International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 6 (2014) pp. 515-527

Refinement of Group Recommendations Using User Preferences and Item Attributes

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

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

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

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

***Abstract*: Providing Group Recommendations is an open research area. In the proposed scheme, the combined entropy based similarities using positive and negative preference ratings among training users are used to extract Similar Taste Users (STUs). Such STUs build a group for the target user from which group recommendations are generated. Using information gain, it further computes top N individual recommendations from these group recommendations based on opposite user preferences.**

In this paper, a method is proposed to overcome the sparsity among preferences of group members. Genre based similarity (based on implicit multi criteria information) among target user and each group member generates genre based profile of target user which in turn increases the density of preferences among group members. Movie Lens dataset is used for experiments. It shows significant improvements in overcoming sparsity problem in group recommender systems and performance measures used shows improvement in recommendation quality.

Keywords: Group Recommendation, Sparsity Problem, Genre Based Similarity, Entropy, Information Gain, User Rating Preferences.

I. Introduction

A. Background

Recommender systems [1] are widely used in helping people to make decisions in exploring their interest in various information domains like books, movies, web pages, etc. In fact, the items recommended by the recommender systems are mostly used by the groups rather than by individuals, for example recommending a movie or a television show to a target user who may watch the recommended movie with family or with work colleagues, or friends. A Group Recommender System is a recommender system aimed at generating a set of recommendations that will satisfy all the members of the group. For example, music selection in public places [2], tourist attractions [3], holiday destinations [4], movies [5], and TV programs [6], are few examples of group recommendations.

One of a well known technique adopted in today's group recommendation system is Collaborative Filtering [7, 8, and 9] which suggests items, based on aggregated ratings of group members for all available items. Actually, it recommends to a target user those items which the target user has not rated as yet and has been liked by most of his/her group members.

If two users have similar rating patterns then they are referred as Similar Taste Users 'STUs'. Such STUs can be combined to form a group. The quality of group recommendations depends on selection of STUs within groups.

When a movie is liked by two users, it reflects positive preference of these two users for the movie; whereas if both users dislike a movie, it reflects their negative preference for the movie. In both the cases, the two users reflect similar taste. Actually, positive and negative preferences together govern the degree of similar taste between two users with respect to a set of movies. In this paper, combinations of these positive and negative preferences are used in order to find STUs which forms a group. A target user is assigned that group in which maximum number of group members is demographically similar to the target user. For example, if target is a male, then that group which has maximum males will be assigned as most promising group to this target user.

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 [10]. 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 [6]. The second strategy aggregates the recommendations of all users in the group into a single recommendation list [11, 12]. In the proposed approach, individual user profiles (preferences) have been aggregated rather than individual recommendations.

One of the aggregation strategy [13, 14] is to average the preferences of individual users in order to obtain Top N group recommendations for the target user. In this paper, local popularity of items within the group is used to aggregate the preferences of group members.

Further, the generation of individual recommendations from group recommendations is possible. The authors intend to filter individual recommendations from group recommendations without user intervention. In contrast to similar taste, if a target user likes a movie and any of his group member dislikes this movie or vice versa, then these opposite preferences makes up dissimilar taste. The generation of individual recommendations is possible by analyzing dissimilarity of preferences among target user and the group members. In this paper, the authors exploit these opposite preferences between a target user and the group members, which act as a filter on group recommendations to generate individual recommendations. In this paper, group formation and individual recommendation are obtained using Information Entropy and Information Gain respectively.

B. *Problem Statement*

In the group recommendation algorithm, the rating of group members of demographically similar group G_T directly affects the quality of predicted recommendations. In other words, the effectiveness of Group Recommender Systems highly depends on the profile completeness (ratings) of group members within the group. The higher is the number of not null entries in the Group Rating Table, the better are the profiles of group members within the group.

On a platform like World Wide Web, there might be millions of items that these group members can access. But, they can't experience even 1% of them. For instance, Amazon.com has several millions of items and 1% of them are more than 10,000 items which these group members can hardly access [15]. Thus, group Rating Table may have many missing entries and end up with incomplete group information. The low ratio of rated items to the total of available items in the group Rating Table is referred as sparsity problem. The Sparsity Level of a group is expressed in (1).

$$\mathit{Sparsity}\mathrm{Level} = 1 - \frac{|\,\mathrm{Not\,Null Entries in\,Group Rating Table}\,|}{|\,\mathrm{Entries\,in\,Group Rating Table}\,|} \quad (1)$$

If the data is too sparse in group rating table, it is possible that good quality group recommendations cannot be predicted because the aggregated group ratings will not depict the true picture of group tastes. This is one of the main limitations in group recommendation. In such a scenario, despite the fact that, group members (STUs) have similar tastes among them on few rated items, but still no quality recommendations can be predicted for the target user. In order to generate group recommendations from such Sparse Groups, the sparsity of the group should be reduced.

Although generation of group recommendations has drawn the attention of academics and practitioners since long, methods and techniques for alleviating sparsity problem remain an open problem for the researchers. Early typical user profile building approaches, although successful at the academic level, rely upon implicitly or explicitly acquired rating data denoting users' interest but are rarely concerned with demographic genre attributes of items being rated by the group members. In case of movies, genre attribute can be action, romantic, children, horror etc. These demographic genre attributes of items can play an important role in reduction of sparsity.

In this paper, the proposed technique will focus on reduction of sparsity problem of identified group thereby extending rating profiles of group members. The proposed technique demonstrates how item genre attributes can be incorporated in the demographically similar group to alleviate sparsity among group members.

In this paper, the major contributions are as follows: 1) partitioning of STUs into groups using Entropy, 2) reduction of sparsity in demographically similar groups based on genre attributes of items and 3) generation of individual recommendations from group recommendations using Information Gain. We have also evaluated the efficiency of our approach using a Movie Lens dataset based on movie ratings.

The remainder of the paper is organized as follows. Section 2 highlights the research related to sparse ratings and group recommender systems. Section 3 details the proposed approach to generate group recommendations with emphasis on reduction of sparsity of the group based on demographic genre attributes of items in the group. The proposed group recommender uses Information Gain to filter individual recommendations from group recommendations for a target user. Section 4 highlights the testing procedure and discusses the results based on Movie Lens data set. Section 5 concludes the paper and provides an overview of future work.

II. Related Work

A. *User Similarity based on User Preferences*

Collaborative filtering based on rating data is a widely used recommendation technique in many fields, such as news, Movie, book, CD, video, joke, etc [15]. A survey on collaborative recommender can be found in [1]. Goldberg used collaborative filtering to build a system called Tapestry for filtering emails [16]. Group Lens [17] first introduced an automated collaborative filtering system using a neighborhood based algorithms. Sarwar proposed a new CF algorithm based on similarity of items instead of neighbors [18]. Its accuracy was better than neighbor based CF. All these systems generate recommendations in which, the items to be recommended come from other people with similar taste [19].

User profiling is usually based on positive preferences (rating pattern between users). The resulting profiles are usually incomplete due to the impatience or forgetfulness of the users. Negative preferences could potentially be applied to information filtering and collaborative recommender systems. There has been a range of research which uses negative preferences. For instance, the Adaptive Radio [20] is one example of work that explored the value of explicitly modeling negative preferences for group recommendation. They used negative preference profiling to determine which solutions are unsatisfactory to individual users and assumed that remaining solutions are satisfactory.

Many Recommender Systems have explored a range of combination of positive and negative preferences [21, 22] to capture user's current interests and preferences for various items like movie, book, web page, etc. In [21], the authors observed that negative ratings were also as important as positive ratings in the recommender systems. They used a tunable parameter to obtain linear combination of these ratings. In [22], the authors observed that by taking harmonic mean of positive and negative preferences, success rate of reciprocal recommendations was increased in comparison to using the positive model of preferences alone.

Depicting similarity between two arbitrary users is an important task in recommender systems. Similarity can be computed by analyzing positive and negative ratings among a user pair for various movies. Measures such as correlation, cosine, adjusted cosine and entropy can be used to compute combined similarity among arbitrary users. Information Entropy is a better similarity measure than other similarity measures [23] because it gives low Means Absolute Error values as compared to any other method. It can be useful to reduce prediction error in collaborative recommendation systems. In [24], entropy based collaborative filtering algorithms provided better recommendation quality than user based algorithm and achieved recommendation accuracy comparable to the item based algorithm. So, the authors have used entropy based combined inter user similarity measure to generate group recommendations.

But these methods are not adapted to generate group recommendations for sparse data. Information Gain can be used to generate individual recommendations from group recommendations. It is derived from entropy and can be used as an attribute selection method. In text classification, it can be understood as the expected entropy reduction by knowing the importance of an attribute [25]. In this paper, the authors use Information Gain to generate highly liked attributes by the target user, in order to generate individual recommendations, without user intervention.

B. *Group Recommender Systems*

Group Recommender Systems [26] 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 [2], movies or TV programs [5, 6], tourism [3, 14], and others. Some well known Group Recommender Systems are MUSICFX [2], POLYLENS [5], INTRIGUE [3] and YU'S TV RECOMMENDER [6]. MUSICFX selects a radio station for background music in a fitness centre that suits the taste of people working out at a given time. POLYLENS recommends movies where an individual's tastes are inferred from ratings and 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.

Till date, group recommendations have been mostly generated using two approaches: aggregating individual preferences into group models or aggregating individual predictions into group predictions. INTRIGUE and POLYLENS aggregate recommendations, while MUSICFX and YU'S TV RECOMMENDER aggregate profiles. Masthoff [13, 14] 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. POLYLENS uses the Least Misery Strategy. INTRIGUE uses a weighted form of the Average strategy. MUSICFX uses a variant of the Average without Misery Strategy. YU'S TV RECOMMENDER also uses a variant of the Average Strategy.

In the proposed approach, group recommendations are generated by aggregating individual preferences. Average Strategy based on local popularity on items within the group is used to aggregate the preferences for items produced for each group member into a single group recommendations' list.

C. *Sparsity Problem*

Usually, web surfers are reluctant in specifying their preferences, explicitly or implicitly, for the items they purchase (online shopping), view (web pages) or rate (products like movies). Also in usual practice, web surfers purchase few common items from an endless list of items available online for shopping. Also, they visit and tend to rate entrant pages of websites more as compared to deeply seated web pages on the websites. Further, in case of rating movies, web surfers view and rate common movies more as compared to less popular movies. So, profile of web surfer has many unknown preferences leading to sparsity. The similarity computations are highly affected by these sparse profiles of two web surfers.

In practical application, rating data is very sparse [1], which leads to construction of group. The sparsity problem has been acknowledged as the most important drawback in collaborative filtering algorithms [27, 28, 29 and 30]. The group recommender inherits this problem too. The problem can be resulted for too few rating data which make it hard to aggregate the predicted ratings of each group member and make it impossible to determine good quality group recommendations. Actually, if user pair has positive or negative preferences for only few items, then the contribution of these ratings is marginal in combined inter user similarity. Providing effective recommendations in presence of sparsity among users is of fundamental importance to collaborative recommender systems.

The item-based collaborative filtering method was proposed to address sparsity problems, and it has proven to outperform the user-based method in recommendation qualities [8]. Previous Works based on overcoming sparsity problem include combining content based filtering and collaborative filtering to link contextual information among items [31, 32 and 33]. Clustering items or users to reduce dimensionality are also used to alleviate the sparsity problem [34]. And Papagelis et al. [35] applies trust inferences between users which refer to as social network to provide additional information that helps to deal with the sparsity problem. In [36], existing similarity measures were weighed by using information entropy. The results not only showed reduction in MAE but was also found to be robust for sparse dataset. The authors in [36] concluded that, if information entropy is applied to the formation of nearest neighborhood and predicting preference score, recommendation error will decrease.

By considering multi criteria information in a recommendation generation process, the quality of recommendations improves [37]. But, by explicitly asking users to provide their preferences on several criteria's (genre attributes) of available items may not be feasible. The genre attribute of items along with user preferences formulates a set of useful implicit dataset to support multi criteria information [38]. The authors in reference [38] showed that such implicit multi criteria dataset extended the capabilities of traditional single criteria recommender system in less time and effort.

In this paper, the proposed group recommender aims to increase density of preferences among members of the most promising group of the target user by using genre attribute of items along with user preferences. The purpose is to demonstrate how some basic demographic genre attributes of items when applied to group members can help in providing additional multi criteria information that helps to alleviate sparsity problems of group recommender systems.

III. Proposed Scheme

The architecture of the proposed Group Recommender System is shown in Figure 1. The target user provides demographic information such as age, gender and occupation during the registration process. After few clicks, the target user awaits for recommendations from the system. The preferences (composed of likes and dislikes) of the target user on these few clicks which define the target user's initial preference profile. The proposed framework not only generates Top N Group Recommendations for the target user, but also Individual Recommendations are generated for this target user. The main components of the proposed Group Recommender System are Offline Unit, Interface Unit, and Group Recommendation Unit.

Offline Unit creates knowledge base, which is used by Group Recommendation Unit to generate recommendations. The backbone of Offline Unit is the Group Formation Unit which generates various groups ({G_1, G_2… G_n }; stored in G_{Set}), where each group G_i consist of similar users. The selection of users in the same group is determined by entropy based similarity of positive and negative preferences between training users. A set of demographic classes (D_{Set}) is formed on all possible values of demographic attributes of a user. For example, if gender is the only attribute considered for a user, taking either value 'M' if user is male or 'F' if user is female; then two demographic classes are formed which are collectively stored in $D_{Set} = \{D_{Male}, D_{Female}\}$. Group Formation Unit finds support of each group G_i for all possible demographic classes and stores this information in the knowledge base along with G_{Set}.

Target User and Group Recommender Unit are two basic entities in any group recommendation generation process. Interface unit acts as an interface between these entities. It fetches the demographic attributes of the target user along with initial preference profile i.e. Click Stream Pattern from the current session of target user and sends the request to the Group Recommendation Unit, where Top N Group and Individual Recommendations are furnished. Finally, the Interface Unit displays the aggregated Top N Group Recommendations along with Top N Individual Recommendations for the target user during his/her current session.

Group Recommendation Unit consists of Group Identification Unit, Group Modification Unit, Group Aggregation Unit and Group Filter Unit. The Group Identification Unit identifies a particular group G_T (such that $G_T \in G_{Set}$) for the target user U_T based on his/ her corresponding demographic class D_T (such that $D_T \in D_{Set}$). The sparsity problem in Collaborative Filtering, leads to sparsity in the identified group G_T. For reflecting sparsity in G_T, it is referred as 'SparseG_T'.

Group Identification Unit passes the SparseG_T group to Group Modification Unit, which intends to reduce the sparsity in the SparseG_T group based on various genre classes of an item like Action centric, Romantic centric; Children centric, Horror centric (as discussed in section 4). It initially constructs genre attribute based profile of the target user ($GAPU_T$) based on initial user profile. It then constructs genre attribute based profile of each member ($GAPGM_x$) of the SparseG_T group. These profiles of group members are stored in Group Genre Attribute Profile Set ($GGAP_{Set}$). Using this set, genre based similarity, 'GenreSim' of each group member is obtained with the target user using correlation and stored in GSim. It then constructs complete genre attribute based profile of the target user ($CGAPU_T$) based on classic user based Nearest Neighbor Collaborative Filtering. This complete profile of the target user is used to fill in the missing entries in the original identified group (SparseG_T), in order to get a new group (DenseG_T).

In order to derive group recommendations for the target user, the Group Modification Unit passes the DenseG_T group to the Group Aggregation Unit which extracts unrated preferences (items) of the target user from DenseG_T group and stores these items in group Recommendation set (GR_{Set}). It uses local popularity of items in the DenseG_T group to aggregate the preferences of group members for these items. The average values reflect the weight of each recommendation in GR_{Set}. These weights are stored in the GR_{Set}. The GR_{Set} is sorted in decreasing order of weight. Finally, top N group recommendations are passed to the interface unit.

In order to derive individual group recommendations for the target user, the group modification unit passes $CGAPU_T$ to the Group Filter Unit along with DenseG_T. Group Aggregation Unit passes GR_{Set} to Group Filter Unit which wishes to identify those attribute values which differentiate most promising individual recommendations from the set of group recommendations stored in GR_{Set}. Using Information Gain, it exploits opposite preferences of target user with respect to preferences of members of DenseG_T group to produce highly informative attribute values, reflecting target user's hidden liking for these selected attribute values. Finally, in order to derive individual recommendations for the target user, Group Filter Unit obtains score of each recommendation in GR_{Set}. It then sorts these recommendations in decreasing order of score. These recommendations are stored in IR_{Set} from which top N individual recommendations are passed to the interface unit. The remainder sub sections, discusses the working of Group Formation Unit, Group Identification Unit, Group Modification Unit, Group Aggregation Unit and Group Filter Unit with a suitable example.

A. *Group Formation Unit*

Movie Lens uses users' ratings matrix '*M*' (1 to 5 likert scale) depicting users likes (rating greater than equal to 4 on likert scale) and dislikes (rating less than equal to 3 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 [39, 40]. 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

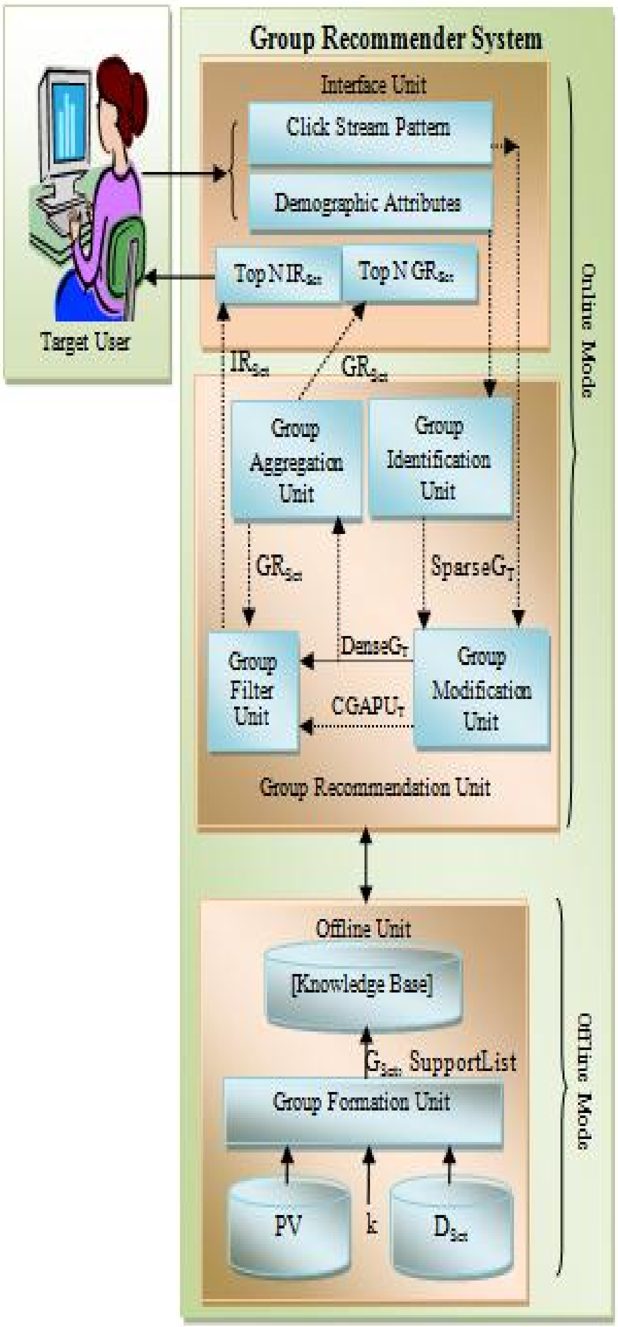

Figure 1. Proposed Framework

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 2. 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.

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

where,

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

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

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

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 (3a) and (3b).

$$P_k(U_t,U_x) = \begin{Bmatrix} 1 \text{ if } \mathrm{M(U_t,k) = 'L' \& M(U_x,k) = 'L'} \\ 0 \text{ Otherwise} \end{Bmatrix} \quad (3a)$$

$$N_k(U_t,U_x) = \begin{Bmatrix} 1 \text{ if } \mathrm{M(U_t,k) = 'D' \& M(U_x,k) = 'D'} \\ 0 \text{ Otherwise} \end{Bmatrix} \quad (3b)$$

The similarity between these two users rests on the similarity computation at both the levels, which decides the degree of similar taste between them. By taking into consideration, the similarity of negative preferences along with positive preferences throughout the session (i.e. at both the levels), in order to find members of a group, one lands up finding strongly related users in a group.

Defined simply as a set of users, a user group can be formed on a recurring basis, for e.g., friends who meet regularly for dinner are most likely to fall in same group. 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 4. 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.

$$Support\left(G_i, \mathrm{D}_c\right) = \frac{No \text{ of Users} \in \text{Class } \mathrm{D}_c}{No \text{ of Users in } G_i} \quad (4)$$

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 Є D_{Set}
SupportList(G_i,C)=Support(G_i, D_C)
Repeat for each Gi Є 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 among similar groups.

The following example illustrates step by step analysis of Group Identification Unit. Suppose the knowledge base 'KB' formed by Group Formation Unit is as shown in Table 1. The G_{Set} is made up of three groups {G_1, G_2, G_3} consisting of 5, 6 and 3 members respectively. For simplicity, we assume that there are only two demographic classes D_{Male} and D_{Female}, stored in D_{Set} instead of 18 Demographic classes as shown in section 4. The support of these three groups for the two demographic classes is found using (3) and stored in SupportList. If a target user U_T, a male, enters into the system, then his demographic class D_T is assigned as D_{Male}. From the supportList, stored in the KB, it is clear that Group G_1 has maximum support for D_{Male}. Thus, target user U_T is assigned group G_1 as identified sparse group $SparseG_T$. The ratings of group members of $SpaseG_T$ group are shown in Table 2.

Table 1. KB built by Group Formation Unit.

GSet	**Group Statistics**			**Support of Group**	
	Group Size	**No. of Males**	**No. of Females**	$\mathbf{D_{Male}}$	$\mathbf{D_{Female}}$
G_1	5	3	2	0.6	0.4
G_2	6	1	5	0.1	0.8
G_3	3	1	2	0.3	0.6

Table 2. Preferences of Group Members in $SparseG_T$

$\mathbf{SparseG_T}$	**Items**						
Group Members	$\mathbf{I_1}$	$\mathbf{I_2}$	$\mathbf{I_3}$	$\mathbf{I_4}$	$\mathbf{I_5}$	$\mathbf{I_6}$	$\mathbf{I_7}$
GM_1	3	-	4	5	-	1	2
GM_2	2	1	-	4	-	5	-
GM_3	-	3	-	2	1	-	-
GM_4	4	5	-	-	-	2	-
GM_5	5	-	2	-	-	-	1

C. Group Modification Unit

Group Identification Unit passes the $SparseG_T$ group to Group Modification Unit, which intends to reduce the sparsity in the $SparseG_T$ group. To reduce the sparsity in $SparseG_T$ group, genre based correlation (similarity) is found between target user and each member of $SparseG_T$ on the basis of genre classes viz Action centric, Romantic centric, Children centric and Horror centric. An item can have more than one genre attribute. Each item x can be represented as ($Item_x$ = $genre_{Action}$, $genre_{Romantic}$, $genre_{Children}$, $genre_{Horror}$) where $genre_i$ is 1 if $genre_i$ is present in $Item_x$. For example, if a movie is romantic and as well as action centric then I_x = (1100). In the above KB, the authors assume that, there are in total 7 items. The items with their genre information are stored in the I_{Set} as shown in Table 3.

Table 3. Items and their Genres

	Genre			
Items	**Action**	**Romantic**	**Documentary**	**Fantasy**
I_1	1	1	0	0
I_2	0	1	1	0
I_3	1	0	0	1
I_4	0	1	1	0
I_5	1	0	0	1
I_6	1	1	0	1
I_7	0	1	0	1

Group Modification unit find Genere Affinity for each genre attribute (GA_{Genere}) of target user and as well as each group member (GM_x) of $SparseG_T$ group using (5).

$$GA_{Genere} = \frac{\sum_{i \in ISet} Genere(i) * Rating(i)}{\text{total rated Items}} \quad (5)$$

It then constructs genre attribute based profile of target user ($GAPU_T$) and each group member (GM_x) of $SparseG_T$ group ($GAPGM_x$), collectively stored in ($GAPGM_{Set}$). After this, correlation based genre similarity, 'GenereSim', is found between target user and each group member. For example, if the target user in the same example rates three items ($Item_1$, $Item_3$ and $Item_5$) out of a total of 7 items in the KB, as shown in Click Stream Pattern 'CSP_T' (Table 4) reflecting initial user profile. The last line of the table 4 shows target user's Genere Attribute Profile ($GAPU_T$) which is obtained using (5). Similarly by using (5), genre attribute profile of each member GM_x of the $SparseG_T$ group is obtained in Group Genere Attribute Profile Set ($GGAP_{Set}$) as shown in Table 5. After this, correlation between the genre profile of the target user and each group member is obtained in Genere Similarity (GSim) as shown in Table 6.

Table 4. Genre Attribute Profile of Target User

CSP_T	Act.	Rom.	Doc.	Fan.	Rating
I_1	1	1	0	0	3
I_3	1	0	0	1	4
I_5	1	0	0	1	1
GA_{Genre}	2.6	1	0	1.6	$\leftarrow GAPU_T$

Table 5. Genre Attribute Profile of Group Members

	Genre			
Group Members	Action	Romantic	Doc.	Fantasy
GM_1	1.6	2.2	1	1.4
GM_2	1.7	3	1.2	1.2
GM_3	0.3	1.6	1.6	0.3
GM_4	2	3.6	1.6	0.6
GM_5	2.3	2.3	0	1

Table 6. Genre Attribute Similarity of Target User with Group Members

GSim	
$GSim(U_T,GM_1)$	0.6695
$GSim(U_T,GM_2)$	0.9604
$GSim(U_T,GM_3)$	1.847
$GSim(U_T,GM_4)$	1.093
$GSim(U_T,GM_5)$	0.287

Next, predicted rating $P(U_T, i)$ is calculated using a classic User based Nearest Neighbor Collaborative Filtering Approach, proposed in [41]. It predicts the rating for each item i that was not rated by the target user, considering the rating $r_{GMx,i}$ of each group member GM_x for the item i. Equation (6) gives the formula used to predict the ratings. Here, values $\overline{r_{U_T}}$ and $\overline{r_{GMx}}$ represent the mean of the ratings expressed by target user U_T and group member GM_x respectively.

$$\text{if } i \in CSP_T \text{ then } P(U_T,i) = r(U_T,i)$$
otherwise
$$P(U_T,i) = \overline{r_{U_T}} + \frac{\sum_{GM_x \in G_T} GSim(U_T,GM_x)(r_{GMx,i} - \overline{r_{GMx}})}{\sum_{GM_x \in G_T} GSim(U_T,GM_x)} \quad (6)$$

Using data from table 1, 2 and 6 in (6), rating values for unrated items $Item_2$, $Item_4$, $Item_6$ and $Item_7$ are obtained. These values convert the incomplete click Stream pattern (CSP_T) into complete genre attribute profile of the target user ($CGAPU_T$) as shown in Table 7.

Table 7. Complete Genre Attribute Profile of Target User

	Items						
Profile	I_1	I_2	I_3	I_4	I_5	I_6	I_7
CSP_T	3	-	4	-	1	-	-
$CGAPU_T$	3	2.33	4	2.109	1	1.445	0.3

Using these rating values from CGAPUT, the missing values in $SparseG_T$ are filled up using (7) and stored into $DenseG_T$ group as shown in Table 8.

$$\begin{aligned}
&\text{if } SparseG_T(GM_x,i) \text{ is not Null then}\\
&\qquad DenseG_T(GM_x,i) = SparseG_T(GM_x,i)\\
&\text{else if } SparseG_T(GM_x,i) \text{ is Null then}\\
&\qquad DenseG_T(GM_x,i) = CGAPU_T(GM_x,i)
\end{aligned} \quad (7)$$

Table 8. Preferences of Group Members in $SparseG_T$

$DenseG_T$	Items						
Group Members	I_1	I_2	I_3	I_4	I_5	I_6	I_7
GM_1	3	2.33	4	5	1	1	2
GM_2	2	1	4	4	1	5	0.3
GM_3	3	3	4	2	1	1.445	0.3
GM_4	4	5	4	2.109	1	2	0.3
GM_5	5	2.33	2	2.109	1	1.445	1

D. Group Aggregation Unit

In order to generate group recommendations for target user U_T, Group Aggregation Unit uses the positive and negative ratings of existing items given by all the users in $DenseG_T$ group having maximum support for the demographic class D_T. It is obtained from Group Identification Unit.

Popularity of a movie indicates how frequently users have rated that movie. It is defined in (8).

$$P_i = \frac{\sum_{u=1}^{n} r_{u,i}}{N_i} \quad (8)$$

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

Local Popularity of item is calculated with respect to ratings given by all those users who belong to $DenseG_T$ group having maximum support for D_T and is defined in (9).

$$LP_{i,DenseG_T} = \frac{\sum_{u=1}^{n} r_{u,DenseG_T,i}}{N_i} \quad (9)$$

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

Let us refer the same example once again. Group recommendation set (GR_{Set}) consists of those items of DenseGT group which have not been rated by the target user i.e. $Item_2$, $Item_4$, $Item_6$ and $Item_7$. Local popularity of these items is obtained using (9), which reflects the weight of each recommendation in GR_{Set}. The items with higher local popularity are considered to be most promising group recommendations. Top 3 group recommendations in this example are $Item_4$, $Item_2$ and $Item_6$ which is depicted in the last row of table 9. These group recommendations are passed to the interface unit.

Table 9. Group Recommendation Set (GR_{SET})

Items	**I_2**	**I_4**	**I_6**	**I_7**
Local Popularity	2.732	3.043	2.178	0.78
Sorted Items	{I_4, I_2, I_6, I_7}			
Top 3 G. Rec.	{I_4, I_2, I_6 }			

E. Group Filteration Unit

Information Filter takes Group Recommendation Set (***GR_{set}***) which is obtained by Group Aggregation Unit. It exploits mysterious preferences between complete genre attribute based profile of target user ($CGAPU_T$) and preferences of group members GM_x stored in $DenseG_T$ group. It attempts to select recommendations from ***GR_{set}*** using Information Gain. Dissimilar taste depicting opposite user preferences between target user and group members can be found by using (10).

$$O_k(U_T, GM_x) = \begin{cases} 1 \text{ if } M(U_T,k)='L'\&M(GM_x,k)='D' \\ \text{or viceversa} \\ 0 \quad\quad\quad \text{Otherwise} \end{cases} \quad (10)$$

where, $O_k(U_T, GM_x)$ corresponds to binary '1' if both users reflect opposite preference for movie 'k'. The set of such 'k' movies is stored in Opposite Preference Taste Set (***OPT_{set}***).

OPT_{set} gives no direct indication whether a group recommendation stored in ***GR_{set}*** will turn out to be successful individual recommendation or not. But, the genre attribute values attached to the movies of ***OPT_{set}*** can reveal some hidden information that may contribute to improve the quality of individual recommendations in ***GR_{set}***.

Information Gain can be used to assign weight (gain) to genre attributes with respect to a set of movies in the ***GR_{set}***. The amount of information provided by a genre attribute GA is expressed as "***Gain (OPT_{set}, GA)***". It is calculated for each genre attribute. The genre attributes with high gain values provide more reliable information as compared to low gain genre attributes. ***Gain (OPT_{set}, GA)*** allows refreshing ***GR_{set}*** by retaining only those recommendations that belong to the subset of highly informative attribute values. Such recommendations serve as individual recommendations for the target user.

Let c={$Opp^{like}_{dislike}$, $Opp^{dislike}_{like}$ } be the set of two possible classes in ***OPT_{set}***. $Opp^{like}_{dislike}$ are those movies which are liked by the target and disliked by any of his group member GM_x. $Opp^{dislike}_{like}$ are those movies which are disliked by the target and liked by any of his group member GM_x. The authors are interested in fetching those attributes which give a clear distinction among these classes.

The expected information, "***Info(OPT_{set})***" needed to classify a movie of ***OPT_{set}*** into these two predefined classes (in general represented by variable 'n') is the entropy of the entire ***OPT_{set}*** as shown in (11).

$$Info(OPT_{set}) = -\sum_{c=1}^{n} p_c \log_c(p_c) \quad (11)$$

where, p_c is the probability of ***OPT_{set}*** belonging to class c. If a genre attribute GA has v distinct values, then average amount of information contributed by a genre attribute GA in class c is expressed as "***$Info_{GA}$(OPT_{set})***" as shown in (12).

$$Info_{GA}(OPT_{set}) = \sum_{v \in values(GA)} p_v \left[-\sum_{c=1}^{n} p_{c,v} \log_2 p_{c,v} \right] \quad (12)$$

where, p_v is the probability of ***OPT_{set}*** having genre attribute GA with the value v. $p_{c,v}$ is the probability of ***OPT_{set}*** having genre attribute GA with value v and which is in category c. ***Gain (OPT_{set}, GA)*** is the expected reduction in entropy caused by partitioning the movies of ***OPT_{set}*** according to the genre attribute GA as shown in (13).

$$Gain\ (OPT_{set}, GA) = Info(OPT_{set}) - Info_{GA}(OPT_{set}) \quad (13)$$

In order to obtain usefulness of group recommendations in ***GR_{set}***, authors calculate the score of each group recommendation as expressed in (14).

$$Score(\text{GRec}) = \left[\sum_{j=1}^{GA} Gain(OPT_{set}, GA_j) * f(\text{GRec}, GA_j) \right] * wt(\text{GRec}) \quad (14)$$

where, *Gain(OPT_{set}, GA_j)* is the gain value of j^{th} genre attribute, j are the total number of genre attributes of a movie and

$$f(\text{GRec}, GA_j) = \begin{cases} 1 \text{ if attibute value } GA_j \text{ occurs in GRec} \\ -1 \text{ otherwise} \end{cases}$$

Group recommendations are arranged in descending order of Score and are stored in individual recommendation set (IR_{Set}). Finally, top N recommendations are given to the target user. The main purpose of Information Filter is that rather than attempting to recommend the items that user want to watch, the system avoids recommending movies having genre attributes that they don't want to watch. The advantage of information filer is that the group recommendations are refined to generate individual recommendations without user intervention.

Continuing with the above example, the items in $CGAPU_T$ and $DenseG_T$ group can be labeled 'like' or 'dislike' as shown in Table 10. Using (10), OPT_{Set} generated consist of {$Item_1$, $Item_2$, $Item_3$, $Item_4$, $Item_6$}. Considering the two classes as mentioned earlier, $Item_1$ and $Item_3$ are assigned to $Opp^{like}_{dislike}$ class whereas $Item_4$, $Item_6$ and $Item_2$ are assigned to $Opp^{dislike}_{like}$ class as shown in Table 10.

Table 10. Opposite Preference Taste Set (OPT_{SET})

User	Items						
User	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$CGAPU_T$	L	D	L	D	D	D	D
GM_1	L	D	L	L	D	D	D
GM_2	D	D	L	L	D	L	D
GM_3	L	L	L	D	D	D	D
GM_4	L	L	L	D	D	D	D
GM_5	L	D	D	D	D	D	D
OPT_{Set}	Y	Y	Y	Y	N	Y	N
$Opp^{like}_{dislike}$	Y	-	Y	-	-	-	-
$Opp^{dislike}_{like}$	-	Y	-	Y	-	Y	-

Using (11), (12) and (13), information gain of each genre attribute is obtained as shown in Table 11, followed by score calculation using (14) as shown in table 12. These values are stored in Individual Recommendation Set (IR_{Set}). All the items are arranged in decreasing order of score. The items with higher score are considered to be the most promising individual recommendations. Top 3 individual recommendations in the example are $Item_7$, $Item_6$ and $Item_4$ as shown in last row of Table 12.

Table 11. Information Gain

Genre	Action	Romantic	Doc.	Fantasy
Gain	0.418	0.3212	0.418	0.97

Table 12. Individual Recommendation Set (IR_{SET})

Items	I_2	I_4	I_6	I_7
Score	-1.772	-1.974	-1.413	0.355
Sorted Items	{I_7, I_6, I_4, I_2}			
Top 3 Ind. Rec.	{I_7, I_6, $I4_6$ }			

From Table IV, by observing the GA_{Genere} values stored in $GAPU_T$, any item that have documentary genre attribute should be filtered or pushed at last in the IR_{Set} as compared to GR_{Set}. For example, $Item_4$ and $Item_2$ having documentary genre attributes are pushed at the end in the IR_{Set}. 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}
I_{Set} = {Genere Attributes of Items in KB}
CSP_T = {Click Stream Pattern of Target User 'U_T'}

Process:
Step 1: Group Identification
$SparseG_T$= max(Support(G_i, D_T) where, $G_i \in G_{Set}$
Step 2: Group Modification
Evaluate $GAPU_T$
For each group member GM_x in $SparseG_T$
$GGAP_{Set}(GM_x)$ = $GAPGM_x$
$GSim(U_T, GM_x)$ = $GenreSim(GAPU_T, GAP_{Set}(GM_x))$
End For
Evaluate $CGAPU_T$ and $DesneG_T$
Step 3: Group Aggregation
For each movie 'k' in $DenseG_T$
If k not in CSP_T then insert k into GR_{Set}
End For
Do
Weight (i) = $LP_{i,DenseGT}$
Store Weight (i) in GR_{Set}
Repeat for each item 'i' in GR_{Set}
Pass Top N recommendations from GR_{Set}
(in decreasing order of Weight to Interface Unit)
Step 4: Group Filtration
For each Group Member GM_x in $DenseG_T$
For each movie 'k' rated by GM_x
If $O_k(U_T, GM_x)$ =1then
insert k and its attribute values into IGList
End If
End For
End For
OPT_{Set}= Unique(InfoGainList)
For each Genere Attribute 'GA'
Evaluate $Gain(OPT_{Set}, GA)$
End For
Highly Selected Attributes = {Top N attributes}
(in decreasing order of Gain)
Do
$IR_{Set}(i)$= Score(i)
Repeat for each item 'i' in GR_{Set}
Pass Top N recommendations from IR_{Set}
(in decreasing order of Score to Interface Unit)
Output:
Top N Group and Individual Recommendations for Target User 'U_T'

Figure 3. Algorithm for Group Recommendation Unit

IV. Experiment

A. Dataset

In this paper, authors performed off-line evaluations where, groups are sampled from the users of a traditional (i.e., single user) recommender system. 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 individually tailored recommendations. 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 authors carried out experiments using the Movie Lens dataset, taken from a research recommendation site being maintained by the Group Lens project. It contains 100,000 ratings, scaling from 0 to 5, derived from 943 users on 1682 movies where each user has rated at least 20 movies. MATLAB 7.0 [42] was used for the experiments. The authors 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. In the proposed group recommender system, the active user's preferences are denoted by rating data (1 to 5 likert scale in Movie Lens database). When using the available ratings, the recommendation algorithm transfer rating data into user item taste matrix. t_{ij} is assigned literal 'L' if user i has given higher rating (no smaller than 4 in 1 to 5 rating structure) to item j and t_{ij} is assigned literal 'D' otherwise.

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 inputs 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 young. If age is between 31 and 50, user is mature. If age is above 50, user is old. Using this categorization, 18 demographic classes were formed as shown in Table 13. When a target user enters the system, then he/she is assigned to one of these attribute classes. For example, if a male user of age 45 years working as a lawyer enters the system, he is assigned demographic class D_2.

Also, when a target user enters in the system, he is allowed to specify few ratings. These ratings are stored in 'CSP_T' which depict his interest's thereby building partial profile of the target user. Experiments were conducted with novice users (whose profile was made of three ratings only) and veteran users (whose profile was made of six ratings).

Table 13. 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, two different kinds of metrics were employed. Statistical accuracy metric 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 (15).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - a_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \qquad (15)$$

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 .

Classification accuracy measure determines the success of a prediction algorithm in correctly classifying items. Precision is the number of relevant items a search retrieves divided by the total number of items retrieved, while recall is the number of relevant items retrieved divided by the total number of existing relevant items that should have been retrieved. Suppose, the set of items that are viewed by the target user are Relevant Items and those items that are recommended by the recommender are Retrieved Items, then Precision Ratio and Recall Ratio are obtained using Eq. (16) and (17) respectively. Usually, Precision and Recall scores are not evaluated in isolation. Instead, they are combined into a single measure, such as the F1 Measure, which is often used in the field of information retrieval for measuring search performance. It is the weighted harmonic mean of Precision and Recall. It measures the effectiveness of retrieval with respect to a target user who attaches equal importance to both of them and is depicted by Eq. (18).

$$F1\ Measure = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \qquad (18)$$

C. Results

The authors first generated synthetic groups (based on combined inter user similarity), then generated group recommendations for the target 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, Precision, Recall and F1 Measure across all users. Computing average MAE, Precision, Recall and F1 Measure in this way counts all the users equally, rather than biasing the result towards users with more ratings. Finally, based on opposite preferences, using information gain, individual recommendations from these group recommendations were filtered.

During formation of groups, k was set to 75. Group size is defined as number of members in a group. These 75 groups varied in size ranging from 1 to 10. Based on demographic data of test users, each test user was assigned most promising demographically similar group 'SparseG_T' by the group identification unit. The quality of group recommendations is affected by the group size. These test users were divided into two classes' viz. small groups and large groups. Small groups consisted of those test users whose group size ranged from 1 to 5 whereas large groups consisted of those test users whose group size ranged from 6 to 10. Group recommendations were generated for each test user in small and large groups by the group aggregation unit, giving results for Top N recommendations where N= {10, 20, 30}. Precision@N, Recall@N and F1Measure@N was measured for both the groups. Table XIV shows the improvement (in percentage), when N was increased.

From the table 14, as Top N size increased, in both the groups, Precision@N increased almost two fold whereas Recall@N decreased very little. Improvement in F1Measure@N is almost two fold. It is evident from the results that improvement is higher in small groups (5.8% - 2.1% = 3.7%) as compared to larger groups (6.1% - 3.1% = 3%), when recommendation size N increased. The quality of group recommendations is better, when the group size is small. Thus, for subsequent experiments, small groups were preferred.

Table 14. Improvement in Performance Measures

Performance	Small Groups		Large Groups	
	Increment in Top N			
	10 to 20	10 to 30	10 to 20	10 to 30
Prec@ N	2.1% (↑)	5% (↑)	2.3% (↑)	4.4% (↑)
Rec @ N	4.5% (↓)	4.5% (↓)	1.1% (↓)	1.5% (↓)
F1 @ N	2.1% (↑)	5.8% (↑)	3.1% (↑)	6.1% (↑)

From small groups, only those test users were chosen whose group size was 5. The group modification unit generated $CGAPU_T$ and DenseG_T for these test users whose click stream pattern consisted of 3 clicks. Group aggregation unit generated group recommendations using SparseG_T and as well as DenseG_T. MAE was obtained for each test user. Figure 4 shows the MAE values for these test users. MAE of Group recommendations found using DenseG_T was lower as compared to MAE of group recommendations found using SparseG_T.

Group recommendations can be refined to generate individual recommendations using Information Gain. Authors have obtained score of each group recommendation using (14) and fetched Top N individual recommendations from these group recommendations. Precision and Recall of individual Top 3 recommendations was higher than of group Top 3 recommendations (Figure 5). It was found that precision and recall of individual recommendations was increased (for example test users 2, 3, 5 and 9) or was comparable to precision and recall of group recommendations respectively.

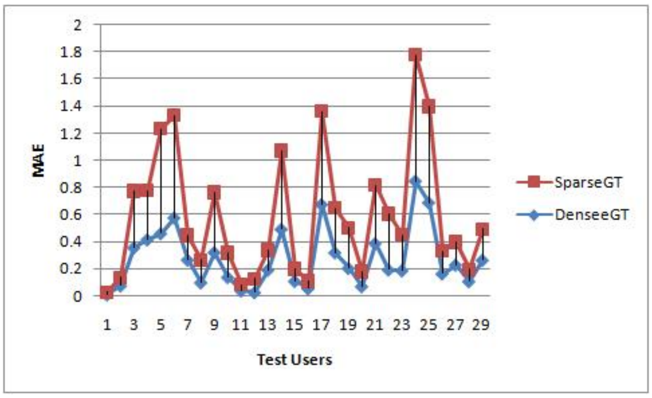

Figure 4. MAE of Test Users

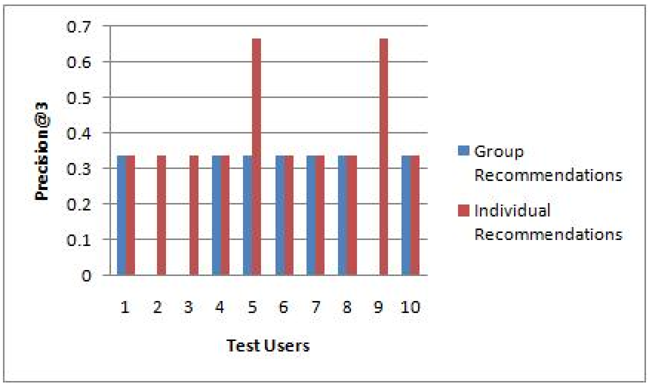

Figure 5. Precision of Test Users

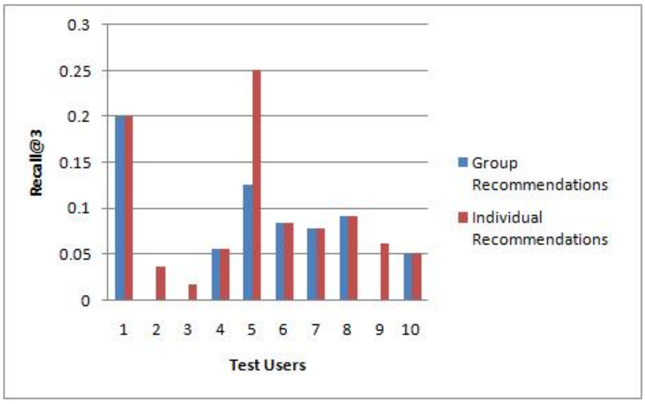

Figure 6. Recall of Test Users

V. Conclusion

Nowadays, Group Recommender Systems are widely used. But its sparsity problem often leads to difficulty in aggregating the preferences of group members resulting into error group recommendations. In this paper, a method (based on implicit multi criteria information) to increase the density of preferences among group members (STUs) is presented thereby enhancing the quality of predictions.

In the proposed scheme, positive and negative preferences between users are used to build group of similar users. Demographic attributes of target user and each member of the group determine the most promising group for the target user. It aggregates the local popularity of individual preferences of all the group members to generate Top N group recommendations. Opposite preferences between target user and the members of demographically similar group filters individual recommendations from group recommendations. Target user's initial rating profile is enhanced by incorporating genre attributes of items rated by target user and group members. This complete genre attribute based profile of target user fills the missing entries in the profile of each group member thereby reducing the sparsity among group members. Entropy has been used to depict the presence of a user in a group. Information Gain has been used to filter individual recommendations from group recommendations.

Results show considerable improvement in accuracy after sparsity reduction. In future, the authors intend to apply multi criteria ratings of group members for further refinement of recommendations.

References

[1] G. Adomavicius, A. Tuzhilin. "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". In *IEEE Transactions on Knowledge and Data Engineering, Vol 17, Issue 6*, pp 734-749, 2005.

[2] 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.

[3] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. "Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices". In *Applied Artificial Intelligence, Vol 1, Issue 8-9,* pp. 687-714, 2003.

[4] K. McCarthy, L. McGinty, and B. Smyth. "Case-based group recommendation: Compromising for success". In *Int. Conf. on Case Based Reasoning Research and Development*,Vol 4626, pp 299-313, 2007.

[5] 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*, *Kluwer Academia*, pp 199-218, 2001.

[6] Z. Yu, X. Zhou, Y. Hao, and J. Gu. "Tv program recommendation for multiple viewers based on user profile merging". In *User Modeling and User-Adapted Interaction, Vol 16, Issue 1*, pp. 63-82, 2006.

[7] R. Bruke, "Hybrid recommender systems; Survey and experiments". In *user modeling & user adapted interaction*, Vol 12, pp. 331-370, 2002.

[8] B. Sarwar, G. Karypis, J. Konstan, J. Riedl. "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering", In *Proc. of 5th International Conference on Computer and Information Technology*, 2002.

[9] M. Ghazanfar and A. Prugel-Bennett. "A Scalable, Accurate Hybrid Recommender System", In *Third International Conference on Knowledge Discovery and Data Mining, Thailand*, 2010.

[10] A. Jameson, B. Smyth. "Recommendation to groups", In *The Adaptive Web*, *Springer Verlag Berlin, Heidelberg,* pp 596 – 627 , 2007.

[11] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, C. Yu. "Group recommendation: Semantics and efficiency". In *PVLDB, Vol 2, Issue 1*, pp. 754-765, 2009.

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

[13] J. Masthoff. "Group Modelling: Selecting a sequence of television items to suit a group of viewers, In *User Modelling and User Adapted Interaction, Vol 14, Issue 1*, pp. 37-85, 2004.

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

[15] G. Linden, B. Smith, J. York. "Amazon.com Recommendations: Item to Item collaborative Filtering". In *IEEE Internet Computing 7*, pp 76- 80, 2002,.

[16] D. Goldberg, D. Nichols, B. M. Oki, D. Terry . "Using collaborative filtering to weave an information tapestry". In *Communications of the ACM*, *Vol 35, Issue 12,* pp 61-70, 1992.

[17] P. Resnick, N. Iacovou, M. Suchak, Bergstrom , J. Riedl. "*Grouplens: An open architecture for collaborative filtering of netnews*". In *Proceedings of the ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, pp 175-186, 1994.

[18] B. Sarwar, G. Karypis, J. Konstan, J. Riedl. "Item-Based collaborative filtering recommendation algorithms". In *Proceedings of the 10th International World Wide Web Conference*, pp 285-295, 2001.

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

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

[21] W. Zeng, M. S. Shang . "Effects of negative ratings on personalized recommendation." In *5th International Conference on Computer Science & Education Hefei, China*, pp. 24–27, 2010.

[22] L. Pizzato, T. Rej, K. Yacef, I. Koprinska and J. Kay. "Finding someone you will like and who won't reject you". In *Proc. of 19th International Conference on User Modeling, Adaptation and Personalization, Spain*, 2011.

[23] H. J. Kwon, T. H. Lee and K. S. Hong. "Improved memory based collaborative filtering using entropy based similarity measures", In *proceedings of International Symposium on web information systems and applications (WISA '09)*, pp 29-34, 2009.

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

[25] Y. Yang, J. Pederson, "A comparative study on feature selection in text categorization". In *proceedings of 14th international conference on machine learning*, pp 412-420, 1997

[26] 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, New York, NY, USA*, pp. 97-106, 2010.
[27] D. Billsus, M. Pazzani. "Learning collaborative information filters". In *Proceedings of the 15th National Conference on Machine Learning*, pp. 46–54, 1998.
[28] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, M. Sartin. "Combining content-based and collaborative filters in an online newspaper". In *Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, CA,* 1999.
[29] D. M. Pennock, E. Horvitz, S. Lawrence, L. Giles. "Collaborative filtering by personality diagnosis: a hybrid memory and model-based approach". In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, San Francisco*, pp. 473–480, 2000.
[30] Y. H. Chien, E. George. "A bayesian model for collaborative filtering". In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, Fort Lauderdale, Florida*, 1999.
[31] P. Melville, R. J. Mooney, R. Nagarajan. "Content boosted collaborative filtering for improved recommendations", In *Proc. Of 18th National conference on Artifical Intellegence, Edmonton, Canada* , pp. 187-192, 2002.
[32] S. Maneeroj, H. Kanai, K. Hakozaki. "Combining dynamic agents and collaborative filtering without sparsity rating problem for better recommendation quality". In *Proc. Of 2nd DELOS Network of Excellence Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.
[33] M. Kwak and D. S. Cho, "Collaborative Filtering with Automatic Rating for Recommendation," In *Proceeding of 2001 IEEE International Symposium on Industrial Electronics*, pp. 625-628, 2001.
[34] M. O' Connor, J. Herlock. "Clustering items for collaborative filtering". In *Recommender Systems Workshop at 1999 Conference on Research and Development in Information Retrieval, Berkeley, CA,* 1999.
[35] M. Papagelis, D. Plexousakis, and T. Kutsuras, "Alleviating the Sparsity Problem of Collaborative Filtering using Trust Inferences," In *Proceedings of I-Trust*, pp.224-239, 2005.
[36] H. J. Kwon, T. H. Lee, J. H. Kim and K. S. Hong. "Improving Prediction accuracy using entropy weighting in collaborative filtering", In *Symposia and workshops on Ubiquitous, Automatic and Trusted Computing, IEEE,* 2009.
[37] G. Adomavicius, N. Manouselis, and Y. Kwon. "Multi criteria recommender systems", In *Recommender system handbook, Springer,* pp 769- 803, 2011.
[38] W. Premchaiswadi and P. Poompuang, "Hybrid profiling for hybrid multi criteria recommendation based on implicit multi criteria information". In *Applied artifical Intelligence: An International Journal, Taylor and Francis, Vol 27*, pp 213-234, 2013.
[39] H. Mehta, V. S. Dixit and. P. Bedi. " Weighted Difference Entropy based Similarity Measure at Two levels in a Modified Memory based Collaborative Filtering Framework". *In ICACCI, WCI 2013 (Accepted), 2013.*
[40] H. Mehta, V. S. Dixit and P. Bedi, "Refinement of Recommendations based on User Preferences", In *Intelligent System Design and Applications (ISDA)*, pp 515-520, 2012.
[41] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering recommender systems", in *The adaptive Web: Methods and Strategies of Web personalization, vol 4321 of LNCS, chap 9,* Springer, pp 291-324, 2007
[42] Matlab http://www.mathworks.com
[43] C. Willmott, K. Matsuura. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in accessing average model performance," In *Climate Research, Vol 30*, pp. 79-82 , 2005.

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.

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.

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, personalization, steganography and steganalysis.