

# Recommendation Based On Object Typicality

Yi Cai\*, Ho-fung Leung†, Qing Li\*, Jie Tang‡ and Juanzi Li‡

\*Department of Computer Science, City University of Hong Kong, Hong Kong, China  
{yicai3, itqli}@cityu.edu.hk

†Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China  
lhf@cuhk.edu.hk

‡Department of Computer Science and Technology, Tsinghua University, Beijing, China  
jietang@tsinghua.edu.cn, ljz@keg.cs.tsinghua.edu.cn

## ABSTRACT

Current recommendation methods are mainly classified into content-based, collaborative filtering and hybrid methods. These methods are based on similarity measurements among items or users. In this paper, we investigate recommendation systems from a new perspective based on object typicality and propose a novel typicality-based recommendation approach. Experiments show that our method outperforms compared methods on recommendation quality.

## Categories and Subject Descriptors

H.4.m [Information Systems Applications]: Miscellaneous

## General Terms

Algorithms, Theory, Human Factors

## Keywords

Recommendation System, Object Typicality

## 1. INTRODUCTION

As the development of Electronic Commerce and World Wide Web, recommendation system becomes more and more popular. The task of recommendation system is to estimate preference or ratings of a user on items that have not been seen by the user.

There has been a lot of works done both in industry and academia on the development of recommendation system in decades. Most of these works focus on developing new methods of recommending items to users, such as recommending movies to Web site visitors or recommending customers for books [1]. Current recommendation methods are usually classified into collaborative filtering (e.g., [10] [19]), content-based (e.g., [18] [16]) and hybrid methods (e.g., [15] [3] [13]) based on their recommendation mechanisms. These methods try to find out a user's favorite patterns and recommend items that are similar to the user's favor patterns; or find out a group of users who have similar favor patterns with the user and recommend items that other users in the same group like to the user. Although these recommendation methods are widely used in E-Commerce,

there is still a need of improving recommendation quality. People require recommendation systems to predict users' preferences or ratings as more accurately as possible. More preferences or ratings predicted by recommendation systems are as the same as the actual preferences or ratings given by users, the better the recommendation systems.

In cognitive psychology, psychologists find that objects (items) have different typicality degrees in concepts in real life [7]. For instance, a sparrow is more typical than a penguin in the concept 'bird'. People are more interested in typical objects in a concept than atypical ones [17]. For example, at most times, while a user retrieves the information about clothing, the information of pants and shirts is considered to be more relevant than that of belts and gloves. This is because pants and shirts are more typical than belts and gloves in people's mind [4]. The Measurement of typicality provides a mechanism to rank objects in a way that is closer to human thinking and feeling. For the reason that recommendation systems try to predict users' preferences on items, we consider that it is necessary and helpful to take how people thinking and feeling objects in concepts into consideration to improve the recommendation systems.

In this paper, we propose a novel typicality-based recommendation method. A distinct feature of the typicality-based recommendation method is that it predicts ratings based on user typicality and item typicality, which makes it differ from previous methods. We conduct experiments to validate the proposed method and compare it with previous methods. Experiments show that our method can improve the recommendation quality.

## 2. RELATED WORK

### 2.1 Prototype View and Object Typicality

According to the study of cognitive psychology, object typicality is different from object membership. the former is a measure of the goodness degrees of objects as exemplars in a concept [7], and the later is a measure of degrees of objects belonging to a concept. Psychologists find that people generally are more interested in typical objects than atypical ones in concepts [17]. In the prototype view [14], a concept is represented by a best prototype or a property list which has all the salient properties of the objects that are classified to this concept. An object is considered more typical in a concept if it is more similar to the prototype of the concept. Vanpaemel et al. [23] propose a model which extends the prototype view with multi-prototypes. Cai and Leung [6] [5] propose a method to measure object typicality in ontology.

### 2.2 Recommendation System

There has been many works on recommendation systems and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.  
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...\$10.00.

most of these works focus on developing new methods of recommending items to users, e.g., works in [13] [24]. Currently, recommendation methods are mainly classified into collaborative filtering (CF), content-based (CB) and hybrid methods [1].

**Content-based Recommendation Systems.** The inspiration of this kind recommendation methods comes from the fact that people had their subjective evaluations on some items in the past and will have the similar evaluations on other similar items in the future.

The descriptions of items are analyzed to identify interesting items for users in content-based recommendation systems. Based on items that a user has rated, a content-based recommender learns a profile of the user's interests or preferences. According to a user's interest profile, the items which is similar to the ones that the user preferred or rated high in the past will be recommended to the user. For content-based recommendation systems, it is important to learn the profiles of users. Various learning approaches have been applied to construct profiles of users. For example, Pazzani and Billsus [18] use naive Bayesian classifier to classify unrated Web pages as 'relevant' or 'irrelevant' for users. Mooney and Roy [16] adopt text categorization methods in LIBRA system to recommend books.

**Collaborative Filtering Recommendation Systems.** This kind recommendation methods predict the preferences of active users on items based on the preferences of other similar users or items. For the reason that collaborative filtering methods do not require well-structured item descriptions, they are more often implemented than content-based methods [1] and many collaborative systems are developed in academia and industry. For example, GroupLens [10], PHOAKS [22] and so on. There are two kinds of collaborative filtering methods which are user-based CF approach and item-based CF approach [1].

The basic idea of user-based CF approach is to provide recommendations on an item for a user based on the opinions of other like-minded users on that item. The user-based CF approach first finds out a set of nearest 'neighbors' (similar users) for each user, and these neighbors share similar favorites or interests with the corresponding user. It tries to predict the favorites or rating of an unrated item for a particular user based on ratings given by the user's neighbors on the item. For example, Aggarwal et al. [2] present a graph-based technique in which nodes are users, and edges between nodes indicate users' similarity for collaborative filtering. Herlocker et al. [8] present an algorithmic framework for performing collaborative filtering and new algorithmic elements which can increase the accuracy of prediction algorithms. Sarwar et al. [19] discuss different techniques for measuring item similarity and obtaining recommendations for item-based CF.

**Hybrid Recommendation Systems.** Several recommendation systems (e.g., [15] and [20]) use a hybrid approach by combining collaborative and content-based methods, which helps to avoid some limitations of content-based and collaborative systems. A naive hybrid approach is to implement collaborative and content-based methods separately, and then combine their predictions by a combining function, such as a linear combination of ratings or a voting scheme or other metrics. Melville et al. [15] use a content-based method to augment the rating matrix and then use a collaborative filtering method for recommendation. Fab [3] is a hybrid recommendation system combining the content-based and collaborative methods to exploit the advantages of the two approaches and avoid their shortcomings.

Some hybrid recommendation systems combine item-based CF and user-based CF. For example, Xue et al. [25] propose a cluster-based Pearson Correlation Coefficient method (SCBPCC). Wang et al. [24] use similarity fusion to unify user-based CF and item-

based CF. Ma et al. [13] propose an effective missing data prediction (EMDP) for combining item-based CF and user-based CF. Recently, Li et al. [12] propose a transfer learning-based recommendation method by using cross-domain information.

### 3. TYPICALITY-BASED RECOMMENDATION

In current recommendation system, there are a set of users denoted by  $U$ , and a set of items denoted by  $O$ . Each item can be represented by some properties. For example, directors, actors and keywords are properties used to describe movies. We use an **item property vector** to represent an item.

**Definition 1.** The **item property vector**  $\vec{p}_a$  of an item  $a$  is a vector of property:value pairs.

$$\vec{p}_a = (p_{a,1} : l_{a,1}, p_{a,2} : l_{a,2}, \dots, p_{a,k} : l_{a,k})$$

where  $k$  is the number of properties of the item in the domain, and  $l_{a,i}$  is a real number between 0 and 1, which indicates the fuzzy degree to which item  $a$  possesses the property  $p_{a,i}$ . That  $l_{a,i} = 0$  means item  $a$  does not possess property  $p_{a,i}$  at all and that  $l_{a,i} = 1$  means  $a$  definitely possesses  $p_{a,i}$ . For example, a laptop  $a$  is represented as following:

$$\vec{p}_a = ('has intelCPU':1, \dots, 'has wideScreen':0.8)$$

For all items, we can cluster them into several fuzzy clusters and each cluster is considered as a group of similar items or a kind of similar items. Items belong to each cluster to some degrees. For example, while we use content keywords as properties to describe movies, they can be clustered into Action movies, War movies and so on. Each movie belongs to each kind of movies to a degree. We name each cluster of similar objects as **item group**. Each item group is considered as a concept and items have different typicality degrees in item groups.

**Definition 2.** An **item group** denoted by  $k_i$  is a fuzzy cluster of similar objects as following:

$$k_i = \{O_1^{w_{i,1}}, O_2^{w_{i,2}}, \dots, O_m^{w_{i,m}}\}$$

where  $m$  is the number of items,  $O_x$  is an item and  $w_{i,x}$  is typicality degree of item  $O_x$  in the cluster  $k_i$ .

Items are related to different item group to some degrees (have different typicality degrees in different item groups). For example, the movie 'Titanic' is more related to Romance movie and also a sad tragedy, but little related to a Action movie for the reason that there is just few action scenes in the movie.

According to [6] [17], each item group can be represented by a single prototype extracted from it. A prototype of a concept is an abstract representative of a group of similar instances and is represented by a property vector of the prototype. Such a representative is extracted from instances of the corresponding group of similar instances.<sup>1</sup> We define a **prototype property vector** to represent an item as follows.

**Definition 3.** The **prototype property vector**  $\vec{t}_{k_j}$  of the prototype of item group  $k_j$  is a vector of property:value pairs.

$$\vec{t}_{k_j} = (p_{k_j,1} : r_{k_j,1}, p_{k_j,2} : r_{k_j,2}, \dots, p_{k_j,m} : r_{k_j,m})$$

where  $m$  is the number of the properties of the prototype of concept (item group)  $k_j$ , and  $r_{k_j,i}$  is a real number between 0 and 1, which indicates the fuzzy degree to which the prototype of concept  $k_j$  possesses the property  $p_{k_j,i}$ . That  $r_{k_j,i} = 0$  means concept  $k_j$

<sup>1</sup>Generally, the representative object of a cluster is considered as the mean or the median or the mode of all objects in the cluster [11] [17].

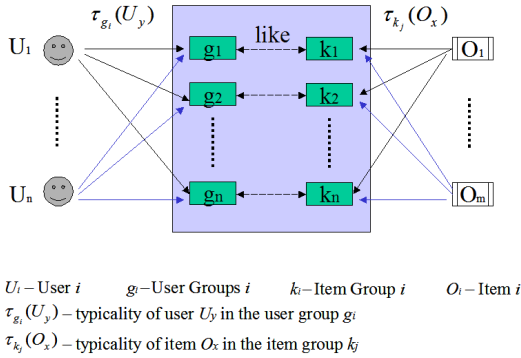
does not possess the property  $p_{k_j,i}$  at all and that  $r_{k_j,i} = 1$  means  $k_j$  definitely possesses  $p_{k_j,i}$ .

For users, they may have their specific favorites or interests on some kinds of items (i.e., item groups). For instance, Bob may be interesting in War movies and Action movies while his wife Amy likes Romance movies and their boy Tom likes Animation movies. For each item group  $k_i$ , we can find a fuzzy set of users who like the corresponding item group and users like such a kind of items to some degrees. For example, Bob like War movies and Action movies to a higher degree and like Romance movies to a lower degree. We name such a fuzzy set of users corresponding to an item group  $k_i$  as a **user group** for  $k_i$ , and consider it as a fuzzy concept ‘users who like the items in the particular item group  $k_i$ ’ which is denoted by  $g_i$ . Users have different typicality degrees in each  $g_i$ .

**Definition 4.** An **user group**  $g_i$  is a fuzzy set of users who like the corresponding item group  $k_i$  as following:

$$g_i = \{U_1^{v_{i,1}}, U_2^{v_{i,2}}, \dots, U_m^{v_{i,m}}\}$$

where  $m$  is the number of users,  $U_x$  is a user and  $v_{i,x}$  is typicality degree of user  $U_x$  in user group  $g_i$ .



**Figure 1: The relations among items, item groups, users, user groups in our method**

There are four roles in our recommendation system, which are items, users, item groups and user groups. For each item, it has different typicality degrees in different item groups. Similarly, a user has different typicality degrees in different user groups. There is a corresponding user group  $g_i$  for each item group  $k_i$ , in which users like the items in  $k_i$ .

According to studies in cognitive psychology, people will be more interested in typical objects than atypical ones in a concept. The principle and the main idea of recommending items to users based on object typicality in our method is as following: For all items, we can obtain several item groups based on some clustering methods; when recommending items to users, if an item  $O_i$  is a more typical object in an item group  $k_x$ , and a user  $U_j$  is a more typical in the corresponding user group  $g_x$  which is considered as a fuzzy concept of ‘users who like the items in the particular item group  $k_x$ ’, then we should recommend the item  $O_i$  to the user  $U_j$  with a higher recommendation score. Figure 1 shows the relations among items, item groups, users and user groups in our method. For example, ‘Titanic’ is a very typical Romance movie and Amy is a very typical person who like Romance movies, then we will recommend ‘Titanic’ to Amy with a very high recommendation score. Thus, the recommendation score of an item  $O_i$  for a user  $U_j$  depends on how typical  $O_i$  is in each item group and how typical  $U_j$

is in each user group. If  $O_i$  is typical in some item groups, and  $U_j$  is also typical in the corresponding user groups for the item groups in which  $O_i$  is typical, then the recommendation score of  $O_i$  for  $U_j$  will be high, and vice versa. The recommendation score of  $O_i$  for  $U_j$  is given by a function which is denoted by  $RS(\vec{O}_i, \vec{U}_j)$ .

$$RS : O \times U \rightarrow [0, 1]$$

where  $O$  is the set of item typicality vectors and  $U$  is the set of user typicality vectors.

We present a  $RS$  function for calculating the recommendation score of item  $O_i$  for user  $U_j$  as following:

$$RS(\vec{O}_i, \vec{U}_j) = \frac{\sum_{x=1}^n w_{x,i} \cdot v_{x,j}}{n} \quad (1)$$

where  $n$  is the number of item groups (also the number of user groups),  $w_{x,i}$  is the typicality degree of item  $O_i$  in item group  $k_x$  and  $v_{x,j}$  is the typicality degree of user  $U_j$  in user group  $g_x$  which is corresponding to  $k_x$ . Please refer to [6] for object typicality measurement.

## 4. EVALUATION

To evaluate our recommendation method, we use the MovieLens data set in the experiments, and this data set is widely used in previous papers such as [19]. MovieLens data set we used contains 100,000 ratings, assigned by 943 users on 1682 movies. Each user has rated at least 20 movies, and the ratings follow the 1(bad)-5(excellent) numerical scale. The sparsity level of the data set is  $1 - \frac{100,000}{943 \times 1682}$ , which is 0.9369. We extract keywords of movies from the Internet Movie Database (IMDB)<sup>2</sup>, and consider these keywords as the descriptions of movies. Each keyword is as a property of the movies.

To measure statistical accuracy we use the *mean absolute error* (MAE) metric defined as the average absolute difference between predicted ratings and actual ratings [1]. The MAE is computed by average the all summing of the absolute errors of the  $n$  corresponding ratings-prediction pairs, and formally it is as following:

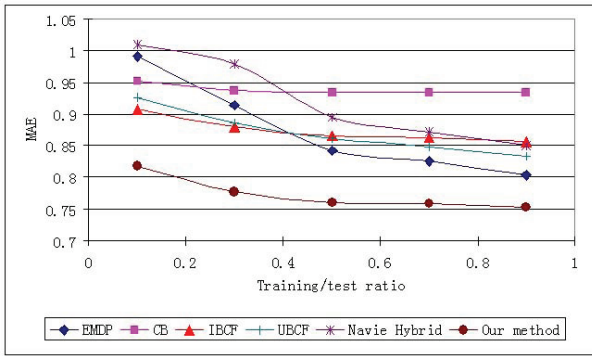
$$MAE = \frac{\sum_{i=1}^n |f_i - h_i|}{n}$$

where  $n$  is the number of rating-prediction pairs,  $f_i$  is a true user-specified rating on an item and  $h_i$  is the prediction for a user on an item given by the recommendation system. If the MAE is lower, it means that the recommendation method can predict users’ ratings more accurately.

In our experiments, we use the Topic Model-based clustering [21] [9] to cluster the movies with keywords. Based on the clustering result, we can extract a prototype for each item group (i.e., movie cluster). The prototype of each cluster is a set of keywords which are used to describe such a kind of movies, and we use top 20 frequently used keywords in a movie cluster as the prototype to represent the movie cluster.

We adopt several classic recommendation methods which are content-based (CB) method with cosine similarity function, user-based collaborative filtering with Pearson Correlation Coefficient (UBCF), item-based collaborative filtering with Pearson Correlation Coefficient (IBCF), naive hybrid method, and a CF method with effective missing data prediction (EMDP) in [13] for the comparison. In our experiments, we divide the whole data set into two parts, one is training set and the other one is test set. We obtain the recommendation prediction based on training set and use test

<sup>2</sup><http://www.imdb.com/interfaces>



**Figure 2: Comparison of Typicality-based method with baseline methods on MAE**

set to evaluate the accuracy of our recommendation method. Besides, we try to test the sensitivity of different scales of training set and test set on recommendation results. In [19], a variable named training/test ratio denoted by  $x$  is introduced, which is used to determine what percentage of data is used as training and test sets. A value of  $x = 0.8$  means 80% of data is used as training set and other 20% of data is used as test set. We randomly choose user-movie-rating tuples to form training and test sets for different training/test ratios. We conduct a 5-fold cross validation and take the average of the MAE.

Figure 2 shows the comparison results of typicality-based recommendation method with baseline methods with different training/test ratios on MAE. According to figure 2, we can find that our typicality-based recommendation method outperforms all other five current methods in all training/test ratios on MAE. For example, for training/test ratio  $x = 0.9$ , the MAE of typicality-based method is 0.7529 while those of EMDP (the second best results when  $x = 0.9$ ) is 0.8040; for  $x = 0.3$ , the MAE of typicality-based method is 0.7774 while that of IBCF (the second best results when  $x = 0.3$ ) is 0.8803. It clearly shows that typicality-based recommendation method has better recommendation quality than all compared methods.

## 5. CONCLUSION

In this paper, we investigate the recommendation system from a new perspective and present a novel recommendation method based on object typicality. A distinct feature of the typicality-based recommendation method is that it predicts ratings based on user typicality and item typicality. The higher typicality degrees of users and items in the corresponding user and item groups, the higher recommendation scores. To the best of our knowledge, it is the first work which applies typicality to handle recommendation problem. Evaluation experiments show that the typicality-based method outperforms previous recommendation methods on recommendation quality.

## 6. ACKNOWLEDGEMENT

The research described in this paper has been supported primarily by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. CityU 117608), and partially by a grant from City University of Hong Kong (Project No.7008043). It is also partially supported by a CUHK Direct Grant for Research. Juanzi Li and Jie Tang are supported by the Natural Science Foundation of China (No. 60703059, No. 60973102), Chinese National

Key Foundation Research (No. 60933013), National High-tech R&D Program (No. 2009AA01Z138).

## 7. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
- [2] C. C. Aggarwal, J. L. Wolf, K.-L. Wu, and P. S. Yu. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In *KDD '99*, pages 201–212, New York, NY, USA, 1999. ACM.
- [3] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.
- [4] L. W. Barsalou. Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(4):629–654, October 1985.
- [5] Y. Cai and H. fung Leung. Formalizing object typicality in context-aware ontology. In *ICTAI (2)*, pages 233–240. IEEE Computer Society, 2008.
- [6] Y. Cai and H. f. Leung. Multi-prototype concept and object typicality in ontology. In *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*, pages 470–475. AAAI Press, 2008.
- [7] K. M. Galotti. *Cognitive Psychology In and Out of the Laboratory*. Belmont, CA: Wadsworth, third edition, 2004.
- [8] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99*, pages 230–237, New York, NY, USA, 1999. ACM Press.
- [9] T. Hofmann. Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAI'99*, pages 289–296, 1999.
- [10] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
- [11] M.-J. Lesot, L. Mouillet, and B. Bouchon-Meunier. Fuzzy prototypes based on typicality degrees. In *Proc. of the 8th Fuzzy Days'04*. Springer, 2005.
- [12] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In C. Boutilier, editor, *IJCAI*, pages 2052–2057, 2009.
- [13] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46, New York, NY, USA, 2007. ACM.
- [14] D. L. Medin and E. E. Smith. Concepts and concept formation. In *Annual Review of Psychology*, volume 35, pages 113–138, 1984.
- [15] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.
- [16] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, New York, NY, USA, 2000. ACM.
- [17] G. L. Murphy. *The big book of concepts*. MIT Press, 2002.
- [18] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Mach. Learn.*, 27(3):313–331, 1997.
- [19] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.
- [20] I. M. Soboroff and C. K. Nicholas. Combining content and collaboration in text filtering. In *In Proceedings of the IJCAI'99 Workshop on Machine Learning for Information Filtering*, pages 86–91, 1999.
- [21] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD '08*, pages 990–998, New York, NY, USA, 2008. ACM.
- [22] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. Phoaks: a system for sharing recommendations. *Commun. ACM*, 40(3):59–62, 1997.
- [23] W. Vanpaemel, G. Storms, and B. Ons. A varying abstraction model for categorization. In *CogSci2005*, pages 2277–2282, Mahwah, NJ, 2005. Lawrence Erlbaum.
- [24] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, USA, 2006. ACM.
- [25] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*.