Evaluating Group Recommendation Strategies in Memory-Based Collaborative Filtering

Nadia A. Najjar
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC, USA
nanajjar@uncc.edu

David C. Wilson
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC, USA
davils@uncc.edu

ABSTRACT

Group recommendation presents significant challenges in evolving best practice approaches to group modeling, but even moreso in dataset collection for testing and in developing principled evaluation approaches across groups of users. Early research provided more limited, illustrative evaluations for group recommender approaches, but recent work has been exploring more comprehensive evaluative techniques. This paper describes our approach to evaluate group-based recommenders using data sets from traditional single-user collaborative filtering systems. The approach focuses on classic memory-based approaches to collaborative filtering, addressing constraints imposed by sparsity in the user-item matrix. In generating synthetic groups, we model 'actual' group preferences for evaluation by precise rating agreement among members. We evaluate representative group aggregation strategies in this context, providing a novel comparison point for earlier illustrative memory-based results and for more recent model-based work, as well as for models of actual group preference in evaluation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Algorithms, Experimentation, Standardization

Keywords

Evaluation, Group recommendation, Collaborative filtering, Memory-based

1. INTRODUCTION

Recommender systems have traditionally focused on the individual user as a target for personalized information filtering. As the field has grown, increasing attention is being given to the issue of group recommendation [14, 4]. Group recommender systems must manage and balance preferences from individuals across a group of users with a common purpose, in order to tailor choices, options, or information to the group as a whole. Group recommendations can help to support a variety of tasks and activities across domains that have a social aspect with shared-consumption needs. Common examples arise in social entertainment: finding a movie or a television show for family night, date night, or the like [22, 11, 23]; finding a restaurant for dinner with work colleagues, family, or friends [17]; finding a dish to cook that will satisfy the whole group [5], the book the book club should read next, the travel destination for the next family vacation [19, 2, 13], or the songs to play at any social event or at any shared public space [24, 3, 7, 9, 18].

Group recommenders have been distinguished from single user recommenders primarily by their need for an aggregation mechanism to represent the group. A considerable amount of research in group-based recommenders concentrates on the techniques used for a recommendation strategy, and two main group recommendation strategies have been proposed [14]. The first strategy merges the individual profiles of the group members into one group representative profile, while the second strategy merges the recommendation lists or predictions computed for each group member into one recommendation list presented to the group. Both strategies utilize recommendation approaches validated for individual users, leaving the aggregation strategy as a distinguishing area of study applicable for group-based recommenders.

Group recommendation presents significant challenges in evolving best practice approaches to group modeling, but even moreso in dataset collection for testing and in developing principled evaluation approaches across groups of users. Early research provided more limited, illustrative evaluations for group recommender approaches (e.g., [18, 20, 17]), but recent work has been exploring more comprehensive evaluative techniques (e.g., [4, 8, 1]). Broadly, evaluations have been conducted either via live user studies or via synthetic dataset analysis. In both types of evaluation, determining an overall group preference to use as ground truth in measuring recommender accuracy presents a complementary aggregation problem to group modeling for generating recommendations. Based on group interaction and group choice outcomes, either a gestalt decision is rendered for the group as a whole, or individual preferences are elicited and combined to represent the overall group preference. The former lends itself to user studies in which the decision emerges from

group discussion and interaction, while the latter lends itself to synthetic group analysis. Currently, the limited deployment of group recommender systems coupled with the additional overhead of bringing groups together for user studies has constrained the availability of data sets that can be used to evaluate group based recommenders. Thus as with other group evaluation efforts [4], we adopt the approach of generating synthetic groups for larger scale evaluation.

It is important to note that there are two distinct group modeling issues at play. The first is how to model a group for the purpose of making recommendations (i.e., what a group's preference outcome *will be*). We refer to this as the *recommendation group preference model* (RGPM). The second is how to determine an "actual" group preference based on outcomes in user data, in order to represent ground truth for evaluation purposes (i.e., what a group's preference outcome *was*). We refer to this as the *actual group preference model* (AGPM). For example, it might be considered a trivial recommendation if each group member had previously given a movie the same strong rating across the board. However, such an agreement point is ideal for evaluating whether that movie should have been recommended for the group.

In evaluating group-based recommenders, the primary context includes choices made about:

- the underlying recommendation strategy (e.g., content-based, collaborative memory-based or model-based)
- group modeling for making recommednations — RGPM (e.g., least misery)
- determining actual group preferences for evaluative comparison to system recommendations — AGPM (e.g., choice aggregation)
- choices about metrics for assessment (e.g., ranking, rating value).

Exploring the group recommendation space involves evaluation across a variety of such contexts.

To date, we are not aware of a larger-scale group recommender evaluation using synthetic data sets that (1) focuses on traditional memory-based collaborative filtering or (2) employs precise overlap across individual user ratings for evaluating actual group preference. Given the foundational role of classic user-based [22] collaborative filtering in recommender systems, we are interested in understanding the behavior of group recommendation in this context as a comparative baseline for evaluation. Given that additional inference to determine "ground truth" preference for synthetic groups can potentially decrease precision in evaluation, we are interested in comparing results when group members agree precisely in original ratings data.

In this paper, we focus on traditional memory-based approaches to collaborative filtering, addressing constraints imposed by sparsity in the user-item matrix. In generating valid synthetic groups, we model actual group preferences by direct rating agreement among members. Prediction accuracy is measured using root mean squared error and mean average error. We evaluate the performance of three representative group aggregation strategies (average, least misery, most happiness) [15] in this context, providing a novel comparison point for earlier illustrative memory-based results, for more recent model-based work, and for models of actual group preference in evaluation. This paper is organized as follows: section 2 overviews related researches. Section 3 outlines our group testing framework. Section 4 provides the evaluation using the proposed framework. Finally section 5 outlines our results and discussion.

2. RELATED WORK

Previous research that involves evaluation of group recommendation approaches falls into two primary categories. The first category employs synthetic datasets, generated from existing single-user datasets (typically MovieLens[1]). The second category focuses on user studies.

2.1 Group Aggregation Strategies

Various group modeling strategies for making recommendations have been proposed and tested to aggregate the individual group user's preferences into a recommendation for the group. Masthoff [16] evaluated eleven strategies inspired from social choice theory. Three representative strategies are average strategy, least misery, and most happiness.

- Average Strategy: this 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. Let n be the number of users in a group and *rij* be the rating of user j for item i, then the group rating for item i is computed as follows:

$$Gr_i = \frac{\sum_{j=1}^{n} r_{ji}}{n} \tag{1}$$

- Least Misery Strategy: this aggregation strategy is applicable in situations where the recommender system needs to avoid presenting an item that was really disliked by any of the group members, i.e., that goal is to please the least happy member. The predicted rating is calculated as the lowest rating of for any given item among group members and computed as follows:

$$Gr_i = \min_j r_{ji} \tag{2}$$

- Most Happiness: this aggregation strategy is the opposite of the least misery strategy. It applies in situations where the group is as happy as their happiest member and computed as follows:

$$Gr_i = \max_j r_{ji} \tag{3}$$

2.2 Evaluation with Synthetic Groups

Recent work by Baltrunas [4] used simulated groups to compare aggregation strategies of ranked lists produced by a model based collaborative filtering methodology using matrix factorization with gradient descent (SVD). This approach addresses sparsity issues for user similarity. The MovieLens data set was used to simulate groups of different sizes (2, 3, 4, 8) and different degrees of similarity (high, random). They employed a ranking evaluation metric, measuring the effectiveness of the predicted rank list using Normalized Discounted Cumulative Gain (nDCG). To account for the sparsity in the rating matrix nDCG was computed only over the items that appeared in the target user test set. The effectiveness of the group recommendation was measured as the average effectiveness (nDCG) of the group members where a higher nDCG indicated better performance.

[1]www.movielens.org

Chen et al. [8] also used simulated groups and addressed the sparsity in user-rating matrix by predicting the missing ratings of items belonging in the union set of items rated by group members. They simulated 338 random groups from the MovieLens data set and used it for evaluating the use of Genetic Algorithms to exploit single user ratings as well as item ratings given by groups to model group interactions and find suitable items that can be considered neighbors in their implemented neighborhood-based CF.

Amer-Yahia et al. [1] also simulated groups from MovieLens. The simulated groups where used to measure the performance of different strategies centered around a top-k TA algorithm. To generate groups a similarity level was specified, groups were formed from users that had a similarity value within a 0.05 margin. They varied the group similarity between 0.3, 0.5, 0.7 and 0.9 and the size 3, 5 and 8. It was unclear how actual group ratings were established for the simulated groups or how many groups were created.

2.3 Evaluation with User Studies

Masthoff [15] employed user studies, not to evaluate specific techniques, but to determine which group aggregation strategies people actually use. Thirty-nine human subjects were given the same individual rating sets from three people on a collection of video clips. Subjects were asked to decide which clips the group should see given time limitations for viewing only 1, 2, 3, 4, 5, 6, or 7 clips, respectively. In addition, why they made that selection. Results indicated that people particularly use the following strategies: Average, Average Without Misery and Least Misery.

PolyLens [20] evaluated qualitative feedback and changes in user behavior for a basic Least Misery aggregation strategy. Results showed that while users liked and used group recommendation, they disliked the minimize misery strategy They attributed this to the fact that this social value function is more applicable to groups of smaller sizes.

Amer-Yahia et al. [1] also ran a user study using Amazon's Mechanical Turk users, they had a total of 45 users where various groups were formed of sizes 3 and 8 to represent small and large groups. They established an evaluation baseline by generating a recommendation list using four implemented strategies. The resulting lists are combined into a single group list of distinct items and were presented to the users for evaluation where a relevance score of 1 was given if the user considered the item suitable for the group and 0 otherwise. They employed an nDCG measure to evaluate their proposed prediction lists consensus function. The nDCG measure was computed for each group member and the average was considered the effectiveness of the group recommendation.

Other work considers social relationships and interactions among group members when aggregating the predictions [10, 8, 21]. They model member interactions, social relationships, domain expertise, and dissimilarity among the group members when choosing a group decision strategy. For example, Recio-Garcia et al. [21] described a group recommender system that takes into account the personality types for the group members.

Berkovsky and Freyne [5] reported better performance in the recipe recommendation domain when aggregating the user profiles rather than aggregating individual user predictions. They implemented a memory-based recommendation approach comparing the performance of four recommendation strategies, including aggregated models and aggregated predictions. Their aggregated predictions strategy combined the predictions produced for each of the group members into one prediction using a weighted, linear combination of these predictions. Evaluation consisted of 170 users where a 108 of them belonged to a family group with size ranges between 1 and 4.

2.4 Establishing Group Preference

A major question that must be addressed in evaluating group recommender systems is how to establish the actual group preference in order to compare accuracy with system predictions. Previous work by [4, 8, 1] simulated groups from single-user data sets. Their simulated group creation was limited to groups of different sizes (representing small, medium and large) with certain degrees of similarity (random, homogeneous and heterogeneous). Chen et al. [8] used a baseline aggregation as the ground truth while [4] compares the effectiveness of the group-based recommendation to the effectiveness of the individual recommendations made to each member in the group. This led to our work in investigating ways to create synthesized groups from the most commonly used CF single-user data sets taking into consideration the ability to identify and establish ground truth. We propose a novel Group Testing Framework that allows for the creation of synthesized groups that can be used for testing in memory-based CF recommenders. In the remainder of the paper we give an overview of our proposed Group Testing Framework and we report on the evaluations we conducted using this framework.

Overall, larger-scale synthetic evaluations for group recommendation have not focused on traditional memory-based approaches. This may be because it is cumbersome to address group generation, given sparsity constraints in the user-item matrix. Moreover, only limited attention has been given to evaluation based on predictions, rather than ranking. Our evalution approach addresses these issues.

3. GROUP TESTING FRAMEWORK

We have developed a group testing framework in order to support evaluation of group recommender approaches. The framework is used to generate synthetic groups that are parametrized to test different group contexts. This enables exploration of various parameters, such as group diversity. The testing framework consists of two main components. The first component is a group model that defines specific group characteristics, such as group coherence. The second component is a group formation mechanism that applies the model to identify compatible groups from an underlying single-user data set, according to outcome parameters such as the number of groups to generate.

3.1 Group Model

In simulating groups of users, a given group will be defined based on certain constraints and characteristics, or *group model*. For example, we might want to test recommendations based on different levels of intra-group similarity or diversity. For a given dataset, the group model defines the space of potential groups for evaluation. While beyond the scope of this paper, we note that the group model for evaluation could include inter-group constraints (diversity across groups) as well as intra-group constraints (similarity within groups).

3.1.1 Group Descriptors

Gartrell et al. [10] use the term "group descriptors" for specific individual group characteristics (social, expertise, dissimilarity) to be accounted for within a group model. We adopt the *group descriptor* convention to refer to any quantifiable group characteristic that can reflect group structure and formation. Some of these group descriptors that can reflect group structure are user-user correlation, number of co-rated items between users and demographics such as age difference. We use these group descriptors to identify relationships between user pairs within a single user data set.

3.1.2 Group Threshold Matrix

A significant set of typical group descriptors can be evaluated on a pairwise basis between group members. For example, group coherence can be defined as a minimum degree of similarity between group members, or a minimum number of commonly rated items. We employ such pairwise group descriptors as a foundational element in generating candidate groups for evaluation. We operationalize these descriptors in a binary matrix data structure, referred to as the *Group Threshold Matrix* (GTM). The GTM is a square $n \times n$ symmetric matrix, where n is the number of users in the system, and the full symmetric matrix is employed for group generation. A single row or column corresponds to a single user, and a binary cell value represents whether the full set of pairwise group descriptors holds between the respectively paired users.

To populate the GTM, pairwise group descriptors are evaluated across each user pair in a given single-user dataset. The GTM enables efficient storage and operations for testing candidate group composition. A simple lookup indicates whether two users can group. A bitwise-AND operation on those two user rows indicates which (and how many) other users they can group with together. A further bitwise-AND with a third user indicates which (and how many) other users the three can group with together, and so on. Composing such row- (or column-) wise operations provides an efficient foundation for a generate-and-test approach to creating candidate groups from pairwise group descriptors.

3.2 Group Formation

Once the group model is constructed it can be applied to generate groups from any common CF user-rating data models as the underlying data source. The group formation mechanism applies the set of group descriptors to generate synthetic groups that are valid for the group model. It conducts an exhaustive search through the space of potential groups, employing heuristic pruning to limit the number of groups considered. Initially, individual users are filtered based on group descriptors that can be applied to single users (e.g., minimum number of items rated). The GTM is generated for remaining users. Baseline pairwise group descriptors are then used to eliminate some individual users from further consideration (e.g., minimum group size). The GTM is used to generate-and-test candidate groups for a given group size.

To address the issue of modeling actual group preferences for evaluating system predictions, the framework is tuned to identify groups where all group members gave at least one co-rated item the exact same rating among all group members. Such identified "test items" become candidates for the testing set in the evaluation process in conjunction with the corresponding group. We note that there are many potential approaches to model agreement among group members. In this implementation we choose the most straightforward approach, where the average rating among group members is equal to the individual group member rating for that item as a baseline for evaluation. We do not currently eliminate "universally popular" items, but enough test items are identified that we do not expect such items to make a significant difference. A common practice in evaluation frameworks is to divide data sets into test and target data sets. In this framework the test data set for each group would consist of the identified common item or items for that group.

4. EVALUATION

4.1 Baseline Collaborative Filtering

We implement the most prevalent memory-based CF algorithm, neighborhood-based CF algorithm [12, 22]. The basis for this algorithm is to calculate the similarity, w_{ab}, which reflects the correlation between two users a and b. We measure this correlation by computing the Pearson correlation defined as:

$$w_{ab} = \frac{\sum_{i=1}^{n}[(r_{ai} - \overline{r}_a)(r_{bi} - \overline{r}_b)]}{\sqrt{\sum_{i=1}^{n}(r_{ai} - \overline{r}_a)^2 \sum_{i=1}^{n}(r_{bi} - \overline{r}_b)^2}} \quad (4)$$

To generate predictions a subset of the nearest neighbors of the active user are chosen based on their correlation.

We then calculate a weighted aggregate of their ratings to generate predictions for that user. We use the following formula to calculate the prediction of item i for user a:

$$p_{ai} = \overline{r}_a + \frac{\sum_{b=1}^{n}[(r_{bi} - \overline{r}_b) \cdot w_{ab}]}{\sum_{b=1}^{n} w_{ab}} \quad (5)$$

Herlocker et al. [12] noted that setting a maximum for the neighborhood size less than 20 negatively affects the accuracy of the recommender systems. They recommend setting a maximum neighborhood size in the range of 20 to 60. We set the neighborhood size to 50 we also set that as the minimum neighborhood size for each member of the groups we considered for evaluation. Breese et al. [6] reported that neighbors with higher similarity correlation with the target user can be exceptionally more valuable as predictors than those with the lower similarity values. We set this threshold to 0.5 and we only consider the ones based on 5 or more co-rated items.

4.2 Group Prediction Aggregation

Previous group recommender research has focused on several group aggregation strategies for combining individual predictions. We evaluate the three group aggregation strategies which are outlined in section 2.1 as representative RGPMs. We compare the performance of these three aggregation strategies with respect to group characteristics: group size and the degree of similarity within the group.

4.3 Data Set

To evaluate the accuracy of an aggregated predicted rating for a group we use the MovieLens 100K ratings and 943 users data set. Simulated groups were created based on different thresholds defined for the group descriptors. The two

Table 1: Degrees of Group Similarity

Similarity Level	Definition $\forall i, j \in G$
High	$w_{ij} \geq 0.5$
Medium	$0.5 > w_{ij} \geq 0$
Low	$0 > w_{ij}$

Table 2: Similarity Statistics for Test Data Set

Degree of Similarity	Number of Valid Correlations	Average User-User Similarity
High	39,650	0.65
Medium	192,522	0.22
Low	95,739	-0.25

descriptors we varied were group size and degree of similarity among group members. We presume the same data set that is used to create the simulated groups is the same data set used to evaluate recommendation techniques.

By varying the thresholds of the group descriptors used to create the group threshold matrix we were able to represent groups of different characteristics, which we then used to find and generate groups for testing. One aspect we wanted to investigate is the affect of group homogeneity and size on the different aggregation methods used to predict a rating score for a group using the baseline CF algorithms defined in section 4.1. To answer this question we varied the threshold for the similarity descriptor and then varied the size of the group from 2 to 5. We defined three similarity levels: high, medium and low similarity groups as outlined in Table 1 where the inner similarity correlation between any two users i, j belonging to group G is calculated as defined in equation 1.

To ensure significance of the calculated similarity correlations we only consider user pairs that have at least 5 common rated items. For the MovieLens data set used we have a total of 444153 distinct correlations (943 taking two combinations at a time). For the three similarity levels defined previously the total correlation and average correlation are outlined in Table 2.

Table 3 reflects the GTM group generation statistics for the underlying data set used in our evaluation. Total combinations field indicate the number of possible group combinations that can be formed giving user pairs that satisfy our group size threshold descriptor. The valid groups field indicates the number of possible groups that satisfy both the size and similarity threshold whereas the testable groups are valid groups with at least one identified test item as described in section 3.2. As we increase the size of the groups to be created the number of combinations the implementation has to check increases significantly. We can also see that the number of testable groups is large in comparison to the number of groups used in actual user studies. As of this writing and due to system restrictions we were able to generate all testable groups for group size 2 and 3 across all similarity levels, group size 4 for low and high similarity level and group size 5 for the high similarity level.

4.4 The Testing Framework

The framework creates a Group Threshold Matrix based on the group descriptor conditions defined. In our implementation of this framework the group descriptors used to define inputs for the group threshold matrix are the user-user correlation and the number of co-rated items between any user pair. This forms the group model element of the testing framework. For the group formation element we varied the groups size and for each group the similarity category, 5000 testable groups were identified (with at least one common rating across group members). A predicted rating was computed for each group member and those values were aggregated to produce a final group predicted rating. Table 3 gives an overview of the number of different group combinations the framework needs to consider to identify valid, and testable groups. The framework exploits the possible combinations to identify groups where the group descriptors defined are valid between every user pair belonging to that group this is then depicted in the GTM.

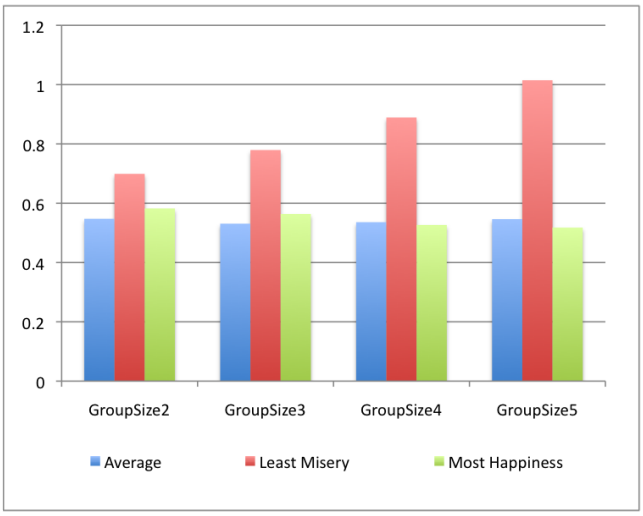

Figure 1: RMSE - High degree of similarity.

We then utilized the testing framework to assess the predicted rating computed for a group based on the three defined aggregation strategies in section 4.2. We compared the group predicted rating calculated for the test item to the actual rating using MAE and RMSE across the different aggregation methods.

It is worth noting here that just like any recommendation technique quality depends on the quality of the input data, the quality of the generated test set depends on the quality of the underlying individual ratings data set when it comes to the ability to generate predictions. For example, prediction accuracy and quality decrease due to sparsity in the original data set.

5. RESULTS AND DISCUSSION

Our evaluation goal is to test group recommendation based on traditional memory-based collaborative filtering techniques, in order to provide a basis of comparison that covers (1) synthetic group formation for this type of approach, and (2) group evaluation based on prediction rather than ranking. We hypothesize that aggregation results will support previous research for the aggregation strategies tested. In doing so, we investigate the relationship between the group's coherence, size and the aggregation strategy used. Figures 1-6 reflect the MAE and RMSE for these evaluated rela-

Table 3: Group Threshold Matrix Statistics

		2	3	4	5
High Similarity >= 0.5	Total Combinations	39,650	1,351,657	40,435,741	1,087,104,263
	Valid Groups	39,650	226,952	417,948	390,854
	Testable Groups	37,857	129,826	129,851	71,441
Medium >=0 < 0.5	Total combinations	192,522	30,379,236	3,942,207,750	434,621,369,457
	Valid groups	192,522	17,097,527		
	Testable groups	187,436	11,482,472		
Low similarity < 0.0	Total combinations	95,739	7,074,964	421,651,608	21,486,449,569
	Valid groups	95,739	1,641,946	6,184,151	
	Testable groups	87,642	470,257	283,676	

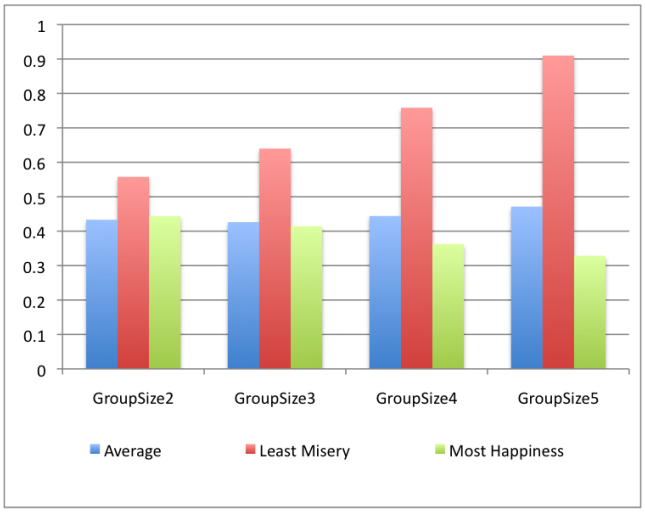

Figure 2: MAE - High degree of similarity.

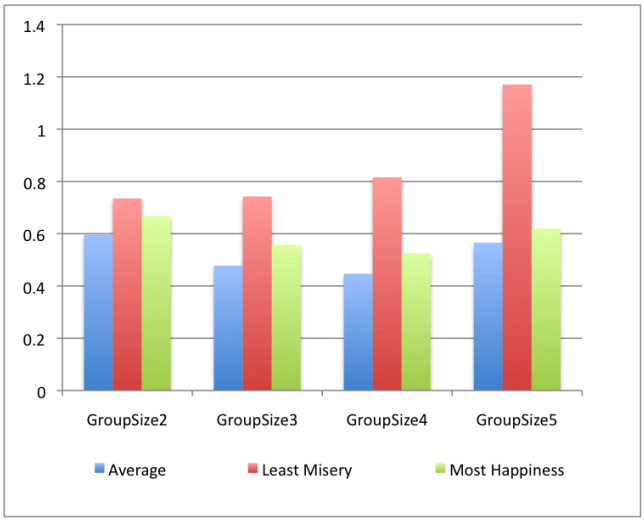

Figure 3: RMSE - Medium degree of similarity.

tionships. Examining the graphs for the groups with high similarity levels, Figures 1 and 2 show that average strategy and most happiness perform better than least misery. We conducted a *t-test* to evaluate the results significance and found that both MAE and RMSE for average and most happiness strategies, across all group sizes, significantly outperform the least misery strategy ($p<0.001$). For group sizes 2 and 3 there was no significant difference between the average and most happiness strategies ($p>0.01$). For group sizes 4 and 5 most happiness strategy performs better than the average strategy ($p<0.001$). Both least happiness and average strategies performance decreases as the group size grows. This indicates that a larger group of highly similar people are as happy as their happiest member.

Figures 3 and 4 show the RMSE and MAE for groups with medium similarity levels. The average strategy performs significantly better than most happiness and least misery across group sizes 2,3 and 4 ($p<0.001$). For the groups of size 5 there was no significant difference between average and most happiness strategies ($p>0.01$). For groups with medium similarity level the least misery strategy performance is similar to the groups with high coherency levels.

Figures 5 and 6 show the results for the groups with low similarity level. Examining the RMSE and MAE in these graphs the average strategy performs best across all group sizes compared to the other two strategies. MAE and RMSE for the average strategy for all group sizes with low

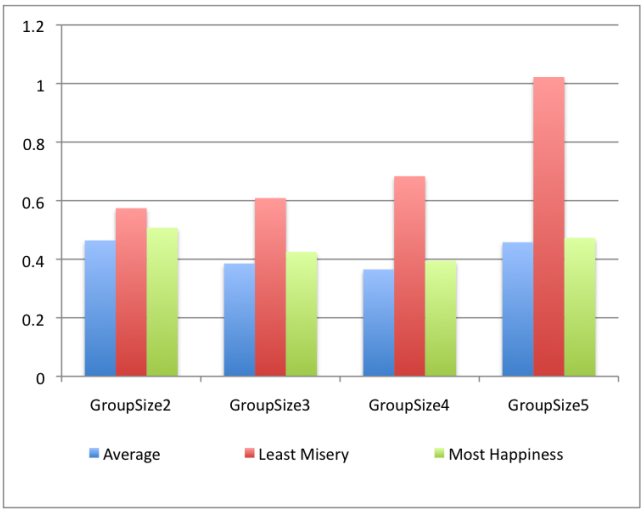

Figure 4: MAE - Medium degree of similarity.

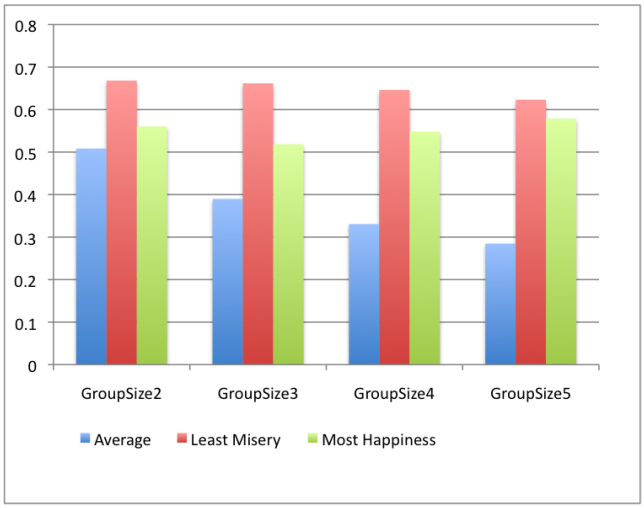

Figure 5: RMSE - Low degree of similarity.

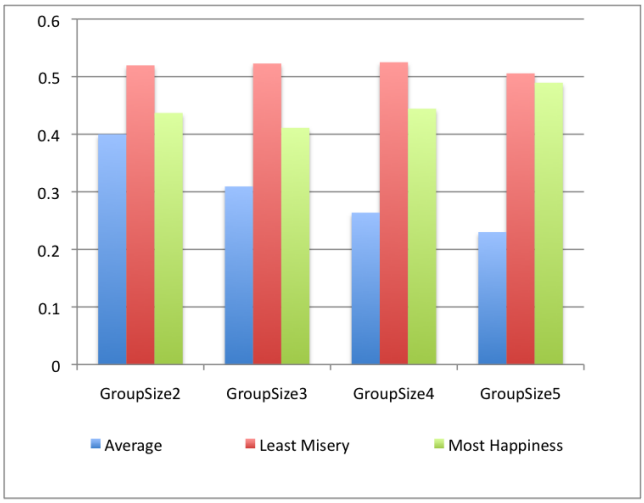

Figure 6: MAE - Low degree of similarity.

coherency had a statistically significant p value ($p<0.001$) compared to both least misery and most happiness strategies. Inconsistent with the groups with high coherency, for groups with low coherency the most happiness performance starts to decrease as the group size increases while the performance of the least misery strategy starts to increase.

These evaluation results indicate that in situations where groups are formed with highly similar members most happiness aggregation strategy would be best to model the RGPM while for groups with medium to low coherency average strategy would be best. These results using the 5000 synthesized groups for each category coincide with the results reported by Gartrell using real subjects. Gartrell defined groups based on the social relationships between the group members. They identified three levels of social relationships (couple, acquaintance and first-acquaintance) that might exist between group members. In their study to compare the performance of the three aggregation strategies across these social ties, they reported that for the groups of two members with a social tie defined as couple the most happiness strategy outperforms the other two. For the acquaintance groups, these groups had 3 members, the average strategy performs best, while for the first-acquaintance, they had one group with 12 members, the least misery strategy outperforms the best. It is apparent that their results for the couple groups performance is equivalent to our high-coherency groups, the acquaintance groups maps to the medium-coherency groups while the first-acquaintance groups follow the low-coherency groups. Masthoff studies reported that people usually used average strategy and least misery since they valued fairness and preventing misery. It is worth noting that her studies evaluated these strategies for groups of size 3 only without any reference to coherency levels.

6. CONCLUSION

As group-based recommender systems become more prevalent, there is an increasing need for evaluation approaches and data sets to enable more extensive analysis of such systems. In this paper we developed a group testing framework that can help address the problem by automating group formation resulting in generation of groups applicable for testing in this domain. Our work provides novel coverage in the group recommender evaluation space, considering (1) focus on traditional memory-based collaborative filtering, and (2) employs precise overlap across individual user ratings for evaluating actual group preference. We evaluated our framework with a foundational Collaborative Filtering neighborhood-based approach, prediction accuracy, and three representative group prediction aggregation strategies. Our results show that for small-sized groups with high-similarity among their members average and most happiness perform the best. For larger size groups with high-similarity performs most happiness performs better. For the low and medium similarity groups, average strategy has the best performance. Overall, this work has helped to extend the coverage of group recommender evaluation analysis, and we expect this will provide a novel point of comparison for further developments in this area. Going forward we plan to evaluate various parameterizations of our testing framework such as more flexible AGPM metrics (e.g. normalizing the ratings of the individual users).

7. REFERENCES

[1] S. Amer-yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *Proceedings of The Vldb Endowment*, 2:754–765, 2009.

[2] 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*, pages 687–714, 2003.

[3] C. Baccigalupo and E. Plaza. Poolcasting: A social web radio architecture for group customisation. In *Proceedings of the Third International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, pages 115–122, Washington, DC, USA, 2007. IEEE Computer Society.

[4] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 119–126, New York, NY, USA, 2010. ACM.

[5] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 111–118, New York, NY, USA, 2010. ACM.

[6] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14^{th} Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[7] D. L. Chao, J. Balthrop, and S. Forrest. Adaptive radio: achieving consensus using negative preferences. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, GROUP '05, pages 120–123, New York, NY, USA, 2005. ACM.

[8] 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.

[9] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: intelligent group music recommendation. In *Proceedings of the 7th international conference on Intelligent user interfaces*, IUI '02, pages 184–185, New York, NY, USA, 2002. ACM.

[10] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work*, GROUP '10, pages 97–106, New York, NY, USA, 2010. ACM.

[11] D. Goren-Bar and O. Glinansky. Fit-recommend ing tv programs to family members. *Computers & Graphics*, 28(2):149 – 156, 2004.

[12] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5:287–310, October 2002.

[13] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 48–54, New York, NY, USA, 2004. ACM.

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

[15] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14:37–85, February 2004.

[16] 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.

[17] J. F. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. pages 1–10, 2002.

[18] J. F. McCarthy and T. D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *CSCW*, page 348, 2000.

[19] K. McCarthy, M. SalamÃş, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Cats: A synchronous approach to collaborative group recommendation. pages 86–91, Melbourne Beach, Florida, USA, 11/05/2006 2006. AAAI Press, AAAI Press.

[20] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: a recommender system for groups of users. In *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, pages 199–218, Norwell, MA, USA, 2001. Kluwer Academic Publishers.

[21] 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.

[22] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *1994 ACM Conference on Computer Supported Collaborative Work Conference*, pages 175–186, Chapel Hill, NC, 10/1994 1994. Association of Computing Machinery, Association of Computing Machinery.

[23] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In P. De Bra, A. Kobsa, and D. Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 40–51. Springer Berlin / Heidelberg, 2010.

[24] D. Sprague, F. Wu, and M. Tory. Music selection using the partyvote democratic jukebox. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 433–436, New York, NY, USA, 2008. ACM.