A Collaborative Filtering based Re-ranking Strategy for Search in Digital Libraries

Rohini U and[1]and Vamshi Ambati[2]

[1] International Institute of Information Technology
Hyderabad, India
rohini@research.iiit.ac.in
[2] Institute for Software Research International, Carnegie Mellon University, Forbes Avenue, PA
vamshi@cmu.edu

Abstract. Users of a digital book library system typically interact with the system to search for books by querying on the metadata describing the books or to search for information in the pages of a book by querying using one or more keywords. In either cases, a large volume of results are returned of which, the results relevant to the user are not often among the top few. Re-ranking of the search results according to the user's interest based on his relevance feedback, has received wide attention in information retrieval. Also, recent work in collaborative filtering and information retrieval has shown that sharing of search experiences among users having similar interests, typically called a community, reduces the effort put in by any given user in retrieving the exact information of interest. In this paper, we propose a collaborative filtering based re-ranking strategy for the search processes in a digital library system. Our approach is to learn a user profile representing user's interests using Machine Learning techniques and to re-rank the search results based on collaborative filtering techniques. In particular, we investigate the use of Support Vector Machines(SVMs) and k-Nearest Neighbour methods (kNN) for the job of classification. We also apply this approach to a large scale online Digital Library System and present the results of our evaluation.

1 Introduction

Digital Libraries(DLs) have received wide attention in the recent years allowing access of digital information from anywhere across the world, providing additional services typically not available in a traditional digital library. Examples include alerting services and recommendation, search facilities, [4] [14] and others. Earlier work on DLs has focussed more on making digital content available to a generic user. With the tremendous progress achieved, DLs should now move from serving a generic user to customizing and adapting to a specific user's interests [8][2][6].

In practice, users in a information resource, use the same resource over and over to gather the required information. But, the time consuming effort put in

searching the resource is often forgotten and lost. Hence it requires a repetition of the manual labor of searching and browsing, every time the information is accessed. This can be avoided if the system learns the user's needs from his interactions with the system and adapts to his requirements. Though the complete set of interests of a user usually differs from any other user, there is a great possibility of overlap of interests among the users if the information in the information source matches their requirements, expectations and motivation. Hence, users benefit by sharing information, experiences and awareness eventually evolving into a community, a group of people who share common interests. Such sharing of information has been widely used in collaborative filtering methods also called as Community Based methods. These methods became popular for recommending news [15], audio CDs[19], movies [1] music, research papers [12] etc. Recommendations to a particular user are typically computed using the feedback namely in the form of item ratings taken from the other users in the community. It is advantageous if the users in a DL can collaborate in a similar way and share the information. This could save the labourious effort put by a user in finding the book to a great extent.

Users in a digital library search for books by querying on the metadata of the book, called 'Metadata Search' or search for information in the pages of the book by querying using keywords, called 'Content Search'. Due to the myriad of information available, the search engines in a digital library that perform the afore mentioned searches, often return a huge list of results. Though these results might contain the query terms, the results are not necessarily relevant to the user and are often not presented in the order of relevance to the user. Re-ranking the results to contain the most relevant documents on the top is useful and is a well known problem in the area of information retrieval. We aim to improve the relevance of the results to a given user by re-ranking them using the profiles of the user and the profiles of other users in the community he belongs to. A profile of a user is a representation of his interests and is learnt from his interactions with the digital library system. A Profile in the case of metadata search is built from the ratings of the books that the user provides explicitly. Similarly in content search, a user profile is built from the content of the pages that have been judged as relevant by the user.

The rest of the paper is organized as follows. Section2 discusses the related work on digital libraries, Section3 discusses the system and the proposed re-ranking strategy in detail, Section4 discusses our Experimental Setup and presents our preliminary Evaluation Results. Section5 presents the Conclusions on our work also briefing our future work.

2 Related-Work

Earlier work on collaboration in DL concentrated on providing alerting services [4], customizable and personalized arrangement of folders in which the Digital

[1] http://movielens.umn.edu/

library Objects(DLOs) are stored, sharing of these folders and group recommendations based on organization of the DLOs in folders [14], additionally recommending users [20]. PASS[2] provided personalized service in a Digital Library by computing similarity between user profiles and documents in a pre-classified research domain. The system described in [6] emphasized collaborative functions of DLs by grouping users based on their profiles. [11] discusses the problem of collaborative search across a large number of digital libraries in a peer-to-peer (P2P) environment where both digital libraries and users are equally viewed as peers. [16] discusses a number of hybrid appraoches combining collaborative and content based methods for recommending research papers to users in a digital library of research papers like CiteSeer[1]. [10] describes an approach for personalized search exclusively for a medical digital library by re-ranking the search results using modified cosine similarity.

The objective of this work is to reduce the user's effort in two search processes in a digital library namely the metadata search and the content search. We believe that this will greatly improve the user's search experience in a DL. We approach this by providing customized results to the user by filtering and rearranging the search results. We investigate the use of community based methods and machine learning techniques for the same. In particular we apply techniques like K-Nearest Neigbour (kNN) and Support Vector machines (SVMs). kNN methods has been very popular in collaborative filtering [3] for finding user and item similarities. SVMs are a popular classification technique backing in statistical theory[21]. It has been applied with great success in various text applications like textclassification[22][9], webpages classification and others. Recently, they have been used for Text Retrieval[5] and achieved performance comparable and even better than the traditional approaches [17][7][18].

3 The Proposed Approach

In this section we describe the re-ranking approach and also show how it is applied to an online digital library system to enhance the search process in it. Re-ranking of search results involves calculating and assigning a score to the results of search returned by a search engine, using the profile of the particular user built from his feedback. However due to the large multitude of data present in today's DLs, it is not possible for a user to provide feedback to at least a significant proportion of them. Hence usage of a community to share across the relevance feedback becomes crucial and useful. The interests of a user regarding the books is learnt in a phase called profile learning through the ratings of the books provided by the user. We apply this approach in digital libraries. As mentioned earlier, users of a digital library search for books by querying on the metadata of the book, or search for information in the pages of the book by querying using keywords. The metadata search in a digital library returns books which the user could then provide feedback by rating them. This profile is the user's MS-Profile. The same profile can not be used to decide the relevance of

[1] http://citeseer.ist.psu.edu/

information on a page during a content search. Because, search results in the case of metadata search are books and in content search it is a page in the book. The relevance of a page merely talks about interest of some facts or pieces of information in the page and does not necessarily correspond to the whole book. Hence separate user profile is learned for content search using the relevance judgements of the pages given by the user. This profile is called the user's CS-profile. To summate, the approach consists of learning user profile and using the particular user profile along with the profiles of other users in the community to re-rank the search results intiated by the particular user.

In sub sections 3.1 and 3.2 we discuss how profile learning and re-ranking of search results takes place in in the metadata search and content search respectively, and in 3.3 we describe the architecture of the search process in a digital library that is enhanced by the proposed approach.

3.1 Collaboration based re-ranking in metadata search

Learning the user's MS-profile Users in a DL search for books using the metadata search and read them online. The user then rates the book on a scale of 1-5. These ratings reflect how much the book has been interesting to him. For each user, the system records these ratings in a database. This constitutes the user's MS-profile, essentially a vector of ratings of all the books given by him. Similarly, profiles are learnt for all the users. In the current work, only explicit feedback is assumed.

Re-ranking of search results Re-ranking of results is done reflecting the user's interests of the books. Ranking in done in descending order based on the ratings of the book if present in the given user's MS-profile. Otherwise, a prediction of the given user's rating of the book is computed using the particular user's MS-profile and the profile of other users in the community. The predicted rating is a weighted combination of prediction using only the particular user's MS-profile and prediction using the MS-profiles of other users in the community. For computing prediction of a book in the former case, kNN is used to pick the k most similar books to a given book among all the books rated by the user.

To describe the computation of these predictions in detail, we first introduce commonly used notation. Books in the DL are typically described by metadata information like 'Title of the Book','author' etc. Let a be a user in the DL, then the rating of the book b given by the user is denoted by $r_{a,b}$ and $p_{a,b}$ denotes the prediction of the book computed by the system. Then the predicted rating of a book b is computed as

$$p_{a,b} = \alpha p1_{a,b} + (1 - \alpha) p2_{a,b}$$

where $p1_{a,b}$ is the prediction calculated using on the user a's MS-profile and is computed as

$$p1_{a,b} = \frac{\sum_{k_b} r_{a,B_i} * Sim_{B_i,b}}{\sum_{k_b} Sim_{B_i,b}}$$

where k_b denotes the k most similar books to the book b. The similarity of the books is based on the metadata content of the book. For the particular books B_i B_j, the similarity is calculated as

$$Sim_{B_i,B_j} = \sum_{f \epsilon F} g(B_i(f), B_j(f))$$

where g is an appropriate scoring function and F is a set containing the metadata features of the book. These similarities are computed offline. Hence they do not add any overheard to the computation of prediction of rating of a book. $p2_{a,b}$ is the predicition calculated using the MS-profiles of the other users in the community and is computed similar to other works in Collaborative Filtering.

$$p2_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{U}(r_{u,i} - \overline{r_u}) X S_{a,u}}{\sum_{u=1}^{U} S_{a,u}}$$

and

$$S_{a,u} = \frac{\sum_{i=1}^{B}(r_{a,i} - \overline{r_a}) X (r_{u,i} - \overline{r_u})}{\sum_{i=1}^{B}(r_{a,i} - \overline{r_a})^2 X (r_{u,i} - \overline{r_u})^2}$$

U is the set of users in the community that the user belongs to, $S_{a,u}$ denote the similarity between the active user a and the user u in the community. B is the total number of books considered in the DL. To summarize, the rank of a book is computed as follows

$$Rank_{b,a} = \begin{cases} r_{a,b} & \text{if the book has already been rated by the user a} \\ p_{a,b} & \text{otherwise.} \end{cases}$$

The book results of the metadata search are then arranged in descending order of the ranks calculated as above.

3.2 Collaboration based re-ranking in Content Search

Learning the user's CS-profile A user in a DL searches for information in the books using the content search and also gives relevance judgement of the page. As mentioned earlier, CS-profile represents the facts or information inside the book that are of interest to the user. In the current work, user's CS-profile learning involves training a classifier on the pages using their relevance judgements given by the user as the class labels. The relevance judgements of the pages are gathered through interactions from the user which are boolean values consisting of a 1 which symbolizes relevant class or -1 which symbolizes irrelevant class. We investigated the use of SVMs in this work. Training SVM on the pages and their feedback results in a model which consists of two sets of hyperplanes , one hyperplane going through one or more examples of the non-relevant class and one hyperplane going through one or more examples of the relevant class. This model forms the user's CS profile and is now capable of classifying a page of a book into a relevant class or non relevant class. In this section, we use the terms documents and pages interchangeably.

For training the SVM, each document is represented as a vector with TF(Term Frequency) representations of the words in the pages of the books as features. After eliminating stop words and words not occuring in at least 5 pages, we obtained about 3000000 dimensions. There were a total of 36,000 books and 14 million pages. We use libsvm [1] in our work for experiments on SVM and Lucene[1] is the search engine.

Re-ranking of search results For re-ranking of search results of a given query posed by the particular user a, we first get all the documents matching the query using the search engine. Let U represent the set of all the users in the community that the user belongs to. Relevance factors for the documents are calculated using the CS-profile of the user a and the CS-profiles of U The Relevance factors represent the systems predictions of how much the page is of interest to the user. These factors are essentially the probability estimates of the membership of the document in the relevant class given by the SVM. Given the user a's CS-profile ,the CS-profiles of U, the search engine's $TFIDF-rank$ and the rating of the book $r_{a,B}$ new rank of the document to the user a is computed as

$$Rank_{a,d} = \alpha(\mathcal{P}_{a,d} + r_{a,B}) + \beta(\rho_{C,d}) + \gamma TFIDF - rank$$

where d is a page in a Book B, $\mathcal{P}_{a,d}$ is the probability estimate given by the the user a's profile. and α, β, γ values are parameters which can be selected by the user reflecting the relative importance given to the predicted rank of the page, the TFIDF rank etc.

3.3 Architecture

We test our approach by applying the re-ranking strategy to an online digital library. The digital library consists of a metadata database and the data servers where the actual content of the books is stored. A web based interface for the metadata search engine helps the user query for books using metadata. Also the content servers are indexed and search engine with a web interface is provided which queries for pages based on keywords. Both the search engines are enabled to gather feedback from the users. The re-ranking strategy is defined in the following two main components-

1. Profile Manager: The profile manager receives the feedback from the user and creates or edits the corresponding profile for the user. The profiles are saved in the corresponding profile database for the search.
2. Reranker: The re-ranker uses the profile database created by the profile manager and executes the ranking strategy. The search engines pass the result sets of a search to the re-ranker, which then re-orders the results and returns them.

[1] http://jakarta.apache.org/lucene

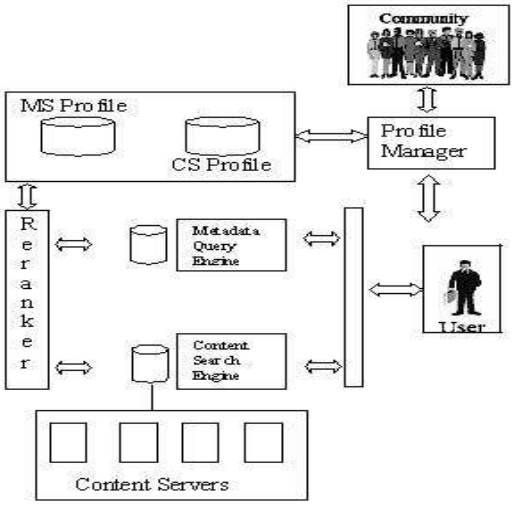

Fig. 1. Architecture of the enhanced retrieval process

4 Experimental Results

We have conducted preliminary experiments evaluating the Proposed Approach by applying it to the search process in The Digital Library of India Project. The project aims at digitizing books and enabling them online for the easy access to information for everyone around the world. The library currently contains about one tenth of a million books with close to 25 million pages. To ease the computations and processing we have only used a quarter portion of these books in our experiments. However the strategy and the algorithm scales and holds good for any number of books and pages. In this section, we describe the experiment setup used in evaluating the proposed approach and then present the metrics used and the evaluation results.

4.1 Experimental Setup

The first step in the evaluation involves discovering the communities. In this work, we have assumed static and pre-defined communities. For example, economy of India, religion, rocket science etc. The user joins the communities of interest to him of which he then becomes a member. A user can join more than one community depending on his interests. However, in the experiments that follow, we assume that the user explicitly chooses a community of interest before initiating a query. In this section, we use the terms participants and user interchangeably.

For user profile learning, we asked the users to search for books using the metadata specifying the community. All the books results returned by the metadata search engine were then shown to the user. The user then provides feedback on a significant number of books among the top 30. This was repeated for a few searches and the user's MS profile was learnt using the ratings provided. After profile learning, whenever the user queries the system for books, two sets

of results are returned. The first list of results are the results returned by the meta search engine and second list consists of results re-ranked using our ranking strategy described in Section 3.1. Again the participants were asked to rate the books in the top10 books presented in the two lists.

For evaluation of the effectiveness of our profile learning and ranking strategy, we use the following metrics borrowed from IR and Recommendation Systems and modify appropriately[13] 1. RelevanceRatio-N: proportion of books relevant among the top N. 2. Novelty Ratio-N: proportion of relevant books retrieved that have not yet been rated by the user in top N. RelevanceRatio-N describes the number of results relevant to the user out of the top N results. The Novelty Ratio-N symbolizes the number of new relevant books discovered by the system for which the user has not yet given feedback. A high RelevanceRatio-N value indicates that a good proportion of the results are relevant to the user. A high Novetly Ratio-N indicates that the system is able to learn and generalize the user's interests. Both of these measures combined determine the effectiveness of our ranking strategy. We followed similar procedure and experimental set up for evaluating the effectiveness of CS-profile learning and ranking strategy in content search. The user gives relevance judgements of the pages in this case instead of ratings. User profile learning and re-ranking of the search results is done as described in Section 3.2.

4.2 Evaluation Results

The Tables 1 and 2 show the averaged RelevanceRatio-N value over all the users. For simplicity of comparison, we showed the RelevanceRatio-N values for 5 users in the graph. For each user, the RelevanceRatio values shown in the graph are the values averaged over all the queries issued by the user in the experiments conducted. The NoveltyRatio-N helps us to assess the improvement of the search due to the re-ranking strategy and the evaluations proved it to be useful. In all these experiments, we considered N = 10 , ie only the top 10 search results were considered. As it can be seen from the tables, our ranking approach showed

Method	Average RelevanceRatio-10
Without re-ranking	0.2
Proposed Re-ranking	**0.404**

Table 1. Evaluation of Metadata Search

Method	Average RelevanceRatio-10
Without re-ranking	0.318
Proposed Re-ranking	**0.767**

Table 2. Evaluation of Content Search

significant improvement in the RelevanceRatio-10. With the number of users

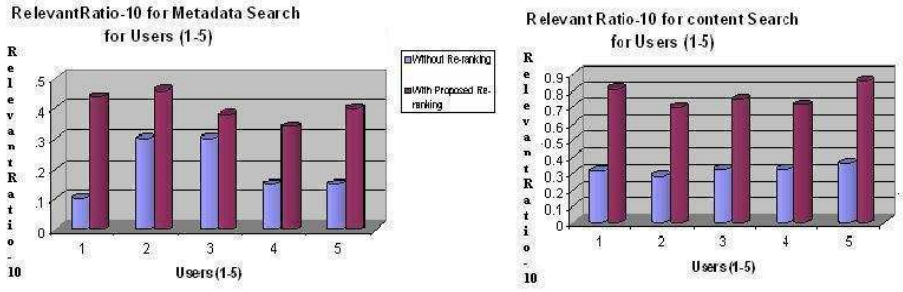

Fig. 2. RelevanceRatio-10 of Metadata search and Content search

in the community increase and the feedback increases, the results would show constant improvement.

5 Conclusions and Future Work

In this paper we have identified two types of searches performed in a digital library and have shown that by providing relevance feedback and by sharing of user experiences in a user community, the relevance of the results returned by these search could be improved. In particular we have experimented machine learning techniques like SVM in learning user profile models. We have also applied the techniques to the Digital Library of India project which is an evolving digital library consisting of about one tenth of a million books and also presented the results of evaluation. In future we would like to experiment other machine learning techniques and compare the results. Also the success of the system discussed above depends largely upon the user and the community that he belongs to. We would like to define the communities based on user behaviour studies and also discover the similar users dynamically instead of requiring the user to pre define it.

References

1. Chih-Chung Chang and Chih-Jen Lin: LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
2. Chun Zeng., Xiaohui Zheng., Chunxiao Xing., Lizhu Zhou : Personalized Services for Digital Library. In Proceedings of Fifth International Conference on Asian Digital Libraries, ICADL (2002) 252-253.
3. Cohen, W., Fan, W. 2000: Web-collaborative ltering: Recommending music by crawling the web. In Proceedings of WWW-2000.
4. Daniel Faensen., Lukas Faulstich., Heinz Schweppe., Annika Hinze., Alexander Steidinger. Hermes: A notification service for digital libraries. In ACM/IEEE Joint Conference on Digital Libraries, pages 373.380, 2001.
5. Drucker, H., Shahraray, B. and Gibbon, D.: Relevance feedback using support vector machines. In Proceedings of the 18th International Conference on Machine Learning, pages 122 129, 2001.

6. Elena Renda, M., Umberto Straccia : A Personalized Collaborative Digital Library Environment. In Proceedings of Fifth International Conference on Asian Digital Libraries, ICADL (2002) 262-274.
7. Harman, D. : Relevance feedback revisited, Proceedings of the Fifth International SIGIR Conference on Research and Development in Information Retrieval,1992, (pp. 1-10).
8. Jamie Callan., Alan Smeaton., : Personalization and Recommender Systems in Digital Libraries, Joint NSF-EU DELOS Working Group Report,May 2003
9. Joachims, T. : Text categorization with support vector machines: learning with features, European Conference on Machine Learning,1998, pp. 137-142.
10. Kathleen R. McKeown., Noemie Elhadad., Vasileios Hatzivassiloglou: Leveraging a Common Representation for Personalized Search and Summarization in a Medical Digital Library ,In Proceedings of the Third ACM/IEEE Joint Conference on Digital Libraries (JCDL 2003)
11. Matthias Bender., Sebastian Michel., Christian Zimmer., Gerhard Weikum: Towards Collaborative Search in Digital Libraries Using Peer-to-Peer Technology
12. McNee, S., I. Albert, D. Cosley, P. Gopalkrishnan, S.K. Lam, A.M. Rashid, J.A. Konstan, and J. Riedl: On the Recommending of Citations for Research Papers. In Proceedings of the ACM 2002 Conference on Computer Supported Cooperative Work (CSCW 2002), New Orleans, LA, 2002, pp. 116-125.
13. Nicholas J. Belkin., Gheorghe Muresan: Measuring Web Search Effectiveness: Rutgers at Interactive TREC. WWW2004 Conference Workshop
14. Norbert Fuhr., Norbert Gvert., Claus-Peter Klas: Recommendation in a Collaborative Digital Library Environment. Technical Report, University of Dortmund, Germany (2001).
15. Paul Resnick., Neophytos Iacovou., Mitesh Suchak., Peter Bergstrom., John Riedl. : GroupLens : An Open Architecture for Collaborative Filtering of Netnews. ,In Proceedings of the ACM 1994 Conference on Computer Supported Collaborative Work (CSCW .94), Chapel Hill, NC, 1994, pp. 175-186.
16. Roberto Torres., Sean M. McNee., Mara Abel., Joseph A. Konstan., John Ried: Enhancing Digital Libraries with TechLens+ In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries (JCDL 2004), June 2004, pp. 228-237
17. Rocchio, J.J. : Relevance feedback in information retrieval, The SMART Retrieval System: Experiments in Automatic Document Processing, ed: Gerald Salton, Prentice Hall,1971, 313-323.
18. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society of Information Science, 1990, 41:288 297.
19. Shardanand, U., P. Maes: Social information Filtering, Algorithms for automating "word of mouth". In Proceedings of the 1995 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 95), Denver, CO, 1995, pp. 210-217.
20. Theobald, M., Klas, C. P.: BINGO! and Daffodil: Personalized Exploration of Digital Libraries and Web Sources, In 7th International Conference on Computer-Assisted Information Retrieval,RIAO (2004)
21. Vladimir N. Vapnik: The Nature of Statistical Learning Theory. Springer, 1995.
22. Yang, Y. and Liu, X.: A Re-examination of Text Categorization Methods. In: Proceedings of the 22nd Annual International ACM Conference on Research and Development in Information Retrieval (1999) 42.49.