are:

$$D(\pi, \sigma) = \sum_{i=1}^{m} \mid \pi(i) - \sigma(i) \mid\,, \quad (2)$$

$$S(\pi, \sigma) = \sum_{i=1}^{m} (\pi(i) - \sigma(i))^2\,. \quad (3)$$

π and σ are regarded as rankling lists here. Diaconis et al.[3] also suggest other two measures. One roughly seems similar to D, and the other is unsuitable for general use, having very small variance about a mean very close to its maximum value. Therefore, we choose D and S here.

Inspired by[5], we define a weighted balanced bipartite graph $G = (V_1 \cup V_2, W)$. $V_1 = r_1, r_2, \cdots, r_m$ is a set of search results to be ranked. $V_2 = p_1, p_2, \cdots, p_m$ is the m available position in the ranking list. For any two vertices $r \in V_1$ and $p \in V_2$, rp is an edge in G, thus G is also a complete bipartite graph. The weight $W(r,p) =$ is the total distance of a ranking value that places r at position p, given by $\sum_i^n \mid A_i(r) - p \mid$ or $\sum_i^n (A_i(r) - p)^2$. Minimizing the total distances to n could be solved by the well-known Hungarian algorithm that finds a minimum cost perfect matching in the bipartite graph. The time complexity is $O(m^3)$.

表 2 Procedures of Evaluation Experiments

1.	Issuing the query submitted by an online user to the Google API module ;
2.	Applying our rank aggregation based on the current user profile and then going into the Log module;
3.	Adapting the user profile to click-history data provided by the Log module through our strategies:
4.	For the long-term model updating the structure and the degree of user interests by the "adding" operation;
5.	For the short-term model, updating web pages in the PHB by the LFUPR algorithm;
6.	If needed, degrading the long-term model according to the changes of the short-term model by the "deleting" operation.
7.	Waiting until the online user submits a new query, and then going to 1.

6. Experiments

6.1 Experimental Setup

We choose the Google Directory Search as our baseline in that Google applies its patented PageRank technology on the Google Directory to rank the sites based on their importance. It is convenient for us to combine and evaluate our methods with Google. The necessary steps are depicted in Table 2. Main modules in the experiments are listed as follows:

(1) Google API module: Given a query, we are offered titles, snippets, and page-associated Google directories beside the URLs by the Google API ☆.

(2) Log module: We monitor user click behaviors, recording the query time, clicked search results, associated topics.

(3) User profile: It has been described in the former section.

6.2 Dataset

For each search, the Google API module got the top 20 Google results due to the limited number of the Google API licenses we have. We randomized the order of the results before returning the them to the online user. For evaluation, 12 invited subjects to search the web through our system are graduate students (5 females and 7 males) researching in several fields, i.e., computer, chemistry, food engineering, electrical engineering, art design, medical, math, architecture, and law.

In the first four days, the subjects input the queries on their majors, and then in the next three days the queries on their hobbies were searched. Finally, in the last three days, the subjects were required to repeat some queries done before. This repeated procedure gave a clear performance comparison between the current and earlier systems, as user profiles were updated search by search. After the data were collected over a ten-day period (From October 23nd, 2006, to November 1st, 2006), we got a log of about 300 queries averaging 25 queries per subject and about 1200 records of the clicked web pages in total.

6.3 Evaluation Metrics

1. **AvgRank** indicates the average rank of search results. An effective rank mechanism should place relevant search results close to the top of the rank list. We ask the subjects to select the search results they considers relevant to their interests. The measure is defined as follows:

$$AvgRank(u, q) = \sum_{p \in S} R(p)/|S| \,. \quad (4)$$

Here S denotes the set of the search results selected by subject u for query q, $R(p)$is the position of p in the ranking list, and $|S|$ is the number of selected search results. A smaller *AvgRank* represents a better quality.

2. **DCG**[7)] gives more weight to highly ranked search results, defined as follows:

$$DCG(i) = DCG(i-1) + G(i)/log(i) \,, \quad (5)$$

where if $i = 1$, $DCG(1) = G(1)$. In the experiments, we used $G(i) = 2$ for highly relevant web pages, $G(i) = 1$ for relevant web pages, and $G(i) = 0$ for non-relevant search results. A larger *DCG* means a better quality.

6.4 Experimental Results

6.4.1 Methods of Rank Aggregation

By the DCG measure[7)], we compared the qualities of our techniques and a simply linear combination (SLC) of measures[2)]. The results of the average improvements over all subjects are illustrated in Table 3. The DCG of SLC is 1.80557. Our rank aggregations brought better search results compared with SLC. The largest improvement is 14.9% produced by Biparitie_S. Borda's Rule is a positional method of ran aggregation, so its ad-

☆ http://code.google.com/apis/soapsearch

表 3 Quality of Rank Aggregation Measured

	Methods	DCG	Improvement
Borda	L_1 norm	1.93948	7.42%
	L_2 norm	1.95184	8.10%
	Median	1.97047	9.13%
	Geometric mean	1.98762	10.1%
Footrule	Bipartite_D	1.88534	4.42 %
	Bipartite_S	2.07534	14.9%

vantage is that it is linear complexity. But the problems with the positional method are that it neither optimizes any distance criterion nor satisfy the Condorcet proterty[22]. Thus, L_1 norm, L_2 norm, Median and Geometric mean performed worse than Bipartite_S. The Hungarian algorithm that finds a minimum cost perfect matching in the bipartite graph is $O(m^3)$ complexity, but showed the best results by the distance measure in Equation 3.

6.4.2 Changing Trend of Quality of Our Search System

Under *AvgRank*, Figure 2 illustrates the changing of the average improvement over all users by *Biparitie_S*. The system in the first three days is under a learning procedure, thus, the quality is lower than the other days, but keeping improved day by day (i.e., the difference between the average rank of Biparitie_S and that of Google Directory is decreasing). As a result of requiring the subjects to change queries from their majors to hobbies, we could see that from the fourth day to the fifth day, the improvement of our scheme experiences a sudden decrease. But after three days on learning the changes, our system reach the highest improvement (57.71%) in the tenth day, while for the fifth day the improvement becomes relative low (around 2%). This difference demonstrates that the changes of user interests will lower the improvement that our scheme could achieve, and further indicates the importance to learn the changes of user interests.

7. Conclusions and Future Work

In this paper we introduced how to capture the changes of user profiles from click-through data and how to utilize the two attributes of the user profiles to aggregate the ranking lists, thus creating personalized views of the web. First, we designed independent models for short-term and long-term

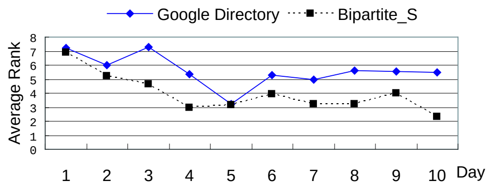

図 2 Changing Trend of Quality of Our Search System (Lower is better)

user interests to consist of a user profile. Then, we adapted the user profile, including the content and the structure, to the accumulation and degradation changes of user interests. Finally, we proposed a set of techniques for rank aggregation. Experimental results on real data demonstrate the effectiveness of our methods. We can also see that the approaches originated from social choice theory and graph theory produce a broadly acceptable ranking list in terms of various attributes of user interests, thus improve the quality of the web search.

In the future, when computing for the node distance in the tree, we plan to consider the edge distance, assigning a different weight for each edge, because each pair of two nodes linked by an edge has different semantic similarity. We could further mine the click-through data to extract more user-centered information and optimize the web search in terms of user satisfaction.

参 考 文 献

1) D. Billsus and M. J. Pazzani. A hybrid user model for news story classification. In *Proc. of the 7th Int'l Conf. on User modeling (UM'99)*, pages 99–108, Secaucus, NJ, USA, 1999.
2) P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using ODP metadata to personalize search. In *Proc. of the 28th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05)*, pages 178–185, Salvador, Brazil, 2005.
3) P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.

4) S. T. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: A system for personal information retrieval and re-use. In *Proc. of the 26th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'03)*, pages 72–79, Toronto, Canada, 2003.
5) C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. of the 10th Int'l Conf. on World Wide Web (WWW'01)*, pages 613–622, 2001.
6) P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Proc. of the 14th Int'l Conf. on World Wide Web - Special interest tracks and posters (WWW'06)*, pages 801–810, Chiba, Japan, 2005.
7) K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proc. of the 23nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'00)*, pages 41–48, 2000.
8) R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *Proc. of the 15th Int'l Conf. on World Wide Web (WWW'06)*, pages 477–486, Edinburgh, Scotland, UK, 2006.
9) W. Lam, S. Mukhopadhyay, J. Mostafa, and M. J. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proc. of the 19th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'96)*, pages 317 – 325, Zurich, Switzerland, 1996.
10) L. Li, Z. Yang, and M. Kitsuregawa. Aggregating user-centered rankings to improve web search. In *Proc. of the 22nd AAAI Conf. on Artificial Intelligence (AAAI'07)*, pages 1884–1885, Vancouver, British Columbia, Canada, 2007.
11) L. Li, Z. Yang, and M. Kitsuregawa. Personalizing web search via modelling adaptive user profile. In *Proc. of Data Engineering Workshop(DEWS'07)*, Hiroshima,Japan, 2007.
12) Y. Li, Z. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.*, 15(4):871–882, 2003.
13) B. Markines, L. Stoilova, and F. Menczer. Bookmark hierarchies and collaborative recommendation. In *Proc. of The 21st National Conf. on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference (AAAI'06)*, Boston, Massachusetts, USA, 2006.
14) F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proc. of the 15th Int'l Conf. on World Wide Web (WWW'06)*, pages 727–736, Edinburgh, Scotland, 2006.
15) V. Schickel-Zuber and B. Faltings. Inferring user's preferences using ontologies. In *Proc. of The 21st National Conf. on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference (AAAI'06)*, Boston, Massachusetts, USA, 2006.
16) X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proc. of the 2005 ACM CIKM Int'l Conf. on Information and Knowledge Management (CIKM'05)*, pages 824–831, Bremen, Germany, 2005.
17) S. J. Soltysiak and I. B. Crabtree. Automatic learning of user profiles- towards the personalisation of agent services. *BT Technology Journal*, 16(3):110–117, 1998.
18) M. Speretta and S. Gauch. Personalized search based on user search histories. In *Proc. of the IEEE / WIC / ACM Int'l Conf. on Web Intelligence (WI'05)*, pages 622–628, Compiegne, France, 2005.
19) J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of the 28th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05)*, pages 449–456, Salvador, Brazil, 2005.
20) D. H. Widyantoro, T. R. Ioerger, and J. Yen. Learning user interest dynamics with a three-descriptor representation. *JASIST*, 52(3):212–225, 2001.
21) H. P. Young. An axiomatization of Borda's rule. *Journal of Economic Theory*, 9(1):43–52, 1974.
22) H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988.