Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 55 (2015) 732 – 740

Procedia Computer Science

Information Technology and Quantitative Management (ITQM 2015)

Improving Ranking-based Recommendation by Social Information and Negative Similarity

Ying Liu[a,b,]* Jiajun Yang[a]

[a]School of Computer and Control, University of Chinese Academy of Sciences, Beijing 100049, China
[b]Key Lab of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100049, China

Abstract

Recommender system is able to suggest items that are likely to be preferred by the user. Traditional recommendation algorithms use the predicted rating scores to represent the degree of user preference, called rating-based recommendation methods. Recently, ranking-based algorithms have been proposed and widely used, which use ranking to present the user preference rather than rating scores. In this paper, we propose two novel methods to overcome the weaknesses in VSRank, a state-of-the-art ranking-based algorithm. Firstly, a novel similarity measure is proposed to make better use of negative similarity; secondly, social network information is integrated into the model to smooth ranking. Experimental results on a publicly available dataset demonstrate that the proposed methods outperform the existing widely used ranking-based algorithms and rating-based algorithms considerably.

Peer-review under responsibility of the Organizing Committee of ITQM 2015

Keywords: Recommendation system, Ranking-based recommendation, Collaborative filtering

1. Introduction

As the amount of information available to us grows dramatically, we are drowning in it and lack of ability to process it. Techniques that help users sift through huge amount of information efficiently are becoming very important to overcome the information overload problem. Recommendation system is one of the promising techniques that can generate item recommendations from a huge collection of items based on users' preferences.

In traditional recommendation systems, the degree of preference is measured by a rating score. For example, Collaborative Filtering (CF) based recommendation algorithms predict the rating scores of unrated items. Given a database of users' past ratings on a set of items, CF can predict the ratings that a user would assign to the unrated items, so that items can be recommended to the user by the predicted ratings in descending order.

* Corresponding author.
E-mail address: yingliu@ucas.ac.cn.

1877-0509
Peer-review under responsibility of the Organizing Committee of ITQM 2015
doi:10.1016/j.procs.2015.07.164

Recently, some new methods have been proposed, which use ranking information instead of rating scores, called ranking-based algorithms. Ranking-based algorithms do not need to predict the rating scores, but directly address the ranking problem without going through the intermediate step of rating prediction [1].

A very promising ranking-based algorithm, VSRank, was proposed recently [2]. It adapts vector space model to CF, resulting a state-of-the-art NDCG performance. Although it shows good NDCG, VSRank still has some weakness that can be improved. In this paper, we propose two methods to overcome the weakness of VSRank: 1) Cosine similarity is used in VSRank to produce a list of the most similar neighbors of a given user. VSRank does not consider the impact from the neighbors who have negative similarities with a given user. However, actually, such information really provides useful hint of the user's interest. Thus, we propose a novel relative preference measure by including the negative pair-wise user similarity; 2) In order to tackle the weakness of VSRank for uncertainty relative preference, we propose a relative preference smoothing method which allows social network information to be included into the measure of user relative preference.

Experiments are conducted on a public data set, Epinions, which not only contains user ratings for items, but also contains the social network information of the users. In order to evaluate the effectiveness of the two proposed methods, we implemented VSRank and Collaborative Filtering, respectively. Normalized Discounted Cumulative Gain (NDCG) is measured for all the three methods. The experimental results showed that both of the proposed methods outperform VSRank and Collaborative Filtering.

1. Related work

Recommendation algorithms can be categorized into two types, rating-based and ranking-based. Collaborative filtering uses the known preferences of a group of like-minded users to make recommendations or predictions for the unknown preferences of other users [4]. In contrast, ranking-based algorithms address the item ranking problem directly without going through the intermediate step of rating prediction. Ranking-based algorithms can be categorized into two subtypes, similarity-ranking method (SRM) and model-ranking method (MRM).

SRM addresses the ranking problem by a set of neighboring users who are similar to the target user. The difference between SRM and CF is the measure of the pair-wise user-to-user similarity. Kendall Rank Correlation Coefficient (KRCC) [1] is used in SRM, which uses relative preference as the measure of pairwise user-to-user similarity. VSRank uses cosine distance of degree-specialty [2] as the measure of pairwise user-to-user similarity.

MRM is a type of machine learning-based approach. It consists of two phases, model learning and rank generating. CoFiRank [5] is a learning-to-rank model which optimizes NDCG metric for ranking. ListRank [6] adopts cross entropy loss and uses the list-wise learning-to-rank algorithm [7] to optimize the matrix factorization. LRHR [8] defines some features and uses Ranking SVM [9] to train a ranking model. SoRank [10] integrates social network information with ListRank and obtained good performance.

Overall speaking, ranking-based recommendation algorithms have achieved a great progress in the most recent years.

2. VSRank

2.1. Vector space model

Vector space model is an algebraic model commonly used in information retrieval (IR). It regards a textual document as a bag of words, disregarding grammar and even word order. It typically uses TF-IDF to weight the terms. Then each document is represented as a vector of TF-IDF weights. Cosine distance is used to compute similarity between document vectors. Large similarity indicates high relevancy of documents.

Term frequency $TF_{t,d}$ of term t in document d is defined as the number of times that t occurs in d. It positively contributes to the relevance of d to t. Inverse document frequency IDF_t of term t measures the rarity of t in a given corpus. If t is rare, the documents containing t are more relevant to t. Formally, $IDF_t = \log\left(\frac{N}{\mathrm{DF}_t}\right)$, where N is the total number of documents and DF_t is the document frequency of t, i.e., the number of documents containing t. The *TF-IDF* value of a term is defined as the product of its *TF* and *IDF* values, that is, $TF\text{-}IDF_{t,d} = TF_{t,d} \times IDF_t$.

VSRank adapts vector space model to collaborative filtering algorithm. It regards the users as documents and the pairwise relative preferences between items as words. The key point is the degree-specialty weighting scheme which simulates the *TF-IDF* scheme. VSRank uses cosine distance between two vectors as the measure of user-to-user similarity. Eventually, the preferences of the top N most similar neighbors of user u are used to recommend items for u.

2.2. User-to-user similarity

Let $r_{u,m}$ denote the rating score of user u to item m. The degree of two items, m and n, for user u is defined in Equation (1), and the specialty of m and n is defined in Equation (2).

$$Degree_{u,i_m\theta i_n} = \log_2(1 + |r_{u,m} - r_{u,n}|) \tag{1}$$

$$Specialty_{i_m\theta i_n} = \log_2\left(\frac{N_{i_m>i_n} + N_{i_m<i_n}}{N_{i_m\theta i_n}}\right) \tag{2}$$

Where $i_m\,\theta\,i_n$ ($\theta \in \{<, >\}$) denotes u's preference between m and n; $N_{i_m>i_n}$ is the number of users who rated item m higher than n; degree-specialty of $i_m\theta i_n$ for user u is the defined as the product of degree and specialty, as shown in Equation (3):

$$DS_{u,i_m\theta i_n} = Degree_{u,i_m\theta i_n} \times Specialty_{i_m\theta i_n} \times I(r_{u,m} - r_{u,n}) \tag{3}$$

Where $I(x)$ is an indicator function, defined as Equation (4):

$$I(x) = \begin{Bmatrix} 1 & x \geq 0 \\ -1 & x < 0 \end{Bmatrix} \tag{4}$$

Let vector V_u denote the *DS* value of user u where every element in the vector is the pair-wise *DS* value of items. Thus, the user-to-user similarity between user u and w is defined as the cosine distance between the two vectors, as shown in Equation (5):

$$S_{u,w} = \frac{V_u \cdot V_w}{\| V_u \| \times \| V_w \|} \tag{5}$$

2.3. Ranking prediction

VSRank consists of two phases:

1) Neighbor discovery. For each user, it discovers his/her most similar neighboring users;
2) Item ranking prediction. It adopts the ranking prediction method in Eigen-Rank [1]. Firstly, it determines the pair-wise relative preferences of user u based on the preferences of his/her most similar neighbors. Secondly, such estimated pair-wise preferences are aggregated into a complete ranking of all the items via a greedy method [2].

2.4. Weakness of VSRank

Although VSRank is an outperforming algorithm, it still has some weaknesses.

1. Loss of similarity

As seen in Equation (3), (4) and (5), user-to-user similarity may be negative when they have opposite relative preferences on many items. Particularly, the user-to-user similarity will be -1 when the following conditions are all satisfied:

- The number of users whose preference $i_m > i_n$ is equal to that of $i_m < i_n$, that it, $N_{i_m > i_n} = N_{i_m < i_n}$;
- The degree of any pair of items is equal, that is, $Degree_{u,i_m \theta i_n} = Degree_{w,i_m \theta i_n}$;
- The relative preference is opposite for two users, that is, $I(r_{u,m} - r_{u,n}) \times I(r_{w,n} - r_{w,m}) = -1$.

Any pair of users who satisfy all the above three conditions will have similarity score -1; on the other side, there must be pairs of users having similarity score +1. So, the range of the similarity between a pair of users is $[-1, 1]$. The score range can be segmented into three parts: 1) (0, 1] means that the two users have positive correlation, indicating similar preference to a certain degree; 2) {0} means there is no correlation between the two users at all; 3) $[-1, 0)$ means that the two users have opposite preference. In traditional methods, the users whose similarities are negative will not have chance to be selected as they will be at the end of the sorted list. However, in reality, it is a common sense that the users who have opposite interests in the history may always have opposite interests. So, negative similarity which is ignored by VSRank may have positive impact to recommendation. As VSRank does not make full use of the similarity, we call this problem *loss of similarity*.

2. Relative preference deficiencies

In the ranking prediction phase, VSRank uses a greedy algorithm [1] to aggregate the relative preference and outputs a rank list of recommendation items. The greedy algorithm chooses the max π_i, firstly, as shown in Equation (6) :

$$\pi_i = \sum_{i \in I} lp(i,j) - \sum_{j \in I} lp(j,i) \tag{6}$$

Where *i* and *j* are the item indices, *I* is the item set, $lp(i,j)$ is the relative preference. $lp(i,j)$ for user *u* is predicted by $lp(i,j)$ of the top *N* most similar neighbors. Every π_i would be updated whenever an item is selected, as shown in Equation (7), where *t* is the index of the item that was selected in the last iteration:

$$\pi_i = \pi_i - lp(i,t) + lp(t,i) \tag{7}$$

However, in some circumstance, the relative preference of a pair of items is predicted to be zero because of his neighbors' uncertainty relative preferences. For such cases, the greedy algorithm will not perform well. For example, assume we have three items, $i_1,\ i_2,\ i_3$ and their relative preferences are

$$\mathrm{lp}(1,2) = 1$$
$$\mathrm{lp}(1,3) = 1$$
$$\mathrm{lp}(2,3) = 0$$

Then, we get $\pi_1,\ \pi_2,\ \pi_3$ by Equation (6)

$$\pi_1 = \big(lp(1,2) + lp(1,3)\big) - 0 = 2$$
$$\pi_2 = lp(2,3) - lp(1,2) = -1$$
$$\pi_3 = 0 - \big(lp(1,3) + lp(2,3)\big) = -1$$

We select i_1 and update $\pi_2,\ \pi_3$ by Equation (7)

$$\pi_2 = \pi_2 - lp(2,1) + lp(1,2) = 0$$
$$\pi_3 = \pi_3 - lp(3,1) + lp(1,3) = 0$$

It is obvious that $\pi_2 = \pi_3$ which makes the greedy algorithm hard to choose at the moment. It will randomly choose one out of the tie items. As a result, the quality of the recommendation will be degraded.

3. Proposed methods

In order to overcome the weakness of VSRank, we propose two novel methods.

3.1. Use of negative similarity

We would like to include negative similarity (as stated in Section 3.4), the range of $[-1, 0)$, into the calculation of the relative preference of two items. So, we propose a novel function to allow the "negative similar" users to move forward in the neighbor list. In contrast to Equation (5), we propose to use the absolute value as the user-to-user similarity, as shown in Equation (8):

$$S'_{u,w} = \frac{|V_u \cdot V_w|}{\| V_u \| \times \| V_w \|} \quad (8)$$

Thus, the relative preferences of u are determined by the top *N* most similar neighbors' similarity by following Equation (9):

$$lp(i,j) = \frac{\sum_{v \epsilon U_u^{i,j}} (S_{u,v} \times I_{v,i,j})}{\sum_{v \epsilon U_u^{i,j}} S'_{u,v}} \quad (9)$$

Where $U_u^{i,j}$ denotes the top *N* neighbors who rated both item *i* and *j*; $I_{v,i,j}$ denotes the output of Equation (4) with parameter$x$ equal to $r_{v,i} - r_{v,j}$, that is to say， $I_{v,i,j} = I(r_{v,i} - r_{v,j})$. Note we use $\sum_{v \epsilon U_u^{i,j}} S'_{u,v}$ as the divisor instead of $\sum_{v \epsilon U_u^{i,j}} S_{u,w}$ in Equation (9).

Let's use an example to illustrate the idea. Assume we have four users, $u_1,\ u_2,\ u_3, and\ u_t$. Assume u_t is our target user. The relative preference $lp(i,j)$ for $u_1,\ u_2,\ u_3$ is -1, 2, -1, and thus, the three user-to-user pair-wise similarities are $S_{1,t} = -0.99$, $S_{2,t} = 0$, $S_{3,t} = 0.01$. So, the similar user list is $\{u_1,\ u_3,\ u_2\}$. Assume we use top two neighbors for prediction, $lp(i,j)$ for u_t is predicted as follows:

$$lp(i,j) = \frac{-0.99 \times (-1) + 0.01 \times (-1)}{0.99 + 0.01} = 0.98$$

Without using negative similarity, the similar user list is $\{u_3,\ u_2,\ u_1\}$, $lp(i,j)$ is predicted as

$$lp(i,j) = \frac{0.01 \times (-1) + 0 \times 2}{0.01 + 0} = -1$$

Without using negative similarity, $S_{3,t}$ is just 0.01, which is meaningless but actually determines $lp(i,j) < 0$. In other words, the algorithm recommends items based on the user who shares "**weak commom**" interest with the target user. By using negative similarity, $S_{1,t}$ makes contribution to $lp(i,j) > 0$. That is, the algorithm recommends items based on the user who has "**strongly opposite**" interest to the target user. Intuitively, strong opposite interest may contribute more than weak common interest which will be verified in the experiments in the subsequent section.

3.2. Relative preferences smoothing

VSRank uses relative preference and greedy algorithm to make prediction. For the case of "zero relative preference", VSRank does not work well. So, we propose a novel smoothing method to address the problem.

In real-world applications, it is a common sense that if your close friends are interested in a particular topic, it is highly probable that you are interested in it too. Based on this observation, we propose to integrate social network information into the model of VSRank. In detail, we would like to use social network information to smooth the "zero relative preference". The proposed relative preference calculation method is presented in Equation (10).

$$lp'(i,j) = \frac{\sum_{v \epsilon R_u^{i,j}} (P_{u,v} \times I_{v,i,j})}{\sum_{v \epsilon U_u^{i,j}} |P_{u,v}|} \quad (10)$$

Where $lp'(i,j)$ denotes zero relative preference; $R_u^{i,j}$ is the set of social friends who rated both item i and j; $P_{u,v}$ is the weight of the connection between user v and u, and u is the target user. The weight between a pair of users indicates the closeness of their relationship. A parameter α is used to balance the impact of social information and preference similarity. The relative preference method is summarized in Equation (11):

$$\mathrm{lp}''(\mathrm{i},\mathrm{j}) = \begin{Bmatrix} \alpha\,\mathrm{lp}(\mathrm{i},\mathrm{j}) & \mathrm{lp}(\mathrm{i},\mathrm{j}) \neq 0 \\ (1-\alpha)\mathrm{lp}'(\mathrm{i},\mathrm{j}) & \mathrm{else} \end{Bmatrix} \tag{11}$$

It is evident that the power of our proposed relative preference calculation method is determined not only by the interest similarity between users but also by the social network information. The more social information provided, the better the recommendation quality.

4. Experimental results

4.1. Data set

Epinions, downloadable at http://www.public.asu.edu/ jtang20/datasetcode, is used in our experiment. It contains 22,166 users who expressed 922,267 ratings for 296,277 items. In addition to the rating scores, user friendship information is also provided. The total number of friendship connections is 355,754. In our experiment, we ignored the users who have less than 50 friend connections. 80% of the rated items are used for training and the remaining 20% for testing. The top 50 similar users are used for relative preference caluculation.

4.2. Evaluation methodology

Since the goal of our proposed methods is to improve item ranking rather than rating prediction, we employ *Normalized Discounted Cumulative Gain* (*NDCG*) [6] as the measure. *NDCG* is popularly used in information retrieval in recent years. Specifically, *NDCG* is evaluated over the top *n* items on the ranked recommendation list. Let *U* be the set of users and $r_{u,p}$ be the rating score of the item at the *pth* place in the recommendation list, and Z_u be a normalization factor calculated which makes the max value of *NDCG* be 1). *NDCG* at the *nth* position with respect to user *u* is defined as follows：

$$NDCG_u@n = Z_u \sum_{p=1}^{k} \frac{2^{r_{u,p}} - 1}{log(1+p)}$$

For a set of users *U*, the average *NDCG* at the *nth* position is:

$$NDCG_{avg}@n = \frac{1}{|U|} \sum_{u \epsilon U} NDCG_u@n$$

The value of *NDCG* ranges from 0 to 1. The higher the value, the better the ranking. *NDCG* is very sensitive to the ratings of the highest ranked items [2].

In order to give a fair comparison, we implemented user-based Collaborative Filtering (CF) [4], VSRank [2]. Comparisons between CF, VSRank and our proposed methods are presented in the subsequent section.

4.3. Performance

1. SVSR vs. Collaborative Filtering and VSRank

Table 1. Impact of the negative similarity

	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
SVSR	0.724815	0.734868	0.744346	0.752244	0.763357
CF	0. 645656	0. 64765	0. 650566	0. 65176	0. 658461
VSRank	0.68961	0.703477	0.717509	0.728599	0.743331

We test the impact of our proposed method in Section 4.1, called SVSR, against VSRank on NDCG@1~ NDCG@5. SVSR improves VSRank by incorporating negative similarity. Results in Table 1 show that SVSR outperforms VSRank and CF significantly.

2. Smoothing vs. VSRank

We evaluate the effectiveness of social network information factor α. Figure 1 shows the *NDCG* measure when varying α from 1 to 0.001. It is evident that the best *NDCG* is achieved when α is set at 0.01. However, it is also evident that the improvement of *NDCG* is not remarkable when α is over 0.5. Therefore, α=0.5 is a good choice.

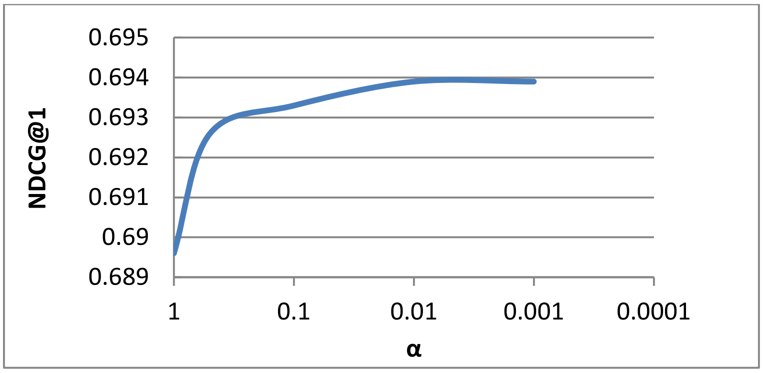

Fig. 1. Impact of social network information factor α to NDCG.

3. EVSR vs. Collaborative Filtering and VSRank

Finally, we integrate the two proposed methods into a single model, called EVSR. We compare EVSR with CF and VSRank. The social information factor α is set at 0.5. Figure 2 shows the experimental results on NDCG@1, NDCG@3, NDCG@5. It is evident that EVSR achieved remarkable improvement in terms of *NDCG* over Collaborative Filtering and VSRank. For example, on NDCG@1, CF achieves 0.6456, VSRank achieves 0.6896 and EVSR achieves 0.7272 which is 12.64% over CF and 5.45% over VSRank.

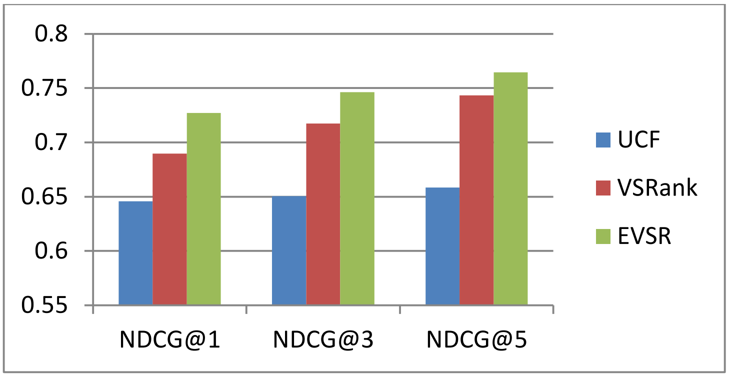

Fig. 2. Performance comparison between EVSR, Collaborative Filtering and VSRank.

5. CONCLUSION AND FUTURE WORK

Ranking-based recommendation algorithms have achieved a better performance in recommendation system than rating-based recommendation algorithms in recent years. It is attracting more and more attention from researchers and gradually becoming a research hot spot. A lot of ranking-based algorithms have been proposed. In this paper, we proposed two novel methods to overcome the weaknesses in VSRank, which is a state-of-the-art ranking-based algorithm. Firstly, negative similarity between users is incorporated into the relative preference between users. Secondly, social network information is incorporated to smooth the relative preferences. The effectiveness of our ideas is evaluated on a public available data set *Epinions* and remarkable improvement in terms of *NDCG* is observed over VSRank and Collaborative Filtering.

In the future, we plan to verify the effectiveness of EVSR on more datasets. Other methods to smooth the relative preferences will be explored and investigated. We would like to investigate the effectiveness of other similarity functions, such as KRCC. In addition, other factors might be integrated into the user-to-user similarity measure, such as demographic information, etc.

Acknowledgements

This project is partially supported by grants from Natural Science Foundation of China #61202312/70621001/70921061.

References

[1] N. N. Liu, Q. Yang. Eigenrank: A ranking-oriented approach to collaborative filtering. SIGIR 2008.
[2] Wang, S., Sun, J., Gao, B.J., Ma, J.,. Adapting vector space model to ranking-based collaborative filtering. In: Proceedings of the 21st ACM Conference on Information and Knowledge Management (CIKM 2012)
[3] R. A. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc.,Boston, MA, USA, 1999.
[4] C. M. K. John S. Breese, David Heckerman. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificia Intelligence (UAI), 1998.
[5] M. Weimer, A. Karatzoglou, Q. V. Le, A. J. Smola. Cofirank: Maximum margin matrix factorization for collaborative ranking. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS), 2007.
[6] Y. Shi, M. Larson, A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In RecSys, pages 269–272. ACM, 2010.

[7] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F. and Li, H., 2007. Learning to rank: From pairwise approach to listwise ap-proach. Technical Report, MSR-TR-2007-40, Microsoft Re-search.
[8] Sun, J., Wang, S., Gao, B.J.,Ma, J.,2012. Learning to rank for hybrid recommendation. In: Proceedings of the 21st ACM Conference on Information and Knowledge Management (CIKM 2012).
[9] T. Joachims. Optimizing search engines using clickthrough data. In Proceedings of SIGKDD 2002, pages 133–142, 2002.
[10] Weilong Yao , Jing He, Guangyan Huang, Yanchun Zhang. SoRank: Incorporating Social Information into Learning to Rank Models for Recommendation(WWW 2014).
[11] Ying Liu, Wei-keng Liao, Alok Choudhary A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets PAKDD 2005, LNAI 3518, pp. 689 – 695, 2005. © Springer-Verlag Berlin Heidelberg 2005.