Aggregating Top-K Lists in Group Recommendation Using Borda Rule

Sabrine Ben Abdrabbah(✉), Manel Ayadi, Raouia Ayachi, and Nahla Ben Amor

LARODEC, Université de Tunis, ISG Tunis, 2000 Bardo, Tunisia
abidrabbah.sabrine@gmail.com, manel.ayadi@hotmail.com, raouia.ayachi@gmail.com, nahla.benamor@gmx.fr

Abstract. With the democratization of the web, recent works aimed at making recommendations for groups of people to consider the circumstances where the item is selected to be consumed collectively. This paper proposes a group recommender system which is able to support partial rankings of items from different users in the form of top-k lists. In fact, the proposed group recommender system is based on generating recommendation lists for the group members using user-based collaborative filtering, then applying approximation Borda rule to generate group recommendations. Experiments show that the proposed group recommender system using approximate voting rules produced more accurate and interesting recommendations than using the standard voting rules.

Keywords: Group recommender systems · Collaborative filtering · Partial rankings · Voting rules

1 Introduction

Group recommendation is designed to find a trade-off among all the group members' tastes and then, derive the group preference for each item. The popularity of group recommender systems has increased in the last years. In literature, there are two main group recommendation strategies [1] including (i) *preferences aggregation* which consists in combining the group members' prior ratings into virtual user's profile and then generating recommendations and (ii) *recommendations aggregation* which consists in generating the members' individual recommendations using an individual recommendation method, then combining them to return a single recommendation list for the group. In this paper, we are interested in the second category since it typically offers better flexibility [2].

Different aggregation functions have been used to aggregate the individual recommendation lists of group members, we can for instance cite *average* [5], *least misery* [4], *voting rules* [3], etc. These functions usually work with full list of items (i.e. complete linear orders over all possible candidates) and they cannot be able to consider the situations when the orderings may not be total.

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part I, LNAI 10350, pp. 325–334, 2017.
DOI: 10.1007/978-3-319-60042-0_38

For instance, recommendation lists are generally presented in the form of partial rankings/orders of the top-k relevant items of each user out of the set of all the unseen items.

In order to handle partial recommendation lists, we propose to use approximate voting rules adapting standard voting rules to the case of partial rankings (i.e. when the positions of some items are unknown). More precisely, we develop a new group recommender system based on partial voting rules. We start by generating top-k recommendation list for each group member based on *collaborative filtering* approach. Then, we propose to use approximate aggregation methods to combine the individual recommendation lists of group members into a single ranked list that captures the collective preference. Contrary to classical group recommender systems that are based on complete recommendation lists, the proposed recommender system can recommend items which are already seen and appreciated by some group members since it takes into account even the items appearing in at least one recommendation list.

The remaining of this paper is organized as follows. Section 2 gives basic concepts on group recommender systems and voting rules. Section 3 is dedicated to the new proposed group recommendation framework based on partial recommendation lists. Finally, Sect. 4 discusses experimental results.

2 Basic Concepts

This section presents relevant background on group recommender systems and voting rules.

2.1 Group Recommender Systems

Group recommender systems (GRS) have been proposed as an efficient tool to discover group preferences and provide recommendations of items that can better match the group interest and taste [1,3,4]. The main idea of GRS consists in aggregating information from individual user models in order to capture the group model. In literature, there are two main group recommendation strategies:

– *Preference aggregation strategy* consisting in aggregating all users' individual preferences into a single profile representing preferences of all group members on each item. Then, recommendations are generated using a traditional recommender system.
– *Recommendation aggregation strategy* consisting in generating recommendations for each group member using an individual recommender algorithm, then, the recommendation lists are aggregated to produce a single group recommendation list.

In this paper, we are concerned with group recommendations generation using the second strategy i.e. aggregating individual recommendation lists since it is efficient and flexible [2]. Within the most common aggregation methods we can mention: (i) the *Average* function that considers the group preference as the average of all ratings given by group's members per item, (ii) the *Least misery* that considers the group preference as the minimum of the ratings given by the group members per item, (iii) the *average without misery* that eliminates the items having at least one individual rating which is below a certain predefined threshold and it considers the group preference as the average of all group members' ratings.

Recently, few works focused on aggregating the individual recommendation lists using voting rules. These latter have been proved as an effective solution to address the problem of finding "consensus" ranking between items given the individual preference orders of several decision makers [9]. We cite, Baltrunas et al. [3] investigated to produce group recommendations based on rank aggregation. The authors started by generating the recommendation list of each group member using collaborative filtering, then, the individual recommendations are aggregated into a ranked list of recommendations using *spearman rule* and *Borda count* aggregation method. This method is only restricted to the items in the group members' test set (i.e. the items which are not yet seen by the group members) and it does no consider the group members' interactions when generating group recommendations. Furthermore, Boratto et al. [6] proposed a group recommender system which is able to detect groups based on K-Means clustering algorithm. The group preference of the clustered users is modeled using different aggregation strategies including the voting rules (e.g. Borda count, plurality voting method, etc.). It has been pinpointed that Borda count and the average strategy are the best strategies that model group preference.

All these methods used the standard voting rules which are limited to full lists of items (i.e. contain the same items) and consider the totally ordered sets (i.e. all the items are ranked in each recommendation list). Nevertheless, actual recommendation lists cannot contain the same 'K' items and consequently, the ranking (i.e. preference) of an item for some group members may be unknown. So, we consider that the partial information presented in these lists should be delved further when aggregating the individual orderings.

2.2 Approximate Voting Rules

Voting rules consist in aggregating users preferences over a set of items in order to determine a consensus decision or recommendation using a specific voting rule.

Definition 1. *A voting model is defined by three components including:* $U = \{u_1, u_2, ..., u_n\}$ *is the group of users,* $A = \{a, b, ...\}$ *is the set of candidate items, such that* $|A| = m$*; and* $P = (RL_1, ..., RL_n)$ *is the preference profile of users in* U *which corresponds to a collection of complete rankings on* A*.* $RL_u \in P$ *represents the complete preference order of user* u *over* A*. For any* $a, b \in A$*,* $a \succ_i b$ *means that user* u *prefers* a *to* b*. For example, if* $A = \{a, b, c\}$*, a user who prefers* a *to* b *and* b *to* c *(and, thus, has complete preferences) would have preference order* $a \succ b \succ c$*.*

Given a complete preference profile, we consider the problem of selecting a consensus alternative, requiring the design of a *voting rule* f which selects a winner or a set of winners from A given a preference profile P and a set of available candidates. Scoring rules are a broad class of voting rules defined by a non-negative vector $\boldsymbol{s} = (s_1, ..., s_m)$ over a set of candidates of size m such that $s_1 \geq ... \geq s_m$. Each candidate receives s_j points from each voter who ranks her in the j^{th} position, and the score of a candidate is the total number of points she receives from all voters. The winner is the candidate with highest total score over all the votes. The well known scoring rule is Borda [8], for which the scoring vector is $\boldsymbol{s} = (m-1, m-2, ..., 0)$.

Example 1. *Let us consider a setting of 3 users with the following complete preferences over four films $m = \{a, b, c, d\}$: user 1 : $a \succ b \succ c \succ d$, user 2 : $b \succ c \succ d \succ a$, user 3 : $c \succ d \succ b \succ a$. Under Borda voting rule, the score of item a (resp. b, c and d) is equal to 3 (resp. 6, 6 and 3). In this setting, items b and c are the winners.*

Voting with top-k lists. Partial voting consists in allowing the users to provide incomplete preferences over the set of items. One natural form of partial voting is top-k voting where recommendation lists contain the k most preferred items out of m and they are indifferent among the remaining ones.

Definition 2. *Partial voting is defined by three components where: $U = \{u_1, u_2, ..., u_n\}$ is the group of users, $A = \{a, b, ...\}$ is the set of items, such that $|A| = m$; and $R = (RL_1^k, ..., RL_n^k)$ is the partial preference profile of users in U which corresponds to a collection of partial rankings on A. $RL_u^k \in R$ represents the partial preference order of user u who ranks only a k out of m items where $k \in \{1, ..., m-1\}$.*

Standard definitions of many voting rules assume that the users have complete preferences. However, requiring the users to provide a complete ranking over the whole set of candidate items can be difficult and too costly. The necessity to adapt these methods to the case of partial voting is of great importance. This adaptation consists in combining the partial preference orders into a consensus ranking. In fact, in order to handle partial preferences under Borda, one possible way is to transform the scoring vector and score unranked items appropriately. In this way, the voting rule will take as input top-k partial preference orders from users and outputs a non-empty subset of approximated winners. We refer to the voting rule that supports partial preference information by *approximate voting rule*. Two possible schemes can be found in the literature for treating partially ordered preferences in the form of k out of the m items:

- **Zero score:** The number of points given for the users' first and subsequent preferences is determined by the total number of items they have actually ranked, rather than the total number standing. Given top-k partial preference profile, scores are awarded to the ranked items as follows: $(m-1, ..., m-k)$ depending on their position in the vote; and unranked items get 0 points.

This method is known as *Modified Borda Count* [7]. We denote this method by $Borda_0$.

– **Average score:** The items not submitted by the users get an average share of the scores which the users have not exercised. Given top-k partial preference profile, scores are awarded to the ranked items as follows: $(m-1, ..., m-k)$ depending on their position in the vote; and unranked items get the average of the remaining scores $\frac{\sum_{j=k}^{m} s_j}{m-k}$. We denote this method by $Borda_{av}$.

Example 2. *Let us consider the above example with only top-2 recommendation list of each user i.e. each user ranks her 2 most preferred items out of the 4 available* $m = \{a, b, c, d\}$*: user* $1\ :\ a \succ b$*, user* $2\ :\ b \succ c$*, user* $3\ :\ c \succ d$*.*

Under $Borda_0$*, the score of item a (resp. b, c and d) is equal to 3 (resp. 5, 5 and 2). The winners are items b and c. Using* $Borda_{av}$*, each unranked item receives an average score equal to* $\frac{1}{2}$ *i.e.* $\frac{1+0}{2}$*. Then, the score of item a (resp. b, c and d) is equal to 4 (resp. 5.5, 5.5 and 3). The winners are also items b and c since they correspond to the highest score.*

3 Group Preference Modeling Based on Partial Recommendation Lists

Our focus in this paper is to generate the top-k items which can capture the group preference even when the preferences of some group members on some candidate items are unknown. The main idea consists in generating the top-k recommendation list for each group member RL_u^k using user-based collaborative filtering as a first step, then, aggregating the group members' recommendation lists into a relevance or consensus top-k recommendation list RL_G^k using $Borda_{av}$ and $Borda_0$. The whole framework of the proposed group recommendations generation is presented in Fig. 1.

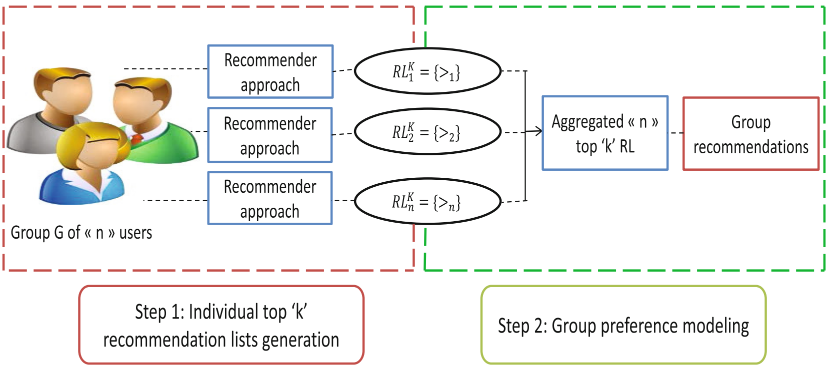

Fig. 1. Group recommendations generation

3.1 Step 1: The Individual Recommendation Lists Generation

In order to capture the group recommendations, we opt to generate the individual top-k recommendation lists of the group members by computing the members-items preferences using user-based collaborative filtering [10] since it is the most commonly used. Formally, let us consider a group G composed of n users $U = \{u_1, u_2, ..., u_n\}$ and a finite set of p items $I = \{i_1, i_2, ..., i_p\}$, the preference prediction of each user u on an item i is then computed as the weighted sum of the ratings given by the users similar to the active user:

$$P_{u,i} = \frac{\sum_{v \in S} r_{v,i} * s(u,v)}{\sum_{v \in S} |s(u,v)|} \tag{1}$$

where S is the set of the most similar users to u, $r_{u,i}$ is the rating given by the user u to the item i and $s(u,v)$ is the similarity degree between users u and v computed based on the *Pearson correlation measure* [11].

$$s(u_1, u_2) = \frac{\sum_{i \in I_{u_1} \cap I_{u_2}} (r_{u_1,i} - \bar{r}_{u_1})(r_{u_2,i} - \bar{r}_{u_2})}{\sqrt{\sum_{i \in I_{u_1} \cap I_{u_2}} (r_{u_1,i} - \bar{r}_{u_1})^2} \sqrt{\sum_{i \in I_{u_1} \cap I_{u_2}} (r_{u_2,i} - \bar{r}_{u_2})^2}} \tag{2}$$

where $I_{u_1} \cap I_{u_2}$ is the set of the co-rated items of users u_1 and u_2, $r_{u_1,i}$ is the rating of user u_1 on item i and $\bar{r}_{u_1}$ is the average rating of user u_1.

Clearly, the items having the highest preference prediction values are selected as the top-k recommendation list. The individual recommendation list RL_u^k represents the preference order of the group member over the top-k items.

3.2 Step 2: The Group Preference Modeling

In this step, the n individual recommendation lists of the group members generated from the previous step will be combined using the approximate voting rule to find out the group ranking RL_G over the m candidate items such that $m = \bigcup_{u=1}^{n} RL_u^k$. The top-k items appearing in RL_G are selected as the group recommendation list RL_G^k as follows:

$$RL_G^k = AVR_{j=1}^{N} RL_j^k \tag{3}$$

where AVR is the voting rule operator to combine the n individual recommendation list into a unique group recommendation list.

Example 3. *Let us consider a group G composed of four members $U = \{u_1, u_2, u_3, u_4\}$ and 50 items $I = \{a, b, c, ...etc.\}$. The following partial preference profile R contains the top '5' individual recommendation lists of the group members: $RL_1^5 : a \succ b \succ c \succ d \succ e$, $RL_2^5 : c \succ f \succ g \succ a \succ d$, $RL_3^5 : h \succ i \succ j \succ f \succ k$, $RL_4^5 : l \succ a \succ d \succ m \succ g$.*

Given the above individual recommendation lists, the candidate items is fixed to 12 (i.e. $m = \{a, b, c, d, e\} \bigcup \{c, f, g, a, d\} \bigcup \{h, i, j, f, k\} \bigcup \{l, a, d, m, g\} = \{a, b, c, d, e, f, g, h, i, j, l, m\}$).

Under $Borda_0$, the group ranking $RL_G = \{a, d, c, f, g, h, l, b, i, j, m, e, k\}$. Then, the group recommendation list RL^5_G contains the top 5 items such as $RL^5_G = \{a, d, c, f, g\}$.

Typically, the classical voting rules only work with complete recommendation lists which contain the same items. Indeed, recommendations are restricted to the items which are not yet seen by any group member since the items appearing in individual recommendation lists are often novel for active users. However, the recommendation list produced by the proposed group recommender system may contain items which are already seen and appreciated by some group members since it takes into account even the items appearing in at least one recommendation list.

4 Experimental Study

This section depicts the experimental study including the data set, the evaluation metrics, the experimental protocol and results.

4.1 Experimental Protocol

To evaluate the performance of the proposed group recommendation generation, we conduct our experiment on MovieLens[1] dataset. Movielens contains 100, 000 ratings collected from 942 users on 1681 movies. Movielens contains quantitative preferences which are scaled from 1 (low liking degree) to 5 (high liking degree).

To evaluate the effectiveness of the group recommendations, we focus on performing an *offline evaluation* which consists at first in dividing the data set chronologically in a training set (80%) and a test set (20%) (i.e. the testing data are selected in such a way that they occur after the training data over time). Then, we randomly organize users into groups of a specific size. For each group of 'n' members, we generate n top-k recommendation lists using user-based collaborative filtering. The individual recommendation lists will be combined into a single consensus ordering using different aggregation rules (i.e. the standard Borda count, $Borda_{av}$ and $Borda_0$). The k items which are classified in the top position of the group ordering are selected as group recommendations. Finally, the recommendations are evaluated individually as in the classical individual recommendation case, by comparing the generated recommendations to the existing data of the test set of each group member. In this experiment, we set 'k' to '5' as it is the most common recommendation list size in this context.

We choose the *Normalized Discounted Cumulative Gain* (nDCG) evaluation metric to compute the effectiveness of the group recommendations as it is one of the most popular IR metric measuring the quality of the ranking produced by a system [12]. The *nDCG* measures the discounted cumulative gain of items positions in a given ranking list (DCG) according to the discounted cumulative gain of items positions in the optimal ordering of the recommendation list (IDCG).

[1] http://movieLens.umn.edu.

Formally, the discounted cumulative gain (DCG) accumulated at a particular rank position p is:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{4}$$

where rel_i is the relevance value given by the user to the item at the position i.

Moreover, we choose the precision to evaluate the accuracy of the group recommendation list. The precision is defined in such a way that it detects the average of the true recommended alternatives relative to the total number of the alternative in the group recommendation list.

4.2 Experimental Results

The first experiment consists in computing the mean nDCG by varying the group size and the aggregation method used to combine the individual recommendation lists of group members.

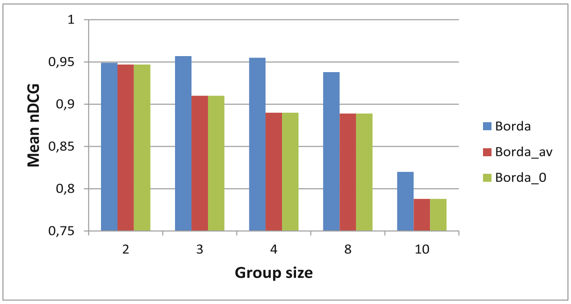

Fig. 2. Mean nDCG of Group recommendations under Borda, $Borda_{av}$ and $Borda_0$

As expected, Fig. 2 shows that the accuracy of the group recommendation list decreases when the group size increases. In fact, finding a consensus order for a large size of a random group is more difficult since the variation of recommendation lists is more significant. Results depicted in Fig. 2 show that the effectiveness of group recommendations when using $Borda_{av}$ and $Borda_0$ with a small group size (i.e. equal to '2') is almost the same as the standard Borda rule. However, with a large group size (i.e. ≥ 2), the effectiveness of the group recommendations in term of ranking quality is less good when using the approximate voting rules ($Borda_{av}$ and $Borda_0$) than the standard Borda rule. This is due to the lack of candidate items that can be considered when generating the group ranking and which consequently affects the mean nDCG.

The second experience is conducted to evaluate the validity of the generated group recommendation using the mean *precision* of the group members.

Figure 3 shows that the effectiveness of the group recommendation list is more significant using both $Borda_{av}$ and $Borda_0$ than the standard Borda rule. In fact,

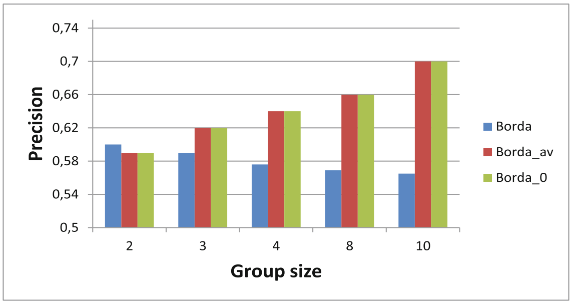

Fig. 3. Precision of Group recommendations under Borda, $Borda_{av}$ and $Borda_0$

by using approximate voting rules, group recommendations may contain more interesting items since it considers even the items with some unknown preferences and which are already seen by some group members. In fact, the group members will probably prefer to retain an item which has been appreciated by at least one user rather than an undiscovered item. However, with the standard Borda, the recommendation process is restricted to the items which are not yet seen by all the group members and with complete preference information. We note also that the precision of recommendations provided by the approximate Borda rules increases when the group size increases. This is due to the fact that the number of candidate items becomes more important with large groups and consequently, there are more chance to select relevant items in top-k recommendation list.

5 Conclusion

This paper proposes a new group recommendation method that offers to groups an efficient tool to support partial rankings contrarily to typical group recommender systems which handle only the total orders. It consists at first in generating top-k recommendation list for each group member. Then, the 'n' individual recommendation lists are aggregated together to produce top-k group recommendation list based on approximate voting rules ($Borda_{av}$ and $Borda_0$). Our experiments show that the group recommendation list generated using the approximate Borda rule is more accurate and interesting compared to group ranking created using the standard Borda. As a future work, one possible way is to consider *fuzzy* preference modeling to deal with incomplete recommendation lists. Additionally, we propose to consider the equally preferable items when aggregating the individual recommendation lists as it is a significant attribute in recommendations context.

References

1. De Pessemier, T., Dooms, S., Martens, L.: An improved data aggregation strategy for group recommendations. In: Proceedings of the 7th ACM Conference on Recommender Systems (RecSys 2013), pp. 36–39 (2013)

2. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007). doi:10.1007/978-3-540-72079-9_20
3. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys 2010), pp. 119–126 (2010)
4. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: PolyLens: a recommender system for groups of users. In: Prinz, W., Jarke, M., Rogers, Y., Schmidt, K., Wulf, V. (eds.) ECSCW 2001, pp. 199–218. Springer, Netherlands (2001). doi:10.1007/0-306-48019-0_11
5. Song, Y., Hu, Z., Liu, H., Shi, Y., Tian, H.: A novel group recommendation algorithm with collaborative filtering. In: International Conference on Social Computing (SocialCom), pp. 901–904 (2013)
6. Boratto, L., Carta, S.: Modeling the preferences of a group of users detected by clustering: a group recommendation case-study. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS 2014), pp. 1–7 (2014)
7. Emerson, P.J.: The politics of consensus: for the resolution of conflict and reform of majority rule. Peter Emerson Belfast (1994)
8. Saari, D.G.: The Borda dictionary. Soc. Choice Welfare **7**, 279–319 (1990)
9. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th International Conference on World Wide Web, New York, USA, pp. 613–622 (2001)
10. Zhao, Z.D., Shang, M.S.: User-based collaborative-filtering recommendation algorithms on Hadoop. In: Proceedings of Third International Conference on the Knowledge Discovery and Data Mining (WKDD 2010), pp. 478–481 (2010)
11. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Analysis of recommendation algorithms for E-commerce. In: Proceedings of the ACM E-Commerce, Minneapolis, Minn, USA, pp. 158–167 (2000)
12. Luberg, A., Tammet, T., Järv, P.: Smart city: a rule-based tourist recommendation system. In: Law, R., Fuchs, M., Ricci, F. (eds.) Information and Communication Technologies in Tourism, pp. 51–62. Springer, Vienna (2011)