Opinion Dynamics-Based Group Recommender Systems

Jorge Castro, Jie Lu, Guangquan Zhang, Yucheng Dong, and Luis Martínez

***Abstract*—With the accessibility to information, users often face the problem of selecting one item (a product or a service) from a huge search space. This problem is known as information overload. Recommender systems (RSs) personalize content to a user's interests to help them select the right item in information overload scenarios. Group RSs (GRSs) recommend items to a group of users. In GRSs, a recommendation is usually computed by a simple aggregation method for individual information. However, the aggregations are rigid and overlook certain group features, such as the relationships between the group members' preferences. In this paper, it is proposed a GRS based on opinion dynamics that considers these relationships using a smart weights matrix to drive the process. In some groups, opinions do not agree, hence the weights matrix is modified to reach a consensus value. The impact of ensuring agreed recommendations is evaluated through a set of experiments. Additionally, a sensitivity analysis studies its behavior. Compared to existing group recommendation models and frameworks, the proposal based on opinion dynamics would have the following advantages: 1) flexible aggregation method; 2) member relationships; and 3) agreed recommendations.**

***Index Terms*—Opinion dynamics, recommender systems (RSs), social influence, weights matrix.**

I. Introduction

CURRENTLY, businesses and individuals often face situations in which they have to chose an alternative from a large range of options. This situation is known as information overload, and limited evaluation resources often lead to the selection of suboptimal alternatives. In information overload scenarios, personalization techniques help by tailoring access to the information. Recommender systems (RSs) are successful tools in personalization that filter relevant items (products or services) according to users preferences to present a reduced list of the most relevant choices, i.e., recommendations. Successful examples of applications are e-learning [1], [2], e-business [3], e-commerce [4], e-tourism [5], [6], financial investment [7], and Web pages [8], [9], among others. The most successful approach for RSs is based on collaborative filtering (CF) [10]. There are several active research lines within RSs, such as context-aware recommendation [11], friend recommendation [12] or group RSs (GRSs) [13], among others. This paper focuses on GRSs, which recommend items to be consumed by groups of users, hence recommendations are targeted to a group of users instead of individuals.

Most of GRSs usually aggregate individual information to produce a group recommendation [13]. Some techniques aggregate individual ratings [14], while others aggregate individual recommendations [15]. Within these approaches, several aggregation strategies are used, such as least misery, most pleasure, or average, among others [13]. However, these aggregation strategies disregard important information about the group [16], such as the relationships between members' preferences. As such, aggregation does not take into account similarity of preferences or overlap of experiences, among others, and this may lead to biased recommendations.

To consider these relationships, this paper aims to develop a new opinion dynamics model and apply it to group recommendations. Opinion dynamics studies the information fusion process within a group of experts [17]. DeGroot's model [18] assumes that individuals change their opinions according to a social influence model, in which each user considers other expert opinions with a certain weight. It seems that this social process could be realistic in GRSs and we propose to integrate DeGroot's model within group recommendation.

DeGroot's model can lead to either consensus or fragmentation of opinions. Consensus has already been studied in group recommendation, and previous research determined that providing a consensus solution benefits recommendations [19]. Some try to achieve consensus through a negotiation process, others use automated consensus reaching processes based on individual predictions that considers members' opinion before aggregating them into a recommendation [20], [21]. However, these works seek consensus by looking only at individual values—they disregard the relationships between members' preferences.

The recommendation process with DeGroot's model has two possible outcomes. Either a consensus is achieved by following

Manuscript received November 1, 2016; accepted March 30, 2017. This work was supported in part by the Australian Research Council under Grant DP-150101645, in part by the Spanish FPU Fellowship under Grant FPU13/01151, in part by the National Natural Science Foundation of China under Grant 71571124, and in part by the Spanish National Research Project under Grant TIN2015-66524-P. This paper was recommended by Associate Editor E. Chen. *(Corresponding author: Jorge Castro.)*

J. Castro is with the Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain, and also with the School of Software, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: jcastro@decsai.ugr.es).

J. Lu and G. Zhang are with the School of Software, University of Technology Sydney, Sydney, NSW 2007, Australia.

Y. Dong is with the Business School, Sichuan University, Chengdu 610065, China.

L. Martínez is with the Computer Science Department, University of Jaén, 23008 Jaén, Spain, and also with the School of Management, Wuhan University of Technology, Wuhan 430070, China

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2695158

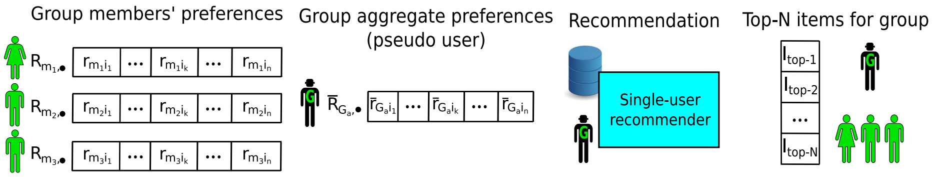

Group members' preferences
$R_{m_1,\bullet}$
$r_{m_1 i_1}$ … $r_{m_1 i_k}$ … $r_{m_1 i_n}$
$R_{m_2,\bullet}$
$r_{m_2 i_1}$ … $r_{m_2 i_k}$ … $r_{m_2 i_n}$
$R_{m_3,\bullet}$
$r_{m_3 i_1}$ … $r_{m_3 i_k}$ … $r_{m_3 i_n}$
Group aggregate preferences (pseudo user)
G
$\bar{R}_{G_a,\bullet}$
$\bar{r}_{G_a i_1}$ … $\bar{r}_{G_a i_k}$ … $\bar{r}_{G_a i_n}$
Recommendation
Single-user recommender
G
Top-N items for group
I_{top-1}
I_{top-2}
…
I_{top-N}
G

Fig. 1. Framework of GRS based on rating aggregation.

the relationships between members, or it is not, and the recommendation may not properly reflect group preferences. Hence, to improve recommendations in the latter case, we propose a GRS based on opinion dynamics with consensus (GROD). This paper proposes the following.

1) Pre-GROD, which extends DeGroot's model to GRS and considers relationships between members' preferences in recommendations.
2) GROD, which extends Pre-GROD by ensuring the conditions to compute consensus recommendations that are accepted by all members.

The remaining of this paper is structured as follows. Section II presents the preliminary concepts needed for the proposal. The new GRS framework is presented in Section III. Section IV presents a set of experiments done to evaluate the proposal's performance, and Section V presents a sensitivity analysis of GROD. Section VI presents the conclusions and highlights future work.

II. Preliminaries

This section introduces the preliminary concepts on GRSs, opinion dynamics, and the study of consensus in DeGroot's model.

A. Group Recommender Systems

GRSs are an extension to individual RSs that compute recommendations targeted to groups instead of individuals [13]. The GRS problem is formalized in the following way:

$$\text{Recommendation}(G, I) = \arg \max_{i_k \in I} \text{Prediction}(G, i_k) \quad (1)$$

where G is the target group, I is the set of available items, and $\text{Prediction}(G, i_k)$ is a function that assigns a utility value for the item i_k regarding group G members.

There are different approaches for making the group predictions. A successful way to address this problem is to reduce it to an individual recommendation problem by means of aggregating individual information. There are two aggregation approaches [22], [23].

1) *Rating Aggregation-Based GRS Frameworks (See Fig. 1):* The members state their preference for a subset of products. These ratings are then aggregated to build a group profile that represents the group preference called "pseudo-user," which is then used in an individual CF.
2) *Recommendation Aggregation-Based GRS Frameworks (See Fig. 2):* Individual recommendations are computed for each member of the group in a CF, which are later combined to produce recommendations targeted to the group.

Several experimental works show that neither approach is better than the other in all scenarios [13], [14]; hence, a study to choose the best technique is required. Moreover, these approaches rely on different aggregation strategies [13].

1) *Least Misery:* Tries to avoid member dissatisfaction with the recommended items. The group must be as satisfied as the least satisfied member, therefore the group preference for one item is the minimum individual preference.
2) *Average:* The group preference is the average of all the individual preferences.
3) *Average Without Misery:* Averages individual ratings after excluding items with individual preferences below a certain threshold.

Each aggregation strategy has its own strengths and weaknesses. For example, the least misery strategy is suitable for small groups. As the groups become larger, the probability of an item having a negative rating increases, which leads to a group profile composed mostly of negative preferences, ultimately biasing the recommendations [22]. Additionally, this aggregation strategy is highly sensitive to new ratings, given that a new negative rating can significantly impact the recommendation. The average strategy aggregates all members' ratings, not just the low ones. To accommodate the fact that lower ratings are more important than high ones, the average without misery strategy does not aggregate low member ratings. Therefore, hated items are avoided with this strategy.

However, as pointed out in the Introduction, these aggregation approaches overlook relationships among members' preferences, such as similarity of preferences or overlap of experiences. In this paper, member relationships are introduced to the process using opinion dynamics. This way, aggregation takes specific features of the group into account when making recommendations.

B. Opinion Dynamics

Opinion dynamics models are used to describe particular aspects of the social behavior of a number of individuals, and to model how the opinion of a group of experts evolves over time. Various approaches have been proposed to model these changes in opinion, given various assumptions within the process.

Our proposal assumes that members update their opinions according to social influences [24]. Therefore, the DeGroot model [18] is in accordance with this view because in it, evolution of opinions is driven by the matrix of weights $A = (a_{u_j,u_k})_{(g \times g)}$, where a_{u_j,u_k} is the weight that expert u_j assigns to the opinion of expert u_k. The value a_{u_j,u_j} denotes the weight that the user assigns to their own opinion, and therefore

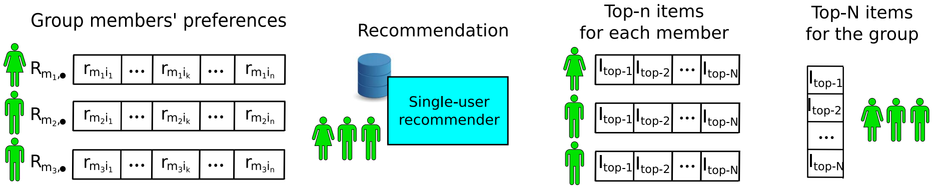

Group members' preferences
$R_{m_1,\bullet}$
$r_{m_1 i_1}$ ... $r_{m_1 i_k}$ ... $r_{m_1 i_n}$
$R_{m_2,\bullet}$
$r_{m_2 i_1}$ ... $r_{m_2 i_k}$ ... $r_{m_2 i_n}$
$R_{m_3,\bullet}$
$r_{m_3 i_1}$ ... $r_{m_3 i_k}$ ... $r_{m_3 i_n}$
Recommendation
Single-user recommender
Top-n items for each member
I_{top-1} I_{top-2} ... I_{top-N}
I_{top-1} I_{top-2} ... I_{top-N}
I_{top-1} I_{top-2} ... I_{top-N}
Top-N items for the group
I_{top-1}
I_{top-2}
...
I_{top-N}

Fig. 2. Framework of GRS based on recommendation aggregation.

is the degree to which they are reluctant to modify their initial preference. A restriction of this matrix is that all the weights of an expert must sum to 1, i.e., $\sum_{u_k \in G} a_{u_j,u_k} = 1$. Opinions are updated in the DeGroot model using the following equation:

$$x_{u_j}^{t+1} = a_{u_j,u_1} x_{u_1}^t + a_{u_j,u_2} x_{u_2}^t + \cdots + a_{u_j,u_g} x_{u_g}^t \tag{2}$$

where $x_{u_j}^{t+1}$ is the opinion of expert u_j in round $t+1$, a_{u_j,u_k} is the influence of expert u_k opinion for u_j, and $x_{u_j}^t$ is the opinion of expert u_j in round t.

Members' opinions are updated until they reach a stability point. As stated by DeGroot [18], the relationship between the initial and final opinions can be determined by analyzing the weights matrix A. The final opinion is a linear combination of the initial opinions of all agents in the form $c = \sum \lambda_{u_j} * x_{u_j}^0$. The coefficients of this linear combination are related to the normalized left eigenvector associated with the eigenvalue 1 of the weights matrix A.

Analysis of the process shows an interesting outcome. If the left eigenvector is unique, then the final opinion converges to one consensus value. If this eigenvector is not unique, then the opinion fragments across several values. Given that consensus recommendations should satisfy all members of the group [20], this paper aims to provide consensus recommendations. Thus, we focus on ensuring consensus by modifying the weights matrix.

C. Ensure Consensus in DeGroot Model

To study the evolution of opinions in DeGroot's model, the relationships between members can be expressed in terms of a weighted directed graph DG(G, S), where $G = \{u_1, \ldots, u_g\}$ is the set of members, and S is the set of directional edges. Thus, s_{u_j,u_k} indicates the degree of the relationship that member u_j has with u_k. The weights matrix A can be computed from S to apply the DeGroot model

$$a_{u_j,u_k} = \frac{s_{u_j,u_k}}{\sum\limits_{u_l \in G} s_{u_j,u_l}}. \tag{3}$$

The normalized eigenvector associated with the left eigenvalue 1 of matrix A is unique as long as there is at least one member accessible by all other members, called the opinion leader. In these cases, the opinion of all members converges to a unique value, since all members' final opinions are influenced by the opinion leaders. Given that we are focusing on whether the set of directional edges S leads to consensus, its analysis can be simplified by considering the accessibility matrix P [see (5)]. P can be used to determine if the weights matrix leads to consensus by checking if at least one member is accessible by all. Therefore, to check if S leads to consensus

$$S \text{ leads to consensus} \iff \exists u_j \text{ s.t. } p_{u_k,u_j} = 1 \ \forall u_k \in G \tag{4}$$

$$P = f\left(S^{(g-1)}\right) \tag{5}$$

where P is the accessibility matrix, S is the matrix with the weighted directional edges, g is the number of experts, and function f is defined as

$$f(s_{u_j,u_k}) = p_{u_j,u_k} = \begin{cases} 1 & \text{if } s_{u_j,u_k} > 0 \\ 0 & \text{if } s_{u_j,u_k} = 0. \end{cases} \tag{6}$$

If S does not fulfil (4), then the opinion fragments across several values. The multiplicity of the left eigenvector, associated with left eigenvalue 1 of the weights matrix A, determines the number of different opinions at the end of the opinion dynamics process. These eigenvectors are orthogonal, and their normalized form determines how members' opinions are grouped into several disjoint opinion subgroups. Moreover, the normalized eigenvectors determine the contribution of member u_j's opinion to the final value.

In a weights matrix A that leads to consensus, the eigenvector is unique. A strictly positive λ_{u_j} in the normalized eigenvector determines the members' opinions that contribute to the final value—the opinion leaders. Therefore, those members whose $\lambda_{u_j} = 0$ simply follow other members opinions. In situations that do not lead to consensus, the eigenvalue has a specific multiplicity, and this determines the number of opinion subgroups formed.

Within each of these opinion subgroups, there is at least one opinion leader. Moreover, followers can always reach at least one opinion leader in their respective subgroup. This way, the members belonging to the same subgroup have the same opinion at the end of the process. However, opinions of other subgroups are different, given that they do not influence on each other.

Therefore, a way to ensure consensus is to connect the opinion subgroups by adding relationships between leaders of opinion subgroups. This way, their eigenvectors orthogonality break and the connected subgroups have the same opinion at the end of the process. To do this, we can take the leader of any opinion subgroup and create a relationship to the leader of a different opinion subgroup.

By doing so iteratively, all opinion subgroups are connected and the group reaches a consensus value with a modified set of directional edges S'. This process needs to be done at least $q-1$ times, where q is the initial number of opinion subgroups. Different combinations of relationships can be added to ensure consensus. In GROD, the opinion subgroups are analyzed to select the best ones and create a relationship between their leaders.

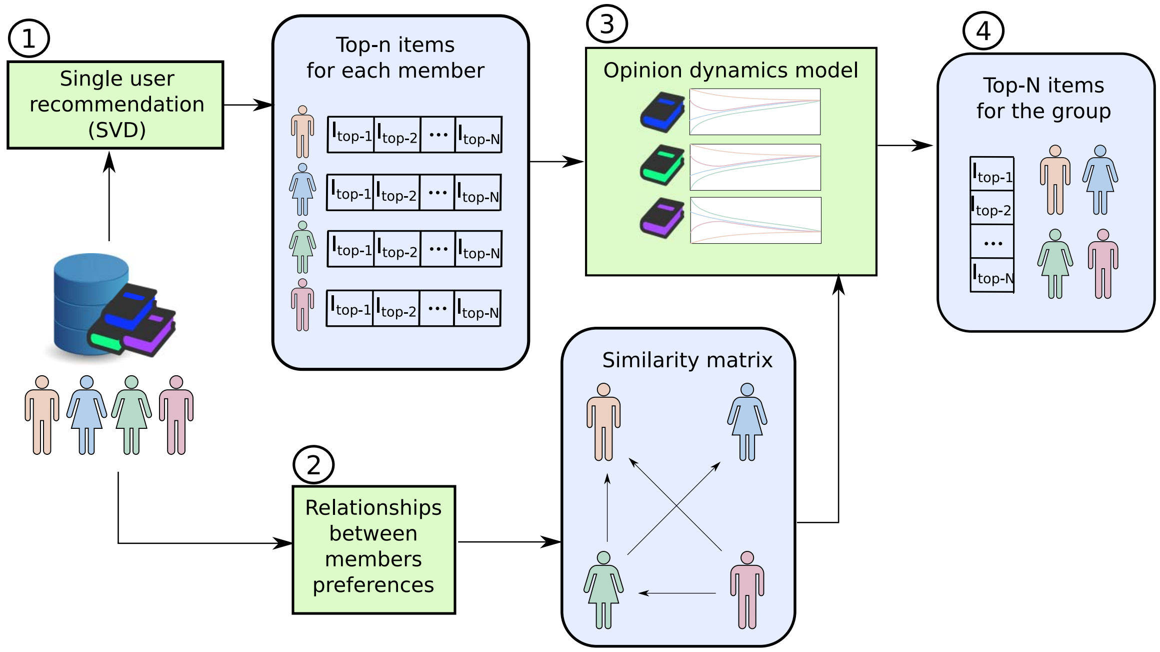

1
Single user recommendation (SVD)
Top-n items for each member
Itop-1
Itop-2
···
Itop-N
3
Opinion dynamics model
4
Top-N items for the group
Itop-1
Itop-2
···
Itop-N
Similarity matrix
2
Relationships between members preferences

Fig. 3. Framework for Pre-GROD.

III. Framework for Group Recommendation Based on Opinion Dynamics

Here, a new framework for group recommendation based on opinion dynamics is introduced. This framework allows us to take relationships between group members' preferences into account, such as similarity of preferences or overlap of experiences, to improve group recommendations. In this framework, individual predictions are combined to produce group recommendations. Unlike traditional aggregation-based GRSs, this framework applies a flexible process to produce a group value, given that it is driven by a matrix of weights between group members. The general framework is depicted in Fig. 3, and comprises the following steps.

1) Compute individual predictions.
2) Compute the relationships between members' preferences.
3) Predict the group rating for each item applying DeGroot's model.
4) Recommend the items with the highest prediction.

Within the above framework two proposals are presented: 1) a GRS based on opinion dynamics (Pre-GROD) and 2) a GROD. Both approaches combine individual predictions using the relationships between member preferences. Pre-GROD follows the scheme depicted in Fig. 3. GROD adds a step to Pre-GROD that analyzes and, if needed, updates the weights matrix to ensure consensus. The remainder of this section details both proposals using the notation in Table I.

TABLE I
Notations

Symbol	Description
$U = \{u_1, \ldots, u_m\}$	set of all users
$I = \{i_1, \ldots, i_n\}$	set of all items
$r_{u_j i_k}$	rating that user u_j gave about item i_k
$\hat{r}_{u_j i_k}$	rating prediction of user u_j over item i_k
$G = \{u_1, \ldots, u_g\} \subseteq U$	target group with g members
$R_{u_j, \bullet}$	set of ratings of u_j
$R_{G^{u_j}, \bullet}$	set of ratings of the members in the opinion subgroup of u_j
$S = (s_{u_j, u_k})_{(g \times g)}$	matrix of directed edges between members
s_{u_j, u_k}	degree of relationship that u_j has with u_k
$A = (a_{u_j, u_k})_{(g \times g)}$	matrix of weights between members
a_{u_j, u_k}	weight that u_j has for u_k opinion
$X^t_{i_k}$	column matrix with the opinion of each member about item i_k at round t
$\hat{r}_{G, i_k}$	rating prediction of group G over item i_k

A. Group Recommender System Based on Opinion Dynamics (Pre-GROD)

The first step computes individual predictions for a given item using an individual RS: the stochastic gradient descent singular value decomposition (SVD) RS [25]. Hence, a prediction exists for each group member. These individual predictions will be used to compute the group prediction. An important consideration when producing the individual predictions, is that the individual RS might not be able to make a prediction for a given user-item pair. To avoid this problem, we use a matrix factorization RS [25] which is able to predict ratings for all user-item pairs as long, as they have ratings.

The second step calculates the relationships between member preferences. Matrix S is produced, which will later be used to drive the opinion dynamics process. The way it is computed defines how the individual predictions are aggregated to obtain the group prediction

$$s_{u_j, u_k} = \text{similarity}\left(u_j, u_k\right) \in [0, 1], \forall u_j, u_k \in G \subseteq U. \quad (7)$$

Remark: The relationship between members' preferences is computed with a similarity measure. This measure can be

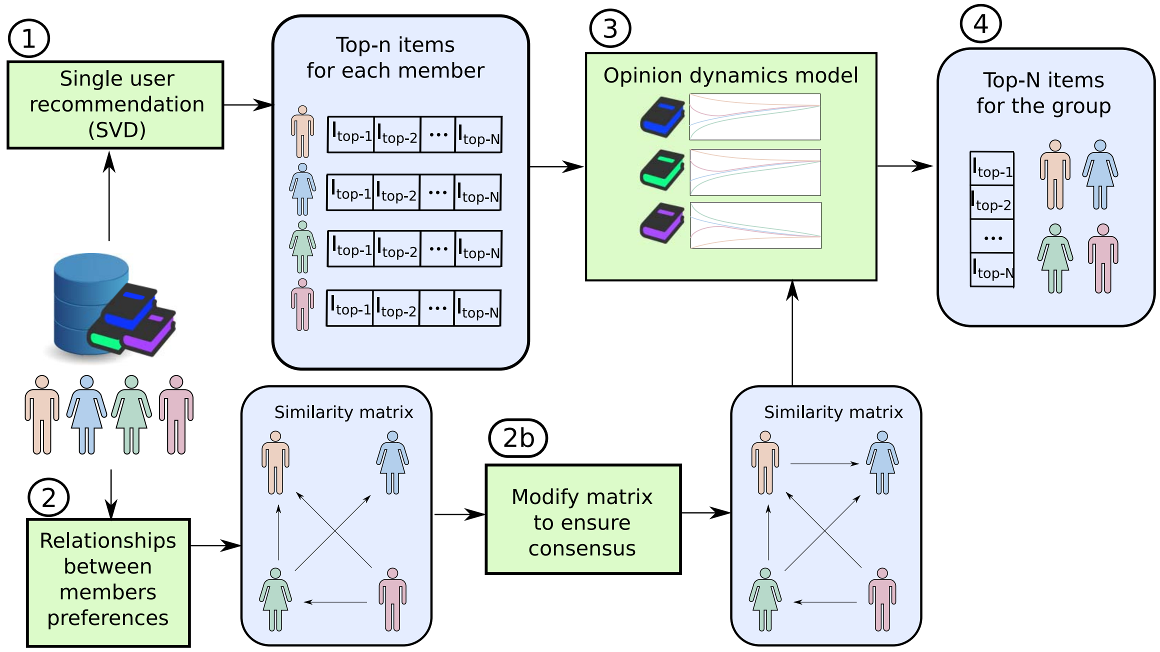

1
Single user recommendation (SVD)
Top-n items for each member
I top-1
I top-2
...
I top-N
3
Opinion dynamics model
4
Top-N items for the group
2
Relationships between members preferences
Similarity matrix
2b
Modify matrix to ensure consensus
Similarity matrix

Fig. 4. GROD.

defined in various ways [26], [27]. In the experiments section, several similarities are evaluated to determine the best one.

The third step calculates the group predictions. First, the weights matrix A reflects the relationships between members' preferences, thus the weights matrix A is computed from the relationships matrix [see (3)]. The DeGroot model is applied to each item to combine the individual predictions (considered to be the initial opinions of each member) and produce the final opinion for that group

$$X_{i_k}^0 = \left(\hat{r}_{u_j,i_k}\right)_{(1\times g)} \tag{8}$$

$$\hat{r}_{G,i_k} = \lim_{t\to\infty} A^t X_{i_k}^0. \tag{9}$$

At this point, all the final opinions of the members for the target item i_k are averaged to obtain the group prediction. This process is repeated for each item to obtain their respective group prediction.

Finally, in the fourth step, the recommendation is determined by the items with the highest group prediction.

B. Group Recommendation Based on Opinion Dynamics With Consensus

The previous section describes Pre-GROD, which does not ensure consensus. This situation may lead to recommendations that not all members agree to, which diminishes members satisfaction. To consider this situation, and correct it, GROD adds a step to the Pre-GROD framework to ensure consensus (see Fig. 4).

In step 2b, the relationships matrix S is analyzed and, if needed, modified to ensure consensus. The weights matrix A is extracted from the relationships matrix, which determines how the opinions are updated in the DeGroot model. As stated in Section II-C, (4) determines whether the relationships matrix S leads to consensus.

If the group does not reach consensus, then the relationships matrix S is modified. The selection of which relationships to add is a key aspect in such modifications for group recommendation.

In these cases, there are q left eigenvalues of weights matrix A and the absolute values of these eigenvalues are all equal to 1. This way, a partition of the members with a contribution to the final opinion is obtained. The aim of adding relationships is to connect those q subsets. Therefore, $q-1$ relationships need to be added in such a way that each subset is connected to at least one other.

There are multiple possible combinations to select the relationships to add. GROD computes the score of each missing relationship using the number of ratings of the target opinion subgroup. Thus, the opinions evolve to those of opinion subgroups with more defined tastes. Hence, the relationship with the highest score is selected and a directional edge with degree of relationship 1 is added

$$\arg \max_{u_j,u_k}\left(\text{score}(u_j, u_k)\right) \tag{10}$$

$$\text{score}\left(u_j, u_k\right) = |R_{G^{u_k}\bullet}| = \sum_{u_l\in G^{u_k}} \left|R_{u_l\bullet}\right| \tag{11}$$

where G^{u_k} is the set of members that are in the opinion subgroup of member u_k, i.e., the members whose lambda is positive in the eigenvector associated to the opinion subgroup.

This process is repeated until the relationships matrix S' leads to consensus, which is used in the remaining steps to obtain the group prediction.

IV. Experiments

To evaluate GROD, two experiments were performed. The first experiment evaluates Pre-GROD for group recommendation in general. The second one evaluates GROD

TABLE II
PARAMETERS FOR THE INDIVIDUAL PREDICTOR

Parameter	Value
Number of features	20
Iterations per feature	20
Learning rate (γ)	0.01
Large values penalty (λ)	0.02

TABLE III
DATASETS USED IN THE EXPERIMENTS

Dataset	Users	Items	Ratings	Sparsity
MovieLens-100k	943	1,682	100,000	93.69%
MovieLens-1m	6,040	3,706	1,000,209	95.53%

for group recommendation in groups that do not reach consensus.

The remainder of this section is structured as follows. First, the settings common to the experiments are described. The datasets and the methods of processing are then detailed, followed by the evaluation measures used to assess the behavior of the compared techniques. Lastly, the results are analyzed.

A. Experimental Procedure

In these experiments, a GRS based on recommendation aggregation using averages is considered to be the baseline. Five similarities between users were identified to compare several notions of relationships between members.

1) *Pre-GROD Cosine:* Cosine coefficient [28].
2) *Pre-GROD Relevance(30):* Relevance factor with $r = 30$ [26].
3) *Pre-GROD Cond. Prob.:* Conditional probability [27].
4) *Pre-GROD Asym. Cos.:* Asymmetric cosine [27].

The evaluated systems used an individual RS to compute the predictions for each member. Stochastic gradient descent SVD [25] is used for this aim, with the configuration shown in Table II.

The second experiment was designed to evaluate the modification of relationships to ensure consensus. Therefore, three systems are compared: 1) the baseline; 2) our proposed Pre-GROD; and 3) our proposed GROD.

B. Datasets

The compared methods were evaluated using datasets readily available to researchers, which are shown in Table III. There are different methods of forming groups, such as users with similar interests or heterogeneous groups, among others. These experiments aim to evaluate techniques for occasional groups, therefore the groups were formed randomly [29], and test different group sizes, ranging from 1 to 10 members.

C. Evaluation Measures

In evaluating the two GRS proposals for accuracy, two widely used evaluation measures were considered: 1) mean absolute error (MAE) and 2) root mean square error (RMSE) [30], [31].

1) *MAE:* Measures the difference between the system prediction and the true value

$$\mathrm{MAE} = \frac{1}{|R|} \sum_{r_{u_j,i_k} \in R} \left| r_{u_j,i_k} - p_{u_j,i_k} \right| \tag{12}$$

where p_{u_j,i_k} is the GRS prediction and r_{u_j,i_k} is the rating in the dataset. The prediction is the same for all the members in the GRS, but their ratings are different.

2) *RMSE:* Similarly, RMSE measures the prediction error, giving more importance to large errors

$$\mathrm{RMSE} = \sqrt{\frac{1}{|R|} \sum_{r_{u_j,i_k} \in R} \left(r_{u_j,i_k} - p_{u_j,i_k} \right)^2}. \tag{13}$$

Given that MAE and RMSE are measures of the error of prediction, they are measures to minimize.

To evaluate the techniques we performed 20 executions of five-cross-fold validation. For each execution, the groups and the training-test partitions were different. The results of each GRS on each execution were averaged to obtain the final value.

D. Experiment 1: Pre-GROD in the General Case

In this experiment, the GRSs described in Section IV-A are evaluated to determine their performance and the suitability of this proposal.

Tables IV and V show the comparative results of the GRSs on MovieLens-100k. Table IV shows the results for MAE and Table V shows the results for RMSE [see also Fig. 5(a) and (b)]. The results are stratified by group size, ranging from 1 to 10 members.

The results show that Pre-GROD with the Asym. Cos. similarity achieved the best performance over the baseline, with a relative improvement of 1% across all group sizes on average. Improvement was higher in smaller group sizes. Relative improvement in MAE was 1.72%, 1.68%, and 1.49% for groups of sizes 2–4, respectively. In RMSE, it was 1.24%, 1.35%, and 1.32%, respectively. Moreover, paired t-tests were performed to determine whether the differences in the results of each technique for each group size are statistically significant. The tests confirmed that most of the differences observed in Tables IV and V are statistically significant. The results for Pre-GROD Asym. Cos. were better than Pre-GROD Cosine and Pre-GROD Cond. Prob. alone. This fact indicates that the combination of exposure to the same experiences, and correlation between the satisfaction on these experiences, improves performance.

In terms of Pre-GROD's performance and the various ways to determine the relationships between members' preferences, asymmetric measures showed better results than symmetric ones. Asymmetric measures, such as Pre-GROD Relevance(30), Pre-GROD Cond. Prob., and GROD Asym. Cos., consistently obtained better results than Pre-GROD Cosine. Symmetric measures obtained results similar to the baseline. This indicates that when asymmetric similarities are considered, the system achieves better results that may be motivated by the inherent asymmetry of relationships between people.

TABLE IV
EXPERIMENT 1: MAE FOR THE EVALUATED GRSs IN MOVIELENS-100K

		Group size									
GRS		1	2	3	4	5	6	7	8	9	10
	Baseline	0.7418	0.7816	0.7931	0.8004	0.8044	0.8050	0.8070	0.8077	0.8082	0.8092
Pre-GROD	Cosine	0.7418	0.7816	0.7927	0.7999	0.8038	0.8045	0.8065	0.8072	0.8077	0.8088
Pre-GROD	Relevance(30)	0.7418	0.7790	0.7887	0.7956	0.7995	0.8004	0.8028	0.8037	0.8045	0.8059
Pre-GROD	Cond. prob.	0.7418	0.7685	0.7802	0.7890	0.7944	0.7966	0.7998	0.8012	**0.8024**	**0.8042**
Pre-GROD	Asym. Cos.	0.7418	**0.7684**	**0.7800**	**0.7887**	**0.7941**	**0.7964**	**0.7997**	**0.8011**	**0.8024**	0.8043

TABLE V
EXPERIMENT 1: RMSE FOR THE EVALUATED GRSs IN MOVIELENS-100K

		Group size									
GRS		1	2	3	4	5	6	7	8	9	10
	Baseline	0.9442	0.9827	0.9954	1.0044	1.0091	1.0103	1.0127	1.0137	1.0144	1.0157
Pre-GROD	Cosine	0.9442	0.9827	0.9951	1.0040	1.0086	1.0099	1.0123	1.0133	1.0141	1.0154
Pre-GROD	Relevance(30)	0.9442	0.9798	0.9904	0.9985	1.0029	1.0043	1.0070	1.0081	1.0091	1.0107
Pre-GROD	Cond. prob.	0.9442	**0.9705**	**0.9821**	**0.9912**	**0.9968**	**0.9991**	**1.0024**	1.0040	1.0054	1.0073
Pre-GROD	Asym. Cos.	0.9442	0.9707	0.9822	0.9913	**0.9968**	**0.9991**	**1.0024**	**1.0039**	**1.0053**	**1.0072**

TABLE VI
EXPERIMENT 1: MAE FOR THE EVALUATED GRSs IN MOVIELENS-1M

		Group size									
GRS		1	2	3	4	5	6	7	8	9	10
	Baseline	0.6751	0.7309	0.7495	0.7590	0.7639	0.7678	0.7709	0.7723	0.7735	0.7747
Pre-GROD	Cosine	0.6751	0.7309	0.7486	0.7579	0.7628	0.7666	0.7699	0.7713	0.7725	0.7738
Pre-GROD	Relevance(30)	0.6751	0.7291	0.7441	0.7523	0.7569	0.7607	0.7643	0.7659	0.7674	0.7690
Pre-GROD	Cond. prob.	0.6751	0.7110	0.7288	0.7403	0.7473	0.7526	0.7577	0.7603	0.7627	0.7648
Pre-GROD	Asym. Cos.	0.6751	**0.7106**	**0.7283**	**0.7398**	**0.7468**	**0.7521**	**0.7574**	**0.7600**	**0.7625**	**0.7647**

TABLE VII
EXPERIMENT 1: RMSE FOR THE EVALUATED GRSs IN MOVIELENS-1M

		Group size									
GRS		1	2	3	4	5	6	7	8	9	10
	Baseline	0.8628	0.9195	0.9398	0.9511	0.9566	0.9613	0.9649	0.9665	0.9681	0.9696
Pre-GROD	Cosine	0.8628	0.9195	0.9390	0.9501	0.9555	0.9602	0.9639	0.9656	0.9672	0.9687
Pre-GROD	Relevance(30)	0.8628	0.9176	0.9338	0.9434	0.9484	0.9529	0.9569	0.9589	0.9608	0.9625
Pre-GROD	Cond. prob.	0.8628	**0.9003**	**0.9189**	0.9313	0.9381	0.9439	0.9494	0.9522	0.9549	0.9572
Pre-GROD	Asym. Cos.	0.8628	**0.9003**	**0.9189**	**0.9312**	**0.9380**	**0.9438**	**0.9493**	**0.9521**	**0.9548**	**0.9571**

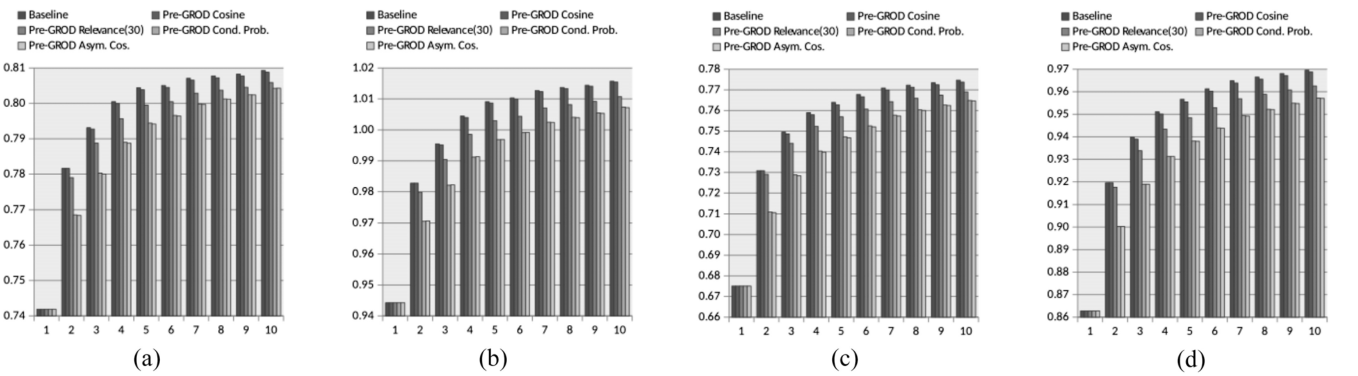
Finally, comparing the results on MAE and RMSE for the different GRSs, in general similar behavior is observed, which means that the balance between small and large errors is similar. An interesting observation is that Cond. Prob. obtained better results for RMSE and Asym. Cos. for MAE. This indicates that the errors of *Cond. Prob.* are smaller than Asym. Cos. ones.

Tables VI and VII show the comparative results of the GRSs on MovieLens-100k. Table VI shows the results for MAE and Table VII shows the results for RMSE [see also Fig. 5(c) and (d)]. To evaluate the proposal, the results have been again stratified by group size from 1 to 10 members.

Overall, the performance of the techniques shows similar behavior to MovieLens-100k. Although, the results obtained in MovieLens-1M were better than MovieLens-100k because this dataset contains more ratings, which increases the quality of the model generated by the RS. Its performance influences the final accuracy of the GRS by decreasing the prediction error.

Mainly, the results show that the proposed method with the Asym. Cos. measure achieved the best performance compared to the other approaches, with an average relative improvement of roughly 2% over the baseline across all group sizes evaluated—a greater improvement than MovieLens-100k. The improvement was also higher for smaller groups. The relative improvement in RMSE was 2.14%, 2.28%, and 2.13% for groups sizes 2–4, respectively. In MAE, it was 2.85%, 2.91%, and 2.59% for the same group sizes. The statistical tests confirmed that most of the differences observed in Tables VI and VII are statistically significant.

We can conclude that Pre-GROD Asym. Cos. improves the results in general recommendation cases, for random groups, compared to the baseline, with an improvement of 1% for MovieLens-100k and 2% in MovieLens-1M. Moreover, the



(a) (b) (c) (d)

Fig. 5. Results for the evaluated GRSs by group size. (a) MAE in MovieLens-100k. (b) RMSE in MovieLens-100k. (c) MAE in MovieLens-1M. (d) RMSE in MovieLens-1M.

TABLE VIII
EXPERIMENT 2: MAE FOR THE EVALUATED GRSS IN MOVIELENS-100K

	Group size									
GRS	1	2	3	4	5	6	7	8	9	10
Baseline	0.7792	0.8091	0.8145	0.8191	0.8205	0.8213	0.8232	0.8255	0.8273	0.8285
Pre-GROD	0.7792	0.8091	0.8088	0.8133	0.8152	0.8166	0.8191	0.8215	0.8234	0.8247
GROD	0.7792	**0.7820**	**0.7999**	**0.8092**	**0.8128**	**0.8152**	**0.8183**	**0.8211**	**0.8232**	**0.8246**

TABLE IX
EXPERIMENT 2: RMSE FOR THE EVALUATED GRSS IN MOVIELENS-100K

	Group size									
GRS	1	2	3	4	5	6	7	8	9	10
Baseline	1.0009	1.0199	1.0262	1.0308	1.0317	1.0323	1.0342	1.0364	1.0381	1.0390
Pre-GROD	1.0009	1.0199	1.0200	1.0240	1.0253	1.0263	1.0287	**1.0310**	**1.0329**	**1.0341**
GROD	1.0009	**1.0019**	**1.0141**	**1.0215**	**1.0239**	**1.0257**	**1.0286**	1.0312	1.0332	1.0345

best configuration of the proposal uses Asymmetric Cosine to compute the relationships between member preferences.

E. Experiment 2: GROD in Groups Without Consensus

This experiment aims to evaluate the improvements achieved by ensuring consensus. To do this, we evaluate the GRSs on groups without opinion leaders by isolating the effect of correcting the weights matrix and measuring the improvement that it provides. In this experiment, we compared three GRSs.

1) A GRS based on recommendation aggregation, the baseline.
2) The proposal without consensus, Pre-GROD.
3) The proposal ensuring consensus, GROD.

Tables VIII and IX show the MAE and RMSE for MovieLens-100k, respectively [see also Fig. 6(a) and (b)]. The results are stratified by group size to determine the improvement of the proposal for each case.

The results show that GROD improved the results over the baseline on groups without consensus for all group sizes. Moreover, GROD improved the results over Pre-GROD for most group sizes evaluated. The statistical tests confirmed that most of the differences observed in Tables VIII and IX are statistically significant. These facts indicate that correction improves recommendations in this scenario.

Furthermore, both measures show that GROD's improvement is greater in smaller groups. This improvement holds for all groups in MAE and for group sizes lower than 7 in RMSE, indicating that GROD is suitable for recommending items to small groups.

An interesting fact is that, for groups of two members, the proposal without correction obtained exactly the same MAE and RMSE as the baseline. The recommendation was the same because the prediction for groups without consensus is computed from the average of the final opinions, as stated in Section IV-A. Given that, in groups of size 2 that do not reach consensus, the weights between both users are necessarily zero, and the opinion of both members does not change in the process, therefore the mean of the individual recommendations is the prediction for the group.

On the other hand, the recommendations obtained by the baseline and GROD for groups sizes of three or more are not the same, because the opinion dynamics process modifies the individual recommendation of the individuals until they reach a stability point. At this point, there are at least two different values in the final opinions. In cases where the number of different final opinions is equal to the group members, the opinion dynamics process does not change the individual predictions, hence the average value is the prediction. However, when there are less different opinions, the opinion dynamics

TABLE X
EXPERIMENT 2: MAE FOR THE EVALUATED GRSs IN MOVIELENS-1M

	Group size									
GRS	1	2	3	4	5	6	7	8	9	10
Baseline	0.6941	0.7359	0.7508	0.7555	0.7667	0.7775	0.7830	0.7854	0.7872	0.7889
Pre-GROD	0.6941	0.7359	0.7378	0.7418	0.7538	0.7652	0.7715	0.7756	0.7782	0.7807
GROD	0.6941	**0.7069**	**0.7267**	**0.7371**	**0.7512**	**0.7635**	**0.7703**	**0.7746**	**0.7777**	**0.7803**

TABLE XI
EXPERIMENT 2: RMSE FOR THE EVALUATED GRSs IN MOVIELENS-1M

	Group size									
GRS	1	2	3	4	5	6	7	8	9	10
Baseline	0.8776	0.9246	0.9414	0.9480	0.9608	0.9728	0.9787	0.9818	0.9839	0.9859
Pre-GROD	0.8776	0.9246	0.9271	0.9333	0.9471	**0.9603**	**0.9669**	**0.9711**	**0.9743**	**0.9771**
GROD	0.8776	**0.9062**	**0.9225**	**0.9324**	**0.9470**	**0.9603**	**0.9669**	0.9712	0.9745	0.9772

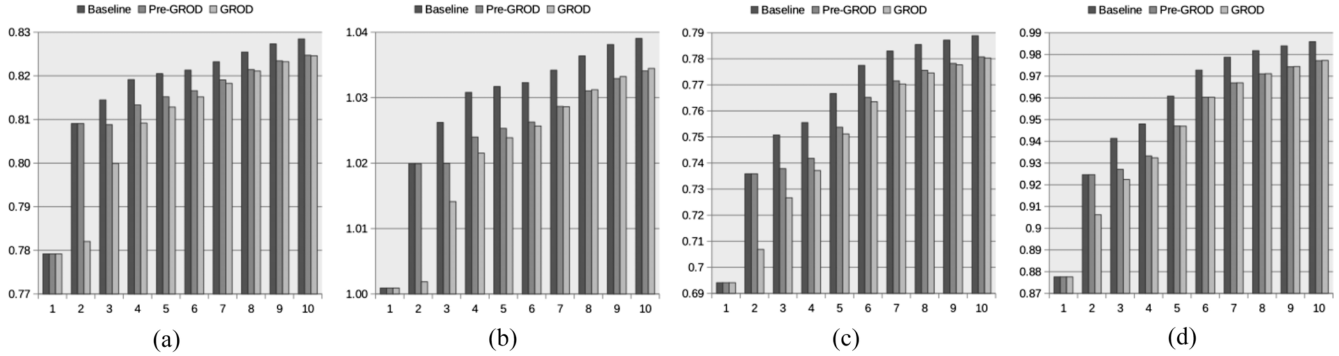


Fig. 6. Results for the evaluated GRSs by group size. (a) MAE in MovieLens-100k. (b) RMSE in MovieLens-100k. (c) MAE in MovieLens-1M. (d) RMSE in MovieLens-1M.

TABLE XII
COMPUTATIONAL COMPLEXITY OF PROPOSAL TASKS

Task	Complexity	Baseline	Pre-GROD	GROD
Individual predictions	$\mathcal{O}(gIk)$	✓	✓	✓
Similarity matrix	$\mathcal{O}(g^2R_g)$		✓	✓
Correct similarity	$\mathcal{O}(g^3)$			✓
Group prediction	$\mathcal{O}(I)$	✓	✓	✓

TABLE XIII
COMPUTATIONAL COMPLEXITY OF EVALUATED APPROACHES TASKS

GRS approach	Computational complexity
Baseline	$\mathcal{O}(max(gIk, \quad\quad\quad)$
Pre-GROD	$\mathcal{O}(max(gIk, \; g^2R_g \quad)$
GROD	$\mathcal{O}(max(gIk, \; g^2R_g, \; g^3 \;)$

process changes the individual predictions for some members. Therefore, the final prediction is not the mean value.

Tables X and XI show the MAE and RMSE, respectively, for MovieLens-1M [see also Fig. 6(c) and (d)]. The results are stratified by group size to determine the improvement of the proposal for each case. The results show similar behavior to that observed for MovieLens-100k. Moreover, the statistical tests confirmed that most of the differences in Tables X and XI are statistically significant.

The results show that the correction of the weights improves the results in both evaluation measures for most group sizes. For both measures, the improvement is greater for smaller groups, which makes it useful for recommending items to small groups, i.e., up to seven members.

F. Computational Complexity

The computational complexity of the evaluated approaches is shown in Table XIII, where k denotes the number of factors in the SVD model, g is the group size, I is the number of items, R_g is the largest amount of ratings of a group member, and q is the number of iterations that the DeGroot model needs to converge. The detailed computational complexity of each task is shown in Table XII.

In a real RS, and in the experiments carried out in this paper, the group size is much smaller than the number of ratings that a user might have, therefore the relationships matrix computation has a higher computational complexity than its correction. This gives GROD a higher computational complexity than Pre-GROD, which is justified by its higher performance in terms of accuracy.

V. SENSITIVITY ANALYSIS

In large-scale systems, computing recommendations is resource consuming. In this scenario we use a sensitivity analysis to determine whether recommendations need to be

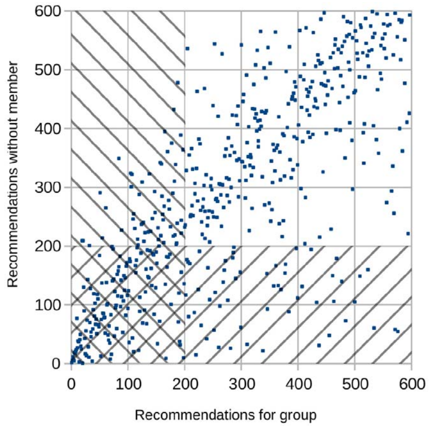


Fig. 7. Changes in the item ranking for group [126, 595, 607, 619, 648] and when user 126 leaves.

updated when inputs change, or whether the recommendation stands because the input changes do not influence the result. This way, certain computations can be avoided to reduce system load and resource needs.

In group recommendation scenarios, there are two main inputs: group members and their ratings. Hence, we consider two sensitivity analyses. The first studies system outputs when a user leaves the group. The second analyzes changes in the recommendation when a user changes their preferences. In both cases, the aim is to determine whether the recommendation must be recalculated or whether the same result can be delivered.

In both analyses, we measure changes to the recommendations when the system inputs change. Specifically, we check the order of the recommended items in the cases being compared, given that changes in the predictions may not change the item rankings within the recommendations. For example, assume that the system recommends products 1–3 with prediction values of 3.9, 3.8, and 3.7, respectively. In a second scenario, the recommended items are 1–3 with prediction values of 5, 4.9, and 4.8, respectively. And, in a third scenario the system returns the items ordered as 3, 2, and 1 with prediction values of 3.9, 3.8, and 3.7, respectively. In the second scenario the predictions change, but the order of the items is the same. However, in the third case, the prediction values change very little but the ranking of the items incur a great change. For this reason we consider rankings instead of prediction values.

Two measures are used to evaluate the changes in the recommendation.

1) *Intersection@size:* The ratio of items in the intersection among the recommended items. This is a measure of whether the recommendations for each input contain the same items.
2) *Spearman@size:* This measures the similarity of the rankings for the items in the intersection of both cases.

The application of these measures is depicted in Fig. 7, which shows the scatter plot of the ranking of items for the complete group and for the group without member 126. The rectangle filled with descending diagonal lines contains the items recommended to the complete group. The rectangle filled with ascending diagonal lines contains items recommended to the group without member 126. The area filled with both diagonal lines contains items recommended in both cases considering a recommendation list of size 200. The remaining area shows the items that are not recommended in either case. Therefore, following Fig. 7, Intersection@200 is the ratio of items in the common area and Spearman@200 analyzes their ranking correlation.

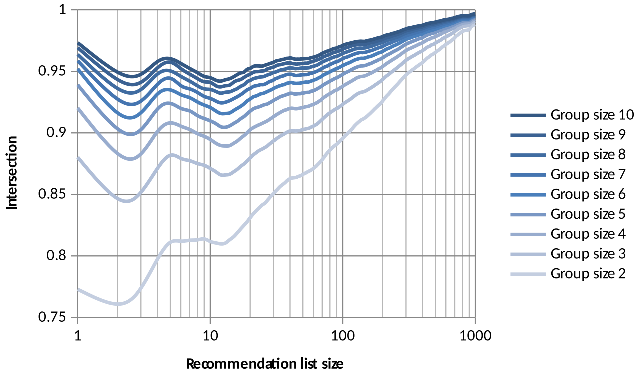


Fig. 8. Results of Intersection@Size for groups of size 2–10 to over the recommendation list.

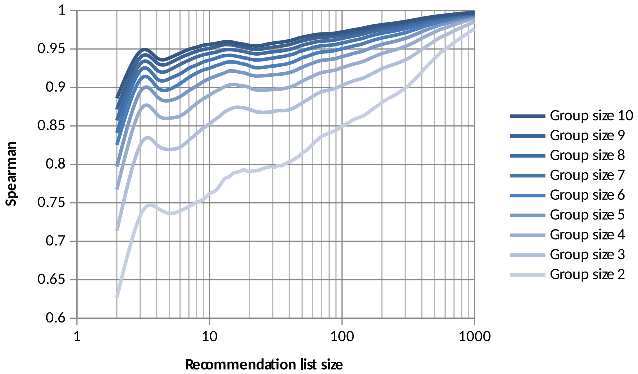


Fig. 9. Results of Spearman@Size for groups of size 2–10 to over the recommendation list.

A. Member Elimination

In this sensitivity analysis, we explore the influence of a member leaving the group. Our aim is to decide when a member leaving the group should trigger an update in the recommendations, or whether the same recommendations can be delivered without incurring a large error.

Figs. 8 and 9 show the results of the Intersection@Size and Spearman@Size for MovieLens-100k. The results are stratified by group size to compare the influence of the member's absence. Notice that the recommendation size is shown in a logarithmic scale to highlight the changes at the beginning of the list.

The results revealed that the changes produced by the absence of a member are higher in smaller groups. This finding was expected, since the fewer members the group has, the larger the influence a user's absence has. In large groups, the influence of a single member in the group recommendation process tends to diminish.

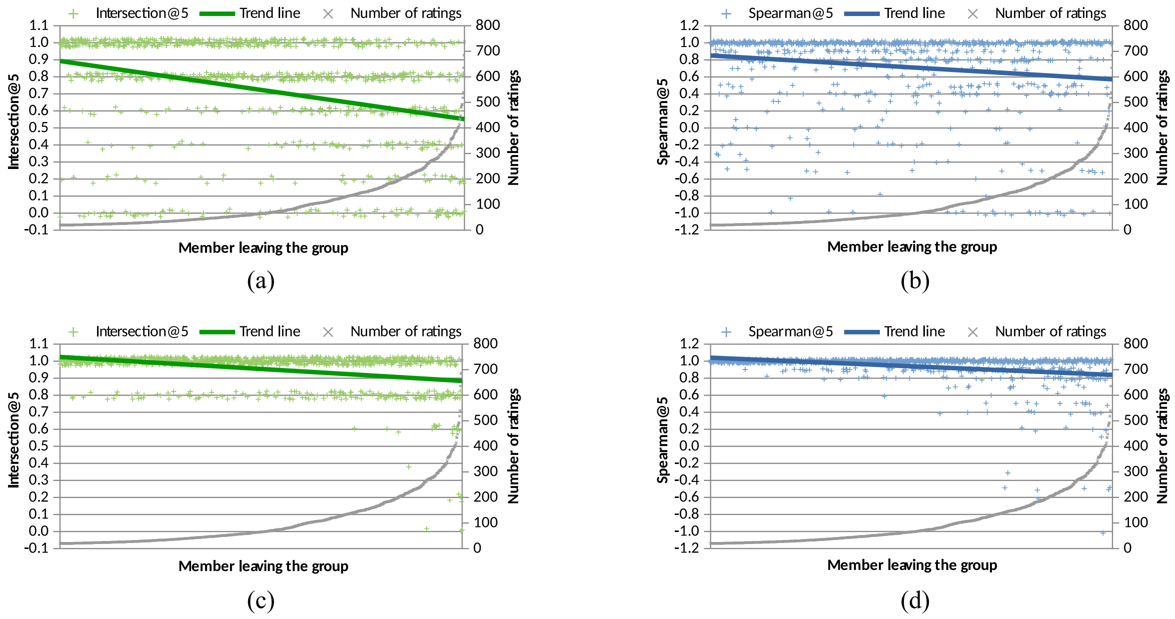


Fig. 10. Changes in the recommendation when member leaves the group sorted by number of ratings. (a) Group size 2, Intersection@5. (b) Group size 2, Spearman@5. (c) Group size 10, Intersection@5. (d) Group size 10, Spearman@5.

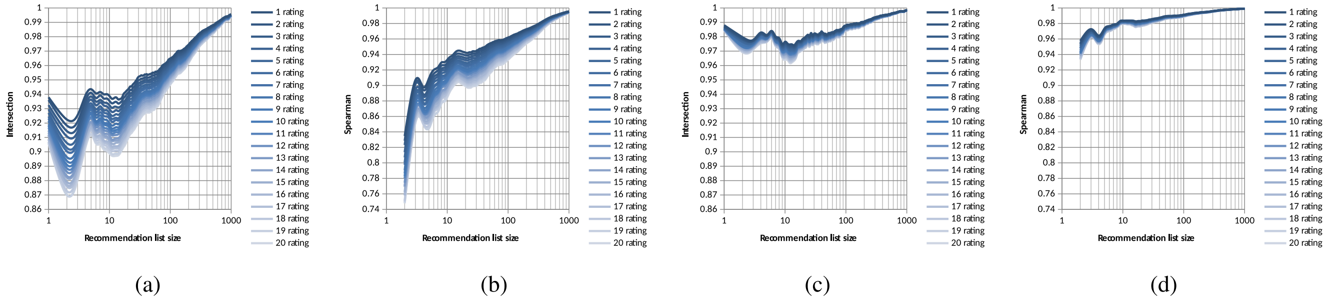


Fig. 11. Changes in the recommendation when member changes ratings. (a) Group size 2, intersection. (b) Group size 2, spearman. (c) Group size 10, intersection. (d) Group size 10, spearman.

Focusing on Intersection@size, the results show that the size of the recommendation list is related to the variability it has when the inputs change. For example, Fig. 8 shows that for all group sizes, recommendations for sizes between 10 and 20 have an Intersection@size lower than sizes between 20 and 30. This fact indicates that the former has more variability when a user leaves the group.

To focus on the impact of the member that is leaving the group given his/her characteristics, Fig. 10 shows the results of executions for each group, which checks recommendations for size 5 and for groups of size 2 and 10 on MovieLens-100k. The cases shown are sorted by the number of ratings of the user leaving the group. This highlights that the number of ratings influences the user's contribution to the final value. To reduce overlap in the data, a jitter value is added to Intersection@5 and Spearman@5.

Fig. 10 shows that the more ratings a member leaving the group has, the more impact its absence has on the recommendation, as highlighted by the trend line in each figure. This tendency is consistent across all group sizes. For the sake of simplicity, only groups sizes 2 and 10 are shown. In each case of a user leaving the group, there is an Intersection@5 and a Spearman@5 value. The data shows that the larger the number of ratings in the profile abandoning the group, the larger the change in the outputs. The rationale for this outcome is that the more ratings a user has, the higher the probability that this user is a leader.

To summarize, the impact on the recommendation is higher when the group is small, and/or when the user leaving the group has a large rating profile. With this evidence we suggest that the recommendations should always be recomputed when the group is smaller than five members. In larger groups, recommendations should be updated only when the user leaving the group has more than 220 ratings. These thresholds have been determined by finding the values for which the Intersection@size never drops below 0.9.

B. Rating Variation

Members' ratings are also other source of variability in group recommendation processes. This sensitivity analysis aims to measure the impact of a group member changing their mind about certain items. Again, the purpose is to determine whether the changes influence the recommendation and it should be updated, or whether the same recommendation can stand to save computational resources.

In this sensitivity analysis, one member of the group changes between 1 and 20 different ratings of items to a new random value. Various group sizes have been analyzed. Given that we are evaluating the results across four dimensions, Fig. 11 only shows the results for group sizes 2 and 10.

The results show that the smaller the group, the more the change has on recommendation. This behavior was expected since the relative weight of a single profile is greater when there are fewer members.

Similarly, the more ratings that are modified, the greater impact on the recommendations. This modification only affects the SVD profile of the user that modifies the ratings, while the remaining members have the same individual recommendations. Therefore, the opinion dynamics process has slightly different initial opinions, which results in a different recommendation. The more initial opinions in the group, the less impact the modified value has in the process. This behavior is consistent across all group sizes.

Overall, recommendations should not be recomputed when a user changes ratings, since the impact of the change is not large enough. The rationale for this is that even in cases with the most impact, such as a group of size 2 [Fig. 11(a)], the change is very small. In this case, Intersection@2 drops below 0.9 only when the user modifies 6 or more ratings. For larger group sizes, Intersection@size never drops below 0.9.

VI. Conclusion

This paper presents a framework to extend opinion dynamics and apply it to GRSs. The proposed framework considers the relationships between members' preferences in recommendations, which improves aggregation. Moreover, the framework ensures consensus in recommendations, which are agreed to by all group members.

Experiments show that the proposed framework improves recommendation results over the baseline. In the first experiment, Pre-GROD is evaluated with different similarity measures, and asymmetric similarities are proven to play an important role in the analysis of members' preferences. This indicates that asymmetry better reflects how the group makes decisions. The second experiment analyzes the effect of ensuring consensus by evaluating GROD in groups without consensus. The results show that ensuring consensus during the recommendation process improves individual satisfaction compared to both baseline and to the proposed framework without ensuring consensus.

A sensitivity analysis studies the impact of recommendations when inputs change to determine whether it is necessary to update them, or whether the same recommendations stand, without incurring in large errors, to conserve computational resources. Two changes are considered: 1) a member abandons the group and 2) a member varies their ratings. The first analysis shows that the recommendation can stay the same in large groups, or if the member leaving the group has a small rating profile. The second analysis shows that a member needs to change a large number of ratings to trigger a significant change in the group's recommendations.

This proposed framework considers that members update their opinions based on the relationships among their preferences. However, in future works, other opinion dynamics models can be explored, using different assumptions for the evolution of opinions. This would allow modeling of other features of the relationships, such as attitudes toward change.

References

[1] D. Wu, J. Lu, and G. Zhang, "A fuzzy tree matching-based personalized e-learning recommender system," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2412–2426, Dec. 2015.

[2] R. Y. Toledo and Y. C. Mota, "An e-learning collaborative filtering approach to suggest problems to solve in programming online judges," *Int. J. Distance Edu. Technol.*, vol. 12, no. 2, pp. 51–65, 2014.

[3] J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, "A Web-based personalized business partner recommendation system using fuzzy semantic techniques," *Comput. Intell.*, vol. 29, no. 1, pp. 37–69, 2013.

[4] D. Rafailidis and A. Nanopoulos, "Modeling users preference dynamics and side information in recommender systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 782–792, Jun. 2016.

[5] J. M. Noguera, M. J. Barranco, R. J. Segura, and L. Martínez, "A mobile 3D-GIS hybrid recommender system for tourism," *Inf. Sci.*, vol. 215, pp. 37–52, Dec. 2012.

[6] M. Al-Hassan, H. Lu, and J. Lu, "A semantic enhanced hybrid recommendation approach: A case study of e-government tourism service recommendation system," *Decis. Support Syst.*, vol. 72, pp. 97–109, Apr. 2015.

[7] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, and G. Lekkas, "Personalized finance advisory through case-based recommender systems and diversification strategies," *Decis. Support Syst.*, vol. 77, pp. 100–111, Sep. 2015.

[8] T. T. S. Nguyen, H. Y. Lu, and J. Lu, "Web-page recommendation based on Web usage and domain knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2574–2587, Oct. 2014.

[9] J. Xuan, X. Luo, G. Zhang, J. Lu, and Z. Xu, "Uncertainty analysis for the keyword system of Web events," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 829–842, Jun. 2016.

[10] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.

[11] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[12] N. Zheng, S. Song, and H. Bao, "A temporal-topic model for friend recommendations in Chinese microblogging systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 9, pp. 1245–1253, Sep. 2015.

[13] J. Masthoff, "Group recommender systems: Aggregation, satisfaction and group attributes," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. New York, NY, USA: Springer, 2015, pp. 743–776.

[14] F. Ortega, A. Hernando, J. Bobadilla, and J. H. Kang, "Recommending items to group of users using matrix factorization based collaborative filtering," *Inf. Sci.*, vol. 345, pp. 313–324, Jun. 2016.

[15] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices," *Appl. Artif. Intell.*, vol. 17, nos. 8–9, pp. 687–714, 2003.

[16] M. Hong and J. J. Jung, "Mymoviehistory: Social recommender system by discovering social affinities among users," *Cybern. Syst.*, vol. 47, nos. 1–2, pp. 88–110, 2016.

[17] Y. Dong, X. Chen, H. Liang, and C.-C. Li, "Dynamics of linguistic opinion formation in bounded confidence model," *Inf. Fusion*, vol. 32, pp. 52–61, Nov. 2016.

[18] M. H. DeGroot, "Reaching a consensus," *J. Amer. Stat. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[19] A. Felfernig *et al.*, *Group Decision Support for Requirements Negotiation*. Heidelberg, Germany: Springer, 2012, pp. 105–116.

[20] J. Castro, F. J. Quesada, I. Palomares, and L. Martínez, "A consensus-driven group recommender system," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 887–906, 2015.

[21] S.-M. Chen and B.-H. Tsai, "Autocratic decision making using group recommendations based on intervals of linguistic terms and likelihood-based comparison relations," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 250–259, Feb. 2015.

[22] L. Boratto and S. Carta, "Art: Group recommendation approaches for automatically detected groups," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 6, pp. 953–980, 2015.

[23] I. Cantador and P. Castells, "Group recommender systems: New perspectives in the social Web," in *Recommender Systems for the Social Web*. Heidelberg, Germany: Springer, 2012, pp. 139–157.

[24] M. Gartrell *et al.*, "Enhancing group recommendation by incorporating social relationship interactions," in *Proc. 16th ACM Int. Conf. Supporting Group Work (GROUP)*, Sanibel, FL, USA, 2010, pp. 97–106.

[25] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[26] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Inf. Retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[27] P. Pirasteh, D. Hwang, and J. J. Jung, "Exploiting matrix factorization to asymmetric user similarities in recommendation systems," *Knowl. Based Syst.*, vol. 83, pp. 51–57, Jul. 2015.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, 2001, pp. 285–295.

[29] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decis. Support Syst.*, vol. 87, pp. 80–93, Jul. 2016.

[30] A. Gunawardana and G. Sani, "Evaluating recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. New York, NY, USA: Springer, 2015, pp. 265–308.

[31] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

Jorge Castro, photograph and biography not available at the time of publication.

Jie Lu, photograph and biography not available at the time of publication.

Guangquan Zhang, photograph and biography not available at the time of publication.

Yucheng Dong, photograph and biography not available at the time of publication.

Luis Martínez, photograph and biography not available at the time of publication.