Expert Systems with Applications 42 (2015) 3801–3812

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Enhancing memory-based collaborative filtering for group recommender systems

Sarik Ghazarian *, Mohammad Ali Nematbakhsh

Software Engineering Department, University of Isfahan, Isfahan, Iran

ARTICLE INFO

Article history:
Available online 29 November 2014

Keywords:
Group recommendation system
Sparsity problem
Collaborative filtering technique
User-based approach
Item-based approach

ABSTRACT

Memory-based collaborating filtering techniques are widely used in recommender systems. They are based on full initial ratings in a user-item matrix. However, most of the time in group recommender systems, this matrix is sparse and users' preferences are unknown. This deficiency may make memory-based collaborative filtering unsuitable for group recommender systems. This paper, improves memory-based techniques for group recommendation systems by resolving the data sparsity problem. The core of the proposed method is based on a support vector machine learning model that computes similarities between items. This method employs calculated similarities and enhances basic memory-based techniques. Experiments demonstrate that the proposed method overcomes the memory-based techniques. It also indicates that the presented work outperforms the latent factor approach, which is very efficient in sparse conditions. Finally, it is indicated that the proposed method gives a better performance than existing approaches on generating group recommendations.

1. Introduction

Over the past years, Individual Recommender Systems (IRSs) have received a lot of attention in research and business communities. They compute the probability of individual's interests using his/her past activities or preferences. In some domains that users are together to carry out an activity as a group, another type of recommender systems called group recommender systems (GRSs) play a major role (Amer-Yahia, Roy, Chawlat, Das, & Yu, 2009; Chen, Cheng, & Chuang, 2008). Some of well-known GRSs, have been developed and used in the last years. MusicFX selects a music station to be played in a fitness center according to audience interests (McCarthy & Anagnost, 1998). Adaptive Radio broadcasts the selected music to users' computers who are present in an environment (Chao, Balthrop, & Forrest, 2005). PolyLens is a Group Recommender (GR) version of MovieLens which is a movie recommender system (O'connor, Cosley, Konstan, & Riedl, 2001). CATS helps a group of friends to decide where to spend their skiing vacation (McCarthy et al., 2006). Intrigue finds suitable sightseeing destination for tourist groups (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003). Unlike IRSs that are in contact with just one user, GRSs are associated with many users that are different in taste and preference. This creates a more complex structure for GRSs. This structure is consisted of users who may or may not share similar preferences, their interests and the way to recommend the best items to the group. The main goal in this systems is to recommend items that would satisfy all users' need as much as possible (Berkovsky & Freyne, 2010). To this end, it is essential to have all group members' preferences and their ratings on the items in order to aggregate their opinions and make recommendation to the whole group (Amer-Yahia et al., 2009; Jameson & Smyth, 2007). In general, it can be said that these systems need to analyze preferences of all group members and attempt to find the most appropriate recommendation for the group to fairly satisfy every member. There are two major strategies for group recommendation: (1) aggregation of individual ratings and (2) aggregation of individual recommendation lists (Christensen & Schiaffino, 2011). In the first approach, for each candidate item, all individual preferences are aggregated by different aggregation functions like average or least misery to compute item's group rating, after that items with the highest group ratings are recommended. While, in the second approach, recommended lists for each member are merged into a single list in order to recommend to a group. Similarly, in both approaches, it is necessary to predict unknown ratings in each member's preference list.

Memory-based collaborative filtering (CF) is a common approach for GRSs, but it has a weakness in fulfilling sparse preference matrices (Huang, Chen, & Zeng, 2004; Sarwar, Karypis,

* Corresponding author. Tel.: +98 913 1074073; fax: +98 313 7932670.
E-mail addresses: sarikghazarian@yahoo.com (S. Ghazarian), mnematbakhsh@eng.ui.ac.ir (M.A. Nematbakhsh).

http://dx.doi.org/10.1016/j.eswa.2014.11.042
0957-4174/

Konstan, & Riedl, 2000; Su & Khoshgoftaar, 2009; Su, Khoshgoftaar, Zhu, Greiner, 2008). Several researchers have attempted to unravel this problem by default voting value approach. This approach fills all unseen preferences with a default value like neutral or average ones (Baltrunas, Makcinskas, & Ricci, 2010; Breese, Heckerman, & Kadie, 1998). Other approaches to overcome this issue are imputation-boosted techniques which use classification algorithms or imputation techniques like mean imputation to fill in unknown ratings (Su, Khoshgoftaar, Zhu, et al., 2008). The major drawback to these researches is that they fill all missing ratings with constant values that are far from the reality and do not consider the actual variances.

This paper attempts to recommending music to a group of randomly presented users who have rated few items. For aggregating members' ratings to make group recommendations, memory-based CF technique is extended to be used efficiently in sparse data situations. One of the major components of our model is a learning model called support vector machine (SVM) to predict similarities between items. Our method uses these obtained values in calculating similarities between users and making predictions. By these approaches, the presented method endeavors to solve the sparsity problem. The main idea is to concentrate on available information as far as possible and pale the limitations in using basic memory-based methods. Unlike previously mentioned approaches that have improved sparsity in IRSs, our method attempts to solve the sparsity problem in GRSs.

The rest of this article is organized as follows: Section 2 presents related works in solving sparsity problem. Section 3 describes the methods that have been used as a basis of comparisons. Section 4 explains the proposed method with details. Section 5 provides experiments on the presented method and compares its accuracy with primary approaches. Finally, Section 6 concludes the paper.

2. Related work

GRSs assist group of users to find the most relevant and suitable items for them based on the existing information about group's taste (Amer-Yahia et al., 2009; Chen et al., 2008; Kim, Kim, Oh, & Ryu, 2010). These systems are expansion of IRSs, which address their recommendations to single users (Amer-Yahia et al., 2009; Berkovsky & Freyne, 2010; Jameson & Smyth, 2007). In many scenarios, presence of GRSs is very important: recommending a TV program to a family in order to watch together, playing music for different users in an environment and choosing the best restaurant for colleagues going out to dine (McCarthy, 2002; McCarthy & Anagnost, 1998; Yu, Zhou, Hao, & Gu, 2006). GRSs face with more challenges than IRSs. Jameson and Smyth (2007) categorized GRS's challenges in four subtasks: getting information about members' preferences, generating recommendations, explaining recommendations and helping users to reach consensus. The first challenge is the first requirement to aggregating users' tastes and recommending the best items for the group. For generating group recommendations there are two main approaches which are: (1) aggregating users' preferences, (2) merging users' individual recommendation lists (Jameson & Smyth, 2007). The first approach aggregates all group members' ratings for each item, and next recommends the items with the highest group ratings. Among all available methods for aggregation process, least misery and average methods are more popularly used (Amer-Yahia et al., 2009; Gartrell et al., 2010; Quijano-Sánchez, Recio-García, & Díaz-Agudo, 2011; Recio-Garcia, Jimenez-Diaz, Sanchez-Ruiz, & Diaz-Agudo, 2009). While second one first computes recommendation lists for every user, then merges all users' recommendation lists and recommends items in the final merging list. It is obvious that in both approaches before the aggregation process, all ratings in the user-item matrix should be known.

There are different researches on group recommendations that have been done to filing all unknown ratings. Chen et al. (2008) predicted item's group rating by using similar items and the subgroups' ratings on those items. They supposed that user-item matrix has enough information and they used it to filling group ratings on the items. In other words, their research supposed group ratings are sparse rather than user's ratings. Dery, Kalech, Rokach, and Shapira (2010) used the minimum required information about users' tastes to determine winner item for recommending to a group. For obtaining this item, they asked users about their preferences on the items. In order to minimize the number of queries, they proposed two heuristic methods. In each step, the heuristic method selects the best pairs of user-item that gives the most important information to obtaining winner item. Dery et al. (2010) filled the unknown ratings in user-item matrix by directly asking the users. In our research, we assume that users do not have any direct contact with recommender system.

In most researches, CF which is one of the most favored and popular approaches for prediction process has been used to fill all the cells of the user-item matrix (Breese et al., 1998; Deshpande & Karypis, 2004; Herlocker, Konstan, Borchers, & Riedl, 1999; Su & Khoshgoftaar, 2009). This approach is based on the phenomenon, which says, users who had alike tastes in the past, will have similar tastes in the future. This is equivalent to "word of mouth" notation (Shardanand & Maes, 1995).

CF is divided into two main approaches: memory-based and model-based (Breese et al., 1998). First approach uses partial information of user-item matrix and contains two following steps: (1) calculating similarities between users or items (2) using a weighted average of ratings in order to calculate preference values. Two subcategories of memory-based CF are user-based (Herlocker et al., 1999; Jin, Chai, & Si, 2004; Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994) and item-based (Deshpande & Karypis, 2004; Sarwar, Karypis, Konstan, & Riedl, 2001) approaches, which focus on users and items, respectively. The second main approach of CF is model-based approach, which uses available information about users to learn a model for predicting unknown ratings. For this purpose, there are many machine learning models like clustering, Bayesian network, and SVM that have been used in many researches (Breese et al., 1998; Su & Khoshgoftaar, 2009; Ungar & Foster, 1998).

Sparsity is one of the major inhibitors for not efficiently utilizing memory-based CF approaches for GRSs (Martín-Vicente et al., 2014; Su & Khoshgoftaar, 2009). In sparseness conditions, most cells of the user-item matrix are empty. The reason is that by increasing the number of items, users are unable to rate millions of them. They cannot tell about all their opinions and preferences about all the items (Dery et al., 2010; Grčar, Fortuna, Mladenič, & Grobelnik, 2006; Huang et al., 2004; Perugini, Gonçalves, & Fox, 2004). In this type of matrices, the accuracy of calculated predictions by applying memory-based CF approaches will be low, since there is not enough information about ratings of the users (Huang et al., 2004). Ntoutsi, Stefanidis, Nørvåg, and Kriegel (2012) applied user-based CF approach in order to predict unknown ratings. First, they partitioned users into clusters. Then for predicting a particular item rating for a user, they considered just the ones in the cluster of the target user instead of all the users in the dataset. They calculated the relevancy of an item to a user, based on the relevancy of that item to similar users in the target user's cluster. Moreover, they involved a support score in the prediction process to be shown how many users in the cluster have rated that item. By using memory-based approaches as the basis, this approach also cannot be used in sparse data situations. Lately, Martín-Vicente et al. (2014) mentioned the sparsity problem as the most