PROCEEDINGS PAPER

A Personalization-Oriented Academic Literature Recommendation Method

Zhongya Wang[1], Ying Liu[1,2], Jiajun Yang[1], Zheng Zheng[3] and Kaichao Wu[3]

[1] School of Computer and Control, University of Chinese Academy of Sciences, Beijing, China

[2] Fictitious Economy and Data Science Research Center, Chinese Academy of Sciences, Beijing, China
yingliu@ucas.ac.cn

[3] Computer and Network Information Center, Chinese Academy of Sciences, Beijing, China

As the number of digital academic items increases dramatically, it is more and more difficult for a student or researcher to find the expected references in a large academic literature database. Although collaborative filtering and content-based recommendation approaches perform well in some applications, they do not produce satisfactory recommendations for academic items because they fail to reflect researchers' unique characteristics in terms of authority, popularity, recentness, etc. In this paper, we propose two novel data structures, ALVector, which expresses various objective attributes of an article, and AUVector, which expresses users' subjective weights for different attributes. Then, we propose a novel personalization-oriented recommendation method that utilizes both the content and non-content attributes in ALVector and AUVector for making recommendations. In order to make the overall best recommendation, the VIKOR algorithm is used with a personalization-oriented method to achieve a compromise solution. A real-world literature data set is used in the experiments. The experimental results show that our method better meets the user's preference in multiple dimensions simultaneously.

1 Introduction

As the amount of digital academic literature, such as conference proceedings, journals, books, magazines, etc., increases dramatically, it has become a time-consuming task for a student or researcher to discover a needed reference. Thus, recommendation systems that can predict items (or rate items) that the user may have an interest in have become a significant research area. Collaborative filtering techniques were proposed in the mid-1990s (Hill, Stead, Rosenstein, & Firnas, 1995). Extensive work has been conducted in business recommendation systems as well as academia over the past two decades (Adomavicius & Tuzhilin, 2005). Generally speaking, these approaches can be classified into two categories, content-based filtering and collaborative filtering.

As all researchers have their own specific research area, research experience, and background knowledge, their research interests are so unique that they cannot be inferred from another researcher's interest. Thus, collaborative filtering approaches that estimate a user's preferred item from an item preferred by users who share similar interests do not perform well in academic literature recommendation. As a researcher's studies progress, his/her preference in academic resources may change in terms of authority, popularity, recentness, etc. However, the content-based approaches, which recommend by content or topic similarity, lack the capability to consider a user's non-content preference and therefore are also not satisfactory in academic literature recommendation. Thus, a recommendation method that allows a user to express both content interest and personalized preference is in demand.

In this paper, we propose two data structures, the *Academic Literature Vector* (*ALVector*) and the *Academic User Vector* (*AUVector*). ALVector is a data structure consisting of two entries (c, e), where c is the content vector of an object and e is an attribute vector composed of a set of attributes of the object. For an academic

article, the first entry of ALVector could be a TF-IDF vector that denotes the major content of the article; the second entry records various attributes of the given article, such as recentness, popularity, and authority, etc. Values in such an ALVector can be extracted from the database in which they are stored. The AUVector expresses a user's personalized preference in research in terms of content similarity, recentness, popularity, authority, etc. Values in each AUVector can be obtained by analyzing the user's web log or cookies where the browsing history provides hints of personalized preference.

In contrast to the traditional content-based approaches that only consider the similarity in content between an article and a user, in this paper, we propose a personalization oriented recommendation method. It considers not only the content similarity but also the non-content attributes included in ALVector and AUVector. ALVector and AUVector give us the chance to match items with users from multiple dimensions in a quantitative manner. The goal of our recommendation method is to maximize the user's expectation at each dimension. However, theoretically, it is infeasible to reach the optimal solution at every dimension (Shi, 2001) simultaneously. Compromise solutions have to be considered as the global optimal in real applications. In order to find the best academic literature in the database that maximize each user's overall satisfaction, we adopted VIKOR, a multi-criterion decision making algorithm (Opricovic & Tzeng, 2004). It ranks a set of alternatives and eventually determines a compromise solution with conflicting criteria that is the closest to the ideal.

We performed experiments on a set of real academic articles collected from the largest publishing platform in China. The experimental results demonstrated that our personalization-oriented recommendation method is able to consider users' personalized preferences comprehensively while the content-based approach ignores utilities other than content similarity.

The rest of this paper is organized as follows. Section 2 gives an overview of related work in recommendation systems and academic literature recommendation methods. Section 3 presents our proposed data structures, ALVector and AUVector. The VIKOR-based recommendation method is presented in Section 4. Section 5 presents the experimental results and analysis. We conclude the paper in Section 6.

2 Related Work

The core task of recommender systems is to predict a user's preference and suggest items by analyzing his or her history. Content-based recommendation has been widely used in a variety of applications, such as recommending web pages, new articles, restaurants, books, etc. Although a number of variations have been proposed, all content-based approaches share a common methodology that recommends items to a user by comparing the text description of an item with the user's profile and determining to what extent they are "similar" (Balabanovic & Shoham, 1997).

Another popular method is collaborative filtering, which recommends items to a user based on the opinions of other like-minded users. User opinions can be obtained explicitly from users or by some implicit measures (Arwar, Karypls, & Konstan, 2001). Also, association rules based recommendation is popular in academic literature recommendation systems. This incorporates a data mining process together with user ratings in making recommendations. Beginning with the real data history, it discovers significant rules that associate academic resources clicked or downloaded by previous users. These rules are later used to infer recommendations (Lin, Alvarez, & Ruiz, 2000). Hybrid algorithms combine the strengths of each filtering approach to address their individual weaknesses (Torres, McNee, Abel, et al., 2004). Nevertheless, none of the above research considers both the similarity in content and the personal characteristics of the users.

In the domain of academic recommendation, citation analysis has a long history in assessing the research performance of an individual scholar or the quality of a publication (McNee, Albert, Cosley, et al., 2002). Citation analysis tends to recommend academic papers that have been cited many times. Some researchers applied the PageRank algorithm, which was most famously used in information retrieval, in improving the ranking of academic search results (Sayyadi & Getoor, 2009) and in measuring the importance of scholarly papers by analyzing the PageRank from a citation network of scholar papers (Chen, Xie, Maslov, et al., 2007). Another approach investigated recommendation satisfaction according to two researchers' levels, masters or PhD student level and professor level. However, this recommendation system modeled each researcher's interest using only one paper that the researcher manually chose (Torres, McNee, Abel, et al., 2004). Tag and social network techniques have been introduced into recommender system research, but these new techniques still pay little attention to a user's customized preferences or interests (Andersen, Borgs, Chayes, et al., 2008). A utility based recommendation approach has been proposed, which adopts a two phase data mining process to find high utility item sets, but it does not take into account the multi-utility ranking problem (Liang, Liu, Jian, et al., 2011).

3 Data Structure
3.1 Academic Literature Vector
In order to express a user's interest in content and personalization, we propose a novel data structure, called the *Academic Literature Vector* (*ALVector*). It consists of two entries (*c, e*), where *c* is the content vector of an article and *e* is the attribute vector composed of a set of non-content attributes of the article.

3.1.1 Content Vector
The content vector of ALVector presents the major content of an article, usually by a set of keywords and their corresponding significance. The TF-IDF measure is the most widely used space vector model in the domain of information retrieval (Salton & Buckley, 1988).

3.1.2 Attribute Vector
The attribute vector of the ALVector records various non-content attributes of an article, such as recentness, popularity, authority, etc.

1. Authority
In terms of academic literature, authority means its technical quality, which can be obtained in many ways. For example, publishing venue is a good indicator for the technical quality of an article. **Table 1** is an example of how to quantify the authority of a given academic article. In **Table 1**, all the publishing venues are divided into five levels, ranking from 4 to 0, the higher the better in technical quality. The corresponding ranking criteria are shown in **Table 1**:

Criteria	rank of the publishing venue
Top journals and top conferences, like Cell Research, Nature, Science, ACM Transactions, IEEE Transactions, etc.	4
International journals and conferences (SCI or EI indexed), like Journal of Semiconductors, Advances In Psychological Science, books, etc.	3
Other journals and conferences	2
Magazines and articles	1
Others	0

Table 1: Levels of publishing venue.

Number of non-self citations is another good indicator for authority. Number of citations can be obtained from some academic indexing systems, such as Science Citation Index (SCI), Engineering Village (EI), etc. The larger the number of citations, the higher the authority (probably). Author or author's affiliation can also be used as an indication of authority. However, in order to rank the authority of an academic article by author or author's affiliation, an evaluation and ranking system of researchers and universities must be available.

Since multiple indicators can be used to measure authority, the overall authority value is defined as the weighted sum of the authorities from different indicators. Note that we define the value of authority to be a number ranging between 0 and 1. Normalization will be performed when necessary.

2. Popularity
Popularity is a non-content attribute indicating to what extent an article is welcomed by peers. Number of downloads or length of browsing time can be used as the popularity indicator. The overall popularity value can be defined as the weighted sum of multiple indicators. Note that we define the value of popularity to be a number ranging between 0 and 1. Normalization will be performed when necessary.

3. Recentness
Recentness is a non-content attribute indicating how recent the article is. The publication date indicates its recentness. For example, the number of months elapsed since the publishing date can be used as the numerical representation of its recentness. Again, we define the value of recentness to be a number ranging between 0 and 1. Min-max normalization is adopted as shown in Eq. (1):

$$Recentness = \frac{12*(year - \min Year) + (month - \min Month)}{12*(\max Year - \min Year) + (\max Month - \min Month)} \tag{1}$$

4. Other attributes

Many other non-content attributes can be used to express the characteristics of an article, such as nationality, language, article types, publishing venue types, etc. All the information can be extracted from the literature database.

In the ALVector, no matter the content vector or the attribute vector, the values are objective values extracted from the academic literature database. Let us use an example to show how to generate the attribute vector of an article. '*Fast Algorithms for Mining Association Rules',* published in the proceedings of the 20th International Conference on Very Large DataBases (VLDB) in September 1994, has been cited by 15880 research papers. As VLDB is the top conference in the database domain, the authority is set at 4; if the largest number of citations in the database is 20000, its popularity will be set at 15880/2000 = 0.794; if the database spans literature from January 1990 to the present, its recentness will be set at $12 * (1994-1990) + (9-1)/12 * (2014-1990) + (7-1) = 0.190$. Thus, the attribute vector of this example is (4, 0.794, 0.190) if we consider only three dimensions.

3.2 Academic User Vector

The *Academic User Vector* (*AUVector*) is proposed to express different aspects of a user's personalized preference. Each element in the vector is a numerical value indicating how important a given attribute is in the user's academic selection, in other words, the weight the user implicitly has assigned to a given factor. The total sum of the weights is 1.

1) Similarity_w

Similarity_w in the AUVector indicates to what extent the user cares about the similarity between the recommended literature and his/her expectation in terms of text. Different users may put different weights on simiarity when looking for references in an academic database.

2) Authority_w

Authority_w in the AUVector indicates to what extent the user is interested in an authoritive article. Senior researchers or professors tend to be more interested in authoritive articles as they tend to require solid background knowledge to be understood. Students or beginners tend to prefer literature that is easier to understand. Therefore, different users may put different weights on authority when looking for references.

3) Popularity_w

Popularity_w in the AUVector indicates to what extent the user is interested in a popular article. Some users are likely to read articles that have been read by many users. Thus, similar to authority, different users may put different weights on popularity when looking for references.

4) Recentness_w

Recentness_w in the AUVector indicates to what extent the user is interested in recent research results. Senior researchers or professors tend to be more interested in more recent articles while students or beginners tend to read more classic articles. Thus, different users may put different weights on recentness.

5) Other attributes

Other non-content attributes, such as nationality, language, article types, publishing venue types, etc., can also be identified from the user's browsing history.

Assume that four attributes are available. An example AUVector (0.5, 0.1, 0.025, 0.375) denotes that the user is serious about similarities in content text and interested in more recent materials but does not care a lot about their authority or popularity. Such values can be derived from the user's literature history. For example, assume that user u downloaded 10 articles from an academic literature database last year, half published in 1994 and the other half published in 2004. According to Eq. (1), the recentness of year 1994 is 0.167 and 0.583 for year 2004. Thus, the recentness_w of u can be estimated as $(0.167 * 5 + 0.583 * 5)/10 = 0.375$. Also, the user's AUVector can be generated and updated using the Rocchio algorithm (Rocchio, 1971) or the Widrow-Hoff algorithm (Widrow & Hoff, 1960) after obtaining the user's rating data. Alternatively, the user can manually set the attributes. In the case of first-time users, as no historial behaviour is available for mining or analyzing, they must set the weights manually; otherwise, a random vector has to be created.

4 Personalization-Oriented Recommendation Method

4.1 VIKOR Method

VIKOR (Opricovic & Tzeng, 2004) was proposed to solve multiple criteria decision making problems (MCDM) that have conflicting and noncommensurable criteria. Assume that compromise is acceptable for conflict resolution, the decision maker wants a solution closest to the ideal, and the alternatives are evaluated according to all established criteria. This method ranks and selects from a set of alternatives in the presence of conflicting criteria and proposes a compromise solution. Eventually, it determines a compromise solution that could be accepted by the decision makers because it provides a maximum group utility for the 'majority' and a minimum of individual regret for the 'opponent'.

Assume that there are m academic items to be evaluated and each item has n attributes. VIKOR selects the items in the following steps:

1) Determine the best and the worst values of all criterion functions.

$$f_j^* = \max(f_{ij}); f_j^- = \min(f_{ij}) \tag{2}$$

where f_j^* is the best value of the jth attribute among the m items and f_j^- is the worst. f_{ij} is item i's jth attribute. Then, the positive ideal solution of the recommendation would be $(f_1^*, f_2^*, f_3^*, f_4^*)$, and the negative ideal solution would be $(f_1^-, f_2^-, f_3^-, f_4^-)$.

2) Compute S_i and R_i of item i by Eq. (3) and Eq. (4):

$$S_i = \sum_{j=1}^{n} w_j (f_j^* - f_{ij}) / (f_j^* - f_j^-), j = 1, 2.., n; \tag{3}$$

$$R_i = \max (w_j (f_j^* - f_{ij}) / (f_j^* - f_j^-)), j = 1, 2.., n; \tag{4}$$

where w_j is the weight of the jth attribute. It could be assigned by a user to express his/her preference in attribute j.

3) Compute Q_i of item i by Eq. (5):

$$Q_i = V^* (S_i - S^*) I (S^- - S^*) + (1 - V) * (R_i - R^*) I (R - R^*) \tag{5}$$

where $S^* = \min(S_i)$, $S^- = \max(S_i)$, $R^* = \min(R_i)$, $R^- = \max(R_i)$. v is a user-predefined parameter indicating the weight of the strategy 'the maximum group utility'; usually v is set at 0.5.

4) Sort all the items by Q_i. The lower the value of Q_i, the better the item. Thus, the top k items in the ranking list are the best compromise results, which best satisfy the user's comprehensive requirements.

4.2 Personalization-Oriented Recommendation Method Using VIKOR

Our personalization-oriented recommendation method can be divided into two steps. In step 1, find the top t items similar to the user's profile in terms of content; in step 2, calculate the Q value for each item and then select the top k $(t \geq k)$ to recommend. Step 1 gives priority to content because content similarity is the most significant factor in any recommendation system. Step 2 adjusts the order of the recommendations by utilizing the non-content attributes. The goal is to find a set of recommendations that best fit the preference of a given user. Assume that the ALVectors of m academic pieces of literature are available as well as the AUVector of user u and his profile. Each content vector consists of n keywords. The personalization-oriented recommendation method can be summarized in the following steps:

1) Calculate content similarity s_i between user u and each article r_i $(i = 1, \ldots, m)$ in terms of content as in Eq. (6):

$$similarity(r_i, u) = \frac{\sum_{j=1}^{n} r_{i,j} * u_j}{\sqrt{\sum_{j=1}^{n} r_{i,j}^2} * \sqrt{\sum_{j=1}^{n} u_j}} \tag{6}$$

where r_i denotes article i's content vector and u denotes the user's content vector in his profile.

2) Sort $s_i (i = 1, \ldots, m)$ in descending order and select the top t articles. t is a user pre-defined parameter, $t \geq k$.
3) For the selected t articles, calculate their VIKOR value Q_i, respectively. The attribute values in u's AUVector are used as the weights in Eq. (3).
4) Sort Q_i in ascending order.
5) Recommend top k articles to u as the final recommendation list.

5 Experiments

In order to evaluate the effectiveness of the proposed personalization-oriented method, theoretically speaking, we should use a benchmark that contains the content description of each paper, user ratings for each paper, and user personality. For each user, we could compare our recommendation list with their ratings of the papers. However, as far as our knowledge goes, no such benchmark is yet available. Due to the nature of recommendation systems, the quality of the recommended items is subjective. Therefore, we decided to conduct our experiments by making a survey. Some students and researchers were invited to use our system, where they registered into the system and rated the papers recommended to them.

5.1 Data Set and Preparation

We collected 9250 Chinese papers in our database from www.cnki.net, which is the largest digital publishing platform in China. Cnki collects all the academic papers published in Chinese journals, conference proceedings, PhD dissertations, MS theses, etc. The attributes of each item include: index, URL, title, name(s) of the author(s), affiliation(s) of the author(s), publishing venue, publication date, database, number of citations, number of downloads from cnki, abstract, keywords, and category.

For each article, we first calculated the frequency of each word. Second, we obtained a stop word list and a keyword list based on the statistics of the words in this particular academic data set. Then, we built the TF-IDF vector for each article to be its content vector in the ALVector. The non-content attributes were extracted and appropriately transformed to normalized numerical values as explained in Section 3.

5.2 Methodology

To evaluate the effectiveness of the proposed recommendation system, we simulated 3 users with different interests. Each researcher was recommended 100 to 150 papers closely related to his or her research interest from the database five times (20 to 30 papers each time). The researcher then rated the papers from 1 to 3 indicating to what degree he/she was satisfied with the recommendations.

Because the number of ratings collected from each user increased over time, the experiments were conducted at five time points. For each experiment, we split the ratings from each user into two subsets: the first 80% of the recommendations were used as training data with which to build the recommendation model; the 20% of the recommendations remaining were regarded as test data.

5.3 Evaluation Metrics

Three widely used metrics were used in our experiments: precision, recall, and Normalized Discounted Cumulative Gain (NDCG).

Precision: the number of recommended relevant documents divided by the total number of recommended documents.

Recall: the ratio of hits divided by the theoretical maximum number of hits owing to the testing set size (McLaughlin & Herlocker, 2004).

Normalized Discounted Cumulative Gain (*NDCG*): measures the usefulness of a document based on its position in the recommended list, where positions are discounted logarithmically (Jarvelin & Kekalainen, 2002). For a given list of size n, the discounted cumulative gain, DCG, is calculated as follows:

$$DCG = rel_1 + \sum_2^n \frac{rel_i}{\log_2 i} \tag{7}$$

where rel_i is the rating of the document at position i in the recommending list. NDCG is normalized DCG given by

$$NDCG = \frac{DCG}{DCG_{ideal}} \qquad (8)$$

where DCG_{ideal} is the ideal DCG.

5.4 Experimental Results

As some statistical analyses have found that readers only read the top recommendations no matter how many items are recommended, we compared the top 10 articles recommended by our personalization-oriented method (PO) with that of the content-based method (CB) (Opricovic & Tzeng, 2004).

Figure 1 presents the precisions of the CB and PO methods. The *x*-axis denotes the time point of the experiments, and the *y*-axis denotes the average precision of all the users at a given point in time. As shown in **Figure 1**, PO significantly outperforms CB in terms of precision. In other words, the recommendations made by PO better fit the users' preferences. In the best case, the average precision of all time points is improved by 100%. At the same time, the average recall of PO is 87% higher than CB. In other words, PO returns more relevant results. This verifies that the proposed non-content attributes in the ALVector and the AUVector do capture the personal characteristics of the users, and the VIKOR-based algorithm does maximize the overall satisfaction rate for each user. In **Figure 1**, as more documents are rated, PO gets more chances to obtain the personal non-content interests or characteristics of a given user, which lead to more accurate recommendations, especially in the top 10 recommended items.

Figure 2 shows the NDCG in our experiments over time. It is evident that PO shows substantial improvement over CB. The reason is that CB only captures the content similarity and lacks the capability to differentiate the personalized preferences among different users. PO tries to overcome this weakness by finding the "optimal" solution in multiple dimensions simultaneously by using VIKOR. The non-content attributes in ALVector and AUVector also contribute to the success of PO in terms of NDCG.
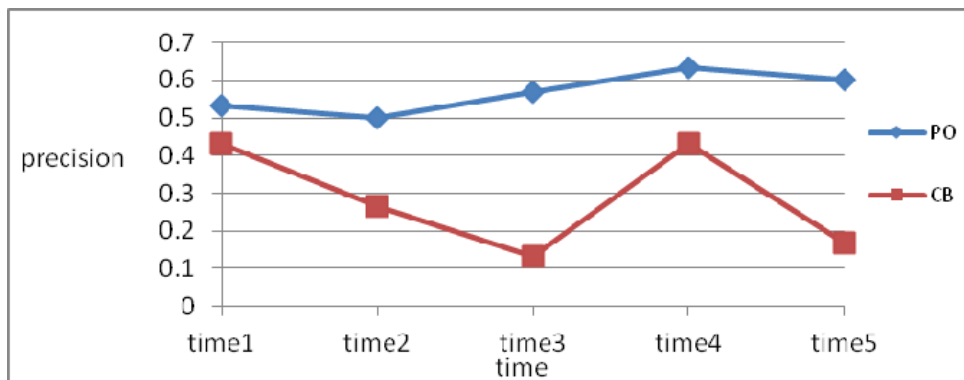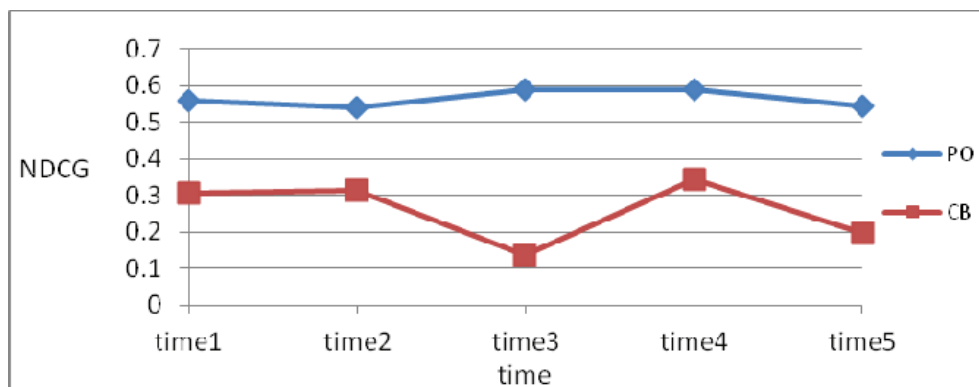

Figure 1: Average precision of all users over time.

Figure 2: Average NDCG of all users over time.

6 Conclusion

In order to overcome the weaknesses of traditional content-based recommendation methods in the context of academic literature recommendation, in this paper, we propose a novel personalization-oriented method, which considers not only content text but also non-content attributes. To achieve this goal, two data structures, ALVector and AUVector, were proposed. The ALVector stores the objective attributes of an article, including the TF-IDF vector, recentness, authority, popularity, etc. The AUVector allows users to express their subjective weights for different attributes. In order to achieve the overall best recommendations, the VIKOR algorithm was adopted to obtain compromise solutions as the optimal solution. A set of real articles downloaded from cnki was used in our experiments. The experiments showed that the recommendations made by the personalization-oriented method outperformed those by content-based methods in precision, recall, and NDCG index. It is observed from the experimental results that the personalized preference of each researcher can be better satisfied by our personalization-oriented recommendation method.

7 Acknowledgement

8 References

Adomavicius, G. & Tuzhilin, A. (2005) Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. IEEE Tran. *Knowledge and Data Engineering 17*(6), pp 734–749.

Andersen, R., Borgs, C., Chayes, J., et al. (2008) Trust-based Recommendation Systems: an Axiomatic Approach. *Proc. of WWW*, NY, USA, pp 199–208.

Arwar, B., Karypls, G., & Konstan, J. (2001) Item-based Collabrative Filtering Recommendation Algorithms. *Proc. of the 10th International World Wide Web Conference,* Hong Kong, China, pp 285–295.

Balabanovic, M. & Shoham, Y. (1997) Fab: Content-based, Collaborative Recommendation. *Communication of the ACM 40*(3), pp 66–72.

Chen, P., Xie, H., Maslov, S., & Redner, J. (2007) Finding Scientific Gems with Google's PageRank Algorithm. *Journal of Informetrics 1*(1), pp 8–15.

Hill, W., Stead, L. Rosenstein, M., & Firnas, G. (1995) Recommending and Evaluating Choices in a Virtual Community of Use. *Proc. Conf. Human Factors in Computing Systems*, Denver, Colorado, USA.

Jarvelin, K. & Kekalainen, J. (2002) Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems 20*(4), pp 422–446.

Liang, S., Liu, Y., Jian, L., et al. (2011) A Utility-based Recommendation Approach for Academic Literatures. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Lyon, France.

Lin, W.Y., Alvarez, S.A., & Ruiz, C. (2000) Collaborative Recommendation via Adaptive Association Rule Mining. *Proc. of International Workshop on Web Mining for E-Commerce*, Boston, MA, USA.

McLaughlin, M. R. & Herlocker, J. L. (2004) A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Sheffield, UK. pp 329–336.

McNee, S. M., Albert, I., Cosley, D., et al., (2002) On the Recommending of Citations for Research Papers, *Proc. of CSCW,* New Orleans, Louisiana, USA, pp 116–125.

Opricovic, S. & Tzeng, G. H. (2004) Compromise Solution by MCDM Methds: a ComparativeAanalysis of VIKOR and TOPSIS. *European Journal of Operational Research 156*(2), pp 445–455.

Rocchio, J. J. (1971) Relevance feedback in information retrieval. In Salton, G. (ed.) The SMART Retrieval System: Experiments in Automatic Document Processing. *Prentice-Hall Series in Automatic Computation*, pp 313–323.

Salton, G. & Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. *Information Processing & Management 24*(5), pp 513–523.

Sayyadi, H. & Getoor, L. (2009) Ranking Scientific Articles by Predicting their Future PageRank. *Proc. of the 9th SIAM International Conference on Data Mining*, Sparks, Nevada, USA.

Torres, R., McNee, S. M., Abel, M., et al., (2004) Enhancing Digital Libraries with TechLens. *Proc. of the 4th ACM/IEEE Joint Conference on Digital Libraries,*. Tucson, Arizona, USA.

Widrow, A. & Hoff, M. (1960) Adaptive Switch Circuits. *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record,* pp 96–104.

OPEN ACCESS