A Conceptual Framework of Using Collaborative Filtering Algorithms to Enhance Keyword Search

Shreyas Deokule
Department of Information Science and Technology
School of Management and Information Systems
University of Missouri-Rolla, USA
ssd436@umr.edu

Wen-Bin Yu
Department of Information Science and Technology
School of Management and Information Systems
University of Missouri-Rolla, USA
yuwen@umr.edu

Bih-Ru Lea
Department of Business Administration
School of Management and Information Systems
University of Missouri-Rolla, USA
leabi@umr.edu

Abstract

This study proposed a Collaborative Filtering (CF) algorithm that generates and recommends a list of alternate keywords based on the keywords originally set by a user when searching documents from an online search engine. The proposed CF algorithm incorporates the similarities among keywords associated with the documents and provides crucial recommendations that may not be considered by the user. The purpose of the proposed CF algorithm is to reduce the efforts in identifying appropriate keywords set to allocate the desired documents more efficiently.

The traditional search algorithms strive to provide closely-matched results to the keywords specified by the user. However, the most exceptional search engine would not provide good quality results if the original keywords selected by the user were not suitable. Therefore, the proposed system aims at suggesting a set of alternative keywords generated based on a user's original keywords to help a user in his/her subsequent search activities.

The proposed system is expected to improve search accuracy and effectiveness for a novice user in the field of interest. For experience users, with the fast-growing availability of information online, who may not be aware of most the updated/critical keywords, the proposed system is also expected to improve search efficiency. Furthermore, the proposed system is flexible and can easily be integrated with other search algorithms to improve search results.

1. Introduction

As the number of published research papers growing at a phenomenal rate every year, it becomes a challenge for a user to find the most relevant or critical information needed from the multitude of publications. It is reported that the number of research papers published each year has increased at a rate of 1% per year since 1986 [20]. National Science Foundation (NSF) found that more than half a million papers were published in more then 1,900 journals worldwide in 1999. As this trend continues, millions of papers will be published over the period of next ten years. As a result, the number of papers published can be overwhelming even for a relatively small or new research area. The problem gets compounded for the interdisciplinary research as the papers are published at different venues.

Keyword querying and matching using online research database is a common acceptable practice for researchers to find the research publications of interest. However, the probability of getting good quality search results depends on the levels of user's domain knowledge. In other words, novice researchers with limited domain expertise and critical keywords often find themselves in a less successful situation and became frustrated in the search process. In addition to the unfamiliarity of proper key words in a particular research domain, the problem can also arise from that certain words have different meanings and might be used in different research domains. Therefore, novice users who used such keywords might get overwhelming results from a different research domain than the one he/she desired and have to go through more key word refining iterations to obtain acceptable search results.

Many researchers in the field of advanced information systems have dedicated their works on developing speedy and/or suitable algorithms in delivering the right information to the users. However, even the most well-designed and –calibrated search system would not deliver the desired information should the keywords entered by the user were irrelevant or inappropriate. It is valuable to provide users with tools that recommend proper keywords during a search attempt.

The objective of this study is to propose a conceptual framework of a keyword recommendation system developed using collaborative filtering algorithms to help users obtain better search results. Collaborative filtering algorithms play an important role in constructing a successful recommender system [5] [17] [23]. Most of such systems can be found in the field of e-commerce applications [5] [23], but very limited applications were found in the research publication search. In the proposed framework, the system will accept user's keywords and return a set of alternate keywords that are generated using a collaborative filtering (CF) algorithm. It is expected that users could reduce the number of search iterations from using the alternate keywords in subsequent search attempts.

The outline of this paper is as follows. Section 2 provides a brief literature review on collaborative filtering algorithm based recommendation systems, similarity computations, and existing research as well as addresses current research gaps. The proposed collaborative filtering algorithm based keyword recommendation system is presented in Section 3. The evaluation methodology and parameters are discussed in section 4. Finally, future research directions are provided.

2. Literature Review

Applications of collaborative filtering algorithms are found mostly in e-commerce. Many e-commerce websites that have implemented a recommendation system provide users with targeted recommendation that users are most likely to accept. Some of the applications of recommendation systems in ecommerce environment are Product Hierarchy-Based Collaborative Filtering [10], Dynamic Profiler [27] or Item-to-item based Collaborative Filtering (used by amazon.com) [17]. Apart from recommending products, collaborative filtering algorithms are also used in non commercial applications. Examples of non-commercial applications include newsgroup posting (Grouplens) [15], Music albums (Ringo for music in general and RACOFI for Canadian music specifically) [25] [1], jokes (Jester) [4] and movies (Movielens) [11].

2.1 Collaborative Filtering Algorithm

Collaborative filtering algorithms are often used in domains where there are multiple users and each user has ratings for multiple items. A rating matrix in Collaborative filtering algorithms is a dataset where user ratings are available for selected items. Various algorithms use different forms of this type of rating matrix. As shown in Figure 1, a user/item rating matrix is can rate (e.g., i_1, i_2, etc.) and the rows represent the users a matrix where the columns represent the items that users themselves (i.e., u_1, u_2, etc.). Each entry (R_{ui}) in a rating matrix is a user's rating for a particular item. For example, $R_{u_z i_k}$ is the rating of user z on item k. There are various ways in which the ratings can be collected. Opinions or ratings can be given explicitly by the user or can be collected implicitly by parsing through the logs generated on the basis of user's activities online or things purchased online. Collaborative filtering aims at filling up the empty spaces in the rating matrix. The entries are calculated based on the similarity between the users. The similarity between users is calculated based on the items they rate.

The goal of collaborative filtering algorithm is to suggest new items or to predict the utility of a certain item to a particular user. In other words compute the entries in the cells of user-item matrix, which is not already filled. This is done based on user's previous likings and opinions of other like-minded users. Refer to figure 1. Let us assume a set of *m* users $U = \{u_1, u_2, u_3 ...... u_m)$ and a list of *n* items $I= \{i_1. i_2. i_3 ...... i_n\}$. Each user u_a has set of items I_{ua} that he/she has expressed their opinion ($r_{ua,i}$) about, where I_{ua} is a subset of I and can also be a null set. Ratings can be within a numeric scale. We consider an active user u_a for whom the system generates the recommendations. The process can be divided into two parts.

1. Prediction is a numeric value P_{i_j, u_a} , expressing the predicted likeliness of item i_j, not a part of I_{ua}, for active user u_a.
2. Recommendation is list of N items that active user will like the most. This is known as Top-N Recommendation. This should not include the items that the user has rated or bought.

The similarity computations methods, explained in the next sub-section, are used to compute the similarities between the users. Thus each user has a pool of similar users. The items, which the pool of similar users has rated high and the present user has not rated, are then recommended to the user.

2.2 Similarity Measure

Similarity computation between two items or users is one of the most important steps in Collaborative Filtering.

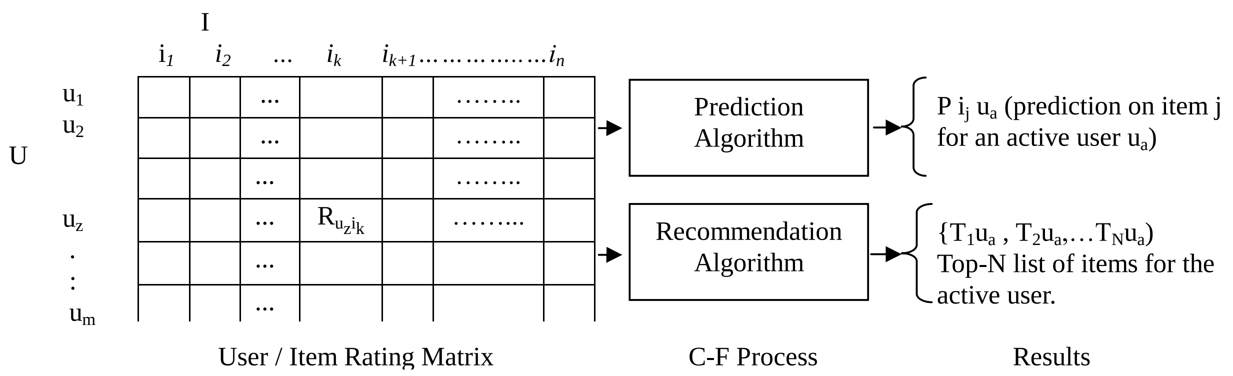


Figure 1: Collaborative Filtering Process

Similarity between two users or items can be calculated using various methods. The most common method is the Cosine-based Similarity computation.

2.2.1 Cosine- based Similarity:

This method calculates the similarity between the two users as the cosine function of their vectors. Similarity between user a and b can be formulated as follow:

$$S_{ab} = cos\ (R_{a,\ I},\ R_{b\ I}) = R_{a,I}\ .\ R_{b,I}\ /\ |\ R_{a,I}\ |\ *\ |\ R_{b,\ I}\ | \qquad (1)$$

where $R_{a,I}$ represents the rating vector for user among all the items (set I).

Cosine-based similarity is one of the simplest methods of evaluating similarity between two vectors. But users never rate the items uniformly. Some users are more liberal in rating items, whereas some users are ungenerous in giving rating. Cosine similarity assumes that the users have rated the items uniformly.

2.2.2. Adjusted Cosine Similarity:

This is a variation of Cosine-based similarity. Users rate the items differently. There are no specific guidelines to rate items. Thus two users equally liking an item might give different rating to it. Also a single user might not be consistent in his/her ratings on different items. The Adjusted Cosine Similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair. The similarity is calculated as:

$$S_{ab} = \frac{\sum_{i \in I}(R_{a,i} - \overline{R_a})(R_{b,i} - \overline{R_b})}{\sqrt{\sum_{i \in I}(R_{a,i} - \overline{R_a})^2}\sqrt{\sum_{i \in I}(R_{b,i} - \overline{R_b})^2}} \qquad (2)$$

where $\overline{R_a}$ is the average of user a’s rating.

2.3 Various types of Collaborative Filtering Algorithms

Some algorithms change the structure of the matrix. The change in structure of matrix gives various forms of the Collaborative Filtering algorithm. The two major types of collaborative filtering algorithms are user-based and item based the Collaborative Filtering algorithms, as described below.

2.3.1 User-based Collaborative Filtering Algorithms

These are the more traditional Collaborative Filtering algorithms. The matrix structure described earlier is used. Similarity between the users is calculated. The items which the similar users have highly rated and the target user hasn’t purchased are recommended to him/her. One of the implementation of User-based Collaborative Filtering is the GroupLens experiment [22]. The CF algorithm is used to help users find articles from the vast pool of articles on net news.

2.3.2 Item-based Collaborative Filtering Algorithms

These algorithms use the complete opposite approach. They are more aimed at finding the similarity between the items rather than users. The similarity between the various items is calculated. Items most similar to the items purchased by the target users are recommended to the user. Sarwar and colleagues [24] have talked implementation of Item-based Collaborative Filtering. Items that are most similar to the items purchased or highly rated by the target user are then recommended to the user. Amazon.com [17] also uses Item-to-item Collaborative Filtering algorithm to recommend items to their patrons.

2.4 Limitations of collaborative filtering algorithms in e-commerce applications

Despite its successful application in various fields, Collaborative filtering algorithms have some inherent drawbacks. The biggest drawback of collaborative filtering is the first rater problem. The item has to be rated by at least one user before it can be considered for recommendation. Also if the item is not recommended, it is less likely to be rated by the user. This can leave out several items without being recommended.

In many domains, there are lot more items then users. Users generally rate a very small percentage of all the items in the domain. This might leave a very small or no overlap between the various users. It gets difficult to find the neighborhood of like minded people. The similarity is very low or almost zero.

While using collaborative filtering algorithms, the designers need to take into consideration the drawbacks. When it comes to recommendation of research papers, researchers use various existing links between the papers and its authors to over come the first rater and sparsity problem.

2.5. Research publication search

Quite a large number of the Internet users today are interested in finding text documents, research material available online. At present most users depend on sites which store information about research texts. Examples of such websites would be ACM Portal [8] or IEEE Xplore [7] or http://www.scholar.google.com/ (still in its beta version) [12]. Such sites collect information which is published by almost all renowned publishers and store them. Users can browse and find interested documents, using the simple text search engine deployed by the site. These sites also use a ranking algorithm which sorts the results in order of relevance to the user’s search. This helps users to get the best results to their query at the beginning. Most sites do provide more help with the detailed information about the authors, abstract of the document, keywords, and references. Few sites give additional information, like ACM portal [8], gives the list of other authors who have collaborated (co-authored some papers) in the past with authors of the document. This information proves useful to understand the expertise of the author and also to find related work from other authors. Google Scholar [13] is designed specifically to help users search scholarly literature, including peer-reviewed papers, theses, books, preprints, abstracts and technical reports from all broad areas of research. Google Scholar searches articles from a wide variety of academic publishers, professional societies, preprint repositories and universities, as well as scholarly articles available across the web. The results thus obtained are then sorted by using the PageRank algorithm [21]. These sorting algorithms sort the results in such way that the most relevant results are displayed first.

There are not many personalization techniques used to aid the users in their search. There have been successful attempts to use the Collaborative Filtering techniques to help users search research material. McNee and colleagues [18] proposed using the citations, used in research paper, to develop relationship between research papers. Each research paper references previously published papers, which can be used to link the papers. There is a rich citation web formed between research papers. The citations between research papers form a graph that can be viewed as a social network known as a *citation web*. For any given paper, it is possible to follow the citation web to see what papers cite it and what papers are cited by it [18]. Using the references found in research papers, it is possible to create citation webs that reflect professional social networks between papers. [18]. Sean McNee and colleagues consider citation as a reference to a paper and paper as a citation for which the entire text is available, including it's complete citation list. Thus if a paper references a certain a paper, the dataset might not have the paper but must have the citation. The citation web thus created is used to find papers similar to any given paper, based on the citations. The main aim here was to find the papers similar to the once the user has likes. Hence if the user reads or accepts a certain paper, the system can recommend similar papers. In short this can be considered analogous to an ecommerce application. We had various users rating various items in ecommerce scenario. In research papers, we have various papers referencing (rating) papers. All three different types of collaborative filtering algorithms were implemented and evaluated.

Torres and colleagues [26] carried forward this approach and created a hybrid algorithm which combined Collaborative filtering with Content Based Filtering. Content Based Filtering is also an algorithm used in recommendation system. Content Based Filtering is generally applied in textual domain. Here a user profile is maintained, which consists of papers the user has refereed or rated highly in the past. Text documents are recommended based on a comparison between their content and a user profile [2]. Content Based Filtering has its own drawbacks [2] [26]. Use of hybrid algorithm eliminated the drawbacks of both the algorithms, if used individually. Collaborative filtering and content based filtering were implemented in a two separate recommender modules. These modules worked independently to generate their results. The systems were applied in sequence one after the other. Different content based and collaborative filtering algorithms were used in each individual recommendation module. Based on the combinations of order in which the modules were used, six recommendation systems were suggested. This system was called as Techlens [26]. They evaluated all forms of the system, based on an offline and online experiments.

3. Proposed System

3.1 Recommending keywords to users

In previous section we reviewed multiple tools which are used to help users search online information. All the tools work to give best results for the user query. But the users are still required to browse through a lot of pages to get all the information. The system works better if the user has definite idea about what kind of papers he/she is looking for. The entire process becomes all the more difficult if the user is not exactly aware of what is required. This is especially true for interdisciplinary field. How can users search for something that they are completely unaware or unheard of?

It can be seen that the present available sites for searching text information uses traditional information retrieval methods, like keyword search. The results thus obtained are then sorted by using various algorithms, the PageRank algorithm [21]. These sorting algorithms sort the results in such way that the most relevant results are displayed first. But here the keyword is very important, as without appropriate keyword the user can not find appropriate results. It doesn't help user if the undesired papers are sorted in order of relevance to the inappropriate keyword. For instance if the keyword is very generic, it might return a lot of results, and user have to search within those results. The recommendation system can recommend papers according to the degree of similarity with user's interest. But if the keywords that are searched for are fuzzy the recommendation system can still face problems in coming up with recommendations. Thus it can be seen that for any search or recommender system to give best results requires as precise keywords as possible.

Users face problems in framing appropriate queries due to various reasons. Users might be completely new to the field, or might not be aware of latest updates in the field. Many keywords mean different things in different domain. In most cases the users try very precise or very vague search keywords. This returns a lot of hits or hardly any hits. Generally when users try to search for information in unknown domains, they start with some generic keywords. The results of early premature searches, gives users a better idea about the domain and the users restructure their query. This is an iterative process which eventually leads the user to the information required. This process can be completed in short amount of time, say minutes or might take longer time, even days. Many a times the users give up after a couple of iterations accepting the results that are generated.

Thus an application which can recommend alternate keywords can go far way in help users search for information. If the users are given recommendations on alternative queries, it might help the users in getting the required information faster. If a system is able to compute similarity between the keywords user has inputted to the other keywords, then most similar keywords to user's keyword can be recommended to the users. For example a user needs to know more about ERP software, but is new to the domain. The user starts with search for SAP, an ERP software, as he/she is aware of it. This is a specific case of ERP and user would like to know about other ERP software too. If the system can compute that ERP and SAP are related, it can recommend ERP as an alternative keyword.

3.2 Generating alternate keywords

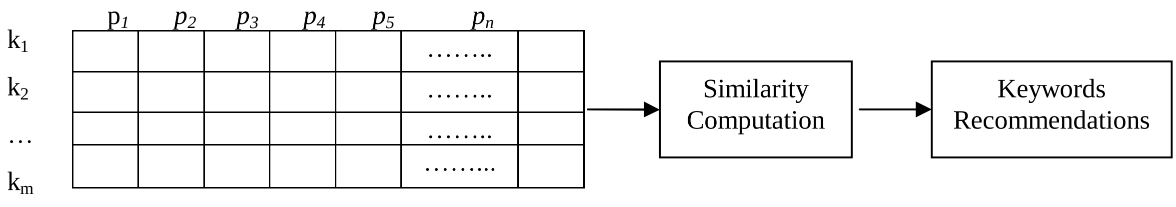
To generate recommendations for keywords, a Collaborative Filtering algorithm is proposed. Each and every research document would have keywords to describe the paper. Most of the journals or conferences proceedings have list of keywords mentioned in each paper. Where keywords are not explicitly mentioned, they can be generated using text-mining methods. The discussion on such methods is beyond the scope of this paper. The results generated by query submitted by the user can be used to compute similar queries. Each paper in the result will add its keywords to a pool of keywords. We can then compute the similarity between the user's keywords to the other keywords in the pool. The keywords most similar to the keywords used by the user will be recommended to the user. Thus the user can go with the results already generated or use the new recommended keywords and start a new search.
This process will continue till the user gets satisfactory results. This can help user to reach the required data in minimum time and iterations. The recommendation system here tries to mimic the actual search procedure followed by the users. In normal searches users refine their queries based on the results of earlier searches. The system would recommend the users with keywords based on the search result of present query. In our domain of recommending keywords, we have multiple papers each having multiple keywords associated with it. Consider *n* number of papers represented by a set $P = \{p_1, p_2, p_3....p_n\}$. Each paper would have *m* keywords associated with it, represented by set $K = \{k_1, k_2, k_3....k_m\}$. Thus we get a new dataset of papers and keywords for every user search. This can be represented as a Keyword/ Paper matrix form.
Construction of a similar matrix was described earlier. The matrix would look like in Figure 2.

Thus we can construct vectors for each keyword. Next step would be to compute similarity for each keyword with the user's keyword. Top of the most similar keywords can be then recommended to the users. This can be considered analogous to the Rating matrix in an ecommerce environment. We have papers here instead of users and keywords instead of items purchased. Thus it forms a classic case for use of collaborative filtering methods. The major drawbacks of Collaborative Filtering algorithms are eliminated here. We intend to develop applications which are based on item-based Collaborative filtering algorithms. As the system has some basic results or keywords to work on, the first rater problem doesn't arise. The sparsity problem can't be eliminated completely, but it won't arise in most cases.

4. Evaluation Methodologies

There can be various algorithms used in generating recommendations for keywords. Each algorithm will have some advantages and disadvantages. All such collaborative filtering algorithms need to be evaluated to work in the domain of recommending keywords. The algorithms which perform the best on most of the parameters can be considered most ideal for the domain. Herlocker and colleagues [5] have devised a strategy for evaluation Collaborative filtering algorithms. The algorithms can be tested using those guidelines. Any new Collaborative Filtering algorithm can be evaluated using following steps:

1. Defining Tasks for the Collaborative filtering: All the tasks that the algorithm needs to perform are to be defined precisely.
2. Finding appropriate Datasets for evaluation of algorithms: Collaborative filtering algorithms are tested on test datasets. Not all the datasets can be considered ideal for evaluating algorithms in every domain. Some Datasets are best suited for certain



Keyword / Paper Rating Matrix

Figure 2: Process of Recommending Keywords

domains and would not be best for use in other domains.

3. Evaluation parameters for accuracy: Accuracy is the most important characteristic for a collaborative filtering algorithm. Thus it is required to define certain parameters that can calculate the accuracy of the recommendations in a particular domain.
4. Evaluating other parameters: There are various other parameters that are important for the domain. The algorithms need to be evaluated based on these domain parameters, other then accuracy.

Next subsections discuss about how the above evaluation model can be implemented on the domain for recommending keywords.

4.1 Defining tasks for the collaborative filtering algorithms within the domain

The algorithm needs to perform various tasks in order to efficient help users find most appropriate keywords

1. The algorithm needs to compute all the related keywords. Here the more important task is not to skip any of the similar keywords. Most of the algorithms are tested to see if they get most similar items. We need to focus on not missing any similar keyword.
2. Give at least some minimum number of alternate keywords to the user. Also help users to construct next query by allowing them to logically connect the keywords.
3. The algorithm must give fast results. The task of recommending keyword is a supportive task to the main task of searching. Users can get impatient if the results take time to be displayed. Thus the algorithm should be as computationally simple and as fast as possible.
4. Users should have option of turning off the keyword recommender. It might happen that the users would be well aware of what they need and won't require any recommendations.
5. It should work in compliment with the actual search and sort algorithm. It is possible that some other recommender algorithm is also used in tandem with the search algorithm. The keyword recommender should be able to work in compliment with these algorithms

4.2 Finding datasets for evaluating the algorithms

The ideal dataset would be some existing repository of research material. ACM portal [8] or IEEE Xplore [7] can be most ideal datasets. But access to these repositories is restricted to only members. Also the datasets are not downloadable. Citeseer dataset [6] is a similar dataset and freely available to users. It is also possible to download the complete Citeseer dataset.

Working on these extensive datasets can get overwhelming for initial evaluation. Traditional CF algorithms testing dataset such as MovieLens [11] may be of use. This dataset stores ratings given by users to movie titles. Each user has rated minimum of fifteen movie titles. This can considered similar to a single paper having multiple keywords. Hence for initial evaluation MovieLens dataset seems to be most appropriate. We have used this to come up with some preliminary results, not discussed here.

4.3 Defining parameters to measure the accuracy of the algorithm

We need to evaluate the performance of the algorithm for the accuracy of its recommendation. This will require user evaluation of the system. In user evaluation accuracy parameter can be gauged by knowing how often the users accept the recommendation. Another interesting parameter would be to measure the number of times the user doesn't accept the recommendation and enters a keyword which is not recommended. It will be further interesting to measure how many times the user enters a new keyword, not recommended but was one of the keywords in the keywords pool from which the recommendations were generated. This is to see if the recommender system is not skipping any keywords.

4.4 Defining parameters other then accuracy

The most important aspect to measure, apart from the accuracy, would be the time required for the algorithm to compute the recommendations. As explained earlier the algorithm needs to be computationally simple and faster to implement. Apart from that we also need to measure the number of iterations required by users to reach the desired results. This can be compared with the number of iteration required by the user in a non recommender environment. Another parameter to measure would be the time taken by user for each of the iterations in recommender environment as opposed to non recommender environment. This will help us know if the system is helping the users in their task of searching.

5. Future Scope

The algorithm suggested here is supposed to work on a well structured dataset. The keywords are mentioned explicitly or can be generated. The future application of such algorithms would be to work in unstructured domain like Internet. It can help user generate queries for almost all the data available online. The major tasks here would be to identify the keywords and computing the results in real time.

6 References

[1] Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D. & McGrath, S. *"A Rule-Applying Collaborative Filtering System"*. Proceedings of IEEE/WIC COLA 2003, Canada,

[2] Balabanovic, M. & Shoham, Y. *"Content Based, Collaborative Recommendation"*. Communications of the ACM, 1997, 40(3), 68-72.
[3] Fisher, D., Hildrum, K., Hong, J., Newman, M., Thomas, M. & Vuduc, R. *"SWAMI: A Framework for Collaborative Filtering Algorithm Development and Evaluation"*. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, 366 – 368.
[4] Goldberg, K. *"Overview of the jester system (invited talk)"*. eigentaste.berkeley.edu. November 1999.
[5] HerLocker, J. L., Konstan, J. A., Terveen, L. V. & Riedl, J. T. *"Evaluation Collaborative Filtering Recommender Systems"*. ACM Transactions on Information Systems, 2004, 22(1), 5-53.
[6] http://citeseer.ist.psu.edu : CiteSeer.IST: Scientific Literature Digital Library.
[7] http://ieeexplore.ieee.org/Xplore/DynWel.jsp IEEE Xplore Release 1.8.
[8] http://portal.acm.org/portal.cfm : ACM Portal containing "The ACM Digital Library" and "The Guide to Computing Literature". It helps to search articles and papers published by ACM or Bibliographic collection from major publishers.
[9] http://www.cs.umn.edu/Research/GroupLens/ : Home page for GroupLens Research Project.
[10] http://www.ieor.berkeley.edu/~goldberg/jester-data/ : Anonymous Ratings Data from the Jester Online Joke Recommender System.
[11] http://www.movielens.org/ : A movie recommender site, formed by GroupLens research team.
[12] http://www.scholar.google.com/ : Beta version of Google search engine to help find research related material.
[13] http://www.scholar.google.com/scholar/about.html : Information and FAQs about Google's scholar search engine.
[14] Kleinberg, J. M. *"Authoritative Sources in a Hyperlinked Environment"*. Journal of the ACM, 1999, 46(5), 604 – 632.
[15] Konstan, J. A., Miller, B., Maltz, D., HerLocker, J., Gordon, L. & Riedl, J. *"GroupLens: Applying Collaborative filtering to Usenet news"*. Communications of the ACM, 1997, 40(3), 77 - 87.
[16] Lawrence, S., Giles, C. L. & Bollacker, K. *"Digital Libraries and Autonomous Citation Indexing"*. IEEE Computer, 1999, 32(6), 67 – 71.
[17] Linden, G., Smith B. & York, J. *"Amazon.com Recommendations: Item to Item Collaborative Filtering*". IEEE Internet Computing, 2003, 7(1), 76 - 80.
[18] McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A. & Riedl, J. *"On the Recommending of Citations for Research Papers"*. Proceedings of the 2002 ACM conference on Computer supported cooperative work, November 2002, 116 - 125.
[19] Niu, L., Yan, X., Zhang, C. & Zhang, S. *"Product Hierarchy- Based Customer Profiles for E-commerce Recommendations"*. Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference, Volume 2, 4-5 Nov. 2002, 1075 - 1080
[20] NSF, *N.S.F. Academic Research and Development*. 1999.
[21] Page, L., Brin, S., Motwani, R. & Winograd, T. *"The PageRank Citation Ranking: Bringing Order to the Web"*. Stanford Digital Library Technologies Project. January 29, 1998
[22] Resnick P., Iacovou N., Suchak M., Bergstrom P. & Reidl J. *"GroupLens: An open architecture for Collaborative Filtering to Usenet news"*. Proceedings of the 1994 ACM conference on Computer supported cooperative work, 1994, 175 - 186.
[23] Sarwar, B. Karypis, G., Konston, J. & Reidl, J. *"Analysis of Recommendation Algorithms for E-Commerce"*. Proceedings of the 2nd ACM conference on Electronic commerce, 2000, 158 - 167.
[24] Sarwar, B. Karypis, G., Konston, J. & Reidl, J. *"Item Based Collaborative Filtering Recommendation Algorithms"*. Proceedings of the tenth international conference on World Wide Web, May 2001, 285 – 295.
[25] Shardanand U. & Maes, P. "Social Information Filtering: Algorithms for Automating "Word of Mouth"". Proceedings of the SIGCHI conference on Human factors in computing systems, 1995, 210 – 217.
[26] Torres, R., McNee, S. M., Abel, M., Konstan, J. & Riedl, J. *"Enhancing Digital Libraries with TechLens"*. Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, 2004, 228 – 236.
[27] Wu, K., Aggarwal, C. C. & Yu, P. S. "Personalization with Dynamic Profiler". Advanced Issues of E-Commerce and Web-Based Information Systems, WECWIS 2001, Third International Workshop on. 21-22 June 2001, 12 - 20.