Information Fusion 28 (2016) 75–82

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

# Efficient recommendation methods using category experts for a large dataset

Won-Seok Hwang [a], Ho-Jong Lee [a,*], Sang-Wook Kim [a,*], Youngjoon Won [a], Min-soo Lee [b]

[a] Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 133-791, Republic of Korea
[b] Ewha Womans University, 11-1 Daehyun-Dong, Seodaemoon-Ku, Seoul 120-750, Republic of Korea

ARTICLE INFO

*Article history:*
Available online 30 July 2015

*Keywords:*
Recommender system
Collaborative filtering
Expert
Performance evaluation

ABSTRACT

Neighborhood-based methods have been proposed to satisfy both the performance and accuracy in recommendation systems. It is difficult, however, to satisfy them together because there is a tradeoff between them especially in a big data environment. In this paper, we present a novel method, called a CE method, using the notion of *category experts* in order to leverage the tradeoff between performance and accuracy. The CE method selects a few users as experts in each category and uses their ratings rather than ordinary neighbors'. In addition, we suggest CES and CEP methods, variants of the CE method, to achieve higher accuracy. The CES method considers the similarity between the active user and category expert in ratings prediction, and the CEP method utilizes the active user's preference (interest) on each category. Finally, we combine all the approaches to create a CESP method, considering similarity and preference simultaneously. Using real-world datasets from MovieLens and Ciao, we show that our proposal successfully leverages the tradeoff between the performance and accuracy and outperforms existing neighborhood-based recommendation methods in coverage. More specifically, the CESP method provides 5% improved accuracy compared to the item-based method while performing 9 times faster than the user-based method.



## 1. Introduction

As the number of online items significantly grows, it becomes a difficult task for users to find items on their own. Good matching of users to suitable items is critical to enhance user satisfaction. It highlights the importance of *recommendation systems*, which automatically suggest items with which a user would be satisfied [1,2]. Previous recommendation systems are often based on *neighborhood-based methods* (NBMs) [3], which predict a rating of an item that an *active user* has not evaluated yet and suggest his/her preferable items based on the predicted ratings. NBMs have several advantages of simplicity, justifiability, efficiency, stability, and clear reasoning behind recommendations [4,5]. It finds the neighbors who rate the items in a range close to the active user and then predicts the rating on a target item by using the ratings of those neighbors [6]. Some NBMs search for the neighbors at every recommendation request to exploit the *latest ratings*. This approach enables to find the users whose preferences are more similar to that of an active user because it uses the latest ratings in finding neighbors. However, it suffers from a high execution time for reflecting latest ratings dynamically, which is more serious in a big data environment. On the other hand, another group of NBMs pre-compute similarities or pre-builds the models by ignoring the latest ratings just for reducing the execution time. However, those NBMs would have an accuracy problem because they ignore a portion of ratings (i.e., the latest ratings). Thus, NBMs have a *tradeoff between performance and accuracy*.

In this paper, we propose new recommendation methods based on 'category experts' rather than neighbors of NBMs. In the real world, users have a tendency to trust experts' opinion, so preferences prediction with the experts could be accurate [7,8]. We define *category experts* as the top-*k* users in giving ratings in a certain category. *k*, the number of experts in a category, is determined by the empirical analysis on each category rating. The CE method, one of the proposed methods aggregates the ratings given by the experts to predict ratings that the user will give to unevaluated items. We also extend the CE method by exploiting two metrics to improve the accuracy: 'similarity' between users and category experts and 'category interest,' defining the degree of interest of the users.

The CE method improves the performance significantly because the experts can be easily identified and maintained by counting the number of ratings given by each user. Also, its accuracy is not

* Corresponding authors.
*E-mail addresses:* hojonglee@agape.hanyang.ac.kr (H.-J. Lee), wook@hanyang.ac.kr (S.-W. Kim).

http://dx.doi.org/10.1016/j.inffus.2015.07.005
1566-2535/

shown worse than the existing methods since people want to look for advice from experts of specific fields in the real world. Thus, using the category experts leverages the tradeoff between performance and accuracy. The CE method and its extensions improve the coverage, a portion of unseen items that are actually predictable, because category experts tend to have rated more items than similar users. For evaluation, we used the datasets from MovieLens and Ciao [9,10]. We analyzed the effect with different numbers of category experts and compared the results with those of various NBMs, such as user-based, item-based, and *k*-Means clustering based methods, in terms of accuracy, performance, scalability, and coverage.

The organization of our paper is as follows. Section 2 presents the related work. Section 3 proposes the CE method and its extensions. Section 4 deals with the performance and accuracy issues with the real-world datasets. Finally, we conclude the paper in Section 5.

## 2. Related work

NBMs are common techniques for collaborative filtering (CF) [11,12]. The user-based method (UBM) looks for $k$ users who have a similar preference with the active user and predicts the rating of an active user on a target item from the ratings given by $k$ users. Its similarity can be calculated via Pearson's correlation coefficient, cosine similarity, or the extended generalized vector-space model [13]. The predicted rating $p_{u,i}$ of item $i$ for user $u$ is computed by $p_{u,i} = \bar{r}_u + \frac{\sum_{v \in S_u} (r_{v,i} - \bar{r}_v) \times |sim(u,v)|}{\sum_{v \in S_u} |sim(u,v)|}$ where $r_{v,i}$ is the rating by user $v$ on item $i$ and $\bar{r}_u$ is the average of all the ratings assigned by an active user $u$. $sim(u, v)$ is the similarity between user $u$ and $v$, and $S_u$ is the set of $k$ users (neighbors) that are the most similar to active user $u$. The similarities between the active user and others are calculated online upon a recommendation request. Herlocker et al. [12] analyzes the accuracy of UBM with respect to the selected neighbors having similar tastes. A few others [14,15] select their neighbors from social network relations. To identify a closer neighborhood, the following approaches also have been proposed: Default voting, inverse user frequency, case amplification [16], and weighted-majority prediction [17,18]. UBM can provide accurate recommendations because it considers all ratings including the latest ratings for searching neighbors. The latest ratings are important because most users assign ratings to only a few items [19–21].

The clustering-based methods (CBM) [22,23] improve the performance by determining the neighbors of each user before the recommendation request. Several CBM variations use various clustering models [24–27] to classify the users off-line. They make user clusters before the recommendation request, and then predict the active user's rating of item $i$ based on the ratings given by those users who belong to the group (i.e., cluster) that the active user belongs to. In addition, there are other model-based methods that use the Bayesian model [28] and the latent semantic model [29] other than clustering models. The item-based methods (IBM) were proposed to overcome a performance problem by pre-calculating similarities among all items [30,31]. It calculates the similarity of items rather than that of users because the relationships between items are *relatively static* [31]. It predicts the active user's rating on a target item by referencing his/her ratings on those items similar to the target item. The predicted rating $p_{u,i}$ assigned by active user $u$ on target item $i$ can be calculated as $p_{u,i} = \frac{\sum_{j \in S_i} \{r_{u,j} \times |sim(i,j)|\}}{\sum_{j \in S_i} |sim(i,j)|}$ where $sim(i, j)$ represents the similarity between user $i$ and $j$, and $S_i$ is the set of items that are similar to item $i$. However, accuracy of CBM and IBM may decrease as time goes by because they exclude the latest ratings after the neighborhood is determined.

There are several previous methods considering experts to improve the accuracy. Amatrin et al. [7] utilize the ratings given by experts in Rotten Tomatoes in order to recommend to Netflix users. Yun et al. [32] also crawl the expert ratings from Rotten Tomatoes and predicts the ratings for users in the other domain via the SVD-based model [33]. However, these methods showed an additional overhead for crawling the expert ratings from other domains like Rotten Tomatoes. If there are no pre-defined experts, they cannot provide recommendations.

Cho et al. [8] define experts as the users who evaluate a lot of items. It predicts the rating on the target item by considering ratings from both experts and neighbors who have a similar taste to the active user. Thus, it suffers from long execution time of searching for experts and neighbors. Liu et al. [34] introduce 'star' users, which are close to experts. The star users are virtual users who represent interests of whole users, and a training algorithm selects the star users through analyzing their ratings.

Pham et al. [35] propose an expert-based method for interactive recommendation systems. This method defines experts for a given active user and his/her attributes (i.e., genre, actor, and director). To define the experts, it needs to analyze the attributes as well as correlation among users. Pham et al. [36] also propose another expert-based method that corrects the users' ratings based on experts' opinions for more accurate recommendations. In order to find the experts, the method uses a ⟨A,V⟩-SPEAR algorithm that refers to an ontology built based on items' attributes. To our knowledge, the existing expert-based recommendation methods ignore the performance issue. In this paper, we propose methods based on category experts instead of neighbors to balance the tradeoff between performance and accuracy of NBMs.

## 3. Category expert methods

This section proposes four recommendation methods using category experts. The *category expert* is a user who is considered to understand well the overall items in a specific category. In this paper, the decision of whether or not a user understands well the item is based on whether or not the user has evaluated the item. The reasoning behind this is, in order to evaluate an item, the user needs to know the item well. Therefore, a user who has evaluated many items in a specific category can be considered as *knowledgeable about the category*, thereby defined as a *category expert*. We formally define the category expert as follows:

**Definition 1.** For category $c$, we define the category experts $E_c$ such that:

$$|I_u| \leqslant |I_v| \ (\forall u \in U - E_c, \forall v \in E_c)$$

where $|I_u|$ indicates the number of items that user $u$ has evaluated. In order to determine the category experts, we compute the number of items in each category evaluated by each user. It is easy to maintain those numbers *incrementally* because we just need to increment the number whenever a user evaluates an item. For this reason, we can reduce dramatically the performance of NBMs by decreasing the effort for finding neighbors (i.e., category experts).

**CE** method: In this paper, we denote the recommendation method that uses category experts as a *CE* method. The CE method predicts the rating with which the active user would assign to the target item, based on the ratings given by the category experts. The intuition behind this is that a user trusts the opinions of experts even if he/she has somewhat different preference with the category experts. The predicted rating for user $u$ on item $i$ included in category $c$ is denoted as $p_{u,i,c}$.