Improving Re-ranking of Search Results using Collaborative Filtering

Rohini Uppuluri
Language Technologies Research Center,
International Institute of Information Technology,
Hyderabad, India
rohini@research.iiit.ac.in

Vamshi Ambati
Regional Mega Scanning Center,
International Institute of Information Technology,
Hyderabad, India
vamshi@iiit.ac.in

Abstract

Search Engines today often return a large volume of results with possibly a few relevant results. The notion of relevance is subjective and depends on the user and the context of search. Re-ranking of these results to reflect the most relevant results to the user, using a user profile built from the relevance feedback has proved to provide good results. Our approach assumes implicit feedback gathered from a search engine query logs and learn a user profile. The user profile typically runs into sparsity problems due to the sheer volume of the WWW. Sparsity refers to the missing weights of certain words in the user profile. In this paper we present an effective re-ranking strategy that compensates for the sparsity in a user's profile, by applying collaborative filtering algorithms. Our evaluation results show an improvement in precision over approaches that use only a user's profile.

Introduction

In general, interactions with current day web search engines could be characterized as "one size fits all". This means that all queries, posed by different users are treated similarly as simple keywords where the aim is to retrieve web pages matching the keyword. As a result, though the user has a focused information need, due to the excess information on the WWW, the amount of results returned for a particular keyword search is enormous. This places burden on the user to scan and navigate the retrieved material to find the web pages satisfying his actual information need. For example, two different users may use exactly the same query "Java" to search for different pieces of information - the Java island in Indonesia or the Java programming language. Existing IR systems would return a similar set of results for both these users. Incorporating the user's interests and focus into the search process is quite essential for disambiguating the query and providing personalized search results.

One way to disambiguate the words in a query is to associate a categorical tag with the query. For example, if the category "software" or the category "travel" is associated with the query "java", then the user's intention becomes clear. By utilizing the selected categories as a context for the query, a search engine is likely to return documents that are more suitable to the user. Current search engines such as Google or Yahoo! have hierarchies of categories to help users to specify his/her categories manually to the query. Unfortunately, such extra effort can not be expected from the user in a web search scenario. Instead it is preferred to automatically obtain a set of categories for a user query directly by a search engine. However, categories returned from a typical search engine are still independent of a particular user and many of the returned document results could belong to categories that may not reflect the intention of the searcher. This demands further personalization of the search results.

In this paper, we propose a two phase strategy to personalize search results over the WWW. We first learn a user profile based on his relevance feedback and use it effectively in a re-ranking phase to provide personalized search results. Since it is difficult to assume that the users will provide the relevant documents ((Kelly & Teevan 2003), (Kelly & Belkin 2001), (Kim, Oard, & Romanik 2000) etc) explicitly, we make use of the implicit feedback given by the users which are captured in search engine interactions as "Query logs" or "Click through data". In the rest of the paper, we use the terms click through data, query log data and implicit feedback interchangeably. Such data consists of the queries, clicked documents and the identity of the user say ip address and is invaluable for research in search personalization.

Liu et al (Liu, Yu, & Meng 2002) successfully built user profiles for re-ranking by incorporating query categories in the learning process. We follow a similar approach to (Liu, Yu, & Meng 2002). We first infer the category for a query using existing search engines and open directory project (ODP). We use the queries,their respective categories and the corresponding clicked documents in the learning of a user profile for the user. The user profile is represented as a matrix containing the pairs (term, category) and their corresponding weights. Machine learning algorithms are used to automatically learn these term weights in the matrix. Each element represents how important the term is when the user is searching for a query of the given category. Re-ranking of the search results based on the user profile thus built, has shown improvement in performance. Though category helps to disambiguate the query, it adds another extra dimension to the user profile. This typically brings in sparsity in the user profile, which was observed in our case. Sparsity refers to the missing weights of certain words in the user profile.

In this paper, we present an effective re-ranking solution that compensates for the sparsity in a user's profile, by collaborative filtering algorithms. A great deal of information overlap exists in web searches among users [(Balfe *et al.* 2004), (Smyth *et al.* 2005), (Balfe & Smyth 2005) (Fitzpatrick & Dent 1997), etc]. This overlap is seen due to users with similar information needs, posing similar queries. To our knowledge, this vast and rich source of information overlap hasn't much been properly exploited for the WWW. Collaborative algorithms work exceptionally well in a community like environment with significant overlap of information needs and interests. The novelty of our re-ranking algorithm lies in addressing the sparsity in the user profile by exploiting the information overlap, using collaborative filtering. Our approach shows an improvement in the overall results when compared to a re-ranking performed based on just the user profile.

The rest of the paper is organized as follows. Section 2 discusses the Related Work, Section 3 discusses the proposed approach of learning user profiles, the re-ranking strategy, and addressing the sparsity in the user profiles. Section 4 describes the experimental setup and evaluation. Section 5 described our conclusions and future work.

Related Work

The related work related to the approach proposed in this paper, can broadly be classified as work done in personalized search and work done in, collaborative filtering applied to search.

Personalized Search

There has been a growing literature available with regard to personalization of search results. In this section, we briefly overview some of the available literature. Page et al (Page *et al.* 1998) proposed personalized PageRank as a modification to the global PageRank algorithm. However, the computation of personalized PageRank in the paper is not addressed beyond the original algorithm. Haveliwala (Haveliwala 2002) used personalized PageRank scores to enable topic sensitive web searches. However, no experiments based on a user's context such as browsing patterns, bookmarks and so on were reported. Pretschner (Pretschner & Gauch 1999) used ontology to model a users interests, which are studied from users browsed web pages. Speretta and Gauch (Speretta & Gauch 2004) used users search history to construct user profiles. Liu et. al (Liu, Yu, & Meng 2002) performed personalized web search by mapping a query to a set of categories using a user profile and a general profile learned from the user's search history and a category hierarchy respectively. Shen et. al (Shen., Tan., & Zhai. 2005) proposed a decision theoretic framework for implicit user modeling for personalized search. They consider the short term context in modeling user. Radlinski and Joachims [(Radlinski & Joachims 2005a), (Radlinski & Joachims 2005b)] learn a ranking function using Support Vector Machines and using it to improving search results.

Collaborative Filtering and Search

Chidlovski et al (Chidlovskii, Glance, & Grasso 2000) describes the architecture of a system performing collaborative re-ranking of search results. The user and community profiles are built from the documents marked as relevant by the user or community respectively. These profiles essentially contain the terms and their appropriate weights. Re-ranking of the search results is done using the term wights using adapted cosine function. The search process and the ranking of relevant documents are accomplished within the context of a particular user or community point of view. However the paper does not discuss much about the experimental details. Sugiyama et.al (Sugiyama, Hatano, & Yoshikawa. 2004) performed personalization by adapting to users interests without any effort from users. Further, they modified the traditional memory based collaborative filtering algorithm to suit to the web search scenario and used it to improve the search results. They constructed a user-term weights matrix analogous to user-item matrix in memory based collaborative filtering algorithms and then applied traditional collaborative filtering predictive algorithms to predict a term weight in each user profile. Lin et. al (Lin. *et al.* 2005) presented an approach to perform personalized web search based on PLSA, Probabilistic Latent Semantic Analysis, a technique which stems from linear algebra. They extracted a co-occurrence triple containing the users, queries, and web pages by mining the web-logs of the users and modeled the latent semantic relationship between them using PLSA. Armin Hust (Hust 2005) performed query expansion by using previous search queries by one or more users and their relevant documents. This query expansion method reconstructs the query as a linear combination of existing old queries. The terms of the relevant documents of these existing old queries are used for query expansion. However, the approach does not take the user into account. In [(Smyth *et al.* 2005), (Balfe *et al.* 2004), (Smyth *et al.* 2003), (Balfe & Smyth 2005), (Smyth *et al.* 2003), (Freyne *et al.* 2004)] a novel approach to web search - Collaborative Web search was introduced. It combined techniques for exploiting knowledge of the query-space with ideas from social networking to develop a Web search platform capable of adapting to the needs of communities of users. In brief, the queries submitted and the results selected by a community of users are recorded and reused in order to influence the results of future searches for similar queries. Results that have been reliably selected for similar queries in the past are promoted. Uppuluri and Ambati (Uppuluri & Ambati 2005) proposed an approach for re-ranking of search results in a digital library scenario. The user profiles were constructed from the documents marked as relevant or irrelevant. Re-ranking of the results is done using the user profile and profile of others users in the community. They assumed and assigned a set of static communities for each user which the user has selected while registering with the system. Also, the user also selects the community before posing the query and the re-ranking is done based on the community selected.

Several other works have made use of past queries mined from the query logs to help the current searcher. see [(Raghavan & Sever 1995), (Ji-Rong Wen & Zhang 2002), (Fitzpatrick & Dent 1997), (Glance 2001), (Hust 2005) etc]

Proposed Approach

The proposed approach to search result personalization consists of two phases. The first is a learning phase and the second is a retrieval/re-ranking phase. We use ”click through data” from a real world search engine, www.alltheweb.com, to build and test our proposed approach. In this section we discuss in detail our approach of learning user profiles and re-ranking search results for personalization.

Mapping Query to a Category

User profiles are learned on implicit feedback data, annotated with the category of the query posed. However, the click through data used here, does not consist of an associated category for the query. We therefore enhance the click through data by assigning category information to all the queries using the ODP, Open Directory Project (http://dmoz.org). The DMOZ Open Directory Project (ODP) is the largest, most comprehensive human-edited web page catalog currently available. It covers 4 million sites filed into more than 590,000 categories (16 wide-spread top-categories, such as Arts, Computers, News, Sports, etc.) Currently, there are more than 65,000 volunteering editors maintaining it. ODP's data structure is organized as a tree, where the categories are internal nodes and pages are leaf nodes.

In our work, we consider only the top most ODP categories in the hierarchy to classify the query into categories. The category information of the query can be obtained by posing the query to one or more of the directory services available (directory.google.com, http://dmoz.org). which returns the related categories to the query. Otherwise, based on the categories of the top 10 documents in the search results, the most common document category is chosen and selected as the query category. Other effective solutions for query categorization exist, like training a text classifier on the documents contained in the ODP data. Such a classifier could be used to categorize the clicked documents in the click-through data. Improvements in performance of query categorization are always possible, and will enhance our proposed approach. However, for simplicity we currently focus on the former approach for query categorization that depends on direct ODP lookup.

Learning User Profiles

We use machine learning algorithms for learning the user profiles from the implicit feedback provided by the user. The input to the learning algorithms is a user's implicit relevance feedback, gathered from the click through data, along with the query and the associated category. Learning the user profile involves learning the weights of certain features extracted from the implicit feedback. The effectiveness of the user profile depends to a large extent on the representation of features. As mentioned earlier, we consider the features (term, category) to effectively represent the context through the category. The weights of the features represent the importance of the term for the respective category.

We considered SVM for learning the weights of the features for its success in various text applications (Vapnik 1995),(Yang & Liu 1999) An SVM is trained using our proposed features and at the end of the training phase, the weights of the features are learned which constitutes our user profile. The procedure of learning weights is similar to Radlinski and Joachims (Radlinski & Joachims 2005a). SVM^{light} (Joachims 1999) has been used for training the SVM.

Re-ranking

Re-ranking of the results is done by first retrieving a set of documents matching the query using a search engine. Then the top documents returned by the search engine are reranked using the userprofile in the following manner. At first the test query category is inferred similar to the learning phase as discussed in section above (Mapping query to a category). Then for each word in the document, the weight of the pair (term, category) is obtained from the user profile. Let c be the identified category of the query, t be a word in the document D_j and w_{t,D_j} be its weight in the document D_j (typically the term frequency TF or TFIDF etc). t_c represent the pair (term, category). UP_{a,t_c} represents the weight of t_c in the user profile of the user a. CP_{a,t_c} represents the predicted weight of t_c using collaborative filtering. Then the document ranks are computed as weighted combination of its term frequency TF in the document and the weight obtained from his user profile as shown in Equation 1. Re-ranking is done by sorting the documents in decreased order of their rank.

$$R_{D_j,Q_c} = \sum_{t \in D_j} \alpha w_{t,D_j} + (1-\alpha) W_{a,t_c} \tag{1}$$

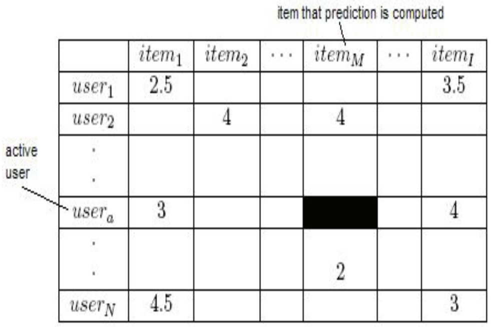
$$W_{a,t_c} = \begin{cases} UP_{a,t_c} & \text{if} \quad UP_{a,t_c} \neq 0 \\ CP_{a,t_c} & \text{otherwise} \end{cases} \tag{2}$$

Based on experimentation, we set the value of α to be 0.7.

Addressing Sparsity in the User Profile to improve Re-ranking

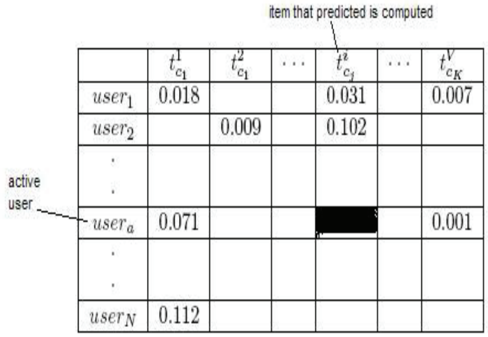
Usage of a (term, category) pairs helps to disambiguate the query and act as good contextual information in building a user profile. However, this typically brings in sparsity in the user profile, due to an added dimension to the user profile - the category of the query. Sparsity refers to missing weights of certain words in the user profile. We address the sparsity in the userprofile using collaborative filtering to improve the re-ranking of the documents. Certain weights of the pairs (term, category) not occurring the user's profile are predicted using the adapted version of the collaborative filtering which we present below.

In the following subsections, we first briefly review the pure collaborative filtering algorithms, especially neighborhood-based algorithms, and then describe the adapted collaborative filtering algorithms to address the sparsity in the userprofiles and then present how we make predictions of the pairs (term, category).


item that prediction is computed
active user

	$item_1$	$item_2$	$\cdots$	$item_M$	$\cdots$	$item_I$
$user_1$	2.5					3.5
$user_2$		4		4		
⋮						
$user_a$	3			■		4
⋮				2		
$user_N$	4.5					3

Figure 1: Sample user-item matrix


item that predicted is computed
active user

	$t^1_{c_1}$	$t^2_{c_1}$	$\cdots$	$t^i_{c_j}$	$\cdots$	$t^V_{c_K}$
$user_1$	0.018			0.031		0.007
$user_2$		0.009		0.102		
⋮						
$user_a$	0.071			■		0.001
⋮						
$user_N$	0.112					

Figure 2: Sample user-(term,category) matrix

Overview of the Pure Collaborative Filtering Algorithm Collaborative filtering is one of the most successful recommendation algorithms. They have been popular for recommending news [(Resnick *et al.* 1994), (Konstan *et al.* 1997)], audio CDs, movies, music (Cohen W 2000), research papers etc. Recommendations are typically computed using the feedback taken from all the users in the community represented in a user-item matrix. The entries in the user-item matrix are the ratings given by the respective users for the respective items. Collaborative filtering can broadly be seen as the problem of predicting missing values in a user-item ratings matrix. Figure 1 shows a simplified example of a user-item ratings matrix. In the neighborhood-based algorithm (Herlocker *et al.* 1999), a subset of users is first chosen based on their similarity to the active user, and a weighted combination of their rating is then used to produce predictions for the active user. The algorithm we use can be summarized in the following steps: 1. Weight all users with respect to similarity to the active user. This similarity between users is measured as the Pearson correlation coefficient between their rating vectors. 2. Select n users that have the highest similarity with the active user. These users form the neighborhood. 3. Compute a prediction from a weighted combination of the neighbors ratings. In step 1, $S_{a,u}$, which denotes similarity between users a and u, is computed using the Pearson correlation coefficient defined below: where $r_{a,i}$ is the rating given to item i by user a, and $\overline{r_a}$ is the mean rating given by user a, and I is the total number of items. In step 2, i.e., neighborhood-based methods, a subset of appropriate users is chosen based on their similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user in the next step 3. In step 3, predictions are computed as the weighted average of deviations from the neighbors mean: where $p_{a,i}$ is the prediction for active user a for item i as shown in Equation (4). $S_{a,u}$ is the similarity between users a and u as described at Equation (3), and n is the number of users in the neighborhood.

$$S_{a,u} = \frac{\sum_{i=1}^{I}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i=1}^{I}(r_{a,i} - \overline{r_a})^2 \sum_{i=1}^{I}(r_{u,i} - \overline{r_u})^2}} \tag{3}$$

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{n}(r_{u,i} - \overline{r_u})S_{a,u}}{\sum_{i=1}^{I} S_{a,u}} \tag{4}$$

Adapted Collaborative Filtering Algorithm In the pure collaborative filtering algorithms described above. we considered a user-item ratings matrix. Similarly, we now consider user-(term,category) matrix see Figure 2. Each row in the matrix represents the entries in the user profile of the respective user. By representing the user profile in this fashion, collaborative filtering algorithms can directly be applied.

The prediction of the (term,category) weights are computed by first identifying a set of similar users (ie users who has similar (term,category) weights as measured using Equation 5. Then using these users, the predictions are computed analogous to pure collaborative filtering as shown in Equation 6

$$S_{a,u} = \frac{\sum_{t_c \in (T,C)}(w_{a,t_c} - \overline{w_a})(w_{u,t_c} - \overline{w_u})}{\sqrt{\sum_{t_c \in (T,C)}(w_{a,t_c} - \overline{w_a})^2 \sum_{t_c \in (T,C)}(w_{u,t_c} - \overline{w_u})^2}} \tag{5}$$

$$p_{a,t_c} = \overline{w_a} + \frac{\sum_{u=1}^{n}(w_{u,t_c} - \overline{w_u})S_{a,u}}{\sum_{t_c \in (T,C)} S_{a,u}} \tag{6}$$

where $p(u, t_c)$ is the predicted computed for term t in query Category c and is equal to CP_{a,t_c}.

Then the re-ranking of the document is done as described in Equation (1).

Experiments

Data and Experimental Setup

Query log data used in the experiments consist of the query, the clicked URLs for the query and the user identifier (ip addresses) and the time of click of the document. Such information though invaluable for research on information retrieval, is not released by major search engines. Recently, Alltheweb.com[1] has made available its search logs for research purposes. The data was collected from queries mainly submitted by European users on 6 February 2001.

[1] http://alltheweb.com

The data set contains approximately a million queries submitted by over 200,000 users and 977,891 unique click URLs. Further information on the data can be found in (Bernard J. Jansen 2005).

We use the query log data released by Alltheweb.com to perform our experiments and to evaluate the proposed approach We first divide the query logs into a large chunk of training clickthrough data, used for learning user profiles and a smaller chunk for testing and evaluating the approach. A direct evaluation experiment of our proposed re-ranking algorithm can not be performed on the present day's search indices of Alltheweb.com or any other search engine for that matter. Document repositories on the WWW have been changing drastically and undergo restructuring. Hence evaluation results can not be based on user profiles learnt from the query logs used in the current experiment. Therefore we first obtain all the documents corresponding to the queries in the testing data by crawling the click URLs and storing them as a repository. We were only successful in retrieving about 40% of the actual click URLs due to broken links and restructing of the WWW. These retrieved documents constitute the document repository used in current test experiments. With the volume of query log data we are working with, this repository could be considered as an analog to the WWW that corresponds to the query logs in discussion. For the purposes of these experiments, we name this repository as the mini-WWW, consisting of about 35,000 documents. We also pick queries from the query log data and pose it to Google to fetch and download the top 100 documents. These documents are added to the mini-WWW. This prevents any kind of bias that may have been introduced in the construction of mini-WWW from click URLs in the query log data. With availability of every day query log data we expect the proposed approaches to scale and be useful in the WWW scenario. We used Lucene [2], an open source search engine for indexing this mini-WWW repository. All the evaluations reported below are obtained by performing our experiments through Lucene's search engine.

Evaluation

The testing data extracted from the clickthrough data is now used for evaluating the performance of our re-ranking approach. The test data consists of 5,000 queries posed by 780 users, with an average repetition of 15.9% in the queries. Repeated experiments have been conducted by using subsets of this training data. Each query from the testing data set is posed to the search engine for the mini-WWW and results obtained are cross-validated with references to the actual clicked documents in the testing data. We follow an evaluation approach similar to the one followed in (Balfe *et al.* 2004) . We compare three methods of re-ranking. Firstly we consider the ranking provided by search engine, in this case Lucene's default ranking. The second approach tested is the ranking based on only the user profile. The third is the proposed approach for addressing the data sparsity problem using collaborative filtering. We refer to them as 'unranked', 'only user profile' and 'collaborative' respectively. The evaluation metrics used for comparison are minimum accuracy and precision @ N, N=5, 10 and 20. We could not evaluate the standard collaborative filtering measures like MAE etc because we assume boolean relevant judgments as opposed to the former which use ratings typically ranging from 0-5.

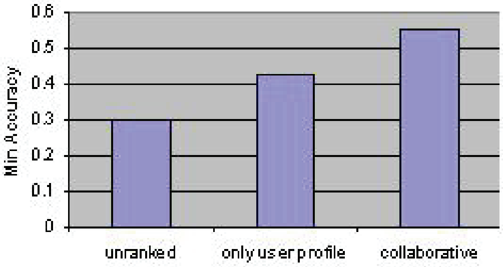


Figure 3: Minimum accuracy comparison

Method	Precision@5	10	20
unranked (UP)	0.09	0.18	0.22
only user profile (PP)	0.15	0.26	0.34
collaborative (CP)	0.23	0.38	0.55
Improvement(PP over UP)	6%	8%	12%
Improvement(CP over PP)	8%	12%	21%

Table 1: Precision@N values for the approaches 'uranked' (UP), 'only user profile' (PP) and 'collaborative' (CP)

Minimum Accuracy Minimum accuracy has been used in (Balfe *et al.* 2004) in evaluation of their approach. It measures the ability of a search engine to return at least a single relevant result in returned results. We compare the top 30 results returned by our ranking approaches in calculating the minimum accuracy. The percentage of the queries for which at least one relevant result is returned is computed. The results are presented in Figure 3.

Precision We used precision at N (p@N) defined as the number of relevant documents at a given cut-off rank at N. It is a widely used metrics for evaluating approaches performing re-ranking of results. The reported value is averaged over all users and queries. We compared the three approaches mentioned above. The precision values for the three approaches 'uranked' (UP), 'only user profile' (PP) and 'collaborative' (CP). The results are shown in the Table 1. As it can be seen from the table, our approach showed an improvement over other approaches.

Conclusions and Future Work

In this paper, we proposed a two phase strategy to personalize search results over the WWW. We first learn a user profile from his "clickthrough data", collected from a real world search engine. This user profile is then used in a re-ranking phase to personalize the search results. We also

[2] http://lucene.apache.org

used query and its category information to in learning the user profile. Category information helps to disambiguate the query and focus on the information need. However, in the scenario of WWW search, it adds another extra dimension to the user profile, typically bringing in sparsity in the user profile. We propose an effective re-ranking strategy that compensates for the sparsity in a user's profile, using collaborative filtering algorithms. We evaluate our approach using standard information retrieval metrics, to show an improvement in performance over earlier re-ranking strategies based on only user profile.

Acknowledgements

We would like to thank Dr. Vasudeva Varma, for dicussions on collaborative filtering and re ranking of search results.

References

Balfe, E., and Smyth, B. 2005. An analysis of query similarity in collaborative web search. In *In Proceedings of the European Conference on Information Retrieval*, 330–344. Springer-Verlag.

Balfe, B. S. E.; Freyne, J.; Briggs, P.; Coyle, M.; and Boydell, O. 2004. Exploiting query repetition & regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research* 383–423.

Bernard J. Jansen, A. S. 2005. An analysis of web searching by european alltheweb.com users. *Information Processing and Management* 41:361–381.

Chidlovskii, B.; Glance, N.; and Grasso, A. 2000. Collaborative re-ranking of search results. In *Proc. AAAI-2000 Workshop on AI for Web Search.*

Cohen W, F. W. 2000. Web-collaborative filltering: Recommending music by crawling the web. In *In Proceedings of WWW-2000.*

Fitzpatrick, L., and Dent, M. 1997. Automatic feedback using past queries: Social searching? In *In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 306–313. ACM Press.

Freyne, J.; Smyth, B.; Coyle, M.; Balfe, E.; and Briggs, P. 2004. Further experiments on collaborative ranking in community-based web search. *Artificial Intelligence Review* 21(3-4):229–252.

Glance, N. S. 2001. Community search assistant. In *In Proceedings of the International Conference on Intelligent User Interfaces*, 91–96. ACM Press.

Haveliwala, T. H. 2002. Topic-sensitive pangerank. In *Proceedings of the 11th International World Wide Web Conference (WWW2002)*, 517–526.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR*, 230–237.

Hust, A. 2005. Query expansion methods for collaborative information retrieval. *Inform., Forsch. Entwickl.* 19(4):224–238.

Ji-Rong Wen, J.-Y., and Zhang, H.-J. 2002. Query clustering using user logs. *ACM Transactions on Information Systems (TOIS)* 20(1):59–81.

Joachims, T. 1999. Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning*.

Kelly, D., and Belkin, N. J. 2001. Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback during interactive information retrieval. In *Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval (SIGIR '01)*, 408–409.

Kelly, D., and Teevan, J. 2003. Implicit feedback for inferring user preference: A bibliography. In *SIGIR Forum*, volume 32.

Kim, J.; Oard, D.; and Romanik, K. 2000. Using implicit feedback for user modeling in internet and intranet searching.

Konstan, J. A.; Miller, B. N.; Maltz, D.; Herlocker, J. L.; Gordon, L. R.; and Riedl, J. 1997. Grouplens: Applying collaborative filtering to usenet news. In *Communications of the ACM*, volume 40 of *3*, 77–87.

Lin., H.; Xue., G.-R.; Zeng., H.-J.; and Yu, Y. 2005. Using probabilistic latent semantic analysis for personalized web search. In *Proceedings of APWEB'05*.

Liu, F.; Yu, C.; and Meng, W. 2002. Personalized web search by mapping user queries to categories. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM '02), ACM Press*, 558–565.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

Pretschner, A., and Gauch, S. 1999. Ontology based personalized search. In *ICTAI.*, 391–398.

Radlinski, F., and Joachims, T. 2005a. Evaluating the robustness of learning from implicit feedback. In *ICML Workshop on Learning In Web Search*.

Radlinski, F., and Joachims, T. 2005b. Query chains: Learning to rank from implicit feedback. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM 2005.*

Raghavan, V. V., and Sever, H. 1995. On the reuse of past optimal queries. *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* 344–350.

Resnick, P.; Iacovou, N.; Suchak, M.; and Bergstorm, J. R. P. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW '94)*, 175–186.

Shen., X.; Tan., B.; and Zhai., C. 2005. Context-sensitive information retrieval using implicit feedback. In *Proceedings of SIGIR 2005*, 4350.

Smyth, B.; Balfe, E.; Briggs, P.; Coyle, M.; and Freyne, J. 2003. Collaborative web search. In *In Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03*, 1417–1419. Morgan Kaufmann.

Smyth, B.; Balfe; Boydell, O.; Bradley, K.; Briggs, P.; Coyle, M.; and Freyne, J. 2005. A live-user evaluation of collaborative web search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, Scotland*.

Speretta, M., and Gauch, S. 2004. Personalizing search based on user search histories. In *Thirteenth International Conference on Information and Knowledge Management (CIKM 2004)*.

Sugiyama, K.; Hatano, K.; and Yoshikawa., M. 2004. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of WWW 2004*, 675 – 684.

Uppuluri, R., and Ambati, V. 2005. A collaborative filtering based re-ranking strategy for search in digital libraries. In *proceedings of 8th ICADL - 2005*.

Vapnik, V. N. 1995. The nature of statistical learning theory. Springer.

Yang, Y., and Liu, X. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, 42–49.