Expert Systems With Applications 82 (2017) 115–127

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Does order matter? Effect of order in group recommendation

Akshita Agarwal[1], Manajit Chakraborty[1,*], C. Ravindranath Chowdary[1]

Department of Computer Science & Engineering, Indian Institute of Technology (BHU), Varanasi 221 005, India

ARTICLE INFO

Article history:
Received 11 November 2016
Revised 9 March 2017
Accepted 29 March 2017
Available online 5 April 2017

Keywords:
Recommender systems
Order in recommendation
Group recommendation
Hungarian method
Least Misery

ABSTRACT

Recommendation Systems (RS) are gaining popularity and they are widely used for dealing with information on education, e-commerce, travel planning, entertainment *etc.* Recommender Systems are used to recommend items to user(s) based on the ratings provided by the other users as well as the past preferences of the user(s) under consideration. Given a set of items from a group of users, Group Recommender Systems generate a subset of those items within a given group budget (*i.e.* the number of items to have in the final recommendation). Recommending to a group of users based on the ordered preferences provided by each user is an open problem. By order, we mean that the user provides a set of items that he would like to see in the generated recommendation along with the order in which he would like those items to appear. We design and implement algorithms for computing such group recommendations efficiently. Our system will recommend items based on modified versions of two popular Recommendation strategies– Aggregated Voting and Least Misery. Although the existing versions of Aggregated Voting (*i.e.* Greedy Aggregated Method) and Least Misery perform fairly well in satisfying individuals in a group, they fail to gain significant group satisfaction. Our proposed Hungarian Aggregated Method and Least Misery with Priority improves the overall group satisfaction at the cost of a marginal increase in time complexity. We evaluated the scalability of our algorithms using a real-world dataset. Our experimental results evaluated using a self-established metric substantiates that our approach is significantly efficient.

1. Introduction

Recommender Systems are a subclass of information filtering systems that predict items to a user or a group of users based on their prior preferences. This information can be obtained either explicitly by collecting users' ratings or implicitly by monitoring the users' behavior (Bobadilla, Ortega, Hernando, & Bernal, 2012). Recommender Systems are often classified based on their design or filtering technique. They are namely, Content-based (Lops, De Gemmis, & Semeraro, 2011), Context-based (Adomavicius & Tuzhilin, 2015; Chakraborty, Agrawal, Shekhar, & Chowdary, 2015), Collaborative (Ricci, Rokach, & Shapira, 2011), Demographic (Pazzani, 1999) and Knowledge-based (Trewin, 2000). Group Recommender Systems are a subclass of general Recommender Systems. Here, the emphasis is on satisfying the needs of a group of users rather than individuals. While a number of techniques have been applied to group recommendation ranging from Collaborative Filtering (Baltrunas, Makcinskas, & Ricci, 2010; Ghazarian & Nematbakhsh, 2015; O'connor, Cosley, Konstan, & Riedl, 2001) to Critiquing-based approaches (McCarthy et al., 2006), it still continues to attract the community because of its varied application in Social Networks (Cantador & Castells, 2012; Gartrell et al., 2010), E-commerce websites (Sarwar, Karypis, Konstan, & Riedl, 2000; Schafer, Konstan, & Riedl, 2001) *etc.* In this paper, we first model the user satisfaction measure and show that introducing order in group recommendation has a positive effect. Consequently, we propose a system, which not only takes user preferences into account but also considers the order in which the user likes them to be presented to her. To the best of our knowledge, this is the first attempt that takes order into account for group recommendation. To prove the efficacy of our proposed system, we adopt two widely known consensus functions for recommendation– Aggregated Voting and Least Misery. Additionally, we provide approximations (*Hungarian Aggregated Method* and *Least Misery Method with Priority*) for the above two consensus functions to suit our purposes. While GReedy Aggregated Method (GRAM) aims to select the best item to maximize the satisfaction of the group, it still is far from providing a perfect solution. Instead, Hungarian Aggregated Method (HAM) offers a maximum satisfaction assignment combinatorial solution. On the other hand, while Least Misery Method (LMM) tends to maximize the minimum least satisfaction of the group, we show with

* Corresponding author.
E-mail addresses: akshita.agarwal.cse13@iitbhu.ac.in (A. Agarwal), cmanajit.rs.cse14@iitbhu.ac.in (M. Chakraborty), rchowdary.cse@iitbhu.ac.in (C.R. Chowdary).

[1] All authors contributed equally, names are listed in alphabetical order.

http://dx.doi.org/10.1016/j.eswa.2017.03.069
0957-4174/

Table 1
Example of schedule choices.

Student	Choice of schedule
Student 1	*Computer Science → Mathematics → Physics → Chemistry*
Student 2	*Mathematics → Geography → Computer Science → English*
Student 3	*Physics → Chemistry → Computer Science → Mathematics*
::	:::

the help of experiments that Least Misery Method with Priority (LMMP) provides better group satisfaction (as high as 30%). We provide a detailed analysis of the effect on three essential variables namely, *Group Budget, Group Size* and *Number of items* pertaining to the inclusion of order in group recommendation. Group budget (also known as *space budget* (Amer-Yahia, Roy, Chawlat, Das, & Yu, 2009) is defined as a subset of the items from the total set of items available. It is the maximum number of items to be recommended. A comprehensive set of experiments was conducted on a real-world dataset *MovieLens* (GroupLens, 2015) which corroborate our claim.

1.1. Motivation

Most Group Recommender Systems suggest a list of items to users, based on some *consensus* function. While generating relevant items to user preferences in the recommended list is a prime requirement, the order in which they appear in the list is important as well. It has been observed through experimentation (as explained later in Section 3.2) that when the satisfaction measure is modeled as a similarity function between user's preference list and the generated recommendation list, there exists a positive correlation between the two. We studied this correlation between users' preferences and generated recommendation list and found out that *order* plays a major role in determining user satisfaction when there is a strict group budget. By *order* we mean that the user provides the set of items that he would like to see in the generated recommendation along with the order in which he would like those items to appear. As such, we propose a system which incorporates the notion of *order* in addition to the existing functionality of recommending to users of a group. This is particularly useful in Recommender Systems like *travel planning* where along with providing the preferences the users would also like to visit places in a particular order depending on the convenience, weather, transportation *etc*. The idea can also be used for planning schedules in educational institutions where the students and teachers would like to suggest an order in which their classes are held or the order in which the examinations are conducted.

Example 1. Suppose there are six subjects in the curriculum– *Computer Science, Mathematics, Physics, Chemistry, Geography* and *English* and for a particular day, the students have to choose four subjects to be taught and also suggest the order in which they would like the classes to be held. Some possible proposed choices are listed in Table 1.

The problem of selecting four subjects and arranging them in order in which they would be held based on some consensus function is quite a challenging task especially if the number of students is large. Now, consider similar constraints on a problem which may include many variables, such as recommending an itinerary to a group of travelers. Clearly, the problem complexity increases manifold. Satisfying each and every user is as important as meeting the satisfaction of the whole group. Another situation could be family members planning for a Harry Potter movie series marathon. While some members might like to watch the critically acclaimed "Prisoners of Azkaban" first, another member might like to start with "Goblet of Fire" but then a Harry Potter purist may stress on watching the series chronologically starting with "Philosopher's Stone". Certainly, this is a conundrum and picking the right order that satisfies everybody's mood is a tough choice. Herein, lies the possibility of a recommender system's inclusion of order to accommodate such variance in choices.

1.2. Our contributions

Our contributions are two-fold:

- We posit the idea of introducing order in group recommendation and study its effect on group satisfaction.
- We propose intelligent approximation algorithms for consensus functions to suit our requirements and analyze them thoroughly using three variables– group size, group budget and number of items.

The rest of the paper is organized as follows. Section 2 gives an overview of related works in existing literature. Section 3 offers an insight into our proposed model. The algorithms and their modified versions are presented in Section 4. Section 5 provides the details of experimental setup. Results and Analysis pertaining to experimentation are presented in Section 6. Finally, Section 7 concludes the article.

2. Related Work

2.1. Group Recommendation

Movie recommendation has been an active research area for the past few decades. But, in recent years there has been an inclination towards building group recommender systems for movies (Christensen & Schiaffino, 2011) to cater the needs of families, friends *etc*.. Pera and Ng (2013) build a group recommender for movies by the name of *GroupReM*. The novelty of this approach is that instead of using the conventional approach of Matrix Factorization for movie ratings, this system employs (personal) tags for capturing the contents of movies considered for the recommendation and group members interests. They formulate a three-pronged approach for the same:

- Employ a merging strategy to explore individual group members interests in movies and create a profile that reflects the preferences of the group on movies.
- Use word-correlation factors to find movies similar in content.
- Consider the popularity of movies on a movie website.

Kagita, Pujari, and Padmanabhan (2015) utilizes the virtual user approach for recommendation generation. They make use of transitive precedence relations among items to generate a virtual user profile that represents the combined profile of the group. This strategy has been applied to two different recommendation techniques– Precedence Mining and Collaborative Filtering. Other existing virtual user strategies take into consideration the set of common items consumed by the users whereas the authors' strategy computes the fuzzy score for the items not consumed by all the users, that gives this method better precision and recall over the other virtual user strategies. Kagita et al. (2015) also propose 'monotonicity', *i.e.* the degree to which a recommendation prevails when new information is added to the training set, as a measure for estimating the quality of a Group Recommender System.

Villavicencio, Schiaffino, Diaz-Pace, and Monteserin (2016) present a multi-agent approach, called PUMAS-GR, for group recommendation. The novelty of their approach is that it leverages on negotiation techniques in order to integrate recommendations