Improving the Performance of Recommender System by Exploiting the Categories of Products

Mohak Sharma[1], P. Krishna Reddy[1], R. Uday Kiran[1], and T. Ragunathan[2]

[1] International Institute of Information Technology Hyderabad, Hyderabad, India
{mohak.sharma,uday_rage}@research.iiit.ac.in, pkreddy@iiit.ac.in
[2] ACE Engineering College, Hyderabad, India
ragunathan@research.iiit.ac.in

Abstract. In the literature, collaborative filtering (CF) approach and its variations have been proposed for building recommender systems. In CF, recommendations for a given user are computed based on the ratings of k nearest neighbours. The nearest neighbours of target user are identified by computing the similarity between the product ratings of the target user and the product ratings of every other user. In this paper, we have proposed an improved approach to compute the neighborhood by exploiting the categories of products. In the proposed approach, ratings given by a user are divided into different sub-groups based on the categories of products. We consider that the ratings of each sub-group are given by a virtual user. For a target user, the recommendations of the corresponding virtual user are computed by employing CF. Next, the recommendations of the corresponding virtual users of the target user are combined for recommendation. The experimental results on MovieLens dataset show that the proposed approach improves the performance over the existing CF approach.

Keywords: Electronic commerce, Recommender systems, Collaborative Filtering, Mass-customization, Classification, Customer loyalty, Cross-sell, Up-sell.

1 Introduction

A recommender system for an E-commerce site receives information from a customer about which products he/she is interested in, and recommends products that are likely to fit his/her needs. Today, recommender systems are deployed on hundreds of sites, serving millions of customers. For example, recommender systems are employed to suggest books on Amazon™(www.amazon.com). Currently, recommender systems have become a key component of modern E-commerce applications. Several research efforts are going on to investigate efficient algorithms for building recommender systems [1,2,3].

In the literature, collaborative filtering (CF) approach has been proposed to build recommender systems [4]. There are two types of CF approaches: memory-based and model-based. Model-based CF approaches compute predictions by

S. Kikuchi et al. (Eds.): DNIS 2011, LNCS 7108, pp. 137–146, 2011.

Table 1. Sample of transactions in a book store

User/Category	S_1	S_2	S_3	S_4	C_1	C_2	C_3
U_1	1	1	1	0	1	1	0
U_2	1	0	0	0	1	1	1
U_3	1	1	0	1	0	0	0
U_4	1	0	1	1	0	0	1
U_5	0	1	0	0	1	1	1
U_6	0	0	1	0	1	1	0

modeling user and item and memory-based CF approaches compute recommendations based on the purchase history of products and users. Improving the performance of CF approach is one of the research issue [1].

In this paper, we make an effort to propose the improved memory-based CF approach by exploiting the categories of products.

The CF approach works by building a database of product ratings given by customers. It recommends products to the customer based on the ratings of other customers who gave similar product ratings. To recommend products to a target user, finding other users who have similar preferences with the target user is an important step. In CF, the recommendation for a target user is computed based on the ratings of k nearest neighbours. Under CF, the product ratings of the target user are compared with the product ratings of other users, and the nearest neighbours are computed based on the similarity values. We have observed the fact that there is an opportunity to improve the performance of CF if we process the ratings of the user by grouping them category-wise. In CF, during the neighbourhood formation step, the similarity values of the product ratings between two users are computed by considering the product ratings of each user as one unit. Normally different sub-groups, based on categories, exist in the set of products rated by a user. It means that users may rate products similarly with respect to certain categories and dissimilar with respect to other categories. In this situation, if we carry out comparison at the user-user level, there is a possibility to miss the close neighbours with respect to categories. So, if we find the neighbourhood by comparing the ratings at the category-level, there is an opportunity to improve the performance of CF. We explain the proposed idea through Example 1.

Example 1. Let U_1, U_2, U_3, U_4, U_5 and U_6 be the users and S_1, S_2, S_3, and S_4 are story books and C_1, C_2 and C_3 are the books related to computer science. Table 1 shows the user-book matrix where the value '1' indicates that the user has purchased the corresponding book (rating = 1) and a '0' indicates that the user hasn't purchased the book (rating = 0). Let U_1 be the target user. If we find the similarity of other users with U_1, the top two neighbours selected for U_1 will be U_2 and U_5. The similarity is computed based on the number of common books purchased. If we compare category-wise, the neighbours of U_1 computed by considering only story books are U_3 and U_4. In this way, the notion of neighbourhood changes if we find the similarity by considering entire user as one unit and each category as one unit.

In the proposed approach, each user ratings are divided into different sub-users by exploiting the categories of products. We consider that the ratings of each sub-group are given by a virtual user. For a target user, the recommendations of the corresponding virtual user are computed by employing CF. Next, the recommendations of the corresponding virtual users of the target user are combined for recommendation. The experimental results on MovieLens dataset show that the proposed approach improves the performance over the existing CF approach.

The rest of the paper is organized as follows. In the next section, we discuss the related work. In section 3, we briefly explain CF. We present the proposed approach in Section 4. Experimental results are presented in Section 5. The last section consists of summary and conclusions.

2 Related Work

A survey on the approaches used for building recommender systems is carried out in [1]. The survey paper discusses various limitations of recommendation methods and suggests possible extensions. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multi-criteria ratings, and a provision of more flexible and less intrusive types of recommendations.

The CF approach is a popular recommendation approach and several variations have been proposed in the literature. The user-based CF is proposed in [4] and the item-based CF is proposed in [5,6]. A fusion framework of both user-based and item-based approaches have been proposed in [2]. Wang [7] has shown how the development of CF can gain benefits from information retrieval theories and models, and proposed probabilistic relevance CF models. Horting [8] is a graph-based recommendation technique in which nodes are consumers, and edges between nodes indicate degree of similarity between two consumers. Bell and Koren [9] have used a comprehensive approach to improve the performance of CF by removing global effects in the data normalization stage of the neighbour-based CF and working with residual of global effects to select neighbours. A user-based CF which is based on an analysis of prediction errors is presented in [10].

A rate-item-pool-based (RIP-based) approach has been proposed in [11]. The RIP-based approach refines the contribution of the global neighbourhood by weighing the impact of global neighbours with a fine-grained similarity metric based on RIPs, and subsets of item ratings in the active user's profile. Ensemble method has been proposed to improve the performance of CF algorithms [12] which combines the predictions of different algorithms (the ensemble) to obtain the final prediction.

In [13], two approaches based on CF namely latent factor models and neighbourhood models are exploited to propose an effective recommendation algorithm. A recursive prediction algorithm is proposed in [14] which suggests that if a nearest-neighbour user has not rated the given item, it's value is estimated based on his/her own nearest neighbourhood. Next, the the estimated rating value is used in the the prediction process for the final active user.

An alternative method to find neighbourhood by exploiting lower-bound similarity is proposed in [15]. A preference-based organization technique has been proposed in [16] to accelerate users' decision process. It suggests that rather than explaining each item one by one, a group of products can be explained together by a category title, provided that they have shared tradeoff characteristics compared to a reference product. A prediction algorithm is discussed in [17] which predicts the ratings of items that they have not rated for every user. The algorithm proposed in [18] visualizes the problem as node selection on a graph, giving high scores to nodes that are well connected to the older choices, and at the same time well connected to unrelated choices.

In this paper, we have made an effort to propose an improved recommendation approach by exploiting the categories of products. The proposed idea is different from the preceding approaches as it exploits a notion of "virtual user" for finding efficient neighbourhood.

3 Collaborative Filtering Framework

In this section we explain CF. It [4] consists of three sub-tasks: data representation, neighbourhood formation and recommendation generation.

1. *Data representation*
   The input data is a collection of products purchased or rated by a user. Assume that there are m users and n products. It is usually represented as a $m \times n$ user-product matrix, R, such that $r_{i,j}$ is '1' if the i^{th} user has purchased the j^{th} product, and '0', otherwise.

2. *Neighbourhood Formation*
   The main goal of neighbourhood formation is to find, for each user u, an ordered list of k users N=$\{N_1,N_2,...,N_k\}$ such that u $\notin$ N and sim(u,N_1) is maximum, sim(u,N_2) is the next maximum and so on. The most extensively used similarity measures are based on correlation and cosine-similarity [5,19,20]. After computing the proximity among users, the next task is to actually form the neighbourhood. Different kinds of neighbourhood formation approaches can be employed. The *Center-based* [4] approach forms a neighbourhood of size k for a particular user c by simply selecting the k nearest other users.

3. *Generation of Recommendation*
   In this step, top-N recommendations ($N > 0$) are computed for a given user. For this, *Most-frequent Item Recommendation* method can be used. The procedure is as follows. It looks into the neighbourhood and scans through the ratings data for each neighbour and performs a frequency count of the products rated. After sorting the products according to their frequency count, it returns the N most frequent products that have not yet been purchased by the target user as recommendation.

4 Proposed Approach

It can be observed that the neighbourhood formation process plays a key role in improving the performance of recommender system. Under CF, a fixed number of neighbours for the target user are selected by considering the ratings of the one user as a single unit. This is appropriate if a typical user rates/purchases the products in all categories in a uniform manner. However, a user may not purchase or rate the products in all categories in a uniform manner in certain kinds of applications. That is, a typical user rates more number of products in certain categories and rates few products among other categories. So, if we consider the ratings of one user as a single unit and find similar users, there is a possibility of missing genuine neighbours. There is a scope to improve the performance, if we divide the user ratings into sub-groups by exploiting categories and build an algorithm by computing neighbourhood at category level.

Similar to CF, the proposed category-based CF (CCF) consists of the following steps: data representation, neighbourhood formation, and generation of recommendation.

The CCF approach divides a target user into several virtual users by considering that each category of products of user are rated by a virtual user. We find neighbours for each virtual user of a target user by employing CF. Next, the recommendations to the target user are computed by combining recommendations of the corresponding virtual users. We explain these steps in detail.

1. *Data representation in CCF*
   In CCF, a user is fragmented into virtual users on the basis of the categories of the purchased products. For instance, a particular user u has purchased n products which can be classified under c categories. The transaction of a user u is divided into c virtual users. Let m, p, c, and v represent the number of real users, products, categories and virtual users respectively. Then, $v = m \times c$. So, $(v \times p)$ virtual user-product matrix will be formed.
2. *Neighbourhood Formation in CCF*
   In CCF, neighbourhoods are formed by processing the ratings of virtual users. For a given real user, neighbourhood is formed for all the corresponding virtual users. The proximity and neighbourhood methods of CF can be used.
3. *Generation of recommendation in CCF*
   The process of recommendation generation in CCF is different from CF. At first, we have to generate recommendation for each virtual user of the corresponding target user. We can follow the most-frequent item recommendation method for this step. Next, recommendations have to be combined to generate final recommendations to the target user. Several options are possible. We present two approaches.
   (a) *Random Approach*: We combine all the recommended products into one set. To find *top-N* recommendations for a particular user, we randomly select N recommendations from this set.

(b) *Ranking Approach*: In this algorithm, we select top ranked P ($P > 0$) virtual users and follow random approach. The ranking approach is as follows. At first, the virtual users of a target user are ranked based on the number of products rated. The virtual user who rates the large number of products receives higher rank. To find *top-N* recommendations for a particular user, we randomly select $\lfloor \frac{N}{P} \rfloor$ recommendations from the corresponding *top-P* virtual users.

5 Experimental Results

We conducted experiments on the data set provided by MovieLens (http://www.grouplens.org/) project. We selected 943 users to obtain 10,0000 ratings on 1682 movies. All ratings follow the 1 (bad) - 5 (excellent) numerical scale. The data set was converted into a user-movie ratings matrix R that had 943 rows (i.e., 943 users) and 1682 columns (i.e., 1682 movies). There are total 18 genres available. The initial dataset was divided into five distinct splits. All the experiments are performed on each of the five splits and average value is reported.

Dataset (each split) was divided into two parts: the training set and the test set. Experiments have been done on the training set, and generated a set of recommendations, we call the *top-N* set. We then look into the test set and match products with our *top-N* set. Products that appear in both sets are members of a special set, we call the *Hit Set* and each match is known as a *Hit*.

We employed recall, precision, and F1-metric [3] as performance metrics. The definitions of precision, recall and F1-metric are as follows.

- Precision. It is defined as the ratio of hit set size to the *top-N* set size, i.e., $precision = \frac{size\ of\ hit\ set}{size\ of\ top-N\ set}$ which can be written as

$$precision = \frac{|test \bigcap top-N|}{|N|}. \tag{1}$$

- Recall. It is defined as the ratio of hit set size to the test set size, i.e., $recall = \frac{size\ of\ hit\ set}{size\ of\ test\ set}$ which can be written as

$$recall = \frac{|test \bigcap top-N|}{|test|}. \tag{2}$$

- F1-metric. It is a combined effect of both recall and precision.

$$F1 = \frac{2 * recall * precision}{recall + precision}. \tag{3}$$

Using the genres of a movie as the categories of a product, 943 users have been fragmented into several other virtual users. The neighbours of a virtual user have been formed using *center-based* neighbourhood method. The *most-frequent item*

recommendation is used for generating recommendations to the virtual users. The total number of recommendations for a virtual user and to a target user has been set at 10, i.e., $N = 10$.

We have conducted experiments by fixing k=70 (number of neighbours for virtual users) and varying the number of virtual users from 1 to 18 (there are only 18 genres in the dataset). So each virtual user can only be split into maximum of 18 virtual users. Figures 1(a), 1(b), and 1(c) show the precision, recall and F1-metric performance of CF, CCF with random and CCF with ranking approaches respectively. It should be noted that the performance of both CF and CCF with random method does not vary with the number of virtual users. The performance curve of CCF with random method indicates the recommendation performance obtained by selecting final recommendations randomly from the recommendations of 18 virtual users. It can be observed that CCF with random method improves the performance over CF. It is due to the fact that by

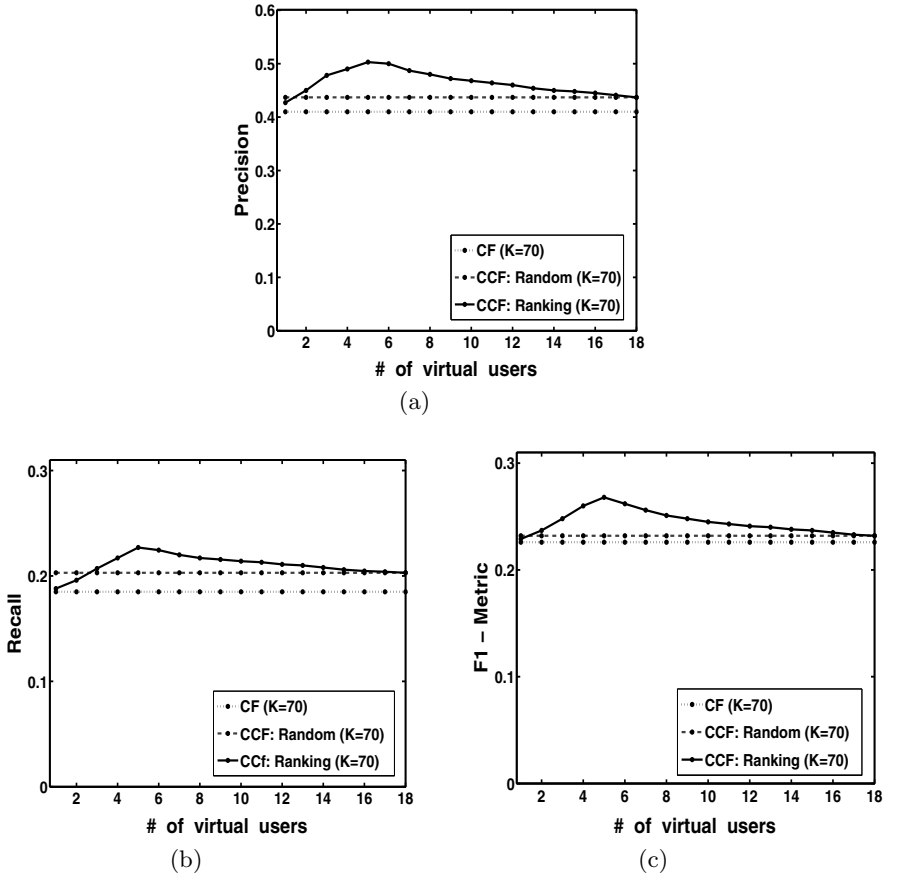

Fig. 1. Performance results of # of virtual users vs (a) Precision, (b) Recall and (c) F1-metric

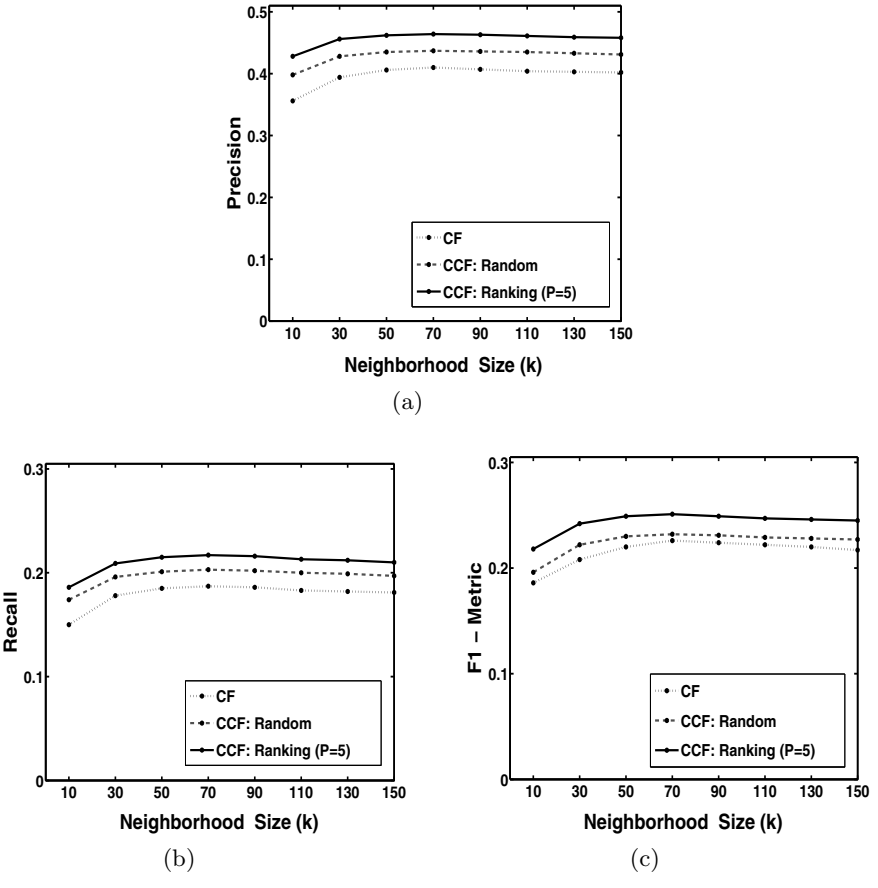

Fig. 2. Performance results of k vs (a) Precision, (b) Recall and (c) F1-metric

computing neighbourhood at the category level, the proposed approach is able to get the efficient neighbourhood as compared to CF. It can be noted that the performance of CCF with ranking method varies based on the number of virtual users. As we increase the number of virtual users, the performance increases gradually to the peak. It then gradually decreases and coincides with the random approach as expected. The results show that the performance of CCF with ranking method is significantly higher than CF. It indicates that by computing the recommendations from the top ranked virtual users, it is possible to capture the neighbourhood based on user interests in an efficient manner.

We have also conducted experiments by fixing number of virtual users to five and varying number of neighbourhood virtual users from 10 to 150. Figure 2(a), 2(b), and 2(c) shows the precision, recall and F1-metric performance respectively. As we increase the number of virtual users in neighbourhood, the performance of CF, CCF with random, and CCF with ranking approaches increases gradually and saturates. It can be observed that CCF with random method improves the

performance over CF. Also, the results show that the performance CCF with ranking method is higher than the other two approaches.

Overall, the experiment results show that the proposed approach improves the performance over CF.

6 Conclusion and Future Work

Recommender system is the main component in E-commerce systems. In this paper, we made an effort to improve the performance the CF approach which is being used to build recommendation systems. We have proposed a framework in which each user is divided into virtual users based on the categories of the products rated. The proposed approach divides each user into corresponding virtual users, computes recommendations for each virtual user and combines these recommendations to give recommendations to the target user. Through experimental results, it has been shown that it is possible to improve the performance of recommender systems using the proposed approach.

As a part of future work, we are planning to conduct extensive experiments by employing different neighbourhood formation and similarity methods. We are planning to investigate improved methods to combine the recommendations computed for virtual users for giving final recommendations to the user. We are also planning to investigate how the notion of virtual user improves the performance of item-based, model-based and other variations of CF approaches.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng., 734–749 (2005)
2. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proc. of 29th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (2006)
3. Roy, S.B., Amer-Yahia, S., Chawla, A., Das, G., Yu, C.: Constructing and exploring composite items. In: Proc. of the 2010 International ACM SIGMOD Conference on Management of Data, pp. 843–854 (2010)
4. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: ACM Conference on Electronic Commerce, pp. 158–167 (2000)
5. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proc. of World Wide Web Conference, pp. 285–295 (2001)
6. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information System, 143–177 (2004)
7. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unified relevance models for rating prediction in collaborative filtering. ACM Transaction on Information Systems 26 (2008)

8. Wolf, J., Aggarwal, C., Wu, K.-L., Yu, P.: Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (1999)
9. Bell, R., Koren, Y.: Improved neighbourhood-based collaborative filtering. In: Proc. of Knowledge Discovery and Data Mining Cup and Workshop (2007)
10. Ding, S., Zhao, S., Yuan, Q., Zhang, X., Fu, R., Bergman, L.D.: Boosting collaborative filtering based on statistical prediction errors. In: Proc. of the 2008 ACM Conference on Recommender Systems, pp. 3–10 (2008)
11. Shi, Y., Larson, M., Hanjalic, A.: Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering. In: Proc. of the 2009 ACM Conference on Recommender Systems, pp. 125–132 (2009)
12. Jahrer, M., Töscher, A.: Combining Predictions for Accurate Recommender Systems. In: Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pp. 25–28 (2010)
13. Koren, Y.: Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proc. of 14th ACM SIGKDD Knowledge Discovery and Data mining (2008)
14. Zhang, J., Pu, P.: A recursive prediction algorithm for collaborative filtering recommender systems. In: Proc. of 2007 ACM Conference on Recommender Systems (2007)
15. Sharma, M., Reddy, P.K.: Using lower-bound similarity to enhance the performance of recommender systems. In: Proc. of ACM Bangalore Compute Conference (2011)
16. Chen, L., Pu, P.: A cross-cultural user evaluation of product recommender interfaces. In: Proc. of the 2008 ACM Conference on Recommender Systems (2008)
17. Schclar, A., Tsikinovsky, A., Rokach, L., Meisels, A., Antwarg, L.: Ensemble methods for improving the performance of neighborhood-based collaborative filtering. In: Proc. of the 2009 ACM Recommeder Systems, pp. 261–264 (2009)
18. Onuma, K., Tong, H., Faloutsos, C.: TANGENT: a novel, 'Surprise me', recommendation algorithm. In: The Proc. of 15th ACM SIGKDD Knowledge Discovery and Data mining (2009)
19. Herlocker, J.L., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. In: Proc. of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2002)
20. McLaughlin, M.R., Herlocker, J.L.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Proc. of 27th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (2005)