Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 1 (2013) pp. 99 - 108

Group Recommendation For Mitigating New User Problem: A Modified OCRG

Harita Mehta[1], Punam Bedi[2] and Veer Sain Dixit [3*]

[1] Department of Computer Science AND College,
University of Delhi, New Delhi, India
harita_mehta@yahoo.com

[2] Computer Science Department, University of Delhi
New Delhi, India
pbedi@cs.du.ac.in

[3] Department of Computer Science ARSD College,
University of Delhi, New Delhi, India
veersaindixit@rediffmail.com
*(*corresponding author)*

***Abstract*: Providing Recommendations to a Group of users rather than individuals is an emerging research field. Group Recommenders can predict the interest level of a user by aggregating rating preferences of group members. The New User Problem is inherited by Group Recommender Systems, because there is relatively little information about his rating preferences. In this paper, a modified Online Cold Recommendation Generator (OCRG) is proposed to find group recommendations for new users.**

Group of similar users is generated based on positive and negative user preferences. Here, OCRG is extended to identify similar user groups based on demographic attributes of the new user. The proposed modified OCRG aggregates the positive and negative ratings of group members using Item Entropy and Item Popularity, to find attraction, repulsion and balanced inclination of new user towards existing items within the group. The experimental results on Movie Lens dataset show significant improvements in overcoming new user problem in group recommender systems using Balanced Inclination aggregation strategy rather than average aggregation strategy.

Keywords: Group Recommendation, Demographic Filtering, New User Problem, Weighted Item Entropy, Item Popularity, Positive and Negative Ratings.

I. Introduction

When the information goes beyond the processing capacity of a web surfer, it leads to stress and loss of time. This state of frustration is commonly termed as information overload problem which affects decision making capacity of a web surfer. Recommender systems [1] have been developed as an effective solution to this problem. “Recommender systems are personalization tools that attempt to provide information which is tailored to individuals based on knowledge about their preferences.” These systems help users to make quality decisions and in turn find the items they would like to favor the most. Examples of such applications include recommending books, CDs, movies, news articles, etc. to users in many e-commerce websites.

Among many alternatives, the Collaborative Recommender Systems [2, 3] are generally accepted to be one of the most successful recommendation techniques. The basic idea of Collaborative Filtering algorithms is to recommend items that users with similar preferences (rating patterns) have liked in the past. Such users are usually referred as Similar Taste Users (STUs). A key advantage of such systems is that it does not rely on content of items and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. In this paper, two issues related to collaborative filtering are discussed as follows.

Firstly, the Collaborative Recommender Systems suffer from New User problem, which describes the situation in which a new user has to provide a sufficient number of ratings so that the recommendation algorithm is able to make reliable and accurate recommendations.

Secondly, the items recommended by the collaborative filtering are often or mostly used by the groups rather than by individuals, for example recommending a movie or a television show to a new user who may watch the recommended movie with family or with work colleagues, or friends.

Pure collaborative filtering cannot help in inferring inclination (attraction and repulsion both) of a new user towards existing items. This is because no rating information is available to form any basis for user taste. Thus, the system must acquire some information about the new user in order to make personalized predictions. Demographic attributes of a user are not related to his/her ratings of various items. Each user’s demographic information (such as age, gender, occupation) can be used to initialize the recommendation generation process for new users. Demographic techniques are similar to collaborative filtering in the way they form “people-to-people” correlations. The only difference is in the nature of data they use. The advantage of the demographic approach over collaborative technique is the ability to provide recommendations without a history of user ratings.

Demographic information is helpful in bridging the gap between existing items and new users by inferring inclination of users for these existing items. In this paper, the proposed approach assumes that every user is required to register and provide his/her demographic information before using the system.

A Group Recommender System is a recommender system aimed at generating a set of recommendations that will satisfy users of the group. As this scenario, (where the recommended items are inherently consumed by groups of users rather than by individuals) increases, we face more and more application domains, in which group recommendations are preferred over individual recommendations. For example, music selection in public places [4], tourist attractions [5], holiday destinations [6], movies [7], and TV programs [8], are few examples of group recommendations. The attempt here is to find group recommendations for a new user based on his demographic attributes.

When switching from individual recommendations to group recommendations, the main challenge is how to record and combine the preferences of many different users in the group. In Group recommenders, there is a need for an aggregation mechanism to represent the group recommendations. To aggregate the preferences of all the group members, two methods in literature are suggested [9]. The first strategy creates a joint profile for all users in the group and provides the group with recommendations computed with respect to this joint profile [8]. The second strategy aggregates the recommendations of all users in the group into a single recommendation list [10, 11]. In the proposed approach, individual user profiles (preferences) have been aggregated rather than individual recommendations.

One of the aggregation strategy [12, 13] is to average the preferences of individual users in order to obtain Top N group recommendations for the new user. The group recommendation quality is highly dependent on the way the individual preferences are aggregated. Group recommender systems must combine and balance preferences from individuals across a group of users with a common purpose, in order to tailor preferences to the group as a whole. Selection of aggregation strategy is a distinguishing area of study applicable for group based recommenders. In this paper, average aggregating method is revised to generate better group recommendations. The aggregation method is termed as Balanced Inclination of a user for an item.

Before the generation of recommendations by a group recommender, users are required to express individual ratings for the items available which reflect their taste. When a user gives positive rating (assuming rating greater than 3 on a scale of 5) to an item, it reflects that the user is attracted towards this item. When a user gives negative rating (assuming rating less than 4 on a scale of 5) to an item, it leads to repulsion of the user from that item. In fact, combination of positive and negative ratings, given by all the users of the group for a particular item, governs the degree of balanced inclination of a new user for that item.

We propose a modified version of Online Cold Recommendation Generator (OCRG) [14] to compute the inclination of new user towards existing items, in order to generate group recommendations. It uses positive and negative ratings of existing items in a group of users along with demographic attributes of new users. In proposed approach, our purpose is to uncover that whether the balanced inclination is more appropriate than average strategy when generating group recommendations.

In this paper, the major contributions are as follows: 1) partitioning of STUs into groups, 2) identification of most promising group and 3) aggregation of user preferences of most promising group to efficiently compute group recommendations for new users without their history information. We have also evaluated both the efficiency and effectiveness of our approach using a Movie Lens dataset of movie ratings.

The remainder of the paper is organized as follows. Section 2 highlights the research related to New User Problem and Group Recommendations. Section 3 details the proposed approach to generate group recommendations with emphasis on balanced inclination of group members on existing items. Section 4 highlights the testing procedure and discusses the results based on real world data set. Section 5 concludes the paper and provides an overview of future work.

II. Related Work

A. Group Recommender Systems

Group Lens [3] and Video Recommender [15] generated predictions using Collaborative filtering algorithms. Amazon.com, Movie Critic and Jester which gives recommendations for books, movies and jokes respectively, are some more examples [16]. PHOAKS is another collaborative recommender system which guides people to find relevant information over the web [17]. All these systems generate recommendations in which, the items to be recommended come from other people with similar taste [18]. Similar taste is not only composed of positive ratings but also negative ratings given by the users. Many Collaborative Recommender Systems have explored a range of positive and negative rating approaches [19] to capture user's current taste for various items like movie, book, web page etc.

Group Recommender Systems [20] were developed to support the recommendation process in activities that involve more than a person. Group-based recommendations are pertinent to many domains and applications, such as music [4], movies or TV programs [7, 8], tourism [5, 13], and others.

Some well known Group Recommender Systems are MUSICFX [4], POLYLENS [7], INTRIGUE [5] and YU'S TV RECOMMENDER [8]. MUSICFX selects a radio station for background music in a fitness centre. This group recommender system suits the taste of a group of people working out at a given time. POLYLENS recommends movies where an individual's tastes are inferred not only from ratings but also from social filtering. INTRIGUE recommends places to visit for tourist groups by further sub grouping within a group. YU'S TV RECOMMENDER predicts a television program for a group to watch. The bases for these recommendations are on the individuals' preferences for program features such as genre, actors, and keywords. Other work considers social relationships and interactions among group members when aggregating the predictions [9, 21, and 22]. Model member interactions, social relationships, domain expertise, and dissimilarity among the group members are modeled when choosing a group decision strategy.

B. New User Problem

It may happen sometimes that the system is unable to find any similar taste users for the target user, because the target user is new to the system. These new users are unaware of existing items and have no historical rating of these items. Providing effective recommendations for such new users is of fundamental importance to collaborative recommender systems.

The group recommender inherits this problem too because it aggregates the predicted ratings for each group member. Solutions to the cold start problem for single person recommenders are summarized in [23]. Solutions include: non-personalized recommendations for cold start users using population averages; intelligent ways to find more ratings [24, 25]; and hybrid recommenders that resort to content-based recommendations when there are insufficient ratings to make collaborative recommendations [26, 27].

These new users can be classified by their personal attributes. Recommendations can be obtained based on their demographic classes. Demographic data increases the quality of information retrieval tasks [28]. Weber and Castillo [29] used demographic information like average income, race, etc. to find difference between groups in a search engine scenario. Moreover, the demographic attributes of users can be used to initialize the recommendation generation process of a new user [30].

There are a many existing hybrid approaches which are able to make new user recommendation. Pazzani [31] proposed a hybrid method that recommends items based on vote of four different algorithms: user-user collaborative filtering, content-based, demographic-based, and collaboration via content. Stern et al. [32] proposed a probabilistic model that combines user and item metadata with users' historical ratings to predict the users' interaction on items. Pennock and Horvitz [33] proposed the use of a “value-of-information” to discover the most valuable ratings information for a user. Another approach to solving the new user problem creates pre-made user categories and quickly assigns new users to one of them. The partitioning can be accomplished by asking pre-determined questions from the user that builds a user preference structure. This helps in entering the user into the system without requiring a substantial number of ratings [34].

C. Group Aggregation Strategy

To date, group recommendations have been mostly generated using two approaches: aggregating individual preferences into group models or aggregating individual predictions into group predictions. Berkovsky and Freyne [2] compared the two approaches in a recipe recommendation problem and found that the first one performs slightly better. But, an extensive comparison of the two approaches is still missing in the literature. INTRIGUE and POLYLENS aggregate recommendations, while MUSICFX and YU’S TV RECOMMENDER aggregate profiles.

Masthoff [12, 13] employed user studies, not to evaluate specific approaches, but to determine which group aggregation strategies people actually use. Results indicated that people particularly use the following strategies: Average, Average without Misery and Least Misery. Average Strategy is the basic group aggregation strategy that assumes equal influence among group members and calculates the average rating of the group members for any given item as the predicted rating. POLYLENS uses the Least Misery Strategy, assuming groups of people going to watch a movie together tend to be small and that a small group tends to be as happy as its least happy member. INTRIGUE uses a weighted form of the Average strategy. MUSICFX uses a variant of the Average without Misery Strategy. YU’S TV RECOMMENDER uses a variant of the Average Strategy and found that their aggregation worked well when the group was quite homogenous, but that results were disliked when the group was quite heterogeneous. A group is homogeneous if all the group members rate similarly for the items whereas a group is heterogeneous if all the group members rate differently for the items.

In the proposed approach, group recommendations are generated by aggregating individual preferences. The main issue in this case is how to aggregate the preferences for items produced for each group member into a single group recommendations’ list.

A new approach is proposed, which is a variant of average strategy and show that it will work well when the group is homogeneous and as well as heterogeneous. Average strategy and the proposed strategy have been tested in our experiments.

Reference [35] explores various techniques to determine the best items which can be recommended to a new user. These techniques make use of strategies based on item popularity, user personalization, item entropy, and also combinations of the above. They found that, item entropy sometimes choose those items which had low item popularity. Thus, they used a balanced technique by combining item entropy and item popularity. They found that the recommendations obtained by using this balanced technique were better as compared to those recommendations that were generated using item entropy alone.

Item popularity and item entropy are usually based on positive ratings of that item given by all the users in the knowledge base. We believe that, the resulting inclination of a new user for this item indicates only the probable attraction of this new user towards this item. Negative ratings of this item could potentially be applied to generate inclination of this new user indicating the probable repulsion from this item. For instance, the Adaptive Radio [36] explored the value of explicitly modeling negative preferences for group recommendation. They used negative preferences to determine which solutions are unsatisfactory to individual users and assumed that remaining solutions are satisfactory. The paper intends to build balanced user inclination towards an existing item, based on positive and negative ratings of this item. This balanced inclination of a user for an existing item is computed using item entropy and item popularity with respect to positive and negative ratings in a combined manner [14].

III. Proposed Scheme

The architecture of proposed Group Recommender System is shown in Figure 1 and Top N Group Recommendations are generated for the new user. The main components are Interface Unit, Offline Unit and Group Recommendation Unit.

Target User and the Recommender System are two basic entities in any recommendation generation process. Interface unit acts as an interface between these entities. It fetches the demographic attributes (gender, age and occupation) from the current session of target user and sends the request to the

Group Recommendation Unit, where Top N Group Recommendations are furnished. Finally, the Interface Unit displays the aggregated Group Recommendations for the target user during his/her current session. Offline Unit creates knowledge base, which is used by Group Recommendation Unit. The backbone of Offline Unit is Group Formation Unit which generates group of similar users. The selection of users in the same group is based on similarity of positive and negative preferences between them. Group Recommendation Unit is composed of Group Identification Unit and Group Aggregation Unit. The Group Identification Unit identifies a particular group for the new user based on his/ her demographic attributes. Group Aggregation Unit aggregates the individual preferences of users in the identified group in order to generate Top N group recommendations. It uses balanced inclination approach, a variant of average strategy to aggregate the preferences of users within the group. The remainder sub sections, briefly discusses the working of Group Formation Unit, Group Identification Unit and Group Aggregation Unit.

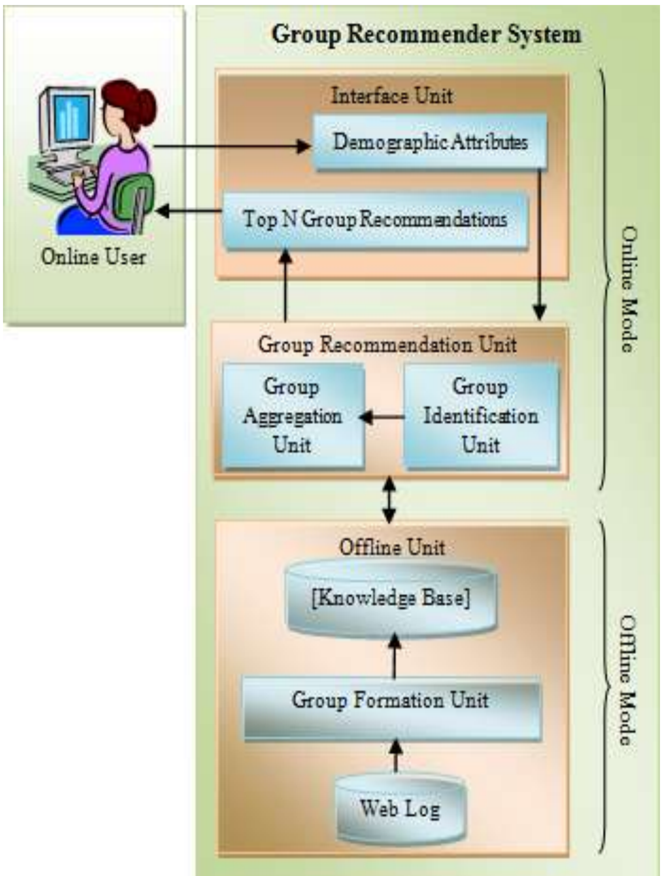

Figure 1. Proposed Framework

A. Group Formation Unit

Movie Lens uses users' ratings matrix '*M*' (1 to 5 likert scale) depicting users likes (rating greater than 3 on likert scale) and dislikes (rating less than 4 on likert scale) to generate personalized recommendations. The user rating matrix 'M' is split into Training (T_1) and Test matrix (T_2); T_1 matrix is further split into training level I matrix (L_I) and training level II matrix (L_{II}) which are required inputs of Offline Unit. The roots of the proposed Offline Unit have been extensively discussed in [37, 38]. The dispersion of positive and negative preferences between target user U_t and user U_x with respect to the set of rated items represents the similarity between these two users. After division of user session into two levels, if $TSim_p(U_t, U_x)$ and $TSim_n(U_t,U_x)$ are both non zero quantities, then the harmonic mean of the of similarity of positive and negative preferences, at both the levels defines the Combined Inter user similarity $Sim_c(U_t ,U_x)$ which is calculated using Equation 1. If $TSim_p(U_t, U_x)$ is zero then the Combined Inter user similarity $Sim_c(U_t ,U_x)$ is reduced to $TSim_n(U_t,U_x)$ only. Likewise, if $TSim_n(U_t, U_x)$ is zero then the Combined Inter user similarity $Sim_c(U_t ,U_x)$ is reduced to $TSim_p(U_t,U_x)$ only.

$$Sim_c(U_t,U_x) = \frac{2 * TSim_p(U_t,U_x) * TSim_n(U_t,U_x)}{(TSim_p(U_t,U_x) + TSim_n(U_t,U_x))} \qquad (1)$$

where,

$$TSim_p(U_t,U_x) = Sim_p^I(U_t,U_x) + Sim_p^{II}(U_t,U_x)$$
$$TSim_n(U_t,U_x) = Sim_n^I(U_t,U_x) + Sim_n^{II}(U_t,U_x)$$

$$Sim_p(U_t,U_x) = 1 - normalize\left(\sum_{k=1}^{Pcount} W_k \left[p\left(P_k(U_t,U_x)\right)\log_2 p\left(P_k(U_t,U_x)\right)\right]\right)$$

$$Sim_n(U_t,U_x) = 1 - normalize\left(\sum_{k=1}^{Ncount} W_k \left[p\left(N_k(U_t,U_x)\right)\log_2 p\left(N_k(U_t,U_x)\right)\right]\right)$$

`

where, Sim_p (U_t, U_x) and Sim_n (U_t, U_x) are Positive and Negative inter user similarities between these two users respectively. These similarities are calculated at both the levels. p($P_k(U_t, U_x)$) and p($N_k(U_t, U_x)$) are the probability density functions of positive and negative preferences between these two users. Total number of positive and negative preferences states between these users, is stored in Pcount and NCount respectively. For Sim_p (U_t, U_x), weight W_k is set to total number of positive preferences, given by all users to item k. Similarly, for Sim_n (U_t, U_x), it is set to total number of negative preferences, given by all users to item k. For a movie say 'k', rated by either target user U_t and / or by another user U_x, positive preference '$P_k(U_t ,U_x)$' and negative preference '$N_k(U_t ,U_x)$' are defined in (2a) and (2b).

$$P_k(U_t,U_x) = \begin{Bmatrix} 1 \text{ if } M(U_t,k) = 'L' \,\&\, M(U_x,k) = 'L' \\ 0 \text{ Otherwise} \end{Bmatrix} \qquad (2a)$$

$$N_k(U_t,U_x) = \begin{Bmatrix} 1 \text{ if } M(U_t,k) = 'D' \,\&\, M(U_x,k) = 'D' \\ 0 \text{ Otherwise} \end{Bmatrix} \qquad (2b)$$

The similarity between any two random users rests on the similarity computation at both the levels. If the similarity of positive and negative preferences at level II is greater than the similarity of positive and negative preferences at level I, then it shows that the user pair had similar positive and negative preferences throughout the session. Hence, $Sim_c(U_t ,U_x)$ decides the degree of similar taste between two users.

Defined simply as a set of users, a user group can be formed on a recurring basis, e.g., friends who meet regularly for dinner. We are mainly motivated by the observation that Combined Inter user similarity $Sim_c(U_t, U_x)$ can be used to cluster users in small groups with strong similarity. This way, our framework applies user clustering for organizing users into groups of users with similar preferences. To do this, we employ k-means clustering algorithm.

The support of each group for all possible demographic classes is calculated using Equation 3. 18 demographic classes were defined on the movie lens dataset (see section 4 for details). These groups along with their Support for each demographic class are stored in the knowledge base. Thereafter, we propose the use of these groups to efficiently locate similar users for a new user.

$$Support(G_i, \mathrm{D}_c) = \frac{No \text{ of Users} \in \text{Class } \mathrm{D}_c}{No \text{ of Users in } G_i} \quad (3)$$

The algorithm for Group formation Unit is depicted in figure 2. It takes three inputs. User preferences in the form of PV matrix are obtained from the movie lens dataset and serves as first input. The second input 'k' is the total number of groups (clusters) formed by applying kmeans clustering on similarity matrix SimMatrix. It is a two dimensional User X User matrix where a cell value at (r,c) depicts the Combined Inter user similarity between user r and user c of PV matrix. The groups are stored in G_{Set}. All possible demographic classes stored in DSet, which is given as third input to Group Formation Unit. Support of each group G_i stored in G_{Set} for each Demographic class D_C stored in D_{Set} is obtained in SupportList. G_{Set} along with SupportList is stored in the knowledge base.

Input:
PV = {Rating Matrix}
k = {No of Groups Required}
D_{Set} = {Demographic Attribute Classes}
Process: Group Formation
Split PV into T1 and T2
Split T1 into L1 and L2
Do
SimMatrix(U_t, U_x) = Sim_C(U_t, U_x)
Repeat for each user pair in PV
G_{Set} = kmeans(SimMatrix, k)
Do
For each $D_C \in D_{Set}$
SupportList(G_i,C)=Support(G_i, D_C)
Repeat for each Gi $\in G_{Set}$
Output:
KB = {G_{Set}, SupportList}

Figure 2. Algorithm for Group Formation Unit

B. Group Identification Unit

When a new user enters into the system, the demographic attributes of this new user defines his/ her demographic class D_T. From all the groups stored in the knowledge base, group having maximum support for demographic class D_T is identified. Finally, Group recommendations for users are produced with respect to the preferences of group members of the identified group G_T, without extensively searching for similar users in the whole database. If more than one group have maximum support for the demographic class D_T, for simplicity the Group Identification Unit obtains first similar group.

C. Group Aggregation Unit

In order to generate group recommendations for new users, Group Aggregation Unit uses the positive and negative ratings of existing items given by all the users in G_T having maximum support for the demographic class D_T. It is obtained from Group Identification Unit. Group Aggregation Unit finds balanced inclination, $I_B(U_T, G_T, i)$, of the new user towards existing movie, reflecting new users inclination for this movie. Balanced inclination of a new user, over all the existing movies helps the system to present promising group recommendations, thus mitigating new user problem.

Popularity of a movie indicates how frequently users have rated that movie. It is defined in (4).

$$P_i = \frac{\sum_{u=1}^{n} r_{u,i}}{N_i} \quad (4)$$

where, $r_{u,i}$ are the rating values given by all those n users who have rated movie i. N_i are the total number of users who have rated movie i. Since the popularity P_i is calculated with respect ratings given by all the users in the knowledge base, it can be referred as global popularity of movie i. In simple terms, it reflects the average of preferences of all the users in the knowledge base, with respect to item i. But here, we intend to find local popularity of movies with respect to the demographic attributes of the new user.

Local Popularity of movies is calculated with respect ratings given by all those users who belong to G_T having maximum support for D_T and is defined in (5).

$$LP_{i,G_T} = \frac{\sum_{u=1}^{n} r_{u,G_T,i}}{N_i} \quad (5)$$

where, $r_{u,G_T,i}$ are the rating values given by all those n users who belong to G_T having maximum support for D_T and have rated movie i. N_i are the total number of users who belong to G_T having maximum support for D_T and have rated movie i. In simple terms, it reflects the average of preferences of all the users within the group G_T, with respect to item i.

Local Popularity of movies, being simple average of rating in the specified group, works well for homogeneous groups. A disadvantage of using local popularity as a measure to generate group recommendations for new users is so that unpopular movies may be hard to recommend. Also, for heterogeneous groups, it can be very difficult to obtain a single recommendation that satisfies every member and the general group satisfaction is not always the average of the satisfaction of its members as different people have different expectations. In such a situation, Entropy may be another measure to generate group recommendations for new users.

In this paper, Entropy [39 and 40] of a movie represents the dispersion of ratings of all the users on that movie. It is defined in (6).

$$H(i) = -\sum_{i=1}^{n} prob(r_i) \log_2 prob(r_i) \quad (6)$$

where, $prob(r_i) = \frac{\rho}{\tau}$

Here, ρ signifies number of users who have given positive rating to movie i. τ signifies number of users who have rated movie i. $prob(r_i)$ is the probability density function of movie i, based on the positive ratings given by n users. Here, one limitation of entropy is quite obvious that it gives the measure of information as the function of only the probability with which a movie is rated, without considering the qualitative weight of that particular movie. To overcome this problem [41] weight W_i of the movie i, can be attached to its individual entropy $H(i)$. In fact, local popularity of movie i can serve as its weight.

Here our aim is to find balanced inclination of a new user towards movie i. Balanced Inclination can be calculated using probable degree of attraction and probable degree of repulsion. Firstly, we find probable degree of attraction $D_A(U_T, G_T, i)$ of new user towards an existing movie i which is defined in (7).

$$D_A(U_T, G_T, i) = -LP^{+ve}_{i,G_T} * (p(G_{T_i}{}^{+ve}) \log_2 p(G_{T_i}{}^{+ve})) \quad (7)$$

where, $p(G_{T_i}{}^{+ve}) = \frac{\rho}{\tau}$, $LP^{+ve}_{i,G_T} = \frac{\sum_{u=1}^{n} r_{u,G_T,i}}{\tau}$

Here, ρ signifies number of users who belong to G_T having maximum support for D_T and have given positive rating to movie i. τ signifies number of users who belong to G_T having maximum support for D_T and have rated movie i. $p(G_{T_i}{}^{+ve})$ is the probability density function of movie i based on the positive ratings given by those n users who belong to G_T having maximum support for D_T. $r_{u,G_T,i}$ are the positive rating values given by n users who belong to G_T having maximum support for D_T and have rated movie i. LP^{+ve}_{i,G_T} is the local popularity of movie i among those users who have given positive rating to movie i and who belong to G_T having maximum support for D_T. The resulting $D_A(U_T, G_T, i)$ of new user U_t with respect to positive ratings only, depicts user interests which are usually incomplete. Secondly, we find probable degree of repulsion, $D_R(U_T, G_T, i)$ of new user towards an existing movie i which is defined in (8).

$$D_R(U_T, G_T, i) = -LP^{-ve}_{i,G_T} * (p(G_{T_i}{}^{-ve}) \log_2 p(G_{T_i}{}^{-ve})) \quad (8)$$

where, $p(G_{T_i}{}^{-ve}) = \frac{\rho}{\tau}$, $LP^{-ve}_{i,G_T} = \frac{\sum_{u=1}^{n} r_{u,G_T,i}}{\tau}$

Here, ρ signifies number of users who belong to G_T having maximum support for D_T and have given negative rating to movie i. τ signifies number of users who belong to G_T having maximum support for D_T and have rated movie i. $p(G_{T_i}{}^{-ve})$ is the probability density function of movie i, based on the negative ratings given by those n users who belong to G_T having maximum support for D_T. $r_{u,G_T,i}$ are the negative rating values given by n users who belong to G_T having maximum support for D_T and have rated movie i. LP^{-ve}_{i,G_T} is the local popularity of movie i among those users who have given negative ratings to movie i and who belong to G_T having maximum support for D_T.

Now, we find balanced inclination of new user towards an existing movie which is defined in (9). If the preferences of all the group members are positive for a particular item, then the group is positive homogeneous with respect to that item. If the preferences of all the group members are negative for a particular item, then the group is negative homogeneous with respect to that item. In case of mixed (positive and negative) preferences, the group is heterogeneous. In reality, the ratio of heterogeneity over homogeneity within a group is high. And as the group size increases, the possible growth in this ratio is obvious.

$$I_B(U_T, G_T, i) = \frac{(LP^{+ve}_{i,G_T})}{\tau} \quad (9a)$$

$$I_B(U_T, G_T, i) = \frac{(LP^{-ve}_{i,G_T})}{\tau} \quad (9b)$$

$$I_B(U_T, G_T, i) = \frac{2 * D_A(U_T, G_T, i) * D_R(U_T, G_T, i)}{D_A(U_T, G_T, i) + D_R(U_T, G_T, i)} \quad (9c)$$

Here, balanced inclination can be interpreted as harmonic mean of probable degree of attraction and repulsion of new user towards an existing movie if the group is heterogeneous as shown in (9c). If the group is positive homogeneous with respect to item i, then balanced inclination towards item i is the ratio of LP^{+ve}_{i,G_T} and τ as shown in (9a). If the group is negative homogeneous with respect to item i, then balanced inclination towards item i is the ratio of LP^{-ve}_{i,G_T} and τ as shown in (9b). The algorithm for the Group Recommendation Unit is depicted in figure 3.

Input:
KB = {G_{Set}, SupportList}
D_T = {Demographic Attribute Class of New User} **Process:**
Step 1: Group Identification
$G_T = \max(\text{Support}(G_i, D_T))$ where, $G_i \in G_{Set}$
Step 2: Group Aggregation
Do
If G_T is positive homogeneous wrt item i
$I_B(U_T,G_T,i) = LP^{+ve}_{i,G_T} / \tau$
If G_T is negative homogeneous wrt item i
$I_B(U_T,G_T,i) = LP^{-ve}_{i,G_T} / \tau$
If G_T is hetrogeneous wrt item i
$N = D_A(U_T,G_T,i) * D_R(U_T,G_T,i)$
$D = D_A(U_T,G_T,i) + D_R(U_T,G_T,i)$
$I_B(U_T,G_T,i) = 2*N/D$
Repeat for each item in G_T
Sort items in descending order of $I_B(U_T,G_T,i)$
Output:
Top N Group Recommendations for New User 'U_T'

Figure 3. Algorithm for Group Recommendation Unit

The algorithm in figure 3 takes two inputs. The knowledge base prepared by the Group Formation Unit is given as first input. The demographic attributes of the new user, supplied at the time of registration, are mapped onto pre defined demographic attribute classes stored in D_C. The corresponding demographic class D_T of new user is given as second input to Group Recommendation Unit. From the groups stored in G_{Set}, the group which has maximum support for D_T is fetched and stored in G_T. In order to achieve group preference for a particular item, the preferences of all the group members are analyzed and the corresponding balanced inclination is obtained using (9). The items are ranked in descending order of balanced inclination and finally Top N group recommendations are presented to the new user.

The following example illustrates step by step analysis of Group Aggregation Unit. Suppose, the total number of items in the PV Matrix is 4. Let the group G_T obtained by Group Identification Unit consist of 5 users (group members). The preferences of these group members, for all 4 items, are depicted in Table 1. By analyzing the preferences of all the group members with respect to each item, the nature of the group can be determined with respect to each item. This is shown in last row of Table 1. For example, all the group members have given negative preferences for item 2, so the group G_T is homogeneous negative with respect to item 2.

Table 1. Preferences of Group Members in G_T.

Group G_T			**Items**	
Users	**Item$_1$**	**Item$_2$**	**Item$_3$**	**Item$_4$**
User$_1$	5	3	5	5
User$_2$	4	2	4	4
User$_3$	5	3	3	5
User$_4$	4	2	2	2
User$_5$	5	1	1	1
Nature Of Group	Pos. Homo.	Neg. Homo.	Hetero.	Hetero.

By applying (9), the authors have obtained Balanced Inclination of the group G_T for all the items, as shown in Table 2. As the group G_T is heterogeneous with respect to item 3 and 4, (9c) is applied for these items. The group G_T is Positive Homogeneous and Negative Homogeneous with respect to item 1 and 2 respectively. Therefore, (9a) and (9b) are respectively used for calculating Balanced Inclination for these items. In order to compare the proposed balanced inclination strategy with the average strategy (Local Popularity as shown in (5)), for the same group G_T, the average preferences are also obtained and depicted in the last row of Table 2.

Table 2. Balanced Inclination and Average Strategy.

Balanced Inclination	**Items**			
Nature Of Group	**Item$_1$**	**Item$_2$**	**Item$_3$**	**Item$_4$**
Positive Homogeneous	0.92	-	-	-
Negative Homogeneous	-	0.44	-	-
Heterogeneous	-	-	0.68	0.50
Average Strategy	4.6	2.2	3	3.4

In order to pass the Top N Group Recommendations for New User U_T, the items of the group G_T should be sorted in descending order of Balanced Inclination or average. The sorted order of items with respect to the two strategies is shown in table 3. Clearly, the order of the proposed Balanced Inclination strategy is different from the average strategy. Thus, Top 2 recommendations with respect to Balanced Inclination strategy are Item 1 and Item 3, whereas Top 2 recommendations with respect to Average strategy are Item 1 and Item 4, as shown in Table 4.

Table 3. Order of Items.

Strategy	**Order of Items**
Balanced Inclination	<Item$_1$, Item$_3$, Item$_4$, Item$_2$ >
Average (Local Popularity)	<Item$_1$, Item$_4$, Item$_3$, Item$_2$ >

Table 4. Top N Recommendations.

Strategy	**Top 2 Recommendations**
Balanced Inclination	<Item$_1$, Item$_3$ >
Average (Local Popularity)	<Item$_1$, Item$_4$ >

IV. Experiment

A. Data Set

In this paper, authors performed off-line evaluations where, groups are formed from the users of a traditional (i.e., single user) recommender system. Group Formation Unit sampled the groups on the basis of Combined Inter User Similarity.

Group recommendations are offered to group members and are evaluated independently by them, as in the classical single user case, by comparing the predicted ratings with the ratings observed in the test set of the user. The group recommendations are generated to suit simultaneously the preferences of all the users in the group and our intuition suggests that they cannot be as good as the recommendations generated for individual users. So, we do not need the joint group evaluations for the recommended items, and we can reuse the most popular single user Movie Lens datasets that contain just ratings of individual users. The Movie Lens dataset contains 100,000 ratings, scaling from 0 to 5, derived from 943 users on 1,682 movies where each user has rated at least 20 movies.

It also contains demographic information about users such as age, gender and occupation. Based on this demographic data, demographic classes were defined and stored in D_{Set} which is one of the input for Group Formation Unit. User gender can take two values viz. male, female. Users were divided into three categories based on occupation. Service Class users were composed of programmer, engineer, health care, librarian, technician, scientist, administrator, executive and educator. Business class users included doctor, entertainment, lawyer, artist, marketing, salesman and writer. Students, retired persons, homemaker, none and others are categorized as miscellaneous users. Similarly, users were divided into three categories based on age. If age is between 10 and 30, user is considered to be young. If age is between 31 and 50, user is considered to be mature. If age is above 50, user is considered to belong to old category. Using this categorization, 18 demographic classes were formed as shown in Table 5. When a target user enters the system, then he/she is assigned to one of these attribute classes. For example, if a female user of age 45 years working as a lawyer enters the system, he is assigned demographic class D_5.

We have used MATLAB 7.0 [42] for our experiments. We have randomly selected 80% of the entire set to constitute the training set and the remaining to constitute the test set. The ratings in the test set are used to test the accuracy of the predictions based upon data in the training set.

Table 5. Demographic Attribute Classes.

D_{Set}	Attribute Class Value
D_1	<male, young, service>
D_2	<male, young, business>
D_3	<male, young, miscellaneous>
D_4	<female, young, service>
D_5	<female, young, business>
D_6	<female, young, miscellaneous>
D_7	<male, mature, service>
D_8	<male, mature, business>
D_9	<male, mature, miscellaneous>
D_{10}	<female, mature, service>
D_{11}	<female, mature, business>
D_{12}	<female, mature, miscellaneous>
D_{13}	<male, old, service>
D_{14}	<male, old, business>
D_{15}	<male, old, miscellaneous>
D_{16}	<female, old, service>
D_{17}	<female, old, business>
D_{18}	<female, old, miscellaneous>

B. Measure Metric

To access prediction quality, Statistical accuracy metric was studied. It measures how close are the numerical values which are generated by the group recommender is to the actual numerical ratings as provided by the user. Keeping this into account, we use Mean Absolute Error (MAE) [43] which measures the average absolute deviation between a recommender system's predicted rating and a true rating assigned by the user. It is defined in (10).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - a_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \qquad (10)$$

It is the sum of the absolute differences between each prediction p_i and corresponding rating a_i divided by the number of ratings n. It is an average of absolute error e_i.

C. Results

The authors first generated synthetic groups (based on combined inter user similarity), then generated group recommendations for the new user using these groups, and finally evaluated these group recommendations. The entire procedure was performed for every user in the test set, and computed an average MAE across all users. Computing average MAE in this way counts all the users equally, rather than biasing the result towards users with more ratings.

The authors aggregated group recommendations using balanced inclination strategy and average strategy. They found average MAE in both the cases. Figure 4 depicts the average MAE@k groups, obtained by these aggregation strategies for k Є {10, 20, 100, and 300} where k represents number of groups. For both the strategies, it was noticed that as the number of groups increased, the quality of group recommendations improved. With the increase in value of k, the size of groups is decreased, thus the strategies predicted more precise ratings. The graph also represented that, for any value of k, the group recommendations generated using the proposed balanced inclination strategy obtained lower MAE@k than the group recommendations generated using average strategy.

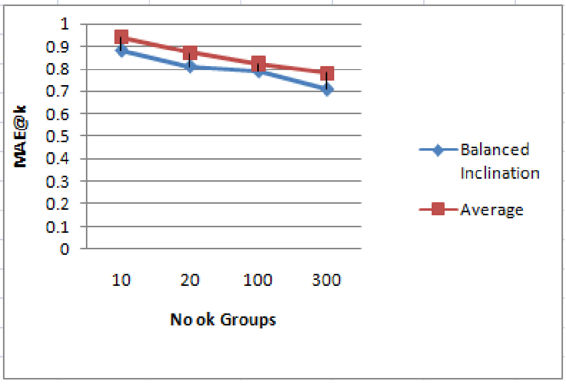

Figure 4. Average MAE of Balanced Inclination and Average Strategy

The authors further evaluated the impact of Top N size of group recommendations on the accuracy of group recommendations generated by both the approaches. The results are shown for k = 10. MAE@10 obtained in both the cases, are shown in table VI. It is clearly evident that as the recommendation size increased, the MAE@10 decreased in both the approaches. Also, from the last row of the table 6, it shows that the percentage of improvement in balanced inclination approach with respect to average strategy is higher at smaller values of Top N sizes. Therefore, in situations where number of group recommendation size is small, the proposed strategy is more appropriate. For example, in case of satellite systems, due to limited bandwidth and number of channels, the size of Top N group recommendation is limited to a smaller number.

Table 6. MAE at Top N Recommendation Sizes.

Strategy	Recommendation Size		
	Top 20	Top 10	Top 5
Balanced Inclination	0.88	0.86	0.82
Average	0.94	0.93	0.91
Percentage of Improvement	6%	7%	9%

V. Conclusion

This paper proposes a modified version of OCRG, to predict the group recommendations for a new user based on his demographic attributes. It aggregates the preferences of group members using balanced inclination group aggregation strategy. Using Movie Lens dataset, group recommendations obtained by balanced inclination had lower MAE as compared to group recommendations obtained by average group aggregation strategy.

In literature, the group recommendation algorithms have usually concentrated on how to model the already existing groups, in order to produce group recommendations that maximize user satisfaction. The preliminary phase of the group recommendation process is to compute a proper identification of similar groups. In the proposed approach, the group identification is based on support of each group member on demographic class of the new user. In future, the authors intend to identify a subset of group members from the identified group, with the aim of suggesting the most suitable group recommendations for a new user. Multi criteria ratings of group members of the identified group will be studied for selection of subset of group members.

References

[1] G. Adomavicius, and A. Tuzhilin, “Toward the next generation of recommender system; a survey of the state of the art and possible extensions”, IEEE transaction on knowledge and data engineering, June 2005, Vol 17, No. 6.

[2] S. Berkovsky, T. Kuik and F. Ricci., “Distributed collaborative filtering with domain specialization”, In Conf. on Recommender Systems, 2007.

[3] Konstan J. A., Miller B. N., Mattz D., Herlock J. L., Gorden L. R. and Riedl J., “GroupLens: Applying collaborative filtering to usenet news,” Communications of the ACM, Vol 40(3), 1997, pp. 77-87.

[4] J. F. McCarthy and T. D. Anagnost. Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In Int. Conf. on Computer Supported Collaborative Work, 1998.

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

[6] K. McCarthy, L. McGinty, and B. Smyth. Case-based group recommendation: Compromising for success. In Int. Conf. on Case Based Reasoning, 2007.

[7] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: a recommender system for groups of users. In European Conf. on Computer Supported Cooperative Work, 2001.

[8] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. User Modeling and User-Adapted Interaction, 16(1):63-82, 2006.

[9] Jameson, A, Smyth, B.:, “Recommendation to groups”, In The Adaptive Web, pp 596 – 627 , 2007.

[10] Amer-Yahia, S., Roy, S.B., Chawla, A., Das, G., Yu, C.: Group recommendation: Semantics and efficiency. PVLDB 2(1), 754-765, 2009.

[11] Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: RecSys. pp. 119-126, 2010

[12] J. Masthoff, “Group Modelling: Selecting a sequence of television items to suit a group of viewers, in User Modelling and User Adapted Interaction, 14(1): 37-85, 2004.

[13] J. Masthoff. Group recommender systems: Combining individual models. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, Recommender Systems Handbook, pages 677–702. Springer US, 2011.

[14] Harita Mehta, Punam Bedi and V. S. Dixit, “OCRG: A proposed Recommender for Mitigating New User Problem”, Information and Communication Technologies (WICT), 2012, pp 515-519.

[15] Hill W., Stead L., Rosenstein M. and Furnas, “Recommending and evaluating choices in a virtual community of use,” In Proceedings of Computer Human Interaction, 1995.

[16] Goldberg K., Roeder T., Gupta D. and Perkins C., “Eigentaste: A constant time collaborative filtering algorithm,” Information Retrieval Journal, Vol 4(2), July 2001, pp. 133-151.

[17] Terveen L., Hill W., Amento B., McDonald D. and Creter J. , “PHOAKS: A system for sharing recommendations,” Communications of the ACM, vol. 40(3), 1997, pp. 59-62.

[18] Shardanand U. and Maes P., “Social Information Filtering: algorithms for automated "Word of Mouth". In Proceedings of Human factors in computing systems, ACM, New York, 1995, pp. 210-217.

[19] Zeng W. and Shang M. S., “Effects of negative ratings on personalized recommendation,” 5th International Conference on Computer Science & Education Hefei, China, 2010, pp. 24–27.

[20] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada, “Enhancing group recommendations by incorporating social relationship interactions”, in proceddings of the 16th ACM International Conference on Supporting group work, Group ’10, pages 97-106, New York, NY, USA, 2010, ACM.

[21] Y.L. Chen, L.C. Cheng, and C.-N. Chuang. A group recommendation system with consideration of interactions among group members. Expert Syst. Appl., 34:2082–2090, April 2008.

[22] J. A. Recio Garcia, G. Jimenez-Diaz, A. A. Sanchez-Ruiz, and B. Diaz-Agudo. Personality aware recommendations to groups. In Proceedings of the third ACM conference on Recommender systems, RecSys ’09, pages 325–328, New York, NY, USA, 2009. ACM.

[23] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, The Adaptive Web, Methods and Strategies of Web Personalization, Lecture Notes in Computer Science 4321, pages 291-324. Springer, 2007.

[24] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In Fourth ACM International Conference on Web Search and Data Mining, 2011.

[25] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In Proceedings of the 2002 International Conference on Intelligent User Interfaces, pages 127-134, 2002.

[26] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. Comms. of the Association of Computing Machinery, 40(3):66-72, 1997.

[27] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In Procs. of the Eighteenth National Conference on Arti_cial Intelligence (AAAI), pages 187-192, 2002.

[28] Krulwich, B,”Lifestyle Finder: Intelligent user profiling using large-scale demographic data,” AI Magazine Vol 18 (2),1997, pp. 37-45.

[29] Weber, I. and Castillo, C, “The demographics of web search,” SIGIR'10. ACM , 2010.

[30] Dai Y. and Gong S., “Personalized recommendation algorithm using user demography information,” WKDD2009, IEEE Computer Society Press, 2009.

[31] Pazzani, M. J, “A framework for collaborative, content-based and demographic filtering,” AI Review Vol 13 (5/6), 1999, pp. 393-408.

[32] Stern D. H. , Herbrich R., and Graepel T., “Matchbox: large scale online bayesian recommendations,” in

proceedings of the 18th international conference onWorld wide web, 2009.

[33] Pennock, D., and Horvitz, E., “Collaborative filtering by personality

diagnosis: A Hybrid Memory- and Modelbased Approach,” Proceedings of UAI , 2000, pp. 473-480.

[34] Ha, V., and Haddawy P, “ Towards case-based preference elicitation: similarity measures on preference structures,” Proceedings UAI 1998, pp. 193-201.

[35] Rashid A., Albert I., Cosley D., Lam S. K., Mcnee S. M., Konstan J. A. and Riedl J., “Getting to Know You: Learning new user preferences in recommender systems,” in Proceedings of the International Conference on Intelligent User Interfaces, 2002.

[36] Chao, D., L., Balthrop, J. & Forrest, S., “Adaptive Radio: Achieving consensus using negative preferences,” In Proc. of the 2005 International ACM SIGGROUP Conference on Supporting Group Work (Sanibel Island, Florida, USA), 2005.

[37] Harita Mehta, Shveta Kundra Bhatia, Punam Bedi and V. S. Dixit, “Collaborative Personalized Web Recommender System using Entropy based Similarity Measure”, International Journal of Computer Science Issues, Vol 8, Issue 6, Nov 2011.

[38] Harita Mehta, V. S. Dixit and Punam Bedi, “Refinement of Recommendations based on User Preferences”, Intelligent System Design and Applications (ISDA), 2012, pp 515-520.

[39] Piao, C-H., Zhao, J. and Zheng, L-J, “Research on entropy based collaborative filtering algorithm and personalized recommendation in e- commerce”, SOCA, Springer, Vol 3, 2009 , pp. 147-157.

[40] Shannon, and E. Claude, “Prediction and Entropy of printed English”, The Bell System Technical Journal, Vol 30, 1951, pp 50-64.

[41] Silviu Guia, “Unweighted Entropy”, in Reports on Mathematical Physics, Vol. 2, No. 3, 1971.

[42] Matlab http://www.mathworks.com

[43] Willmott C., Matsuura K., “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in accessing average model performance,” Climate Research, Vol 30, Dec 2005, pp. 79-82.

Author Biographies

Harita Mehta is an Assistant Professor in the Computer Science Department of Acharya Narender Dev College, University of Delhi. Her research area is web recommender systems and is currently pursuing her PhD under Dr. V.S. Dixit.

Punam Bedi received her PhD in Computer Science from the Department of Computer Science, University of Delhi, India in 1999 and M. Tech in Computer Science from IIT Delhi, India in 1986. She is an Associate Professor in the Department of Computer Science, University of Delhi. She has about 25 years of teaching and research experience and has published more than 110 research papers in national/international journals/conferences. She is a member of AAAI, ACM, senior member of IEEE, and life member of Computer Society of India. Her research interests include web intelligence, soft computing, semantic web, multi-agent systems, intelligent information systems, intelligent software engineering, software security, intelligent user interfaces, requirement engineering, human computer interaction (HCI), trust, information retrieval, personalisation, steganography and steganalysis.

V.S. Dixit is a Senior Assistant Professor in the Department of Computer Science, Atma Ram Sanatam Dharam College, University of Delhi. His research area is queuing theory, peer to peer systems, web usage mining and web recommender systems. He is currently engaged in supervising research scholars for PhD. He is life member of IETE.