# Hierarchy Visualization for Group Recommender Systems

Wei Wang, Guangquan Zhang, and Jie Lu, *Senior Member, IEEE*

*Abstract*—Most recommender systems (RSs), especially group RSs, focus on methods and accuracy but lack explanations, hence users find them difficult to trust. We present a hierarchy visualization method for group recommender (HVGR) systems to provide visual presentation and intuitive explanation. We first use a hierarchy graph to organize all the entities using nodes (e.g., neighbor nodes and recommendation nodes) and illustrate the overall recommender process using edges. Second, a pie chart is attached to every entity node in which each slice represents a single member, which makes it easy to track the influence of each member on a specific entity. HVGR can be extended to adapt different pseudouser modeling methods by resizing group member nodes and pseudouser nodes. It can also be easily extended to individual RSs through the use of a single member group. An implementation has been developed and feasibility is tested using a real data set.

*Index Terms*—Collaborative filtering, data visualization, group recommender systems (GRSs), recommender systems (RSs).

## I. Introduction

IN RECENT years, users have attempted to obtain a variety of personalized services from online providers [1]–[3]. However, rapid and massive information growth has made focusing on specific interests very difficult. Recommender systems (RSs) address this problem by providing personalized information and services for users, based on individual preferences.

Group RSs (GRSs) are proposed as a new form of RSs as online services targeting on groups of users are widely needed. Unfortunately, although many group recommender methods have been studied and used in practical applications, due to most GRSs [4]–[7] only focus on the recommendation list, but provide no reasoning for why the entries have been selected, users have trouble interpreting and/or trusting the systems.

Black box, which means only the final results (recommendations) are accessible, is a natural limitation of RSs for most recommendation methods put a priority on theoretical accuracy. It becomes more challenging and difficult to be addressed for GRSs, because: 1) the further complex recommender process (such as aggregating members' preferences) is more difficult to explain; 2) the vast information raised by multiple active users (group members) is hard to present; and 3) the relationships between group members are various such as a member may wonder the relationships between herself and other members.

Incorporating visualization into RSs [8] is beginning to be studied, and provides a summarized review [9] of some of these methods. These methods make the systems far easier to understand, and provides an effective means of allowing users to interact with systems, such as Knijnenburg *et al.* [10] used visually presents data and its relationships. Not surprisingly, visualization techniques have been used by many researchers to help users understand RSs in an efficient and inherently human way, and many combine a number of different graph layouts for more convincing systems, such as [11]–[13] in music, travel, and academic domains.

Despite these efforts, providing a visual explanation for GRSs is still challenging. First, most methods are proposed for individual RSs and are not appropriate for GRSs. In individual RSs, link-node layout graphs are widely used because they give the intuitive inheritance information presentation for procedures deriving recommendation from observed historical data. However, unlike having a single root node, i.e., the active user, in graphs for individual RSs, in GRSs, some recommendations linking member A may share no common nodes with member B. This means that some graphs for GRSs could be discrete and unorganized; therefore, a new layout is needed to handle all the possibilities. Second, a member may not only wonder about the recommendation information but also about the relationship information between herself and other members. In [14], for example, the distances between group consensus and individual member recommendations are mapped to a 2-D plane using SOM clustering, ensuring that a member can easily compare their preference with that of other members. However, most of these visualizations focus on the specific entity (recommendation in [14]) and lack information explaining the overall process.

In this paper, a novel hierarchy visualization method for group recommender (HVGR) is introduced to address these problems. First, it is the first attempt, to the best of our knowledge, to visualize the GRSs. A hierarchy graph-based model is developed to explain the group recommender process. This method presents the key information, including the detailed information, pseudouser' ratings neighbor similarities, and gives consideration to the possible discrete paths related to

The authors are with the University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: wangwei.sam@gmail.com; guangquan.zhang@uts.edu.au; jie.lu@uts.edu.au).

different members. The concise system structure gives a higher level of abstraction of the recommender procedures, which makes it easy to extend HVGR to other group recommender methods. For example, adapting the width of edges between member nodes (MNs) and pseudouser nodes allow our method to illustrate most group profile modeling methods. The method also allows members to gain an immediate understanding of their relative relationships from different aspects of the process (i.e., pseudouser profile, neighbors, and recommendations) by illustrating personal influence data using pie charts. Pie charts are attached to every entity node and each slice represents a specific member, thus members easily know whether they are directly connected to the node and can track their individual influence compared to other members along every related path. Moreover, members can gain insight with which they can gauge the quality of recommendations.

The contributions of this paper can be summarized as follows.

1) A HVGR is proposed. HVGR aims to provide more information and explanation than simple recommendations by presenting visual information for the group recommender process in an organized way by using multilevel nodes and edges. Experimental results show that this method is feasible on real data sets.

2) Member-based pie charts are introduced in the HVGR method to allow group members to understand their individual influences by mapping members into slices. Members will know whether they are directly connected to a specific node and will understand the relative relationships with other members on that node.

3) HVGR provides a high level of abstraction in recommender procedures, making it easy to extend and adapt the model to other group recommender methods.

The rest of this paper is organized as follows. Section II reviews visualization methods to RS. We provide a detailed description of our HVGR method in Section III. Implementation and experiment are described in Section IV. The conclusion and future work are presented in Section V.

## II. RELATED WORK

GRS aim to generate recommendations which are preferred by a group of users rather than an individual user. Group decision making (GDM) has been developed for many years to find the most acceptable decision. Many techniques are used in GDM including fuzzy GDM [15] and multicriteria GDM [16]. Many applications such as emergency risk management [17] use GDM to help users in making decisions. However, there are two main differences between GDM and GRS. Ratings or other types of feedback are provided in most GDM and GRS to specify individual opinions and they are aggregated into consensus suggestions, but in GDM they are often dense, whereas they are sparse in GRS, which means that different aggregation methods are needed in GRS. Another important difference is that after aggregation, GDM obtains the recommended solution, but GRS only obtains an aggregated group profile, and heuristic methods are used to generate recommendations.

Many methods have been used in GRSs, the most important two types of methods are group preference modeling methods and prediction calculating methods. Group preference modeling methods are used to generate a pseudouser to represent the overall preference, such as average and least misery-based aggregating methods. Collaborative filtering is the most popular method in GRSs to compute the unknown group ratings. Lu *et al.* [18] presented a review of many GRSs and described them in terms of their domains: 1) book, document, and webpage; 2) movie and music; 3) tourism; and 4) TV program. Despite their success in real applications, group recommender methods suffer a number of problems. One of the most significant—the inadequacy of explanations—means they lack persuasion. Design and employ visualization techniques, hybrid with RSs, can effectively address this issue. Many visualization techniques have been used to provide an instinctive understanding of the system and reveal deep-level relationships in data. Systems that show results as graphs are known to engender more trust, and therefore more loyalty, from its owners and users.

Even simple graphs such as line charts and bar charts can improve user understanding. A user profile is often presented using visual components. Musical avatar is proposed in [11] for users to represent their musical preferences by mapping their preferred musical terms into different parts of the body to create their unique avatar. Middleton *et al.* [19] proposed Foxtrot—a system that recommends online academic research papers. They used a profile visualization approach to allow users to interact with system and build the profile in line form with varying time. The profile is represented in ontological terms understandable to the users, when the profile is used to search most related articles. In [20], a bar chart was used to demonstrate the weight of the item and give context to its sources, such as Facebook and Twitter. Although these peer-to-peer comparison methods illustrate a number of aspects of these systems, they do not adequately represent complex relationships between multitype peers.

Many graph-based visualizations are therefore introduced in RSs, especially for neighbor-based RSs. Most neighbor-based recommender methods use known feedback of users to predict unknown preferences. From the information point of view, preference information is inherited and modeled from known historical information. In graph-based methods, an inherited relationship can be easily represented using a directed edge between two nodes, and this makes them suitable and widely employed in RSs. A graph $G$ can be represented as $= (N, E)$, where $N$ represents a set of nodes and $E$ represents a set of edges. The edges connect two nodes in $N$, and are directed from one node to another, if and only if, there is an inherited relationship between two nodes. Hierarchy graphs are therefore well-suited to RSs because the items naturally inherit on users' preferences.

There are many different layouts for representing a graph. The traditional node-link layout is the simplest [21], [22]. It computes positions of each node and draws every edge as a curve. More examples are given in [23]. Layouts created by tree algorithms and spring algorithms fall within this node-link category. The classic tree is straightforward

and provides clear 2-D representations; it was first proposed in [24] and has become one of the base methods for describing data with inherited relationships. A novel explanation technique, based on trees, was presented by Hernando *et al.* [25]. Items are represented as leaves, and the edges between them provide reliable recommendation information. Many tree variations to the classic layout [26]–[28], have been proposed to improve space efficiency and generate compact graphs. Gotz and Wen [29] used a variation on the tree layout, named fanlens, along with many other basic layouts, such as bar, line and scatter charts, to find interaction patterns and adaptively switch the layout of displaying result for users. However, tree layouts are not appropriate in some circumstances because the cyclic graphs are needed to represent relationships. For example, the recommendations for an active user in neighbor-based systems are determined by aggregating the opinions of similar neighbors. In this case, if an item is recommended because it is preferred by two neighbors, then edges between nodes representing the active user, two neighbors and the recommendation form a circle.

The spring layout, first proposed in [30], is well-known as force-directed graph and is also well studied graph layout because of its simplicity. Spring layouts are based on a cost function that models graph edges as "springs" with forces between nodes that pull two nodes together or push them further apart. The graph iteratively changes until it becomes stable. A simple example of a spring layout is a social network graph that moves two nodes closer, or further apart, depending on the closeness of their relationship. The final graph is generated after several iterations to adjust relative positions of each node. Heckerman *et al.* [31] used a network to show predictive relationships. The data are represented as nodes in the network and all the nodes have one or more dependency links to the other nodes that describe the degree of correlation between them. When the degree is less than a user-defined threshold, the link is not displayed in the network. A CF-based RS [32] used a network to show the relationships between recommendations, the active user, neighbors, and recommendations. Some improvements to spring layout focus on different cost function as outlined in [33] and [34]. Even the spring layouts can handle the cyclic relationships, they are over concerning the cost function and thereby generate a low readable graph. For example, a spring layout graph representing members, neighbors and items as nodes, tends to mix and scatter all the nodes and make it difficult for specific member to find those directly related recommendation nodes (RNs) at once. Therefore, the spring layout can explain relationships between nodes but lack organizing in RS.

Map-based visualization techniques are used in many geographic location systems, but this constraint narrows the range of applications. In [35], users can gain a clear image illustrating data of selection for TV shows and music based on canonical maps. In [12], mobile tourism RSs with visualization are introduced. The locations, tours and recommendation lists are visualized to help users gain an intuitive understanding of the system. However, when the geo-information is missing, map-based methods are not feasible.
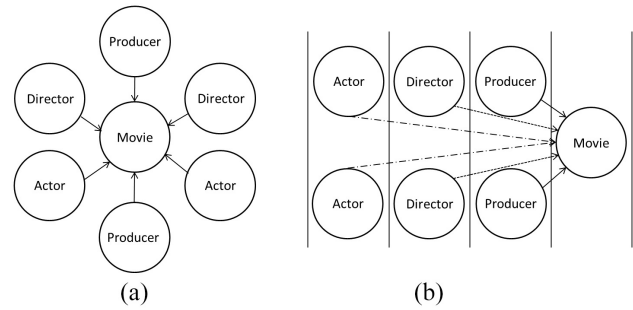


Fig. 1. (a) Example of hierarchy relationship on a movie. Producer, director, and actor are there different types of nodes that a movie inherits from. (b) General hierarchy graph and use different level to represent different type of nodes when different line type to represent different edge type.

When the visualizations in RSs only focus on one kind of relationship, such as social relationships or recommender dependencies, simple spring layouts are sufficient. However, they can only provide limited information because multiple entities and complicated relationships are involved, making it more difficult to understand the overall recommender process. Hierarchy graphs take components organization one step further that each node and each edge has an indicator to specify a type. For example, in Fig. 1(a), a spring layout graph of a movie would show the directors, producers and actors as three types of roles. The movie inherits its script from the producer, its visualization from the director, and its performance from the actors. Fig. 1(b) shows the hierarchy graph using different node levels and line types to represent node and edge types. In [36], a hierarchy graph is used to visualize an individual RS for recommending articles. However, the authors only organize and represent the recommendations, i.e., the articles, using a hierarchy tree graph. The information is insufficient for members who need to know more about the process. A multilevel network is used in [37] to show individual RSs. Profiles, common friends and recommendations are presented as nodes and are allocated to different levels and depicted using different colors. Unfortunately, nodes in same level are unordered and lack readability. Furthermore, using both color and level to represent the type of node is redundant.

## III. HVGR METHOD

In this section, we describe the HVGR method and related recommender algorithms. We first introduce the system structure of the method, then the components in detail.

### A. System Structure

The pseudouser-based group recommendation problem is defined as follows.

*Definition 1:* Let $U$ be all the users of the system $U = \{u_1, u_2, \ldots, u_{|U|}\}$ and $I$ be all the items $I = \{i_1, i_2, \ldots, i_{|I|}\}$. Group $G$ is a subset of $U$ that $G = \{g_1, g_2, \ldots, g_{|G|}\}$, where group members $g_m \in U, m = 1 \cdots |G|$. Let $I_G = \{i_1^*, i_2^*, \ldots, i_{|I_G|}^*\}$ be all the items that have been rated by group members, where $i_n^* \in I, \ n = 1 \cdots |I_G|$. Let $g$ be the pseudouser and the profile $P_g$ represents the overall group preference. $P_g$ is defined as a rating vector and every dimension

of it is $r_{g,i}$, where $i \in \{i_1^*, i_2^*, \ldots, i_{|I_G|}^*\}$ which is obtained by a specific pseudouser modeling procedure *procedure*$(U, G, I)$. Neighbors $N \in \{n_1, n_2, \ldots, n_k\}$ are identified by a procedure *procedure*$(U, P_g)$ where some similarity measures are used to retrieve similar users to generate recommendation candidates according to the pseudouser profile. Lastly, the unknown group ratings $r_{g,i}$, $i \notin I_G$ are predictions from a recommendation prediction procedure, $r_{g,i} = $ procedure$(P_g, i, N)$. The group recommendations are generated by choosing the top-*N* items with the highest predictions.

It can be seen from this definition that, to obtain the final recommendations from known users' information, four types of entities are involved in the pseudouser-based recommender method: 1) group member; 2) pseudouser profile; 3) neighbor; and 4) recommendation. All the entities are represented as nodes and rearranged at four different levels in our HVGR method. The four types of nodes are listed below.

1) *Level 1 Nodes:* The group member nodes (MNs). Every node corresponds to a group member.
2) *Level 2 Nodes:* The pseudouser profile nodes (PNs). The pseudouser profile is presented as a rating vector over the items that have been rated by the members and every node in this level corresponds to an item in the profile.
3) *Level 3 Nodes:* The neighbor user nodes (NNs). Every node corresponds to a neighbor which is determined by the similarities calculated with the PCC method.
4) *Level 4 Nodes:* The RNs. Every node in this level corresponds to a recommendation.

There are three procedures, shown in the definition, for calculating recommendations in pseudouser-based methods: 1) pseudouser modeling; 2) neighbor identification; and 3) recommendation prediction. The pseudouser modeling procedure takes group member entities as the input and pseudouser profiles as the output. The profile is represented as a rating vector on group items (i.e., items that have been rated by members). Pseudouser modeling procedure often defines a specific aggregation method and all the ratings specified on each group item are aggregated accordingly. The pseudouser profile then becomes the input of the neighbor identification procedure. The neighbor identification procedure identifies neighbors that have similar preference patterns to the pseudouser, using a similarity calculation method—typically cosine or Pearson's correlation. The last procedure predicts unknown item rankings and generates the recommendations with the highest rankings according to neighbor preferences. The edges that link two nodes at different levels demonstrate the three main procedures.

1) *MN-PN Edges:* The edges represent pseudouser modeling.
2) *PN-NN Edges:* The edges represent neighbor identification.
3) *NN-RN Edges:* The edges represent recommendation prediction.

With this design, users are given a clear overview of the whole system. In Fig. 2, an overview of the method is given to illustrate the flowchart of the recommendation, including three procedures and four types of entities with corresponding
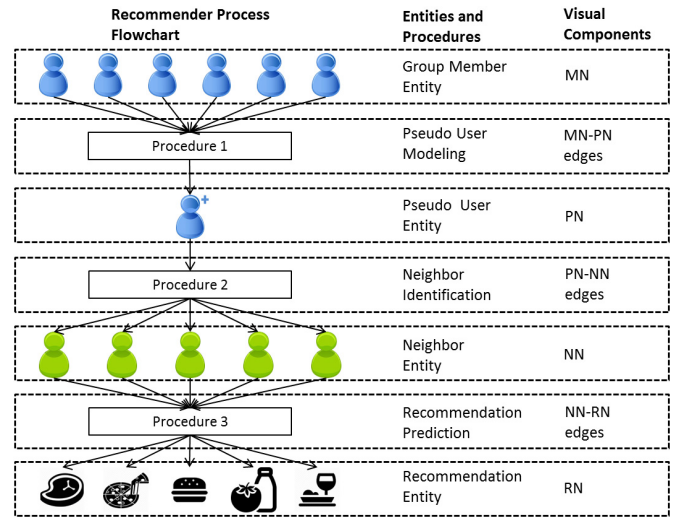


Fig. 2. Flowchart of the recommender process including four entities, three procedures, and corresponding visual components.

visual components. Our HVGR method employs a contribution score-based group recommender method (CS-GR) [38] to calculate recommendations. Components based on the CS-GR method are described in the following section. It is important to point out that HVGR not only visualizes the CS-GR method but it can also support other group recommender methods, as discussed in Section IV.

*B. Components*

*1) Level 1—MNs:* In the CS-GR method each member has an important attribute, its contribution, which is a numeric measurement of the representative of group members. The CS-GR method aims to develop a recommendation method that maximizes group satisfaction for users by modeling group preferences through the analysis of member ratings. The method employs separable non-negative matrix factorization to calculate contributions in terms of a sub-rating matrix.

We use all the item-pair subspaces as our sub-rating matrix to measure the contribution. Let $S$ be the set of all the subspaces, $S_j \in S$, $j \in [1..|S|]$ and the contribution of member $g_m$ in subspace $S_j$ is

$$c_{g_m}^{S_j} = \begin{cases} 1, & m = \mathbf{argmax}_k \mathbf{arctan2}\left(r_{g_k j1}, r_{g_k j2}\right) \\ 0, & \mathbf{else} \end{cases} \quad (1)$$

where $j1$ and $j2$ are two items which form the $S_j$. The overall contribution of member $g_m$ is sum of all subspaces according to (2) as follows:

$$c_{g_m} = \sum_{j=1}^{|S|} c_{g_m}^{S_j}. \quad (2)$$

Every member in $G$ is visualized as an MN and then the MN set is $[\mathrm{MN}_{g_1}, \mathrm{MN}_{g_2}, \ldots, \mathrm{MN}_{g_{|G|}}]$. How the MNs are constructed mainly depends on two factors: 1) the member identifier $m$ and 2) its corresponding contribution $c_{g_m}$. The member identifier helps members distinguish themselves from the others, and, in our system, the identifiers are shown using different colors. The contributions are shown using different

TABLE I
RATING MATRIX OF A GROUP EXAMPLE WITH
TWO USERS AND FOUR ITEMS

|  | $item_1$ | $item_2$ | $item_3$ | $item_4$ |
|---|---|---|---|---|
| $user_1$ | 5 | 4 | 4 | ? |
| $user_2$ | 3 | 1 | ? | 3 |

TABLE II
PSEUDOUSER PROFILE CALCULATION

| $c_{g_m} r_{g_m, i_n^*}$ | $item_1$ | $item_2$ | $item_3$ | $item_4$ |
|---|---|---|---|---|
| $user_1$ | $\frac{2}{5} \times 5 = 2$ | $\frac{2}{5} \times 4 = 1.6$ | $\frac{2}{5} \times 4 = 1.6$ | ? |
| $user_2$ | $\frac{3}{5} \times 3 = 1.8$ | $\frac{3}{5} \times 1 = 0.6$ | ? | $\frac{3}{5} \times 3 = 1.8$ |

TABLE III
MN-PN EDGE WIDTHS

| $W_{MN-PN_{g_m, i_n^*}}$ | $item_1$ | $item_2$ | $item_3$ | $item_4$ |
|---|---|---|---|---|
| $user_1$ | $5 \times 1 = 5$ | $5 \times \frac{1.6}{2} = 4$ | $5 \times \frac{1.6}{2} = 4$ | 0 |
| $user_2$ | $5 \times \frac{1.8}{2} = 4$ | $5 \times \frac{0.6}{2} = 1.5$ | 0 | $5 \times \frac{1.8}{2} = 4$ |

radius of nodes, and the radius is determined by a mapping equation (3). Let $c_{\max} = \max(c_{g_m})$, $m = 1 \cdots |G|$ and the maximum radius is $R$, the radius of $MN_{g_m}$ is

$$R_{MN_{g_m}} = R \frac{c_{g_m}}{c_{\max}}. \tag{3}$$

However, the MN level may contain numerous MNs and these unorganized nodes make members are difficult to obtain relative relation between themselves and the others, therefore they are sorted by level of contribution for clarity of presentation. The nodes are sorted in each subsequent level, based on similar quantitative information, and these details are given in the corresponding explanation of each component.

*Example 1:* A two member group is presented in Table I. The contribution and MNs calculation are presented. In this and following examples, the maximum radius of nodes $R$ is set to 10 and the maximum width of edges $W$ is set to 5.

The contributions of two members are computed according to (1) and (2)

$$c_{user1} = \frac{2}{4}\left(c_{user1}^{I_{1,2}} + c_{user1}^{I_{1,3}} + c_{user1}^{I_{1,4}} + c_{user1}^{I_{2,3}} + c_{user1}^{I_{2,4}} + c_{user1}^{I_{3,4}}\right)$$
$$= \frac{2}{4}(0 + 1 + 0 + 1 + 0 + 0) = 1$$
$$c_{user2} = \frac{2}{4}(1 + 0 + 1 + 0 + 1 + 0) = \frac{3}{2}.$$

After normalizing, the MCS are

$$\begin{cases} c_{user1} = \dfrac{1}{1 + 3/2} = 2/5 \\ c_{user2} = 3/5. \end{cases}$$

The maximum MCS is $c_{\max} = MCS_{user2}$. There are two MNs to represent two members and according to (3), the two radiuses are

$$\begin{cases} R_{MN_{user1}} = 10 \times \dfrac{2/5}{3/5} = 6.67 \\ R_{MN_{user2}} = 10 \times 1 = 10. \end{cases}$$

*2) MN-PN Edges:* Once a member's contribution has been calculated, the member's group rating can be calculated. Let $g_m$ be the member and $c_{g_m}$ be corresponding contribution, for a specific group item $i_n^*$, $r_{g_m, i_n^*}$ is the rating for $g_m$ of $i_n^*$. Since the group rating is a weighted sum of all the members that have rated that item, the part of group rating related to $g_m$ is $c_{g_m} r_{g_m, i_n^*}$. Given a member may not rate all the items in the group, an MN-PN edge is drawn when, and only when, a member $g_m$ has rated item $i_n^*$. From perspective of record, every MN-PN edge represents a historical rating and is only determined by $c_{g_m} r_{g_m, i_n^*}$. In this case, the width of edge is used to visualize $c_{g_m} r_{g_m, i_n^*}$. Let $r_{\max} = \max(c_{g_m} r_{g_m, i_n^*})$, where $g_m \in G \bigwedge i_n^* \in I_G$ has a maximum width of $W$. The width of

the MN-PN edge is calculated according to (4). The width of the edge indicates degree of preference, giving the user insight into pseudouser profile

$$W_{MN-PN_{g_m, i_n^*}} = W \frac{c_{g_m} r_{g_m, i_n^*}}{r_{\max}}. \tag{4}$$

*Example 2:* The pseudouser profile calculation using data in Example 1 is shown in Table II.

The maximum $r_{\max}$ is $c_{user1} r_{user1, item1}$, then the widths of all MN-PN edges are shown in Table III.

*3) Level 2—PNs:* The group rating is calculated with a weighted sum and the equation is shown as follows:

$$r_{g, i_n^*} = \sum_{g_m \in G} c_{g_m} r_{g_m, i_n^*}. \tag{5}$$

Next the pseudouser $g$ is modeled by constructing a rating vector pseudouser profile $P_g$ which assume can represent the preferences of the whole group. $P_G = [r_{g, i_1^*}, r_{g, i_2^*}, \ldots, r_{g, i_{|I_G|}^*}]$. Every element in $P_g$ is visualized as a PN, and the PN set becomes $[PN_1, PN_2, \ldots, PN_{|I_G|}]$. The PN node's construction mainly depends on two factors: the group rating and individual influence within it. A group rating represents the overall preference for an item and is visualized using the radius of the node. Instead draw a node using a solid color, a pie chart is displayed in the node to illustrate the individual influences.

The radius of a PN is determined by a mapping equation (6) considering all the group ratings. Let $r_{\max} = \max(r_{g, i_n^*})$, $n = 1 \cdots |I_G|$ and the maximum radius is $R$, the radius of $PN_n$ is

$$R_{PN_n} = R \frac{r_{g, i_n^*}}{r_{\max}}. \tag{6}$$

*Example 3:* The pseudouser profile is calculated according to (5) using data in Example 1

$$\begin{cases} r_{g, item1} = \dfrac{2}{5} \times 5 + \dfrac{3}{5} \times 3 = 3.8 \\ r_{g, item2} = \dfrac{2}{5} \times 4 + \dfrac{3}{5} \times 1 = 2.2 \\ r_{g, item3} = \dfrac{2}{5} \times 4 = 1.6 \\ r_{g, item4} = \dfrac{3}{5} \times 3 = 1.8 \end{cases}$$

TABLE IV
RATIO CALCULATION OF PNS

| $ratio^m_{PN_n}$ | $item_1$ | $item_2$ | $item_3$ | $item_4$ |
|---|---|---|---|---|
| $user_1$ | $\dfrac{2}{2+1.8}$ $= 0.53$ | $\dfrac{1.6}{1.6+0.6}$ $= 0.73$ | $\dfrac{1.6}{1.6+0}=1$ | $0$ |
| $user_2$ | $\dfrac{1.8}{2+1.8}$ $= 0.47$ | $\dfrac{0.6}{1.6+0.6}$ $= 0.27$ | $0$ | $\dfrac{1.8}{1.8+0}=1$ |

TABLE V
RATING MATRIX OF NEW NONMEMBER USERS

| | $item_1$ | $item_2$ | $item_3$ | $item_4$ | $item_5$ | $item_6$ |
|---|---|---|---|---|---|---|
| $user_3$ | 4 | 1 | ? | 2 | 4 | 3 |
| $user_4$ | 4 | 2 | 3 | ? | 2 | 5 |

TABLE VI
CALCULATION OF SIMILARITIES BETWEEN PSEUDO
USER AND NEW NONMEMBER USERS

| $Sim^{i^*_n}(g,u)$ | $item_1$ | $item_2$ | $item_3$ | $item_4$ |
|---|---|---|---|---|
| $user_3$ | 0.62 | 0.16 | (7) 0 | 0.08 |
| $user_4$ | 0.56 | 0.15 | | |

TABLE VII
PN-NN EDGE WIDTHS

| $W_{PN-NN_{i^*_n,u}}$ | $item_1$ | $item_2$ | $item_3$ | $item_4$ |
|---|---|---|---|---|
| $user_3$ | $5 \times 1 = 5$ | $5 \times \dfrac{0.16}{0.62}$ $= 1.29$ | | $5 \times \dfrac{0.08}{0.62}$ $= 0.65$ |
| $user_4$ | $5 \times \dfrac{0.56}{0.62}$ $= 4.52$ | $5 \times \dfrac{0.15}{0.62}$ $= 1.21$ | $5 \times \dfrac{0}{0.62}$ $= 0$ | |

and the radiuses of the PNs are

$$
\begin{cases}
R_{PN_{item1}} = 10 \times 1 = 10 \\
R_{PN_{item2}} = 10 \times \dfrac{2.2}{3.8} = 5.8 \\
R_{PN_{item3}} = 10 \times \dfrac{1.6}{3.8} = 4.2 \\
R_{PN_{item4}} = 10 \times \dfrac{1.8}{3.8} = 4.7.
\end{cases}
$$

All the PNs, once calculated, are sorted and arranged according to their group ratings. This helps members understand relative relationships within the pseudouser profile.

The individual influences from different members are illustrated by slices in a pie chart and the ratio of each slice is calculated according to the following equation:

$$
\mathbf{ratio}^m_{\mathbf{PN}_n} = \frac{C_{g_m} r_{g_m,i^*_n}}{r_{g,i^*_n}} = \frac{C_{g_m} r_{g_m,i^*_n}}{\sum_{m=1}^{|G|} C_{g_m} r_{g_m,i^*_n}}. \tag{7}
$$

*Example 4:* The ratios of PNs are shown in Table IV. From the table, we can see that $ratio^{user2}_{PN_{item3}}$ and $ratio^{user1}_{PN_{item4}}$ is zero because there are no historical ratings.

*4) PN-NN Edges:* Based on the pseudouser profile, $P_g$, the similarities between the pseudouser and nonmember group users can be calculated. The Pearson correlation coefficient (PCC), widely used in a number of RSs, has been used for the similarity computation. Let $u$ be a nongroup user. Let $I_u$ be the item set that has been rated by $u$. The PCC similarity between pseudouser $g$ and $u$ is computed based on their common ratings as follows:

$$
\mathbf{Sim}(g,u) = \frac{\sum_{i \in (I_G \cap I_u)} (r_{g,i} - \overline{r_g})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in (I_G \cap I_u)} (r_{g,i} - \overline{r_g})^2 \sum_{i \in (I_G \cap I_u)} (r_{u,i} - \overline{r_u})^2}} \tag{8}
$$

where $I_G \cap I_u$ is the set of common items, $\overline{r_g}$ is the average rating of $g$, and $\overline{r_u}$ is the average rating of $u$. It is important to note that for a given $i^*_n$ and $P_g$, the relationship between $g$ and $u$ on item $i^*_n$ can be measured by

$$
\mathbf{Sim}^{i^*_n}(g,u) = \frac{(r_{g,i^*_n} - \overline{r_g})(r_{u,i^*_n} - \overline{r_u})}{\sqrt{\sum_{i \in (I_G \cap I_u)} (r_{g,i} - \overline{r_g})^2 \sum_{i \in (I_G \cap I_u)} (r_{u,i} - \overline{r_u})^2}}. \tag{9}
$$

It is easy to see that every PN-NN edge represents a historical rating of nongroup user on items of pseudouser profile. Let $r_{max} = \max(\mathbf{Sim}^{i^*_n}(g,u))$, where $u$ is a nonmember user and the maximum width is $W$, for specific pseudouser profile item $i^*_n$ and neighbor $u$, the width of PN-NN edge is

$$
\mathbf{W_{PN-NN_{i^*_n,u}}} = W \frac{\mathbf{Sim}^{i^*_n}(g,u)}{r_{max}}. \tag{10}
$$

*Example 5:* Example 1 is extended by introducing two nonmembers, $user_3$ and $user_4$, and items, $item_5$, and $item_6$. The new rating matrix is shown in Table V.

The pseudouser profile obtained in Example 3 is [3.6, 2.2, 1.6, 1.8], then the similarities $\mathrm{Sim}^{item^*_n}(g,u)$, $n \in [1,4]$, between the pseudouser and new users are shown in Table VI.

The PN-NN edge widths are shown in Table VII.

*5) Level 3—NNs:* After all the similarities of nonmember users are determined, a collection of neighbors are selected, either according to a predefined number or a similarity threshold, to calculate the predictions. The selected neighbors set is noted as $N$, where $N = \{n_1, n_2, \ldots, n_{|N|}\}$. This means that only the PN-NN edges that target the selected neighbors need to be drawn. Every element in $N$ is shown as an NN and where the NN set is $[NN_1, NN_2, \ldots, NN_{|N|}]$. The radius of an NN is determined by a mapping function from similarities. Let $sim_{max} = \max(sim(g, n_k))$, $k = 1 \cdots |N|$ be the similarities between the pseudouser and its neighbors with a maximum radius of $R$. The radius of $NN_k$ is

$$
R_{NN_k} = R \frac{\mathbf{sim}(g, n_k)}{\mathbf{sim_{max}}}. \tag{11}
$$

*Example 6:* According to (8), the similarities are $sim(g, user3) = 0.86$ and $sim(g, user4) = 0.71$. The radiuses of the two NNs are then

$$
\begin{cases}
R_{NN_{user3}} = 10 \\
R_{NN_{user4}} = 10 \times \dfrac{0.71}{0.86} = 8.26.
\end{cases}
$$

All the NNs are rearranged according to their similarity value, which provides a clear representation of the neighborhood. For a given neighbor $u$, the influence of a specific pseudouser profile item $i^*_n$ is calculated, using (12), by normalizing

TABLE VIII
RATIO CALCULATION OF NNs

| $ratio^m_{NN_k}$ | $User_3$ | $User_4$ |
|---|---|---|
| $User_1$ | $\frac{0.62}{0.86} \times \frac{2}{2+1.8} + \frac{0.16}{0.86}$ $\times \frac{1.6}{1.6+0.6} = 0.51$ | $\frac{0.56}{0.71} \times \frac{2}{2+1.8} + \frac{0.15}{0.71}$ $\times \frac{1.6}{1.6+0.6} + 0 = 0.57$ |
| $User_2$ | $\frac{0.62}{0.86} \times \frac{1.8}{2+1.8} + \frac{0.16}{0.86}$ $\times \frac{0.6}{1.6+0.6} + \frac{0.08}{0.86} \times \frac{1.8}{1.8}$ $= 0.49$ | $\frac{0.56}{0.71} \times \frac{1.8}{2+1.8} + \frac{0.15}{0.71}$ $\times \frac{0.6}{1.6+0.6} = 0.43$ |

the similarity for every common item

$$\mathbf{ratio}^n_{NN_k} = \frac{\mathbf{Sim}^{i^*_n}(g,u)}{\mathbf{Sim}(g,u)}. \tag{12}$$

Combining the influence of a specific member to $i^*_n$, gives the influence of a specific member to a neighbor. A pie chart demonstrates how much a member has influenced a specific neighbor

$$\mathbf{ratio}^m_{NN_k} = \sum_{n=1}^{|I_G|} \mathbf{ratio}^n_{NN_k} \mathbf{ratio}^m_{PN_n}$$
$$= \sum_{n=1}^{|I_G|} \frac{\mathbf{Sim}^{i^*_n}(g,u)}{\mathbf{Sim}(g,u)} \frac{C_{g_m} r_{g_m, i^*_n}}{\sum_{m=1}^{|G|} C_{g_m} r_{g_m, i^*_n}}. \tag{13}$$

*Example 7:* Ratios of NNs are calculated according to (13) and shown in Table VIII.

*6) NN-RN Edges:* Unknown group ratings can be predicted after the neighbors are selected. This paper adopts user-based collaborative filtering, calculated by the weighted sum of deviations from the average rating of its neighbors. Let $r_{g,i}$ be the unknown group rating for item $i$, and $i \notin I_G$, and $r_{g,i}$ can be computed as

$$r_{g,i} = \overline{r_g} + \frac{\sum_{u \in N} (r_{u,i} - \overline{r_u}) \times \mathbf{Sim}(g,u)}{\sum_{u \in N} |\mathbf{Sim}(g,u)|} \tag{14}$$

where $\overline{r_g}$ denotes the average rating of pseudouser $g$. Clearly, $\overline{r_g}$ is a constant for given group, thus the predictions are mainly determined by

$$\tilde{r}_{g,i} = \frac{\sum_{u \in N} (r_{u,i} - \overline{r_u}) \times \mathbf{Sim}(g,u)}{\sum_{u \in N} |\mathbf{Sim}(g,u)|}. \tag{15}$$

Therefore, for given neighbor $u$ and item $i$, the NN-RN edge is constructed by the mapping (15). Let $r_{max} = \max(\tilde{r}_{g,i})$ for specific neighbor $u$ and the maximum width is $W$, the width of NN-RN edge is

$$W_{NN-RN_{u,i}} = W \frac{\tilde{r}_{g,i}}{r_{max}}. \tag{16}$$

*Example 8:* The $\tilde{r}_{g,i}$ for item$_5$ and item$_5$ according to (15) is shown in Table IX.

The NN-RN edge widths are shown in Table X.

*7) Level 4—RNs:* Usually, the top-$N$ items with highest predictions are selected as the final recommendations using (14). From the perspective of recommendations, only NN-RN edges those target to final results need to be visualized. The recommendation set is denoted as $R$, where $R =$

TABLE IX
$\tilde{r}_{g,i}$ CALCULATION

| | $item_5$ | $item_6$ |
|---|---|---|
| $user_3$ | $\frac{|(4-2.33) \times 0.86|}{|0.86| + |0.71|} = 0.91$ | $\frac{|(3-2.33) \times 0.86|}{|0.86| + |0.71||} = 0.37$ |
| $user_4$ | $\frac{|(2-2.67) \times 0.71|}{|0.86| + |0.71|} = 0.3$ | $\frac{|(5-2.67) \times 0.71|}{|0.86| + |0.71|}$ $= 1.05$ |

TABLE X
PN-NN EDGE WIDTHS

| $W_{NN-RN_{u,i}}$ | $item_5$ | $item_6$ |
|---|---|---|
| $user_3$ | $5 \times \frac{0.91}{1.05} = 4.33$ | $5 \times \frac{0.37}{1.05} = 1.76$ |
| $user_4$ | $5 \times \frac{0.3}{1.05} = 1.43$ | $5$ |

$\{r_1, r_2, \ldots, r_{|R|}\}$. Every element in $R$ is shown as an RN and the RN set is $[RN_1, RN_2, \ldots, RN_{|R|}]$. The radius of an RN is determined by a mapping function from group ratings. Let $r_{max} = \max(r_{g,i})$, $i = 1 \cdots |R|$ and the maximum radius is $R$, the radius of $RN_k$ is

$$R_{RN_k} = R \frac{r_{g,i}}{r_{max}}. \tag{17}$$

*Example 9:* The predictions of item$_5$ and item$_5$ are shown below when the average pseudouser rating is 2.35

$$\begin{cases} r_{item_5} = 2.35 + \dfrac{0.86 \times (4-2.8) + 0.71 \times (2-3.2)}{|0.86 + 0.71|} = 2.46 \\ r_{item_6} = 2.35 + \dfrac{0.86 \times (3-2.8) + 0.71 \times (5-3.2)}{|0.86 + 0.71|} = 3.27. \end{cases}$$

The radiuses of the two RNs are

$$\begin{cases} R_{RN_{item5}} = 10 \times \dfrac{2.46}{3.27} = 7.5 \\ R_{RN_{item6}} = 10. \end{cases}$$

Because the unknown group rating is derived by aggregating predictions from all the neighbors, the influence of specific neighbor $u$ for given recommendation $i$ can be measured by $(r_{u,i} - \overline{r_u}) \times \text{sim}_{g,u}$. Thus, the individual influence from different neighbors for specific $r_{g,i}$ is calculated by (17)

$$\mathbf{ratio}^k_{RN_i} = \frac{|(r_{k,i} - \overline{r_k})\text{sim}_{g,k}|}{\sum_{u \in N} (r_{u,i} - \overline{r_u}) \times \text{sim}_{g,u}}. \tag{18}$$

Combining these ratios with ratio of the member to specific neighbor provides the ratio of the member to specific recommendation

$$\mathbf{ratio}^m_{NN_k} = \sum_{k=1}^{|N|} \mathbf{ratio}^k_{RN_i} \mathbf{ratio}^m_{NN_k}$$
$$= \sum_{k=1}^{|N|} \left( \frac{|(r_{k,i} - \overline{r_k})\text{sim}_{g,k}|}{\sum_{u \in N} (r_{u,i} - \overline{r_u}) \times \text{sim}_{g,u}} \sum_{n=1}^{|I_G|} \right.$$
$$\left. \times \frac{\mathbf{Sim}^{i^*_n}(g,u)}{\mathbf{Sim}(g,u)} \frac{C_{g_m} r_{g_m, i^*_n}}{\sum_{m=1}^{|G|} C_{g_m} r_{g_m, i^*_n}} \right). \tag{19}$$

### TABLE XI
### INDIVIDUAL INFLUENCE OF DIFFERENT NEIGHBORS FOR SPECIFIC RECOMMENDATION

| $ratio^k_{RN_i}$ | $item_5$ | $item_6$ |
|---|---|---|
| $user_3$ | $\frac{\|(4-2.33)\times0.86\|}{\|(4-2.33)\times0.86\|+\|(2-2.67)\times0.71\|}$ $=0.75$ | $\frac{\|(3-2.33)\times0.86\|}{\|(3-2.33)\times0.86\|+\|(5-2.67)\times0.71\|}$ $=0.26$ |
| $user_4$ | $\frac{\|(2-2.67)\times0.71\|}{\|(4-2.33)\times0.86\|+\|(2-2.67)\times0.71\|}$ $=0.25$ | $\frac{\|(5-2.67)\times0.71\|}{\|(3-2.33)\times0.86\|+\|(5-2.67)\times0.71\|}$ $=0.74$ |

### TABLE XII
### RATIO CALCULATION OF NNs

| $ratio^m_{NN_k}$ | $item_5$ | $item_6$ |
|---|---|---|
| $user_1$ | $\frac{0.75}{0.75+0.25}\times0.51+\frac{0.25}{0.75+0.25}\times0.57=0.53$ | $\frac{0.26}{0.26+0.74}\times0.51+\frac{0.74}{0.26+0.74}\times0.57=0.55$ |
| $user_2$ | $\frac{0.75}{0.75+0.25}\times0.49+\frac{0.25}{0.75+0.25}\times0.43=0.47$ | $\frac{0.26}{0.26+0.74}\times0.49+\frac{0.74}{0.26+0.74}\times0.43=0.45$ |

A pie chart shows the individual influence of every member in the RN, and becomes a useful tool for members to track the influence they had on each recommendation.

*Example 10:* Calculations of ratios of RNs according to (18) and (19) are shown in Tables XI and XII.

## IV. IMPLEMENTATION AND DISCUSSION

We demonstrate the usability and interactivity of our proposed hierarchy graph and MC-GR method with an example based on data set MovieLens 100K.

### A. Implementation

*1) Usability:* An overview of the result is shown in Fig. 3, where nodes representing the entities, i.e., the group members, profile items, neighbors, and recommendations, are allocated to different levels. Compared to traditional methods, our method presents more detail and organized visual information, which will improve group members' confidence in accepting the recommendations.

A group consisting of five randomly selecting members was formed. The overall recommendation procedures are represented using different sized nodes and different width edges. First, the MC-GR method calculated each member's contributions and a pseudouser was modeled from the weighted average of them all. As a result, five nodes of different sizes were rendered in the MN level—their names are displayed to the right side of each node. Focusing on members 1 and 5, for simplicity, their contributions were 0.31 and 0.06, respectively, and therefore the radius of member 1's node is much larger than member 5's. Member 1's node has three edge sources showing that member 1 rated three items, i.e., P1, P2, and P4, but did not rate P3 and P5.

The width of the edges represents the strength of inheriting information. The MN-PN edge represents the weighted ratings. Since member 1's contribution is constant, the three edges show that member 1 gave her highest rating to P1 and her lowest to P4. Given the target group rating is the sum of weighted ratings from all the members, the P1 node which is determined by member 1, becomes the biggest because
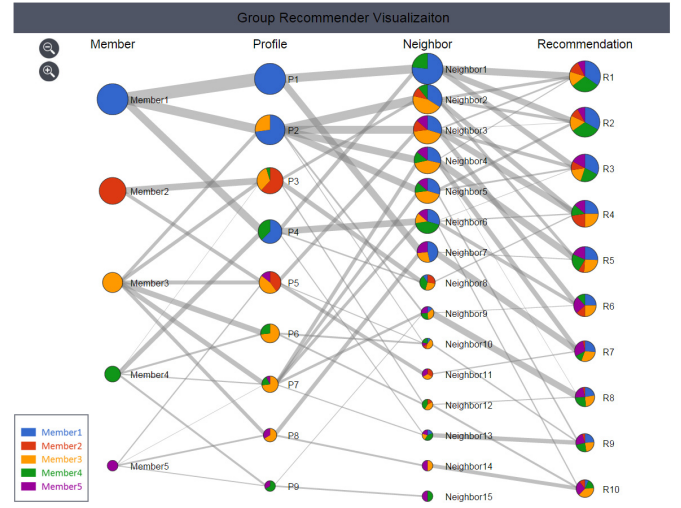


Fig. 3. Visualization example on real data set MovieLens 100K.

of member 1's large contribution. P9's radius is the smallest because the member contributions were small. In this way, all members can easily see what items are in pseudouser profile and how the group ratings were derived.

The PN-NN edges represent the ratings specified by neighbors for items in the pseudouser profile. Take P1's sources an example. Both neighbors 1 and 7 rated this item, but the width of the edge to neighbor 1 is little wider than to neighbor 7. This demonstrates that neighbor 1 preferred P1 more than neighbor 7.

Additionally, all the PN-NN edges target a single NN node to represent the overall preference of a neighbor. For example, the final similarity of neighbor 7 was 0.65 when P1's source was 0.44 and P4's source was 0.21. It is important to point out that a neighbor is selected by the PCC similarity measure, which means it could be no direct relationships between a group item and a neighbor. For example, neighbor 14 only has direct connections with P8, which means it was only selected because he preferred P8 and her opinions about the rest of pseudouser profile items are not known.

The NNs, RNs, and NN-RN edges work very similarly to the MNs, PNs, and MN-PN edges. The only difference is that the numerical measure for members is their contribution and for neighbor it is their similarity. Because our HVGR method employs user-based collaborative filtering to generate unknown ratings, the recommendations have direct relationships with at least one neighbor. The width NN-RN edge represents the degree of preference for recommendations for neighbors and is numerically equal to the product of the similarity of a neighbor and the specified ratings in recommendations. For example, R1 is recommended because the top four neighbors rated it, and the most similar neighbor preferred it the most, providing members with understanding for why these recommendations were generated and why some ratings are higher.

Individual connections are given as pie charts at every level from the MNs to the RNs. This enables members to immediately understand the relative relationships with others
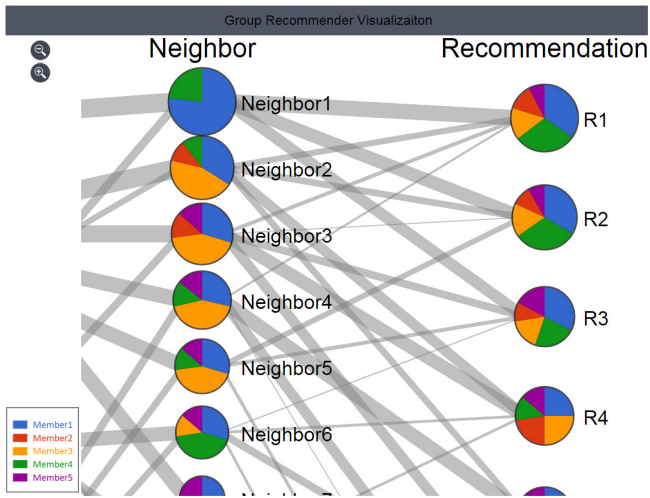
Fig. 4. Interactivity zoom and pan to enable usability when data is large.

| | Question Description |
|---|---|
| Q1 | Easy to access |
| Q2 | Effective for providing members information |
| Q3 | Effective to illustrate the commonality of the group |
| Q4 | Effective to explore the related uses for the group |
| Q5 | Effective for presenting recommendations |
| Q6 | Instinctive to explain the recommending process |
| Q7 | Easy to obtain relative relationships with other group members |

and it is easy for members to track their individual influence throughout the recommendation process. For example, tracking member 1 through P2, then neighbor 2 and lastly to R4 shows member 1's individual influence along this path. At P2, she can see her dominance in the group rating, because member 3's percentage is so much smaller. In the NN for neighbor 2, she sees her percentage, or influence, has decreased but member 3's percentage has increased because neighbor 2 prefers P3, which is dominated by member 3. Lastly, at R4, her influence has continued to decrease because many other neighbors have influenced the prediction. We do not list all the details of profiles of participants shown in the case for each of them in the data set has specified scores of ratings.

*2) Interactivity:* In real circumstances, situations are much more complicated than those described in the previous example. There may be numerous visual components because the group could be larger and entity relationships could be sparse. However, the space used to display the visual result is limited and this will definitely lower the readability for users, making it difficult for them to focus on the content they are interested in.

The interactivities, zoom and pan, are provided to guarantee the usability of the method when the data is scaled. Zoom and pan techniques allow users to explore the data in detail. The interest domain of a large graph can be focused on, and neighboring domains can be easily viewed. This guarantee increases the efficiency of screen usage and guarantees the scalability of our method. Fig. 4 gives a demonstration of focusing on the right-top part of the graph using zoom.

*3) Adaptability:* Furthermore, the fundamental difference among group modeling methods is that the group rating is determined by which member(s). A group rating of an item is calculated by combining all the individual ratings of members specified on it. Many group modeling methods are proposed and generate different combinations. For example, when group ratings for group profile are modeled using average-method, which means a group rating of an item is calculated by the

mean of all the ratings specified on that item. Therefore, pie charts at every PN, the fractions of all the members are same. For the same group presented in Section IV-A, at P2 node, the fractions of members 1 and 3 will be 50% and 50%; at P5 node, the fractions of members 2, 3, and 5 are all 33.3%. The method is easily adaptable to graph other group modeling methods, like plurality voting, least misery, average, etc. The modeling procedure is abstracted using MNs, PNs, and MN-PN edges. The method only needs to adjust radius of PNs and widths of MN-PN edges to adapt to other modeling methods.

Therefore, when abstract method-independent components, i.e., MNs, group rating nodes and edges between them, from these trivial calculation details, our method can easily adapt to other modeling methods by simply adjusting the edges. This method can also be applied to visualize individual RSs, by simply using a 'single member' group.

*4) Extensibility:* The method could also provide solid explanations in collaborative filtering-based recommendation systems, because an individual recommender process can be seen as a "special" group recommender process using group that only contains one member.

Visualization would need to be modified in first two levels to support individual visualization. The first level would contain a single root of only one node, representing the active user. The radius of the circle would represent their average rating and illustrate their rating pattern. For example, a user's trend for giving higher or lower ratings can be easily identified in this way.

The pseudouser profile becomes active user profile, and this profile can still be visualized in PN level. Each node of PN would represent an item has rated by the active user and can be filled with different color, which is similar of members nodes in group visualization. The pie chart in NN and RN level could represent the influence of every group item within a specific recommendation.

### B. Evaluation

To evaluate the effectiveness and assess user satisfaction of the HVGR method, a questionnaire was designed as described in Table XIII. Some criteria are applied to evaluate the performance when the system has been build and allow users to feedback after comparing the results with/without a visualization method. For example, an online shopping system can check if the user will click more and buy more products

TABLE XIV
MAIN FEATURES OF DIFFERENT VISUALIZATION METHODS ON RSS

| | Method | Group recommender systems | | Individual recommender systems | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Member | Profile Modeling | Profile | Neighbor Identification | Neighbor | Recommendation Prediction | Recommendation |
| | HVGR | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [11] | Avatar | | | ✓ | | | | |
| [19] | Line | | | ✓ | | | | |
| [14] | SOM | ✓ | | ✓ | | | | |
| [25] | Tree | | | | | ✓ | ✓ | ✓ |
| [29] | Tree | | | | | ✓ | | ✓ |
| [13] | Link-node | | | | | ✓ | | ✓ |
| [20] | Link-node | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| [31] | Force | | | | | ✓ | ✓ | ✓ |
| [32] | Force | | | | | ✓ | ✓ | ✓ |
| [35] | Map | | | | | ✓ | ✓ | ✓ |
| [12] | Map | | | | | | | ✓ |
| [36] | H Graph | | | | | ✓ | ✓ | ✓ |
| [37] | H Graph | | | ✓ | ✓ | ✓ | ✓ | ✓ |

to test a visualization method. However, our experiments are designed and executed on a public data set, not an online system, therefore we use questionnaire to evaluate our proposal. Totally 30 participants from business intelligence, e-commence, computer science, and other areas, were involved to specify a score for the seven statements listed in the table. The assessment results are summarized in Fig. 5. The results show that users felt that the proposed recommendation method provides an instinctive explanation of a recommending process; they therefore can understand why a particular item is recommended and very likely accept it. For four types of entities in recommender process, comparing to the members information, neighbor users and recommendations, it is the most difficult to understand the group commonality, i.e., group preference, and the relative relationships between entities can be easily obtained. The HVGR method is easy to access for users. From questions 2–6, the feedback shows that the key information and process is intuitively presented. Therefore, users can trust the recommendation more than before for the trust comes from the information presenting and understandability of the black box process. In general, by the survey, the proposed HVGR method can improve transparency of recommendation process, increase user trust and confidence to the recommended items and the system, and enhance effectiveness of GRSs.

We have also compared our methods with exiting methods in both individual RSs and GRSs. Table XIV shows the differences between our HVGR method and other methods using RSs. The member and profile modeling columns are unique to GRSs while the other features are common to individual RSs and GRSs.
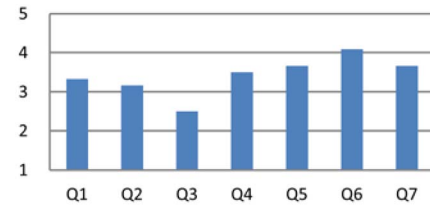


Fig. 5. Average satisfactions (scores) for questionnaire questions.

## V. CONCLUSION

GRSs are hardly to be trusted for lack of information presentation for individual members and procedure explanation since most related works to date focus on traditional RSs. In this paper, we have presented and discussed a novel HVGR to provide well-presented information and explanation. Three results are highlighted when GRSs are combined in the HVGR method. First, to the best of our knowledge, no literature offers either studies or systems that allow users to explore the group recommender process. Our HVGR method organizes and presents information using multilevel nodes and edges, and explains the recommender procedures, which makes a significant contribution to the GRSs area. Second, group members can also gain an understanding of their individual influence and relative relationships with other members by mapping members by the pie chart for every node. Lastly, HVGR provides a high level of abstraction of recommender procedures and shows high expansibility by simply changing the size of the MNs and edges between the MNs and pseudouser PNs. This method is implemented in an e-learning RS [1], which will be reported in another paper.

Our future work will provide greater interactivity, including highlighting all related nodes when a specific node is selected, and supporting deep digging of data. Another future work is to add a new function in GRS which allow users to give feedbacks on every step of process, and the systems can real time response to them.

## 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] Q. Shambour and J. Lu, "A trust-semantic fusion-based recommendation approach for e-business applications," *Decis. Support Syst.*, vol. 54, pp. 768–780, Dec. 2012.

[3] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs, "SigTur/E-Destination: Ontology-based personalized recommendation of tourism and leisure activities," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 633–651, 2013.

[4] A. Nanopoulos, "Item recommendation in collaborative tagging systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 4, pp. 760–771, Jul. 2011.

[5] P. Kazienko, K. Musial, and T. Kajdanowicz, "Multidimensional social network in the social recommender system," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 4, pp. 746–759, Jul. 2011.

[6] W. Wang, G. Zhang, and J. Lu, "Collaborative filtering with entropy-driven user similarity in recommender systems," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 854–870, 2015.

[7] M. Mao, J. Lu, G. Zhang, and J. Zhang, "A fuzzy content matching-based e-commerce recommendation approach," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Istanbul, Turkey, 2015, pp. 1–8.

[8] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proc. ACM Conf. Comput. Supported Cooper. Work*, Philadelphia, PA, USA, 2000, pp. 241–250.

[9] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *Proc. 23rd Int. Conf. Data Eng. Workshop*, Istanbul, Turkey, 2007, pp. 801–810.

[10] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell, "Explaining the user experience of recommender systems," *User Model. User Adapted Interaction*, vol. 22, no. 4, pp. 441–504, 2012.

[11] D. Bogdanov *et al.*, "Semantic audio content-based music recommendation and visualization based on user preference examples," *Inf. Process. Manag.*, vol. 49, no. 1, pp. 13–33, Jan. 2013.

[12] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "Mobile recommender systems in tourism," *J. Netw. Comput. Appl.*, vol. 39, pp. 319–333, Mar. 2014.

[13] K. Verbert, D. Parra, P. Brusilovsky, and E. Duval, "Visualizing recommendations to support exploration, transparency and controllability," in *Proc. Int. Conf. Intell. User Interfaces*, Santa Monica, CA, USA, 2013, pp. 351–362.

[14] 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.

[15] J. Lu, G. Zhang, and F. Wu, "Web-based multi-criteria group decision support system with linguistic term processing function," *IEEE Intell. Inform. Bull.*, vol. 5, no. 1, pp. 34–43, Jun. 2005.

[16] J. Lu *et al.*, "A linguistic multi-criteria group decision support system for fabric hand evaluation," *Fuzzy Optim. Decis. Making*, vol. 8, p. 395, Dec. 2009.

[17] G. Zhang, J. Ma, and J. Lu, "Emergency management evaluation by a fuzzy multi-criteria group decision support system," *Stochastic Environ. Res. Risk Assess.*, vol. 23, no. 4, pp. 517–527, 2009.

[18] 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.

[19] S. E. Middleton, N. R. Shadbolt, and D. C. D. Roure, "Ontological user profiling in recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 54–88, 2004.

[20] S. Bostandjiev, J. O'Donovan, and T. Höllerer, "TasteWeights: A visual interactive hybrid recommender system," in *Proc. 6th ACM Conf. Recommender Syst.*, Dublin, Ireland, 2012, pp. 35–42.

[21] M. Kaufmann and D. Wagner, *Drawing Graphs: Methods and Models*. Heidelberg, Germany: Springer, 2001.

[22] H. C. Purchase, "Which aesthetic has the greatest effect on human understanding?" in *Proc. 5th Int. Symp. Graph Drawing*, Rome, Italy, 1997, pp. 248–261.

[23] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[24] E. M. Reingold and J. S. Tilford, "Tidier drawings of trees," *IEEE Trans. Softw. Eng.*, vol. SE-7, no. 2, pp. 223–228, Mar. 1981.

[25] A. Hernando, J. Bobadilla, F. Ortega, and A. Gutiérrez, "Trees for explaining recommendations made through collaborative filtering," *Inf. Sci.*, vol. 239, pp. 1–17, Aug. 2013.

[26] P. Eades, "Drawing free trees," *Bull. Inst. Combinatorics Appl.*, vol. 5, no. 2, pp. 10–36, 1992.

[27] Q. V. Nguyen and M. L. Huang, "Space-optimized tree: A connection+enclosure approach for the visualization of large hierarchies," *Inf. Vis.*, vol. 2, no. 1, pp. 3–15, 2003.

[28] K. Marriott and P. Sbarski, "Compact layout of layered trees," in *Proc. 13th Aust. Conf. Comput. Sci.*, Ballarat, VIC, Australia, 2007, pp. 7–14.

[29] D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in *Proc. 14th Int. Conf. Intell. User Interfaces*, Sanibel, FL, USA, 2009, pp. 315–324.

[30] P. Eades, "A heuristic for graph drawing," *Congressus Numerantium*, vol. 42, pp. 149–160, May 1984.

[31] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *J. Mach. Learn. Res.*, vol. 1, pp. 49–75, Oct. 2001.

[32] J. O'Donovan, B. Smyth, B. Gretarsson, S. Bostandjiev, and T. Höllerer, "PeerChooser: Visual interactive recommendation," in *Proc. Conf. Human Factors Comput. Syst.*, Florence, Italy, 2008, pp. 1085–1088.

[33] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing," *ACM Trans. Graph.*, vol. 15, no. 4, pp. 301–331, 1996.

[34] K. Misue, P. Eades, W. Lai, and K. Sugiyama, "Layout adjustment and the mental map," *J. Vis. Lang. Comput.*, vol. 6, no. 2, pp. 183–210, 1995.

[35] E. Gansner, Y. Hu, S. Kobourov, and C. Volinsky, "Putting recommendations on the map: visualizing clusters and relations," in *Proc. 3rd ACM Conf. Recommender Syst.*, New York, NY, USA, 2009, pp. 345–348.

[36] A. Hernando, R. Moya, F. Ortega, and J. Bobadilla, "Hierarchical graph maps for visualization of collaborative recommender systems," *J. Inf. Sci.*, vol. 40, no. 1, pp. 97–106, 2014.

[37] B. Gretarsson, J. O'Donovan, S. Bostandjiev, C. Hall, and T. Höllerer, "SmallWorlds: Visualizing social recommendations," *Comput. Graph. Forum*, vol. 29, no. 3, pp. 833–842, 2010.

[38] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decis. Support Syst.*, vol. 87, pp. 80–93, Jul. 2016.
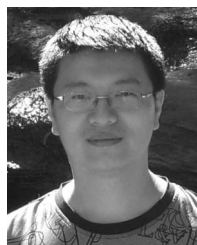
**Wei Wang** is currently pursuing the Ph.D. degree with the School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia.

He is a member of the Decision Systems and e-Service Intelligence Research Laboratory, Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney. He has published papers in conferences and journals, such as *Decision Support Systems*. His current research interests include recommender systems, business intelligence, and data visualization.

**Guangquan Zhang** received the Ph.D. degree in applied mathematics from the Curtin University of Technology, Sydney, NSW, Australia.

He is an Associate Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney. His current research interests include multiobjective, bi-level and group decision making, fuzzy optimization and uncertain information processing, recommender systems, and artificial intelligence. He has published four monographs, four reference books, and over 390 papers in refereed international journals and conference proceedings.

Dr. Zhang was a recipient of seven Australian Research Council discovery grants and many other research grants. He has organized seven special issues in international journals as a Guest Editor.

**Jie Lu** (SM'13) is a Distinguished Professor, Director of Centre for Artificial Intelligence, and the Associate Dean with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia. She has published five research books and 400 papers. Her current research interests include decision support systems, recommender systems, and other learning-based intelligent systems.

Prof. Lu was a recipient of eight Australian Research Council discovery grants and ten other grants. She serves as the Editor-in-Chief for *Knowledge-Based Systems* (Elsevier) and the *International Journal of Computational Intelligence Systems* (Atlantis), an Editor for book series on *Intelligent Information Systems* (World Scientific), and a Guest Editor of 12 special issues for the IEEE and other international journals, as well as delivered 15 keynotes in the IEEE and other international conferences.