Centrality-based Group Formation in Group Recommender Systems

Hamidreza Mahyar* Vienna University of Technology hmahyar@cps.tuwien.ac.at
Elahe Ghalebi K.* Vienna University of Technology eghalebi@cps.tuwien.ac.at
S. Mojde Morshedi Sharif University of Technology mmorshedi@ce.sharif.edu
Saina Khalili Sharif University of Technology mkhalili@cs.sharif.edu
Radu Grosu Vienna University of Technology radu.grosu@tuwien.ac.at
Ali Movaghar Sharif University of Technology movaghar@sharif.edu

ABSTRACT

Recommender Systems have become an attractive field within the recent decade because they facilitate users' selection process within limited time. Conventional recommender systems have proposed numerous methods focusing on recommendations to individual users. Recently, due to a significant increase in the number of users, studies in this field have shifted to properly identify groups of people with similar preferences and provide a list of recommendations for each group. Offering a recommendations list to each individual requires significant computational cost and it is therefore often not efficient. So far, most of the studies impose four restrictive assumptions: (1) limited number of users, (2) number of groups, (3) average number of group members, and (4) full knowledge of the network topological structure. To overcome these limitations, we propose a novel approach which improves the accuracy of recommendations list to each group using network centrality concept. In this approach, the most central users are identified as heads of the groups, and then groups of users with similar preferences are consequently formed. After the group formation, a new group profiling strategy is provided to aggregate preferences of group members relative to their centralities. Our approach is evaluated in different types of group recommender systems compared to several common strategies over the MovieLens-1M dataset. Experimental results demonstrate that our group formation and group profiling, based on the proposed user centrality measure, lead to more accurate recommendations list for each group.

Keywords

Group Recommender Systems, Centrality, Social Networks

*Authors contributed equally

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3055363

1. INTRODUCTION

Massive Increase in the volume and availability of data on the Internet will certainly result in the difficulty of finding interesting choices among various options in a reasonable amount of time. *Recommender Systems* have emerged to solve this problem by helping people to find their preferences effortlessly. Conventional recommender systems are not useable in some domains and contexts when the items' recommendation process involves more than a person [16]. There are lots of situations in real life which demand recommendations to a group of users: A movie recommendation to be watched by a group of friends, or a recommendation for a city to be visited by a family. To this end, *Group Recommender Systems (GRSs)* have been recently proposed to recognize the items having high probability in satisfying a large number of group members [3, 4, 26]. In addition, a recommendation for a group of users can be considered as a solution to the *cold start* problem [25], as well as to *data sparsity* and also to *scalability*. The cold start problem is concerned with users who have recently logged in. Such users only have limited purchase capabilities and options, hence making recommendations to these users quite difficult. Moreover, this difficulty can also appear from an item's perspective: new items have been rated by few users, only. As a consequence, these items will be recommended with lower probability to other users [2, 31]. Data sparsity refers to a few number of available scores in comparison with a significant number of users and items [2]. This problem derives its name from the fact that users usually rate only a small portion of available items [13]. Thus, there is a sparse rating matrix with many zeros through thousands of rows and columns. Since a recommendation process is based on the users' scores, user behavior is hard to predict. Scalability is one of the main challenges for recommender systems. This challenge has received a lot of attention over the past years due to the quick growth in the number of users and items in many real-world systems. Typically, high computational cost is necessary to do recommendations given the massive data produced by the interactions of users with items such as ratings, preferences and reviews [28].

Although a large number of papers focused on recommending items to individual users, little work has been done so far in overcoming aforementioned problems, for group rec-

ommender systems [3, 7, 10]. For example in the cold start problem, a new user could join to a group, then use close preferences of users' group and their recommendation lists. In the data sparsity problem, the more the group recommender systems use the available information about group members, the better the recommendation list produced will be. To improve scalability, since recommendations list to a groups is not recomputed globally and this is done by incrementally updating the recommendation lists, joining new users to the system will be done with lower time complexity [19]. Recently, group recommender systems have been utilized in several applications, such as managing web sites and news pages [27], documents [6], tourist attractions [14], music tracks [10], books [17], television programs [30] and movies [10, 11]. Most of the previous works have developed group recommender systems with prior knowledge about group members, however in many realistic scenarios this information is not generally available [11]. As a consequence, one of the inputs of these systems is a group of predefined users, and only a restricted list of recommendations is presented to all members as output.

Given the growing number of users in many real-world social networks, each with various interests makes providing a list of accurate recommendations difficult. Consequently, group recommender systems have recently received much more attention. There are many situations confirming the need for identification of communities that include people of similar preferences. Because the calculation and prediction of a recommendations list for each individual is usually difficult, costly, and sometimes impossible. However, in such situations, the main goal is to provide a list of high-quality recommendations for a large number of users to maximize the users' satisfaction, without having prior knowledge about the relationship between them. It is noteworthy to mention that the interests of the people may be quite contrasting. For example, consider a large number of students from several countries that attend a conference and a recreational break is considered for them. If one recommendations list is presented to all of them, it will unlikely satisfy all of them. Hence, it should be possible to categorize these students in different groups according to their preferences. Regardless of the limited number of groups, this categorization should be conducted such that it satisfies all the students. These groups can be formed based on the ancient monuments, universities, research areas, or recreational centers of their countries. Similar situations can also be seen in large-scale systems. For example, there are some scenarios in the field of digital marketing confirming the need for finding appropriate communities in recommender systems with the aim of increasing profitability of the enterprises. To name a few of them, one can refer to the Amazon with more than 244 million active users according to the GeekWire report [1].

The idea of finding different user communities with similar preferences seems essential in order to properly organize recommendations of suitable products to users, when we are faced with a large volume of users and products. Each community is considered as a group of users and the list of products, in accordance with the preferences of each group, is recommended. Research in this field has accordingly shifted towards the automatic identification of groups of users with similar interests. Most of the previous studies have considered restrictive assumptions on: number of users, number of groups, average number of members in each group, and full knowledge of the network topological structure. One of the main open problems in group recommender systems is group formation to maximize user satisfaction.

In this paper, to tackle with the aforementioned challenges, we present a novel approach for group formation in group recommender systems which only requires as input a ratings matrix that included the items' scores evaluated by users. The proposed method is based on the concept of *network centrality* which is the term in social network analysis [20]. Centrality refers to identifiers that indicate the most important nodes in a network, and centrality measures quantify the role of nodes from different points of view. To summarize our contributions: (1) A novel group formation method in group recommender systems is proposed using only the ratings matrix without any assumption on the number of users, the number of groups, the average number of members in each group, and full knowledge of the network topological structure. (2) To the best of our knowledge, this work is the first to consider the centrality of users to enhance the recommendation lists by identifying groups of users with similar preferences. (3) A new strategy is proposed for building an accurate group profiling based on the centrality of users. (4) A bayesian similarity metric is used for creating an accurate graph representing similarity between users. (5) Our approach is implemented in different types of group recommender systems and evaluated over a real dataset. (6) A novel analysis is done on how growth of the number of groups and the average number of members in each group are affected by Cosine and Bayesian similarities.

2. RELATED WORK

In the literature, there are three different types of group recommender systems (GRSs) that can be categorized as: **First**, *Model Based* is a type of GRSs that first detects groups of users with similar preferences, then builds a model for each group by considering the preferences of its members, finally predicts group ratings using the model. **Second**, *Merge Recommendations* is another type of GRSs that first detects groups of users with similar tastes, then predicts individual preferences, finally selects the items with the highest predicted ratings for each user and constructs a group model based on union of these items. In fact, group preferences are predicted by modeling the top-l items. **Third**, *Predictions Aggregation* is the next type of GRSs that first detects groups of similar users, then predicts individual preferences, finally builds a group model via aggregating the scores assigned to each item by group members (class 1). In this type of GRSs, another class may exist that is initially responsible for predicting unrated items and thereafter forms the groups more precisely (class 2). A comprehensive review of the literature is given in table 1. It is obvious that each type of GRSs has three general steps: *Group Formation* (*i.e.* Identification of users with similar preferences as group members), *Group Modeling* (*i.e.* Aggregation of group members' preferences), and *Prediction* (*i.e.* Prediction of unrated items).

3. PRELIMINARIES

3.1 Basic Notations and Definitions

Consider network $G = (U, I, R)$ where U represents the set of users with cardinality $|U| = n$, and I is the set of items

Table 1: Review of the related work in group recommender systems

Model Based	Step1- Group Forming			Step2- Group Modeling	Step3- Ratings Prediction for Groups	
	Approach	Method	Similarity Metric	Strategy	Method	Similarity Metric
[11]	Community	Modularity	Cosine	Average Strategy	Item Based Nearest Neighbor Collaborative Filtering	Adjusted cosine Similarity
[12] [10]	Clustering	K-means	Cosine	Average/ Borda Count/ Approval voting/ Least misery/ Most Pleasure/ Average without misery	Item Based Nearest Neighbor Collaborative Filtering	Adjusted cosine Similarity
[7]	Approximation	Greedy Algorithm	-	Average Strategy/ Least Misery	-	-

Merge Recommendations	Step1- Group Forming			Step2- Individual predictions		Step3- Group Modeling
	Approach	Method	Similarity Metric	Method	Similarity Metric	Strategy
[10]	Clustering	K-means	Cosine	Classic User Based Nearest Neighbor Collaborative Filtering	Pearson's correlation	Recommendation top-l items

Predictions Aggregation (class 1)	Step1- Group Forming			Step2- Individual predictions		Step3- Group Modeling
	Approach	Method	Similarity Metric	Method	Similarity Metric	Strategy
[26]	Clustering	Hierarchical Agglomerative/ Friends Finder	-	Classic User Based Nearest Neighbor Collaborative Filtering	Pearson's correlation	Least misery/ Fair/ Most Optimistic Strategies
[8] [10]	Clustering	K-means	Cosine	Classic User Based Nearest Neighbor Collaborative Filtering	Pearson's correlation	Average/Borda Count/ Approval voting/Least misery/ Most Pleasure/Average without misery

Predictions Aggregation (class 2)	Step1- Individual predictions		Step2- Group Forming			Step3- Group Modeling
	Method	Similarity Metric	Approach	Method	Similarity Metric	Strategy
[8] [10]	Classic User Based Nearest Neighbor Collaborative Filtering	Pearson's correlation	Clustering	K-means	Cosine	Average/Borda Count/ Approval voting/ Least misery/ Most Pleasure/Average without misery

with cardinality $|I| = m$. R is an $n \times m$ ratings matrix, where r_u^i is the rate of user $u \in U$ to item $i \in I$ and $r_u^i \in [r_{min}, r_{max}]$. R^I typically contains a discrete set of positive integers. $r_u^i = 0$ is defined as an unrated item. r_u is the vector of item scores rated by user u and $\overline{r_u}$ represents the average of these scores. U_i is the subset of users that have expressed a preference for item i, and U_{ij} represents the set of users that rated both items i and j ($U_i \cap U_j = U_{ij}$). I_u is the subset of items rated by user u and I_{uv} represents the set of items evaluated by both users u and v ($I_u \cap I_v = I_{uv}$).

Definition 1: similarity graph of users. Suppose $\mathcal{G} = (V, E)$. Vertices (nodes) V represent users with cardinality $|V|$, and edges (links) E between vertices with cardinality $|E|$ represent the similarity preferences among users. The presence or absence of edges is determined by setting a threshold θ for the similarity between two users. The similarity graph $\mathcal{G}$ can be displayed by adjacency matrix Adj as:

$$Adj(u,v) = \begin{cases} 1 & Sim(u\ ,\ v) \ \geq \theta \quad (u,\ v \text{ are similar}) \\ 0 & Otherwise \end{cases} \tag{1}$$

where $Sim(u, v)$ is used for exposing similarity interests between two users (two similarity functions are defined in Section 4). The users who are linked to each other in the similarity graph are considered as *neighbors*. N_u represents the neighborhood of user u. $|N_u|$ shows the degree of node u (the number of its neighbors).

Definition (2): user centrality. Centrality measure indicates the importance of users which is based on their position in the similarity graph. Centrality of user u is [32]:

$$C_u = \frac{B_u}{\frac{2E_u}{(|N_u|)(|N_u|-1)}} \tag{2}$$

where E_u is the total number of edges between neighbors of u. B_u denotes the betweenness centrality of u, as [18]:

$$B_u = \sum_{v,w,v\neq w} \frac{\sigma_{vw}(u)}{\sigma_{vw}} \tag{3}$$

where σ_{vw} is the total number of shortest paths between nodes v and w, and $\sigma_{vw}(u)$ is the number of those that pass through node u.

Definition (3): group of users. Consider $G_x \subseteq U$ as a group of users where $x = \{1, 2, ..., k\}$ such that $\bigcup_x G_x = U$ and $G_x \cap G_y = \phi$. Each member of a group is similar to at least one member of that group: $\forall u \in G_x,\ \exists v \in G_x \Rightarrow v \in N_u$. $N_u^{G_x}$ represents the neighbors of user u in the group $G_x$with cardinality $|N_u^{G_x}|$.

Definition (4): group centralization. $C_{Gx} \in [0, 1]$ is representative for the centralization level of group G_x, as [32]:

$$C_{Gx} = \frac{\sum_{u\in G_x} (C'_u - C_u)}{(|G_x| - 1)\max(C'_u - C_u)} \tag{4}$$

where $C'_u = \max_{u\in G_x} C_u$ represents centrality of user u and $|G_x|$ represents the number of group members. According to Equation (4), if all users in a group hold the same centrality, the group is non-core. This means that all group members have the equal importance and impact. In the same way, if a user has the maximum centrality and other users have the same values, the maximum centrality for that group will exist. In the proposed method we utilize this measure to calculate the centrality of a group.

Definition (5): group model/profile. The aggregation process of group member's profile in the same group is usually called group model/profile [25]. As an example, the following user-ratings matrix R can be mapped to group rating matrix G^I in which every row of G^I is aggregated from some rows in R. It should be noted that g_x^i is used for explaining a vector of estimated item's score i for group G_x.

$$R = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & \dots & i_m \end{array} \\ \begin{pmatrix} 3 & 0 & \dots & 0 \\ 0 & 4 & \dots & 5 \\ 2 & 5 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 4 \end{pmatrix} \end{array} \Longrightarrow G^I = \begin{array}{c} \\ G_1 \\ G_2 \\ \vdots \\ G_K \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & \dots & i_m \end{array} \\ \begin{pmatrix} 0 & 3 & \dots & 0 \\ 2 & 0 & \dots & 4 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 1 \end{pmatrix} \end{array}$$

Some of the most popular strategies for building group profile are as follows [9]:

– *Average (Ave. Strategy):* The score assigned to an item i for group G_x is equal to the average score of item i that is

evaluated by all users in G_x: $g_x^i = \frac{\sum_{u \in G_x} r_u^i}{|G_x|}$.

– *Average without Misery (Ave. W. Misery):* The score assigned to an item i for group G_x is equal to the average score of item i that has been evaluated by all users in G_x, where each item score is greater than or equal to a certain threshold *(e.g.* $\gamma = 4$): $g_x^i = \frac{\sum_{u \in G_x} r_u^i}{|G_x|},\ \forall r_u^i \geq \gamma$.

– *Least Misery (Least Misery):* The score assigned to an item i for group G_x is equal to the minimum score of the item i evaluated by users in G_x: $g_x^i = \min_{u \in G_x} r_u^i$.

– *Most Respected Person (Most Res. Per):* The score assigned to an item i for group G_x is equal to the score of the item i evaluated by user u in G_x: $g_x^i = r_u^i,\ u \in G_x$.

– *Most Pleasure Person (Most Ple. Per):* The score assigned to an item i for group G_x is equal to the maximum score of the item evaluated by users in G_x: $g_x^i = \max_{u \in G_x} r_u^i$.

3.2 Problem Statement and Motivation

Group formation or how to place users in appropriate groups is the fundamental problem in group recommender systems. Users should be placed in a group such that its preferences is similar to the other members of that group. Since members of each group receive a similar recommendation list, a proper group formation method should be accurate in order to satisfy most of the users in the group. In real-world GRSs, this problem is raised by receiving only a ratings matrix, including users and items without any prior knowledge about the number of groups, the average number of members in each group, and full knowledge of the network topological structure. Among the main motivations of this paper to tackle with these issues, we should refer to the absence of proper method in the literature of group recommender systems to find optimal groups of users with similar preferences, in order to maximize the users' satisfaction.

3.3 Model and Problem Formulation

The common input of many classical recommender systems is the user ratings matrix $R \subseteq U \times I$, which shows set of users $U = \{u_1, u_2, \ldots, u_n\}$ and set of items $I = \{i_1, i_2, \ldots, i_m\}$. The output of such systems is the recommendation list L_{G_x} which is assigned to the group G_x. A high quality recommendation list satisfies group members and has the following properties: (1) each of these lists contains $|L_{G_x}|$ items with the highest estimated scores. (2) the previously rated items by a user are not used for recommendation to that user again even when that item is in its group recommendations.

4. PROPOSED METHOD

In this section, we propose a novel approach to enhance the recommendation process in real-world systems without restrictive assumptions and priori knowledge such as number of groups, average number of group members, and full knowledge of the network topological structure. Since the proposed method is based on the recognition of users' similarity in their preferences, Section 4.1 initially describes how to create a social network from the ratings matrix. The proposed group formation method in Section 4.2 represents the method of detecting appropriate groups based on users' centrality, because we believe that the most effective individuals among users can be considered as an appropriate foundation in forming the groups of similar users. Section 4.3 contains our group modeling method for properly aggregating group members' preferences. According to the experimental results in Section 5.4.3 the groups formed based on these individuals and the model constructed for these groups will lead to a more accurate recommendations list. Our proposed method is evaluated in various types of group recommender systems in Section 4.4.

4.1 A Social Network Corresponding to The Ratings Matrix

At the beginning, it is necessary to take a series of processing steps on the system's input $G = (U, I, R^I)$ described in Section 3.1. The output of these operations is a similarity graph $\mathcal{G} = (V, E)$ where vertices represent users, and links represent similarity among users as described in Definition (1). To determine if any two users hold similar preferences, a standard metric is required for the comparison of users' preferences. Most of the studies [10, 11, 11, 12] have used the *Cosine Similarity (CS)* metric to determine the degree of similarity between individuals. This metric is defined as:

$$CS(r_u, r_v) = \frac{(r_u \cdot r_v)}{\| r_u \|_2 \times \| r_v \|_2} \tag{5}$$

where $r_u = (\ r_u^1,\ r_u^2,\ \ldots\ ,\ r_u^m)$ is the vector of ratings that user $u \in U$ has given to different items. " $\cdot$ " denotes the dot product (also called the Euclidean inner product) of two vectors, and $\|.\|_2$ represents the Euclidean norm of a vector [5]. Expanding the various products one can get:

$$CS(r_u, r_v) = \frac{\sum\limits_{i \in I_{uv}} r_u^i \times r_v^i}{\sqrt{\sum\limits_{i \in I_{uv}} {r_u^i}^2} \times \sqrt{\sum\limits_{i \in I_{uv}} {r_v^i}^2}}. \tag{6}$$

Based on Equation (6), the similarity between users u and v is calculated by using the vectors of given scores. Although this criterion suffers from considerable shortcomings [15], such as the flat-value problem, the opposite-value problem, the single-value problem, and the cross-value problem, it enables us to compare the proposed approach with previous works. According to the problems of CS and their significant influence on performance, we also use the *Bayesian similarity (BS)* metric to infer a more accurate graph.

The Bayesian similarity metric, proposed in [15], is based on the Dirichlet distribution. This mediator considers both direction (rating distances) and length (ratings amount) of rating vectors and compares the preferences of users u and v more precisely. The *rating distance* is defined as $d_{lev} = |r_u^i - r_v^i|$ where lev represents the level of rating distance. According to this metric, the degree of similarity between users u and v is obtained from the following equation:

$$BS(u, v) = \max\left(BS'_{u,v} - BS''_{u,v} - \delta, 0\right) \tag{7}$$

where $BS'_{u,v} = 1 - \frac{d_{u,v}}{d_{max}}$ is called the *raw similarity* between users u and v. In this relation, $d_{u,v}$ is considered as *users' distance* and is calculated by the weighted average of rating distances d_{lev} according to their importance weights; and d_{max} represents the *maximum rating* distance based on the rate interval. Up to this part of the formula, the calculation of the similarity between users is based on the distributions of rating distances. In $BS''_{u,v} = \prod\limits_{l=1ev}^{max} \left(\frac{\alpha_q}{\alpha_0}\right)^{\gamma_{lev}^0}$, parameter

α_q is considered as the amount of ratings distances located in level d_{lev} and α_0 is the total amount of prior ratings pair. The ratio of $\frac{\alpha_q}{\alpha_0}$ is representative of the prior probability of rating pairs with ratings distance d_{lev}. The *comparison vector* of users' scores can be defined as $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_{lev})$ in which $\gamma_{lev} = 1$ if and only if other values are equal to zero where $d_{lev} = |r_u^i - r_v^i|$. For example, a rating pair $(1, 4)$ on a certain item i can be represented as $\gamma = (0, 0, 0, 1, 0)$ if the rating interval is between 1 and 5. γ^0 represents the i-th component of the j-th observation γ_j and γ_{lev}^0 is the amount of evidences that their rating distances are in the level d_{lev}. In the formula pertaining to the calculation of Bayesian similarity, *user bias* has also been taken into account which is represented by δ.

4.2 Proposed Group Formation Method

The similarity graph and the groups consisting of users with similar preferences constitute input and output of our group formation algorithm, respectively. We adapt the algorithm from [32] which combines heuristic with optimization methods to detect communities based on the core influence of nodes (central nodes) in complex networks. The network node centrality evaluates the user's position in the similarity graph. The most central people are not necessarily the ones that are similar to a high number of users, but they are specified based on the way they are placed in the similarity graph of users. This indicator is obtained by combination of the betweenness centrality as a global measure and the local clustering coefficient as a local measure. This metric describes how important a user is in its community.

Group formation begins with arranging all the users in descending order based on their centrality value in Equation (2). Then, the first user with the highest centrality is selected and placed in the first group together with its all neighbors. After formation of the first group, the second user with the highest centrality is selected. If this user is not already included in the previous group, the user will be considered as the core of the second group. Otherwise, it is necessary to calculate the importance of this user in the current group, and then decide whether the user should be considered as the core of the new group or stay in the former group. The K-function is used to make this decision [32]:

$$K_u = \frac{\frac{|N_u^x|}{|E|}}{\frac{|N_u|}{\sum_{u \in U} |N_u|} \times \frac{\left(\sum_{u \in G_x} |N_u|\right) - |N_u|}{\left(\sum_{u \in U} |N_u|\right) - |N_u|}} \tag{8}$$

where $|N_u^x|$ denotes the number of neighbors of user u in the group G_x to which user u belongs, $|E|$ is the total number of edges in the whole network, $|N_u|$ represents the degree of user u, $\sum_{u \in G_x} |N_u|$ represents the sum of degrees of all users in the group G_x, and $\sum_{u \in U} |N_u|$ is the sum of degrees of all users in the network.

This function indicates that the smaller the value of K_u is, the more likely the user u will be the core of the next group. Similarly, the higher the value of K_u is, the more important the role of user u in its current group will be. For deciding whether a user should remain in the current group or should establish an independent group, a proper interval has to be considered for the K-function. In Experiment 1 of Section 5.4.1, the proper interval selection will be analyzed and the results show that the most suitable one should be between 1.7 and 2.1. After the formation of initial groups, some users can be located in more than one group. These users should be then placed in the group which its user has the highest number of links to the other members in that group. Thus, the user belongs to a group with more similar members. This process is repeated until all the users are placed in a proper group. The pseudo code of the proposed group formation method is shown in *Algorithm 1*.

Algorithm 1 The Proposed Group Formation Algorithm

Input: $\mathcal{G}(V, E)$
$\mathcal{G}(V, E)$: Similarity graph of users
C_u: Centrality of user u
G: Set of groups
G_x: Group x
$G(u)$: Set of groups which user u belong
K_u: K-function for decision making about user u
———— /*Constructing Initial Groups*/ ————
Foreach user $u \in V$ **do**
  C_u = calculate centrality of user u based on Equation (2)
end for
Sort all users in descending order based on C_u
$G = \phi$
$x = 1$
Foreach user $u \in V$ **do**
  if $\big(G(u) = \phi\big)$ **or** $\big(G(u) \neq \phi$ **and** $K_u \in$ Suitable range$\big)$ **then**
    Create a new group
    $G_x = \{u\} \cup N_u$
    $x = x + 1$
    G_x will be added to G
  else
    User u is in its proper group
  end if
end for
———— /*Constructing Final Groups*/ ————
if $|G(u)| > 1$ **then**
  Select G_y which has $\max_{G_y \in G(u)} N_u^{G_y}$ for user u
  $G(u) = G_y$
end if
Output: Groups of users with similar preferences

4.3 Proposed Group Modeling Method

The already established groups according to Section 4.2 constitute the input of this section that aims to build model or profile for the groups as an output. When the number of individuals who have rated a specific item in each group goes beyond the threshold level, the ratings of that item is calculated based on the selected strategy in Definition (5); otherwise, the ratings of that item should be predicted [8, 10–12]. The proposed group modeling strategy is based on the users' centrality, which is indeed a weighted strategy wherein the weight assigned to each user is proportional to its centrality, as follows:

$$g_x^i = \frac{\sum\limits_{u \in G_x} C_u \times r_u^i}{\sum\limits_{u \in G_x} C_u} \tag{9}$$

where g_x^i indicates the predicted score of item i for group G_x and r_u^i represents the score of item i evaluated by user u. The proposed strategy can be analyzed by using the centralization function in Definition (4). It is possible to claim that if the centralization level of a group tends to 0, then the results of the current proposed strategy will be more similar to the results of the "Average strategy". Because group

members have similar impacts on building the profile. If the centralization level of a group tends to 1, then the results will be closer to "Most Respected Person strategy". Because the most important member has a significant impact on group modeling process. The results of the Experiment 2 in Section 5.4.2 confirms this claim.

4.4 Applying The Proposed Methods to Different Types of GRSs

According to Table 1, group recommender systems have three general steps. We proposed new methods for group formation and group modeling steps. In this section the proposed methods are applied on different types of GRSs.

4.4.1 Model Based Group Recommender Systems

Now, the proposed group formation and the group modeling methods are investigated in this type of group recommender systems. Our group formation method finds the central people among all the users and forms the suitable groups of users. After that, the group model is built with a proper weighting based on the importance of the group members. Then, according to the conventional method of Item-Based Nearest Neighbor Collaborative Filtering [16] that has been used in [8, 10, 11], unrated items are also predicted. In such method, in order to predict p_x^i for the unrated item i by the group G_x, the ratings of the most similar items that was evaluated by that group g_x^j are considered. This similarity value is calculated based on *Adjusted Cosine Similarity* [9], which is defined as:

$$p_x^i = \frac{\sum_{j \in RatedItems(G_x)} ItemSim(i,j) \cdot g_x^j}{\sum_{j \in RatedItems(G_x)} ItemSim(i,j)} \tag{10}$$

where

$$ItemSim(i,j) = \frac{\sum_{u \subseteq U_{ij}} (r_u^i - \overline{r_u}) \cdot (r_u^j - \overline{r_u})}{\sqrt{\sum_{u \subseteq U_{ij}} (r_u^i - \overline{r_u})^2} \sqrt{\sum_{u \subseteq U_{ij}} (r_u^j - \overline{r_u})^2}} \tag{11}$$

U_{ij} is a set of users that are in the same group and rates both items i and j. $\overline{r_u}$ is the average of the scores rated by user u. $ItemSim(i,j)$ is used to express similarity between two items. In order to reduce the complexity of the algorithm, there are two solutions: First, a threshold limit should be considered for the similarity metric in Equation (11) and the similar items to the desired item should have $ItemSim$ more than the threshold limit. Second, a certain number of items that are more similar to the desired item can be calculated by Equation (11). In this paper, the second solution is used. Figure 1 shows the implementation of the proposed methods in model based group recommender systems.

4.4.2 Group Recommender Systems Based on Merging Individual Recommendations

The proposed group formation and group profiling methods are also implemented in group recommender systems based on merging individual recommendations. First, the proposed group formation method forms the proper groups based on users' centrality so that the central users consider as heads of the groups. After constructing the group profile, this step predicts a score p_x^i for item i that was not rated by user u, according to the conventional method of User Based Collaborative Filtering Approach [29] as:

$$p_u^i = r_u^i + \frac{\sum_{v \subseteq N_u} (UserSim(u,v)) \times (r_v^i - \overline{r_v})}{\sum_{v \subseteq N_u} UserSim(u,v)} \tag{12}$$

As mentioned earlier, neighbors of each user are the users that are linked together in the similarity graph. In Equation (12), $UserSim(u,v)$ is used to express similarity between users u and v, which is calculated by Pearson's correlation as [9]:

$$UserSim(u,v) = \frac{\sum_{i \subseteq I_{uv}} (r_u^i - \overline{r_u}) \cdot (r_v^i - \overline{r_v})}{\sqrt{\sum_{i \subseteq I_{uv}} (r_u^i - \overline{r_v})^2} \sqrt{\sum_{i \subseteq I_{uv}} (r_v^i - \overline{r_v})^2}} \tag{13}$$

where I_{uv} is the set of items rated by both users u and v. Being extracted from the main method of this algorithm, only the users with the same group are considered for prediction here, although the accuracy is reduced in comparison with the state that all the users are included. For each user, the ratings of unpredicted items are arranged in descending order and then top-l items are selected. The union of the items with the highest predicted ratings for any user constructs group model. The average ratings for an existing item in the list of most of the members in one group is considered for that item.

4.4.3 Group Recommender Systems Based on Aggregation of Individual Predictions

In this type of group recommender systems, we use the same subclasses as of [10] that aggregates individual predictions for group recommendation.

Class 1 - group forming and predict: The proposed group formation and group profiling methods are applied to this type of GRSs. All the tasks for the two first steps are the same as algorithms recently presented in Section 4.4.2. The only difference with the last type of group recommender systems is on the modeling process that considers all the predictions instead of top-l predicted items. In the third step, the group profiles are built based on the proposed group profiling strategy.

Class2 - Predict and group forming: This method has been presented to improve the formed groups, especially when there is significantly smaller number of rated items than the available items. In this type, the unrated item i by user u is predicted based on Equation (12), then the similarity graph is created on the basis of the completed ratings matrix. After that, the groups of users with similar preferences are detected based on the proposed group formation method. The only difference is that the input of this constructed graph is based on the completed ratings matrix. Consequently, the formed groups will be more accurate. Finally, group model is built based on the proposed group profiling strategy.

5. EXPERIMENTAL EVALUATION

5.1 Dataset

Since there is not dataset containing groups of users in the literature of group recommender systems [10], the common datasets for recommender systems are usually used.

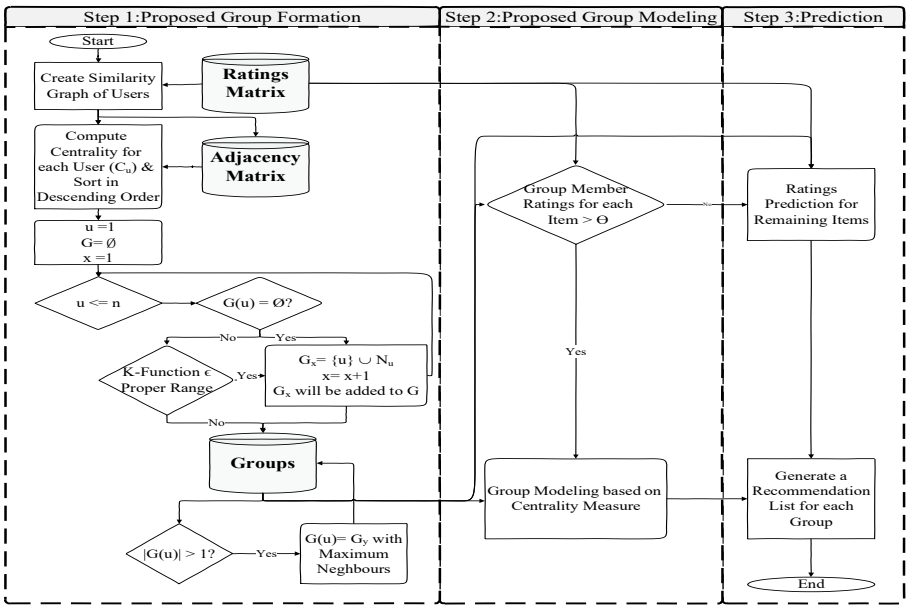

Figure 1: Applying the Proposed Methods to the Model Based GRS.

We evaluate the proposed methods over the MovieLens-1M dataset [25] that contains 1 million ratings of 3952 movies made by 6040 users. This dataset includes: User IDs with values between 1 to 6040 that represent the ID of each user; Movie IDs with values between 1 and 3952 that represent the ID of each movie; Ratings represent the scores assigned to movies by users with the range between 1 to 5. It should be mentioned that each user has rated at least 20 movies. Despite having 6040 users and 3952 items, only 1 million rates are available, thus, the sparsity of this dataset is about 95.81%.

5.2 Settings

For the experiments, the parameters discussed throughout the paper have been organized as follows:

– As mentioned in Section 3.1-1, a threshold level is required to determine the presence or absence of a link in the similarity graph which shows the similarity of two users. This value is considered 0.8 like other similar studies.

– In the calculation of the similarity between two users based on Bayesian criterion, parameter δ represents the value of user bias. This parameter is set to 0.04 according to [15].

– In the Group forming, K-function has been used to decide whether keep the users in their current community or create new one. According to the Experiment 1, the proper range for this function should be between 1.7 to 2.1.

– In the group modeling step, when the number of individuals who have rated a specific item in each group goes beyond the threshold level, the score of that item should be calculated; otherwise, the approximatef value for this score is predicted. The value of this threshold in [11] has been analyzed under the title of co-rating parameter. Accordingly, the amount of 10% is considered for the experiments.

– In the prediction step, a certain number of most similar items to the desired item in $ItemSim(i, j)$ are calculated to reduce the time complexity. These parameters have been analyzed under the name of top-l correlations and trust respectively in [10, 11]. Accordingly, in this paper, top-l correlation parameter is set to 10.

5.3 Metric

Root Mean Squared Error (RMSE) is a standard metric for model errors. This measure is used to assess the quality of predictions made by the proposed method. This criterion calculates the difference between the estimated ratings e_x^i, which represents the rating of i estimated by the proposed algorithm (in group modeling step g_x^i or prediction step p_x^i) for group G_x; and r_u^i, which represents the rating given to the item i by user u who is a member of group G_x. This criterion is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{r=0}^{t} (r_u^i - p_x^i)^2}{t}} \tag{14}$$

where t is the number of available rates in the test collection.

5.4 Evaluation Results

5.4.1 Experiment 1

In the proposed algorithm, K-function has been analyzed to control the amount of modularity or break-off of groups. This function manages the modularity by deciding on whether keep the users in their current community or create new community. Figure 2 shows modularity according to the relative error of recommendation lists to users. The constant number of users and the increase of the number of communities will lead to a drop in the average number of members of each and reduction of error. This experiment is an attempt to find an appropriate range for the optimal modularity in order to have the groups with a reasonable number of people with similar preferences, and then to offer a high quality recommendation list to each group. According to the Figure 2, the amount of error has experienced an ascending trend up to the value of 1.9 where the breaking degree of groups is reasonable up to this point. However, the modularity degree increases as a result of the formation of communities, especially single-user communities and the error experiences a descending fashion. Therefore, the most appropriate range is from 1.7 to 2.1 since a trade-off between group members and the quality of recommendation lists is desired.

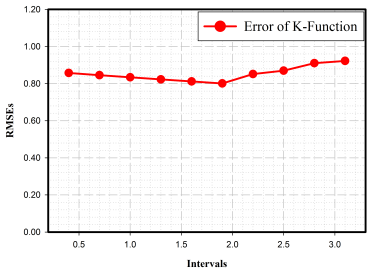

Figure 2: Analysis of the appropriate range for K-function.

Figure 3: Comparison of the proposed strategy with others.

5.4.2 *Experiment 2*

The proposed group profile strategy is a weighted average based on the centrality of each group member to construct a group model. Thus, $C_{Gx} = 0$ represents the similar centrality of all group members and the proposed strategy becomes similar to the average strategy. Placing a user in the core of a group ($C_{Gx} = 1$) results in a state similar to most respected person. These results are shown in Figure 3.

5.4.3 *Experiment 3*

In this experiment, the proposed approach is assessed in different types of group recommender systems with several strategies (Figure 4, Figure 5).

(1) *Comparison of the results for the proposed method in different group recommendation systems.* Figure 4 and Figure 5 represent the accuracy of recommendation lists in various types of group recommender systems with different strategies. The results show that the grouping based on users' centrality with weighted strategy leads to the best recommendation lists. Our proposed approach has the best results compared to all the studies done so far in this area [8, 10–12]. Figure 4 and Figure 5 obtained from the simulation of different GRSs. It leads to the understanding that the quality of second class of Predictions Aggregation is more accurate than other types of GRSs. This is due to the prediction of scores and formation of the input matrix in the first step, which results in the formation of more accurate groups with better recommendations.

(2) *Comparison of the proposed strategy with common strategies.* As it is shown in the simulated strategies in Figure 4 and Figure 5, the proposed strategy outperforms other strategies in structuring groups based on users' centrality. In addition, the recommendation lists in this method is more accurate than that of all the other methods. In group formation, the Least Misery strategy has often led to the worst results since this strategy tries to keep all the people satisfied. Therefore, it considers the smallest rated score as the score of each item in the group model that consequently increases the error.

Table 2: Comparison of the growth in the groups and the group members in various types of Group Recommender Systems

Metric	GRS Type	RMSE (our strategy)	Ave. growth in # of groups (per 100 users)	Ave. growth in # of members (per 100 users)
CS	Model Based	0.9095	≈ 7	≈ 1
	Merge Recommendations	1.8129		
	Predictions Agg.: *class 1*	0.9396		
CS	Predictions Agg.: *class 2*	0.2985	≈ 10	≈ 2
BS	Model Based	0.8876	≈ 3	≈ 7
	Merge Recommendations	1.8095		
	Predictions Agg.: *class 1*	0.8864		
BS	Predictions Agg.: *class 2*	0.2939	≈ 1	≈ 33

5.4.4 *Experiment 4*

As mentioned in Section 4, groups are formed according to the similarity graph and this graph is formed based on the similarity between any two users. In this paper, Cosine and Bayesian similarity metrics are used to measure the similarity of both users. Due to the problems mentioned in Section 4 in relation to Cosine criterion, Bayesian criterion has been selected to determine the similarity between two users. In the following, the advantage of this criterion over Cosine metric is displayed through the simulation of different types of group recommendation systems. The quality of the recommendation lists offered to the formed groups based on both Cosine and Bayesian criteria confirms the superiority of the proposed method and strategy.

Table 2 compares the accuracy of the results of the proposed strategy based on Bayesian and Cosine criterion. According to this table, the proposed method has ended in a high quality recommendation list based on both criteria in all types of group recommender systems. These results seem to be the most ideal ones among all the previously-obtained results. In addition, the increase in the number of groups and group members has been investigated for every 100 users. Looking carefully into the development groups and the members of each group, one will understand that growth of the number of groups in Cosine metric is greater than in Bayesian metric.

As a result, the growth of the number of group members is lower and the recommendation list for this criterion is slightly better than in Bayesian criterion. This is so while Bayesian criterion forms new groups with a much lower growth rate; therefore, a list with an accuracy very close to cosine metric is produced. Since the optimal number of groups is desired in large scale environments, Bayesian criterion will be a suitable choice. It is due to the fact that the lower number of groups and, as a result, accurate recommendation lists. This will help the scalability of the system.

6. DISCUSSION AND FUTURE WORK

Based on the result of Section 5, second class of predictions aggregation owns the most accurate recommendation list offered to users, but the completion of the input matrix in the large scale environment is followed by high complexity. It is possible to select this system and improve the complexity of this method for the enjoyment of the greatest accuracy in generating recommendation lists: (1) the employment of the algorithm with better time order to complete the ratings matrix. (2) attempt to reduce the ratings matrix dimensions whose scores should be predicted; for example, this can be accomplished by grouping of the items and considering some

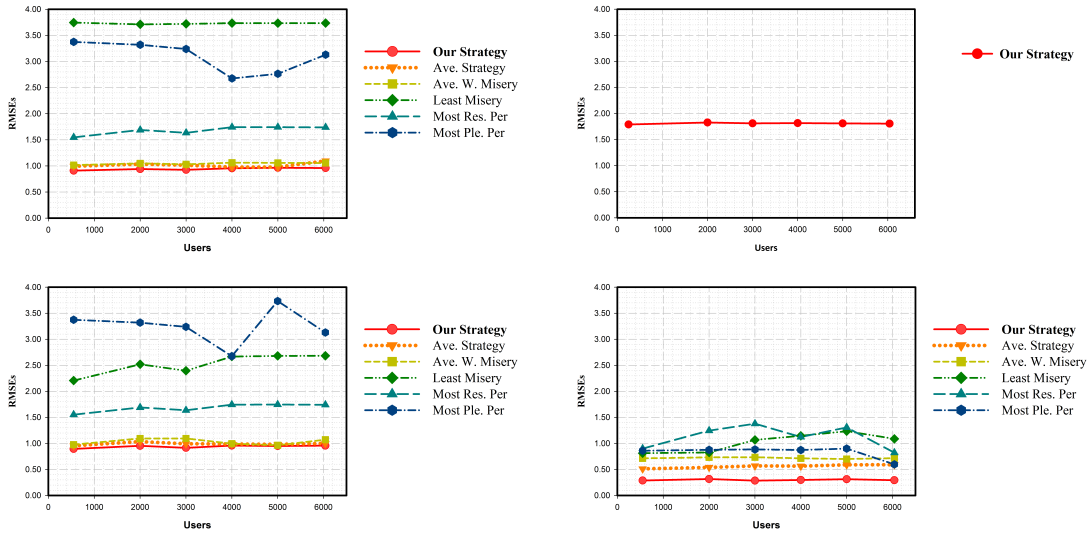

Figure 4: Comparison of the proposed strategy with other strategies with Cosine Metric in (a) Model Based (b) Merge Recommendations (c) Aggregation Predictions: G and P (d) Aggregation Predictions: P and G.

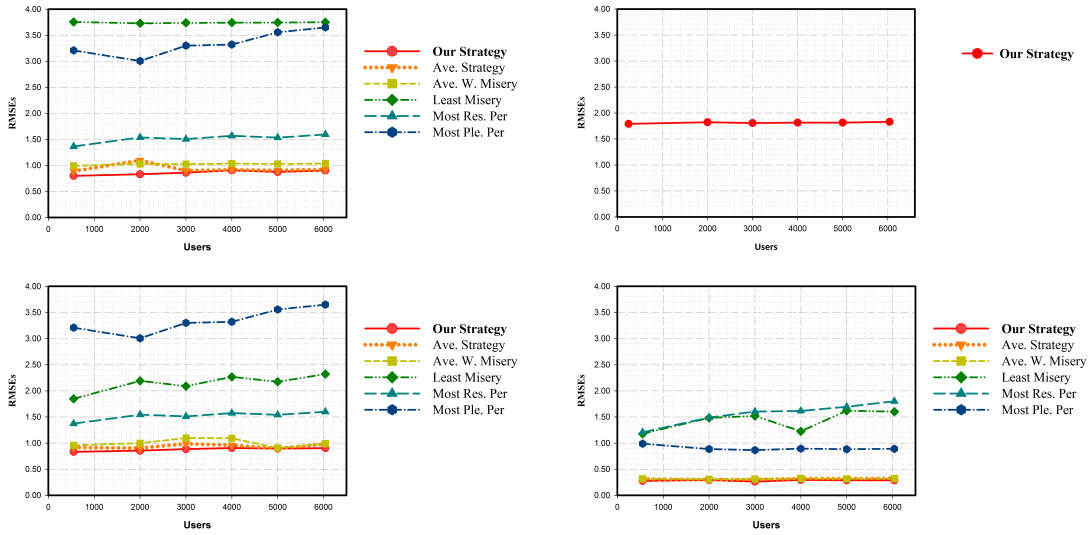

Figure 5: Comparison of the proposed strategy with other strategies with Bayesian Metric in (a) Model Based (b) Merge Recommendations (c) Aggregation Predictions: G and P (d) Aggregation Predictions: P and G.

items as representative of other items. In addition, one of the most impressive aspects of recommender systems is the scalability of these systems. Group recommender systems in the stated context can considerably help improve this aspect. In the proposed method, given that each group contains a model, each added user to the system can join to the appropriate group by comparing between the group models and his/her ratings vector. Therefore, it is possible to identify a series of initial and accurate groups in offline mode and, then, add new users to the formed groups. As a future work, we can identify central nodes for group formation in group recommender systems with indirect measurement using Compressive Sensing Theory [20–23]. Moreover, we can consider the corresponding social network as a weighted graph [24] such that the links are not only binary entities, either present or absent, but here associated a given weight that record their strength relative to one another.

7. CONCLUSION

With a considerable growth in the Internet users, a new aspect of group recommender systems (GRS) has received a lot of attention over the past years. In this paper, we present a novel approach to enhance the recommendation process in large systems without restrictive assumptions and priori knowledge such as number of groups, average number of group members, and full knowledge of network topological structure. To this end, we proposed our group formation and the group profiling methods based on the user centrality. Our extensive experimental s on the real-world Movielens-1M network demonstrated the accuracy and superiority of the proposed methods compared to other strategies in group formation and group profile steps of GRSs. Furthermore, the obtained results suggest that applying the proposed approach to the third type of GRSs has an accurate result in comparison with two other types.

References

[1] Iire. *http://www.geekwire.com/2014/amazon-adds-30-million-customers-past-year*.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[3] I. Ali and S.-W. Kim. Group recommendations: Approaches and evaluation. In *International Conference on Ubiquitous Information Management and Communication*, pages 105:1–105:6, 2015.

[4] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *Proc. VLDB Endow.*, pages 754–765, 2009.

[5] H. Anton. *Elementary Linear Algebra*. John Wiley & Sons, 2010.

[6] J. P. Baskin and S. Krishnamurthi. Preference aggregation in group recommender systems for committee decision-making. In *ACM Conference on Recommender Systems*, RecSys '09, pages 337–340, New York, NY, USA, 2009.

[7] S. Basu Roy, L. V. Lakshmanan, and R. Liu. From group recommendations to group formation. In *ACM SIGMOD International Conference on Management of Data*, pages 1603–1616, 2015.

[8] L. Boratto and S. Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. pages 1–20, 2011.

[9] L. Boratto and S. Carta. Modeling the preferences of a group of users detected by clustering: A group recommendation case-study. In *International Conference on Web Intelligence, Mining and Semantics*, WIMS '14, pages 16:1–16:7. ACM, 2014.

[10] L. Boratto and S. Carta. ART: group recommendation approaches for automatically detected groups. *Int. J. Machine Learning & Cybernetics*, 6(6):953–980, 2015.

[11] L. Boratto, S. Carta, A. Chessa, M. Agelli, and M. L. Clemente. Group recommendation with automatic identification of users communities. In *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 547–550, 2009.

[12] L. Boratto, S. Carta, and M. Satta. Groups identification and individual recommendations in group recommendation algorithms. In *PRSAT@ RecSys*, pages 27–34, 2010.

[13] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. pages 107–144, 2011.

[14] I. Garcia, L. Sebastia, and E. Onaindia. On the design of individual and group recommender systems for tourism. *Expert Syst. Appl.*, 38:7683–7692, 2011.

[15] G. Guo, J. Zhang, and N. Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *International Joint Conference on Artificial Intelligence*, pages 2619–2625, 2013.

[16] A. Jameson and B. Smyth. The adaptive web. chapter Recommendation to Groups, pages 596–627. Springer-Verlag, Berlin, Heidelberg, 2007.

[17] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu. A group recommendation system for online communities. *Int. J. Inf. Manag.*, 30(3):212–219, June 2010.

[18] N. Kourtellis, T. Alahakoon, R. Simha, A. Iamnitchi, and R. Tripathi. Identifying high betweenness centrality nodes in large social networks. *Social Network Analysis and Mining*, 3(4):899–914, 2013.

[19] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou. Recommender systems. *Physics Reports*, 519(1):1–49, 2012.

[20] H. Mahyar. Detection of Top-K Central Nodes in Social Networks: A Compressive Sensing Approach. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015, Paris, France*, pages 902–909, Aug. 2015.

[21] H. Mahyar, H. R. Rabiee, and Z. S. Hashemifar. UCS-NT: An Unbiased Compressive Sensing Framework for Network Tomography. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, Canada*, pages 4534–4538, May 2013.

[22] H. Mahyar, H. R. Rabiee, Z. S. Hashemifar, and P. Siyari. UCS-WN: An Unbiased Compressive Sensing Framework for Weighted Networks. In *Conference on Information Sciences and Systems, CISS 2013, Baltimore, USA*, pages 1–6, Mar. 2013.

[23] H. Mahyar, H. R. Rabiee, A. Movaghar, E. Ghalebi, and A. Nazemian. CS-ComDet: A Compressive Sensing Approach for Inter-Community Detection in Social Networks. In *IEEE/ACM ASONAM*, pages 89–96, Aug. 2015.

[24] H. Mahyar, H. R. Rabiee, A. Movaghar, R. Hasheminezhad, E. Ghalebi, and A. Nazemian. A Low-cost Sparse Recovery Framework for Weighted Networks under Compressive Sensing. In *IEEE International Conference on Social Computing and Networking, SocialCom 2015, Chengdu, China*, pages 183–190, Dec. 2015.

[25] J. Masthoff. Group recommender systems: Combining individual models. pages 677–702, 2011.

[26] E. Ntoutsi, K. Stefanidis, K. Nørvåg, and K. Hans-Peter. Fast group recommendations by applying user clustering. pages 126–140, 2012.

[27] S. Pizzutilo, B. De Carolis, G. Cozzolongo, and F. Ambruoso. Group modeling in a public space: Methods, techniques, experiences. In *WSEAS International Conference on Applied Informatics and Communications*, pages 175–180, 2005.

[28] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., 1st edition, 2010.

[29] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. The adaptive web. chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.

[30] S. Seko, T. Yagi, M. Motegi, and S. Muto. Group recommendation using feature space representing behavioral tendency and power balance among members. In *ACM RecSys '11*, pages 101–108, 2011.

[31] L. H. Son. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*, 58:87–104, 2016.

[32] C. Tong, J. Niu, B. Dai, and Z. Xie. A novel complex networks clustering algorithm based on the core influence of nodes. *The Scientific World Journal*, 2014.