Preference Networks and Non-Linear Preferences in Group Recommendations

Jitendra Kumar[a], *Korra Sathya Babu*[a]

[a] *Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, Odisha, India* 769008

Abstract

Recommendation system has gained great traction in the past decade as it facilitates the user's selection process within a limited time. Traditional Recommendation System caters to the preferences of an individual user. Sometimes, it is important to recommend an item to a batch of people. e.g. model for tourist recommendations, a movie for a family, music for friends, etc. This is where the Group recommendation system comes. It generates recommendations for a group of users with similar taste by combining the preferences of individual members and recommending a list that is liked by most members. In this paper, we analyze different methods for preference aggregation and group choice prediction based on weighting individual preferences. The weight calculation is based on the node centrality score. Multiple centrality techniques are analyzed for score calculation. The experimental results show that the non-linear remapping of preferences yields better group predictions and recommendations.

Keywords: `Group Recommender system(GRS), Collaborative filtering(CF), Centrality, Aggregation model, Preference Networks`

1. Introduction

With more people starting to use applications online. It has become the need of the hour to know the preferences of a user to keep them intact. An important aspect of getting to know the user is to carefully analyze the user's interaction with the system, their preferences, likes, and dislikes. Many systems have been developed lately to collect substantial information about the user to analyze their behavior with the application. The aim is to use this information to study in-depth about the user and develop a system that is capable of predicting the user rating or preference towards a particular feature. And that is exactly what Recommender systems are built for.

A recommender system, or a recommendation system (sometimes replacing 'system' with a synonym such as a platform or an engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.[1, 2]. Recommender systems find their applications at places like music and video services, E-commerce product recommendations, presenting

Email address: `520cs1001@nitrkl.ac.in` (*Jitendra Kumar*[a])

related posts on social media. It also provides suggestions for articles, restaurants, Trips, and matches for people. There are several approaches to building the system. One of them is Collaborative filtering which has extensive use. *Breese et al* [3] Collaborative filtering works on the assumption that the user who agreed on something in the past will agree on similar items in the future. A prime advantage of this approach is that it is capable of recommending complex items such as songs very accurately without requiring an understanding of the item itself. This is accomplished by locating the peer items with rating history similar to the current item. Another familiar approach is content-based filtering. [4, 5] It is based on the content of the item and the preference of the user. Whenever there is not sufficient information about the user, this approach could be leveraged effectively. As it treats recommendations as a user-specific classification problem and learns a classifier for the user's interests based on an item's features. With a varying set of applications that we use every day, Recommender systems play a vital role in helping the business to provide the key insights about the user behavior. A very good example is Spotify, the online music streaming platform. According to this article, Spotify has 350 million monthly active users. Spotify recommendations are one of the finest and one can say that it is the key element to Spotify's successful user base. To unlock this level of recommendation capability, the Recommendation system must not just be considering a user as an isolated node. People are related to one another by geography, common taste, and lifestyle. So should be the nodes where each node represents a user. Conventional Recommender systems weren't intelligent enough to predict the user rating or preferences when there is no prior information. That is usually the case when a new user just enters the system. This poses the cold start problem when the system doesn't know anything about a user. Not just with user's, it is even applicable to a new song that just got added to the Music library. There is no prior data about the song in order to recommend it to a user. Now there is a good chance that the song or the user might not be recommended because the probability of them being recommended is low due to lack of information. In real-world systems, this scenario is very common and it is to be addressed carefully.

Another problem that might arise is Data sparsity when the user rates only a few of the available songs. In terms of scalability, it becomes really difficult to process the data and come up with a useful result for the recommendation. A few years before, researchers realized this challenge, so they started searching for a solution to all these problems. That is when the idea of Group Recommendation Systems (GRS) was put forward. Instead of perceiving a user as an isolated node, GRS treats the user base as groups of many communities. Each community represents a span of people with common interests, so the system aims to come up with a recommendation that satisfies the majority of the group. With GRS, the above-mentioned problems are no longer a challenge. Let's solve the cold start problem. When a new user just logs into the system, they could be added to one of the communities and now all the recommendations which are already computed for that community could be used to recommend to the new user. As new users get added to a community, the system learns more about the user and keeps updating the recommendation list incrementally instead of globally recomputing the list for every single user of the community. This way GRS also provides us a scalable solution with its preference networks. The idea of having communities and computing the user preferences for a group instead of one isolated user has become essential. As in today's modern world, with a multitude of applications that are being used by a varying mix of people across the world. The Recommendations are expected to be fast and accurate. Researchers have been pondering over the years coming with innovative solutions to solve this critical problem that is crucial for businesses to keep their users intact. Now that we have better understood the problem from a broader perspective, let's dive in to get clarity on how exactly the different approaches solve these problems. Each node in a group refers to a user in the network, and as mentioned before the recommendations are computed in such a way that it caters to the most significant node. And to find out these most significant centrality techniques are employed. 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. Once the nodes have been weighted, we aggregate each user's preferences to come with a group model, which is the optimal recommendations list for the group. The stages carried out in the process are described in detail in the Methodology section of the article.

This article focuses on comparing different centrality and group modeling techniques across various datasets and different cluster sizes to maximize the parameter a. To put briefly what will be covered in this article: (1) Data from 3 different datasets, MovieLens-100k, MovieLens-1M, and Netflix-1M are used as inputs (2) Varying cluster sizes in k-means algorithm for k=4,8,12,20,50 (3) Computing the model with degree centrality, closeness centrality, harmonic centrality techniques. (4) Comparing the 4 Group Modeling techniques, average, least misery, average without misery, most pleasure (5) The metrics used to compare are MSE (Mean Square Error) and Precision at K. Both metrics compare the predicted recommendation to the actual recommendation that is desired by the user and gives us a sense of understanding of how accurate the system could predict the user preferences.

1.1. Applications of Recommendation System

Recommendation System finds huge use in a various domains. Some of example, We discuss below.

Table 1: Application Domain of Recommendation System.

Model	Application Domain	User type	Period
MusicFX [6]	Music	Group	1998
FlytrapS [7]	Music	Group	2002
A multi-agent e-government system [8]	E-governmen	Individual	2005
University digital library Recommender system [9]	E-library	Individual	2009
E-tourism [10]	E-group	Group	2002
INTRIGUE [11]	E-group	Group	2006
PolyLens [12]	E-group	Group	2013
TV program recommender [13]	E-group	Group	2009
Application of inhibitors [14]	E-group	Group	2009
Social media analytics [15]	Online based	Individual	2017
text mining [16]	Online based	Individual	2021

2. Related Work

Researchers have been studying individual problems extensively to come up with the best possible solution. And there have been progressions one after another. To summarise it all, Irfan Ali and Sang-Wook Kim discuss the limitations of the existing GRS that exist in their paper [17]. They also suggest to us some possible leads for the development of recommender systems. After performing extensive research, they conclude that no single approach consistently performs better than the other approaches. It is either limited by the group size or group type. [18] Some researchers have also analyzed and proposed formal semantics that accounts for both item relevance to a group and disagreement among the other members in the group. They evaluated their recommendation system through a comprehensive user study conducted on Amazon Mechanical Turk and demonstrate that incorporating disagreements is critical. Here are some papers [19, 20, 21] that dive into a detailed case study of Recommender systems and group formations. Apart from coming up with innovative and better approaches which are essential to building Recommender systems. Researchers have also explored some critical problems like the new user cold-start problem. This paper [21] is a comparative study between the different approaches to solving the problem. In this article, some basic approaches like MIPFGWC-CS, NHSM, FARMS, and HUFCF, are described. After experimenting with these approaches in various settings, it turns out that the NHSM performs achieves better computational time and accuracy compared to the rest.

In group recommendation systems, the idea is to group users and predict a recommendation list that satisfies the majority of the user in the group. While there have been a few approaches that take into account of the already existing user, no system was able to detect intrinsic user communities.

Group Recommendation with Automatic Identification of Users Communities [22] paper describes an algorithm that detects groups of users whose preferences are similar and predicts recommendations for such groups. A modularity-based Community Detection algorithm generates groups of various granularities, allowing a service provider to investigate the trade-off between the degree of personalizing of the recommendations and the number of channels. According to the experimental results, there was a linear increase in the quality of the recommendations. Furthermore, Researchers conducted extensive research on studying the neighborhood-based recommendation methods which are used in the Group Recommendation systems. This chapter [16] presents a comparative study of neighborhood-based methods and the main benefits of such methods as well as their characteristics. Most importantly, it deals with essential decisions that are to be taken in the process to serve the user better recommendation. In the end, it also covers the often-observed problems of sparsity and limited coverage. It becomes really difficult to predict accurate recommendations when there is a limited amount of information about the user/item. In terms of neighborhood, what if the user has just specified the trusted neighborhood and hasn't rated any of them? Or what if users have just rated but not specified as trusted or not? To deal with these problems, Guibing Guo published a paper [23] that proposes a resolution to the cold start problems and data sparsity problem. Their method turned out to be effective on three real-life dataset examples as it achieved the best performance and accuracy as well as other relevant parameters compared with other benchmarks. The only case when their model failed was when the trust and rating are out of reach. In that case, it required more information to rely upon and predict accurate recommendations.

Table 2: Literature Survey on Recommendation System.

Model	Domain	Algorithm	Year
Polylens [24]	Movies	Collaborative filtering	2001
Rule based Rec [25]	Movies	Rule based, heuristics	2010
GroupRem [26]	Movies	Aggregation	2013
COM[27]	Movies	Probabilistic	2014
SparseRec [28]	Music	SVM	2015
CATS [29]	Travel	Collaborative filtering	2006
TV-A [30]	TV programs	Aggregation	2009
Consensual preferences [31]	Study	Cross-validation	2015

Finally, our base paper Centrality-based group formation in group recommender systems [32] discusses the different centrality-based group formation techniques for GRS. Most studies so far have imposed four constraints: (1) a small number of users, (2) a small number of groups, (3) an average number of group members, and (4) complete knowledge of the network topological structure. To address these issues, they suggest a novel approach based on the network centrality principle that enhances the accuracy of recommendation lists for each party. Initially, the heads of the group are identified and consequently, groups of users with similar interests are formed. After the group formation, several group profiling techniques are employed to aggregate the preferences of the individual group members relative to their centralities.

3. Methodology

The approach suggested in [33] limits only to the use of Degree Centrality for weight calculation for individual users in the group and sets a predetermined value of base for exponential remapping of ratings. In this section, we suggest an approach to analyze different methods for preference aggregation and group choice prediction based on weighting individual preferences. Along with this a pathway to determine the value of base for exponential remapping of ratings that yield best results is suggested.

3.1. Approach for Analyzing Centrality:

Here we discuss an approach to analyze 3 different Centrality techniques and their impact on the predictive capability of the Group Recommender System. The following stages are proposed for the process:

- Group Forming
- Network Creation
- Weight Calculation
- Group Modeling

3.1.1. Group Formation

Group formation is suggested as the first step. It is used to group similar users together that have same preferences. Following the grouping method mentioned in [34], K- Means Clustering is applied to categorize each member into their respective preference group. K-Means Clustering is an unsupervised which is used to divide unlabelled data into different clusters. K-Means Clustering uses Euclidean distance was used as a distance metric. The entire input dataset was divided into clusters of 4, 8, 12, 20, and 50. This allows us to measure the impact on predictive capability of a centrality/group profiling technique with decrease in members in group.

3.1.2. Nerwork Creation

Preference Network is constructed as suggested in [33]. Different network is constructed for each group. The nodes of the network represent group members and the edges represent connection between them. These connections are weighed with node-to-node similarities. To calculate the weight of each connection Full Choice-set Distance measure (FullDist) is used. This metric is similar to Spearmans Footrule Distance function [23]. It considers members preferences for the full set of options (ChoiceSet), to compute the weight for connection between two group members u and v. It gives an undirected preferences relationship between pairs of group members:

$$FullDist_{u,v} = \sum_{i \in ChoiceSet} |score_u(i) - score_v(i)| \tag{1}$$

Where $score_u(i)$ is the rating given by user u to item i. This score is an integer that lies between 0 and 5.

3.1.3. Weight Calculation

Now after creation of a network in each group, we can apply network analysis methods *i.e.* centrality techniques to determine the weights of each node in the group. A group member with high centrality score is the person who shares a great deal of preferences with the other members of the group. There are different Centrality strategies for network analysis like Closeness Centrality(CC), Harmonic Centrality(HC) and Degree Centrality(DC) [35]. This three Centrality techniques were analyzed on the given dataset.

- **Degree Centrality:** is defined as the number of links incident upon a node. The degree can be interpreted in terms of the immediate risk of a node for catching whatever is flowing through the network.

  $$DC_u = deg(u) \tag{2}$$

  where DC_u represents the Degree Centrality of u and deg is the in degree of node u.

- **Closeness Centrality:** of a node is defined as the average length of the shortest path between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes. This centrality technique is not applicable to unconnected graph.

  $$CC_u = \frac{1}{\sum_{v \in UserSet} FullDist_{u,v}} \tag{3}$$

  where CC_u represents the Closeness Centrality of u

- **Harmonic Centrality:**also known as valued centrality, is a variant of closeness centrality. It enables the technique to be used in unconnected graphs.

  $$HC_u = \sum_{v \in UserSet1} \frac{1}{FullDist(u,v)} \tag{4}$$

  Where HC_u represents the Harmonic Centrality for user u

Now the weight wu of each member of the group is calculated by normalizing the centrality score CS(u) of user u present in its preference group G:

$$W_u = \begin{cases} 1, & \text{if } max(CS) = min(CS) \quad (5) \\ \dfrac{CS(u) - min(CS)}{max(CS) - min(CS)}, & \text{otherwise} \quad (6) \end{cases}$$

Where min(CS) and max(CS) are the minimum and maximum node centrality score within the graph G.

3.1.4. Group Modelling

The aggregation process of group member's profile in kumar2021surveythe same group is usually called group model/profile [36, 37]. Some popular group profiling strategies like Average, Least Misery, Average without Misery, Most Pleasure, etc are discussed in [19, 37]. Here we analyzed these four preference aggregation methods to generate Group Rating.

- **Average** The score assigned to an item i for group G is equal to the average score of item i that is evaluated by all users in group G:

$$g^i = \sum_{u \in G} \frac{r_u^i}{|G|} \tag{7}$$

- **Average without Misery:** The score assigned to an item i for group G is equal to the average score of item i that has been evaluated by all users in G, where each item score is greater than or equal to a certain threshold:

$$g^i = \sum_{u \in G} \frac{r_u^i}{|G|}, \forall r_u^i > \gamma \tag{8}$$

  Where, γ is threshold value.

- **Least Misery** The score assigned to an item i for group G is equal to the minimum score of the item i evaluated by users in G:

$$g^i = min_{u \in G} r_u^i \tag{9}$$

- **Most Pleasure** The score assigned to an item i for group G is equal to the maximum score of the item evaluated by users in G:

$$g^i = max_{u \in G} r_u^i \tag{10}$$

At the beginning, the group formation stage takes place as mentioned in the section 3.1.1. The entire dataset was divided into clusters of 4, 8, 12, 20 and 50

$$R =$$

$$\begin{pmatrix} r_1^1 & r_1^2 & r_1^3 & \dots & r_1^n \\ r_2^1 & r_2^2 & r_2^3 & \dots & r_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_n^1 & r_n^2 & r_n^3 & \dots & r_n^n \end{pmatrix}. \Longrightarrow \begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ G_K \end{pmatrix}.G_i = \begin{pmatrix} r_1^1 & r_1^2 & r_1^3 & \dots & r_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_x^1 & r_x^2 & r_x^3 & \dots & r_x^n \end{pmatrix}.$$

Where R is the input rating matrix, r_a^b is the rating given by user a to item b, G_i is one of the resultant group from the K-Means method.

Now for each preference group, preference network is created as mentioned in section 3.1.2. Formula(1) is used to determine the edge weights. This results in a network adjacency matrix. A small example for a group of 4 users 5 items is given:

$$\text{G} =$$

$$\begin{pmatrix} 4 & 0 & 0 & 8 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 7 & 10 & 0 \end{pmatrix}. \Longrightarrow N = \begin{pmatrix} 4 & 0 & 0 & 8 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 7 & 10 & 0 \end{pmatrix}.$$

Where N_i is the adjacency matrix of the preference network created from one of the group G_i. Now utilizing this adjacency matrix, we calculate the centrality of each node as mentioned in section 3.1.3

$$CC =$$

$$\begin{pmatrix} 0.0232558 \\ 0.0212765 \\ 0.0232558 \\ 0.020408 \end{pmatrix}, HC = \begin{pmatrix} 0.230952 \\ 0.217553 \\ 0.239898 \\ 0.204278 \end{pmatrix}, DC = \begin{pmatrix} 3 \\ 3 \\ 3 \\ 3 \end{pmatrix}.$$

Weights of each member is calculated using the centrality obtained by utilizing the formula(2).

$$W_{CC} = \begin{pmatrix} 1 \\ 0.304946 \\ 1 \\ 0 \end{pmatrix}, W_{HC} = \begin{pmatrix} 0.748838 \\ 0.372691 \\ 1 \\ 0 \end{pmatrix}, W_{DC} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Now this weight is multiplied by every rating of the corresponding user to give us a new rating matrix.

$$NG_{CC} = \begin{pmatrix} 4 & 0 & 0 & 8 & 0 \\ 0 & 0 & 2.744 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, NG_{HC} = \begin{pmatrix} 2.995 & 0 & 0 & 5.99 & 0 \\ 0 & 0 & 3.554 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 7 & 10 & 0 \end{pmatrix}, NG_{DC} \begin{pmatrix} 4 & 0 & 0 & 8 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 7 & 10 & 0 \end{pmatrix}.$$

Where NG represents weighted rating matrix for group . This weighted rating is now used to create group profile for each group. This done by following each of the profiling strategy discussed in section 3.1.4.

$GP_{CC}^{avg} = \begin{pmatrix} 1 & 0 & 0.686 & 2 & 0 \end{pmatrix}, GP_{HC}^{avg} = \begin{pmatrix} 0.748 & 0 & 0.838 & 1.497 & 0 \end{pmatrix}, GP_{DC}^{avg} \begin{pmatrix} 2.25 & 0 & 4 & 4.5 & 0 \end{pmatrix}$

where $GP_{CC}^{avg}$is the group profile using average strategy and closeness centrality technique, GP_{HC}^{avg} is the group profile using average strategy and harmonic centrality technique. GP_{DC}^{avg} is the group profile using average strategy and degree centrality technique

3.2. Approach for Exponential Remapping:

In the [33], it has already been concluded that exponential remapping of ratings yields in better predictions. In this section we discuss an approach to analyze the effect of different values of the base on prediction. The suggested approach has the following stages:

- Group Formation
- Preference Transformation
- Group Profiling.

3.2.1. Approach for Exponential Remapping:

In the [33], it has already been concluded that exponential remapping of ratings yields in better predictions. In this section we discuss an approach to analyze the effect of different values of the base on prediction. The suggested approach has the following stages: Group Formation (Section 3.1.1) Preference Transformation Group Profiling. (Section 3.1.4)

3.2.2. Preference Transformation:

we transform the group members' individual preferences from a linear scale into a non-linear one. The transformation formula used is:

$$newrating = a^{Oldrating} \tag{11}$$

Now for each group exponential transformation is applied on each group rating matrix. This is done just as discussed in section 3.2.1.

$$G_i = \begin{pmatrix} 4 & 0 & 0 & 8 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 7 & 10 & 0 \end{pmatrix} . \Longrightarrow NRG_i = \begin{pmatrix} 2.856 & 1 & 1 & 8.157 & 0 \\ 1 & 1s & 10.604 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3.719 & 1 & 6.274 & 13.785 & 1 \end{pmatrix} .$$

Where G_i is one of the group and NRG_i represents exponentially transformed rating matrix for group i, nr_i^j is weighted rating of item i by user j. Here a = 1.3, 2.856 is weighted rating for item 1 rates by user 1 and it is obtained from G_i. This exponentially transformed rating is now used to create group profile for each group. This done by following each of the profiling strategy discussed in section 3.1.4.

4. Evaluation

To compare the predictive capabilities of the implemented group profiling strategies along with Centrality technique, we use two different evaluation metrics: Mean Square Error(MSE) and precision of the recommendations at Top 2 (P@2).

$$MSE = \frac{(\sum_{i=1}^{t} |{r_g}^i - \tilde{p_g}^i|)^2}{t} \tag{12}$$

Mean Square Error is a standard metric for calculating model error. It is a standard technique to analyze the predictive capability of various suggested techniques. This metric evaluates the difference between the estimated rating(pi) generated by the group to the actual rating generated(r_i).

4.1. Precision@K

$$P@k = \frac{(\ \#CorrectTopkRecommendations)}{(\ \#TopkRecommendations)} \tag{13}$$

It is defined as the ratio of number of recommendation that was relevant to the k number of recommendation suggested. The numerator catches the number of recommendation that is correctly predicted by the aggregation technique. The denominator is the top k recommendation provided by the aggregation strategy. Precision is discussed [38]

4.2. Experiment 1

In the suggested approach in section 3.1, the various centrality technique has been analyzed in combination with some popular group profiling strategy for different cluster number k. This were analysed on MovieLens 100K. This experiment is an attempt to find the optimal cluster number k. With increase in the number of groups, we notice a reduction in MSE for every combination of centrality technique for weight calculation and group profiling method. Among the group profiling technique for a single centrality technique, it is observed that least misery aggregation provides the least value of MSE followed by the average profiling strategy. Coming to the Precision at 2 metric it is found that it behaves in an erratic fashion. When comparing group profiling technique for a centrality we find that average profiling gives the best result followed by least misery profiling. Comparing among the centrality technique it quite evident that closeness centrality gives the best result. Same conclusion was deduced when using MovieLens 1M dataset. All the results of the experiment is mentioned in table 1.

Table 3: Result for MovieLens 100K dataset on Average Startegy on non-linear Preference

Centrality Technique	Evaluation Metric	Aggregation Technique	K				
			4	8	12	20	50
Closeness Centrality	MSE	AVG	0.1078	0.0913	0.1346	0.1402	0.1296
		LM	0.011	0.027	0.035	0.079	0.111
		AWM	26.5036	24.2705	24.9629	21.8599	18.8918
		MP	2.7178	2.4600	2.4026	2.0934	2.2026
		Hybrid1	3.2381	2.9603	1.8759	2.1197	1.2377
		Hybrid2	5.9290	5.3709	4.1325	4.1122	3.2028
	P@2	AVG	1	0.9375	0.9	0.7857	0.8437
		LM	0.875	0.8125	0.9	0.6071	0.5937
		AWM	0.125	0	0.1	0.0357	0.125
		MP	0.125	0.125	0.05	0.0357	0.0937
		Hybrid1	1	0.8125	0	0.75	0.625
		Hybrid2	0	0.0625	0.05	0	0.0312
Harmonic Centrality	MSE	AVG	0.1185	0.0988	0.1401	0.1438	0.1317
		LM	0.0014	0.0274	0.0356	0.0798	0.1118
		AWM	26.4482	24.2720	25.0539	22.4794	18.8771
		MP	3.0613	2.6450	2.5489	2.1942	2.2920
		Hybrid1	2.9568	2.7165	1.7667	2.0727	1.2033
		Hybrid2	6.0055	5.3052	4.1735	4.1894	3.2762
	P@2	AVG	0.25	0.0625	0.1	0.0714	0.0625
		LM	0.125	0.0625	0.1	0.0714	0.0625
		AWM	0	0	0	0	0
		MP	0	0.0625	0	0	0
		Hybrid1	1	0.75	0.8	0.7142	0.625
		Hybrid2	0	0.0625	0.05	0	0.0312
Degree Centrality	MSE	AVG	0	0	0	0	0
		LM	0	0	0	0	0
		AWM	0	0	0	0	0
		MP	0	0	0	0	0
		Hybrid1	15.2104	13.3787	13.9663	14.4216	11.8262
		Hybrid2	15.2104	13.3787	13.9663	14.4216	11.8262
	P@2	AVG	0.25	0.125	0.1	0.0714	0.0625
		LM	0.25	0.125	0.1	0.0714	0.0625
		AWM	0.25	0.125	0.1	0.0714	0.0625
		MP	0.25	0.125	0.1	0.0714	0.0625
		Hybrid1	1	1	1	1	1

4.3. Experiment 2

In the suggested approach in section 3.2, the effect on prediction capability by varying exponential remapping of ratings is analyzed for various group profiling techniques along with varying the cluster number k. As before, with increasing the number of groups, the MSE error reduces. Varying the base value also affects the performance. An increase the value of a from 0 results in a decrease in MSE till a certain limit then MSE increases. From the experimental table 2 obtained it is clear that having the base value around 1.3 gives the best prediction output for all the aggregation techniques.

Table 4: MSE for MovieLens 100K dataset on Average Startegy on non-linear Preference

Aggregation Technique	Exponent base a	K=4	K=8	K=12	K=20	K=50
	0.01	0.9919	0.8720	1.0104	0.7447	0.3296
	0.1	0.9918	1.0103	0.8718	0.7446	0.3295
Average	0.3	0.9896	1.00797	0.86978	0.74266	0.3289
	0.4	0.9859	1.0043	0.8665	0.7397	0.3279
	0.5	0.9790	0.9977	0.8606	0.7343	0.3260
	0.6	0.9672	0.9866	0.8506	0.7253	0.3228
	1.3	0.7555	0.7898	0.6624	0.5530	0.2621
	1.4	0.8213	0.8451	0.7046	0.5882	0.2768
	1.5	1.0240	1.0201	0.8503	0.71459	0.3264
	1.6	1.4674	1.4057	1.1795	1.0028	0.4376
	0.01	10370.0228	3075.7262	1616.8101	719.9692	199.0663
	0.1	10370.0228	3075.7262	1616.8101	719.9692	199.0663
AWM	0.3	10370.0228	3075.7262	1616.8101	719.9692	199.0663
	0.4	10370.0228	3075.7262	1616.8101	719.9692	199.0663
	0.5	10370.0228	3075.7262	1616.8101	719.9692	199.0663
	0.6	10370.0228	3075.7262	1616.8101	719.9692	199.0663
	1.3	542.7489	167.4985	89.6432	39.9469	11.2677
	1.4	0	0	0	0	0
	1.5	0	0	0	0	0
	1.6	46.9809	15.2853	8.4996	3.9244	1.1659
	0.01	0.1761	0.3224	0.3186	0.3157	0.1639
	0.1	0.1765	0.3228	0.3188	0.3159	0.1640
LM	0.3	0.1802	0.3261	0.3213	0.3180	0.1649
	0.4	0.1843	0.3298	0.3241	0.3204	0.1659
	0.5	0.1916	0.3365	0.3293	0.3246	0.1677
	0.6	0.2066	0.3500	0.3397	0.3330	0.1712
	1.3	0.9998	0.9996	0.8331	0.6995	0.3197
	1.4	0.9998	0.9995	0.8330	0.6992	0.3195
	1.5	0.9998	0.9995	0.8335	0.6993	0.3196
	1.6	0.9999	0.9998	0.8335	0.7008	0.3201
	0.01	9.6171	7.6224	5.7938	4.2589	1.7618
	0.1	9.6169	7.6223	5.7938	4.2587	1.7618
MP	0.3	9.6166	7.6223	5.7938	4.2581	1.7615
	0.4	9.6165	7.6218	5.7937	4.2581	1.7615
	0.5	9.6164	7.6213	5.7933	4.2571	1.7609
	0.6	9.61625	7.6206	5.7930	4.2563	1.7605

Table 5: Precision@2 for MovieLens 100K dataset on Average Startegy on non-linear Preference

Aggregation Technique	Exponent base a	K=4	K=8	K=12	K=20	K=50
Average	0.01	0	0	0	0	0
	0.1	0	0	0	0	0
	0.3	0	0	0	0	0
	0.4	0	0	0	0	0
	0.5	0	0	0	0	0
	0.6	0	0	0	0	0
	1.3	0.875	0.9375	0.75	0.6	0.27
	1.4	0.875	0.9375	0.75	0.575	0.26
	1.5	0.875	0.9375	0.75	0.55	0.25
	1.6	0.875	0.9375	0.7083	0.55	0.24
AWM	0.01	0	0	0	0	0
	0.1	0	0	0	0	0
	0.3	0	0	0	0	0
	0.4	0	0	0	0	0
	0.5	0	0	0	0	0
	0.6	0	0	0	0	0
	1.3	0.75	0.75	0.6667	0.475	0.2
	1.4	1	1	0.8333	0.7	0.32
	1.5	1	1	0.8333	0.7	0.32
	1.6	0.875	0.875	0.791	0.65	0.3
LM	0.01	0	0	0	0	0
	0.1	0	0	0	0	0
	0.3	0	0	0	0	0
	0.4	0	0	0	0	0
	0.5	0	0	0	0	0
	0.6	0	0	0	0	0
	1.3	0.9998	0.9996	0.8331	0.7	0.32
	1.4	0.9998	0.9996	0.8331	0.7	0.32
	1.5	0.9998	0.9996	0.8331	0.7	0.32
	1.6	0.9998	0.9996	0.8331	0.7	0.32
MP	0.01	0.875	0.8125	0.5416	0.45	0.19
	0.1	0.875	0.8125	0.5416	0.45	0.19
	0.3	0.875	0.8125	0.5416	0.45	0.19
	0.4	0.875	0.8125	0.5416	0.45	0.19
	0.5	0.875	0.8125	0.5416	0.45	0.19
	0.6	0.875	0.8125	0.5416	0.45	0.19

5. Conclusion

With considerable growth in Internet users, a new aspect of group recommender systems (GRS) has received a lot of attention over the past years. In this paper, we work on the foundation laid by [33]to analyze different centrality techniques for centrality-based preference networks and the effect of non-linear transformation. We provided a simple and efficient approach to analyze those techniques. With extensive experimentation on real-world dataset MovieLens, 100K and Movielens 1M demonstrate that closeness centrality with average profiling strategy provides the best result. We provide an efficient approach to optimize the parameter for non-linear transformation.

References

[1] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender systems handbook, Springer, 2011, pp. 1–35.

[2] J. D. Garman, L. L. Sample, S. A. Steele, From playboy to prison: When pornography use becomes a crime, Deviant Behavior 42 (1) (2021) 18–36.

[3] J. S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, arXiv preprint arXiv:1301.7363.

[4] C. C. Aggarwal, et al., Recommender systems, Vol. 1, Springer, 2016.

[5] P. Brusilovsky, E. Millán, User models for adaptive hypermedia and adaptive educational systems, in: The adaptive web, Springer, 2007, pp. 3–53.

[6] J. F. McCarthy, T. D. Anagnost, Musicfx: an arbiter of group preferences for computer supported collaborative workouts, in: Proceedings of the 1998 ACM conference on Computer supported cooperative work, 1998, pp. 363–372.

[7] A. Crossen, J. Budzik, K. J. Hammond, Flytrap: intelligent group music recommendation, in: Proceedings of the 7th international conference on Intelligent user interfaces, 2002, pp. 184–185.

[8] P. De Meo, G. Quattrone, D. Ursino, A decision support system for designing new services tailored to citizen profiles in a complex and distributed e-government scenario, Data & Knowledge Engineering 67 (1) (2008) 161–184.

[9] C. Porcel, A. G. López-Herrera, E. Herrera-Viedma, A recommender system for research resources based on fuzzy linguistic modeling, Expert Systems with Applications 36 (3) (2009) 5173–5183.

[10] I. Garcia, L. Sebastia, E. Onaindia, On the design of individual and group recommender systems for tourism, Expert systems with applications 38 (6) (2011) 7683–7692.

[11] L. Ardissono, A. Goy, G. Petrone, M. Segnan, P. Torasso, Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices, Applied artificial intelligence 17 (8-9) (2003) 687–714.

[12] K. Schmidt, L. Bannon, Constructing cscw: The first quarter century, Computer supported cooperative work (CSCW) 22 (4-6) (2013) 345–372.

[13] B. Smyth, P. Cotter, A personalized television listings service, Communications of the ACM 43 (8) (2000) 107–111.

[14] A. Jastrzab, E. Skrzydlewska, Thioredoxin-dependent system. application of inhibitors, Journal of Enzyme Inhibition and Medicinal Chemistry 36 (1) (2021) 362–371.

[15] P. Grover, A. K. Kar, Big data analytics: A review on theoretical contributions and tools used in literature, Global Journal of Flexible Systems Management 18 (3) (2017) 203–229.

[16] S. Garg, S. Sinha, A. K. Kar, M. Mani, A review of machine learning applications in human resource management, International Journal of Productivity and Performance Management.

[17] I. Ali, S.-W. Kim, Group recommendations: approaches and evaluation, in: Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, 2015, pp. 1–6.

[18] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: Semantics and efficiency, Proceedings of the VLDB Endowment 2 (1) (2009) 754–765.

[19] L. Boratto, S. Carta, Modeling the preferences of a group of users detected by clustering: A group recommendation case-study, in: Proceedings of the 4th international conference on web intelligence, mining and semantics (WIMS14), 2014, pp. 1–7.

[20] S. Basu Roy, L. V. Lakshmanan, R. Liu, From group recommendations to group formation, in: Proceedings of the 2015 ACM SIGMOD international conference on management of data, 2015, pp. 1603–1616.

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

[22] A. Hernando, J. Bobadilla, F. Ortega, A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model, Knowledge-Based Systems 97 (2016) 188–202.

[23] L. Boratto, S. Carta, Art: group recommendation approaches for automatically detected groups, International Journal of Machine Learning and Cybernetics 6 (6) (2015) 953–980.

[24] M. Oconnor, D. Cosley, J. A. Konstan, J. Riedl, Polylens: A recommender system for groups of users, in: ECSCW 2001, Springer, 2001, pp. 199–218.

[25] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, K. Seada, Enhancing group recommendation by incorporating social relationship interactions, in: Proceedings of the 16th ACM international conference on Supporting group work, 2010, pp. 97–106.

[26] M. S. Pera, Y.-K. Ng, A group recommender for movies based on content similarity and popularity, Information Processing & Management 49 (3) (2013) 673–687.

[27] Q. Yuan, G. Cong, C.-Y. Lin, Com: a generative model for group recommendation, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 163–172.

[28] S. Ghazarian, M. A. Nematbakhsh, Enhancing memory-based collaborative filtering for group recommender systems, Expert systems with applications 42 (7) (2015) 3801–3812.

[29] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, P. Nixon, Cats: A synchronous approach to collaborative group recommendation, in: Florida Artificial Intelligence Research Society Conference (FLAIRS), 2006, pp. 86–91.

[30] R. Sotelo, Y. Blanco-Fernandez, M. Lopez-Nores, A. Gil-Solla, J. J. Pazos-Arias, Tv program recommendation for groups based on muldimensional tv-anytime classifications, IEEE Transactions on Consumer Electronics 55 (1) (2009) 248–256.

[31] A. K. Kar, A hybrid group decision support system for supplier selection using analytic hierarchy process, fuzzy set theory and neural network, Journal of Computational Science 6 (2015) 23–33.

[32] H. Mahyar, K. Ghalebi, S. M. Morshedi, S. Khalili, R. Grosu, A. Movaghar, Centrality-based group formation in group recommender systems, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 1187–1196.

[33] E. Ntoutsi, K. Stefanidis, K. Nørvåg, H.-P. Kriegel, Fast group recommendations by applying user clustering, in: International Conference on Conceptual Modeling, Springer, 2012, pp. 126–140.

[34] L. Boratto, S. Carta, A. Chessa, M. Agelli, M. L. Clemente, Group recommendation with automatic identification of users communities, in: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Vol. 3, IEEE, 2009, pp. 547–550.

[35] L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in: Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 119–126.

[36] S. Wasserman, Faust. k, Social Network Analysis: Methods and Applications.

[37] J. Kumar, Y. Ramanjaneyulu, K. S. Babu, B. K. Patra, A survey on group modeling strategies for recommender systems, in: New Paradigms in Computational Modeling and Its Applications, Elsevier, 2021, pp. 209–239.

[38] A. K. Kushwaha, A. K. Kar, P. V. Ilavarasan, Predicting information diffusion on twitter a deep learning neural network model using custom weighted word features, Responsible Design, Implementation and Use of Information and Communication Technology 12066 (2020) 456.