

GROUP MODELING, RECOMMENDATION AND EVALUATION IN
COLLABORATIVE FILTERING GROUP-BASED RECOMMENDER SYSTEMS

by

Nadia Abusaymeh Najjar

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2015

Approved by:

Dr. David Wilson

Dr. Heather Lipford

Dr. Anita Raja

Dr. Mohamed Shehab

Dr. Jaime Bochantin

©2015
Nadia Abusaymeh Najjar
ALL RIGHTS RESERVED

ABSTRACT

NADIA ABUSAYMEH NAJJAR. Group modeling, recommendation and evaluation in collaborative filtering group-based recommender systems. (Under the direction of DR. DAVID WILSON)

As the field of recommender systems has grown, more and more attention has been focused on the need for systems that provide and tailor recommendations to groups of users, as opposed to individuals. We have identified open issues in group-based recommender system along three different dimensions: (1) the group recommendation technique employed to generate recommendations, (2) the group modeling strategy for generating group recommendations, and (3) the group evaluation metrics and procedures used for assessment of the recommendations. Group recommendation presents significant challenges in evolving best practice approaches to group modeling, recommendation, and evaluation. Early research provided more limited, illustrative evaluations for group recommender approaches, but recent work has been exploring more comprehensive evaluative techniques.

The main research problem we address is how to improve the prediction accuracy in group-based recommender systems employing a memory-based collaborative filtering technique. We break down this problem along the dimensions we identified in group-based systems to the following questions:

- How to implement a principled approach to evaluate the prediction accuracy in group-based recommender systems using datasets of individual users' preferences?

- Does the evaluation approach affect the results for the prediction accuracy?
- Does rating normalization increase the prediction accuracy for the group?
- Does incorporating the group-context in the neighborhood selection increase the prediction accuracy?
- Does a hybrid group-based strategy increase the prediction accuracy?
- Does incorporating the group context in the group modeling increase the prediction accuracy?

Group-based recommender systems introduce extra overhead in recruiting groups of users to cooperate toward a common goal at the same time. To overcome this limitation researchers have utilized publicly available large-scale datasets derived from individual-based recommender systems by creating synthesized groups and using them in offline evaluations. The need for a principled approach to evaluations utilizing this technique remains an open issue. We address this problem by developing a group testing framework to evaluate group-based recommenders in this context using data sets from traditional, single-user, collaborative filtering systems. We utilize this group testing framework in conducting comparatively large-scale evaluations of our proposed approaches along the problem dimensions of group-based recommenders. We first show the feasibility of an exact overlap constraint for evaluation. We then compare the prediction accuracy of some of the most commonly adopted group modeling strategies and compare to previous research that utilized synthesized groups with an average ground truth and report on the discrepancy of evaluation results between the two approaches. We also show that the choice modeling the ground truth for synthesized groups affects the evaluation results by comparing the prediction accuracy

using various models.

To be able to compare any new approach to previous research in this domain we need to utilize the same baseline they have used for evaluation where they model the actual group preference as an average of the individual group members preferences. We utilize the testing framework to create groups and the training and testing datasets for those groups. Since different users provide preferences on different scales research in individual based recommender has shown that normalizing the ratings in the prediction calculation increased the accuracy of predictions for users. This led us to explore the affect of rating normalization on the prediction accuracy for groups of users. We show the conditions where rating normalization would be beneficial for the group.

Previous research in this context has mainly adopted recommendation techniques validated for single user recommender systems with out considering the group context in the recommendation technique. We believe that utilizing the group context in the recommendation technique and group modeling would result in predictions with higher accuracy for the group. We evaluate a neighborhood model incorporating the group context in the neighborhood selection with a weighted approach based on neighborhood overlap. Our results show a higher prediction accuracy for the group is realized with this finer-grained neighborhood weighting model based on the group context is applied. We also evaluate a hybrid recommendation technique that incorporates the two group-based strategies. Our results show an increase in the prediction accuracy for groups in general. We also evaluate the performance of this approach along different group contexts and identify that this model is more advantageous for

groups with highly similar group members terms of prediction accuracy.

To incorporate the group context in the group modeling previous research proposed incorporating a disagreement component in the model using the predicted ratings for that item. This led us to explore ways to base the disagreement model using a concrete preference rather than a predicted one since we believe this would increase the prediction accuracy for the group. Our proposed disagreement model based on item similarity revealed that it might not be feasible to model disagreement by incorporating similar items, rated by all the group members, to the target item in the prediction calculation. Continuing to think along the lines of the group context as a first-class element of the group model we investigated a Case-based Reasoning approach where cases are matched based on the group context. Our approach is based on group-to-group similarity rather than user-to-user case matching where we retrieve whole previous groups as the starting point for predictions. We confirmed the potential benefit for integrating whole-group retrieval CBR approaches into group recommendation across different case-base and group conditions. We also demonstrated that the benefits of a CBR approach may be found even in straightforward implementations, showing the potential for a broad range of deployments and investigation in the space.

ACKNOWLEDGMENTS

I would like to express my special appreciation and thanks to my advisor, Dr. David Wilson, who has been a tremendous mentor for me. I would like to thank him for encouraging me and for allowing me to grow as a researcher and an academic. Dr. Wilson's advice has been invaluable.

I would also like to thank my committee members, Dr. Heather Lipford, Dr. Mohamed Shehab, and Dr. Anita Raja whose input and feedback during my proposal defense has helped me stay on the right track. I'd also like to thank Dr. Kelly Zellars for her role during my proposal defense. A special thank you to Dr. Jaime Bochantin for taking on the role of the Faculty Graduate Representative on my dissertation defense committee.

A heartfelt, special thanks to Dr. Bill Tolone for serving on my qualifying exam committee and for setting a great example as an educator, researcher, and program coordinator.

I could not have succeeded without the invaluable support of several. Thank you to my parents, Sadi and Fadwa Abusaymeh, for teaching me that education is indispensable and to never give up. Thanks to my siblings, Mudar, Dirar, Nidaa and their families for their love and care through out the years. Thank you to my father and mother in-law, Sakib and Bushra Najjar, for their continuous support, encouragement, and love.

I also like to mention some dear friends who've been there, since as far as I can remember, Ghada Sasa, Ghadir Sasa, Tamara Jaradat and Enas Alrousan. The fun

times we shared together helped me carry on.

To my beloved kids, Faris and Dalia, thank you for being such good kids. Thank you for all the distractions through this journey and for always cheering me up and on.

I owe so much to my husband Sami Najjar. A very special thank you for your practical and emotional support as I added the roles of wife and then mother to the challenging demands of work, study, and continuing personal development. Your support and understanding helped get me to where I am today.

Finally, I dedicate this work to my mother who's truly loved and missed.

TABLE OF CONTENTS

LIST OF FIGURES	xiv
LIST OF TABLES	xvi
CHAPTER 1: INTRODUCTION	1
1.1. Elements of Recommender Systems	5
1.1.1. Personalization	7
1.1.2. Filtering Technique	8
1.1.3. Recommendation	11
1.2. Collaborative Filtering	13
1.2.1. Neighborhood-based CF	15
1.2.2. Model-based CF	18
1.3. Evaluation Metrics for Recommender Systems	20
1.4. Group-Based Recommender Systems	22
1.4.1. Recommendations to Groups	22
1.4.2. Group Modeling	23
1.4.3. Group-based Evaluations	26
1.5. Research Scope and Contributions	27
CHAPTER 2: GROUP RECOMMENDER SYSTEMS BACKGROUND	32
2.1. Recommendation Technique	33
2.1.1. Demographic-based Group-Based Recommender Systems	33
2.1.2. Content-Based Group-Based Recommender Systems	34

2.1.3. Collaborative Filtering Group-Based Recommender Systems	36
2.2. Group Modeling	37
2.2.1. Weighted Linear Combination	38
2.2.2. Social Choice Theory	40
2.2.3. Extending Social Choice Theory Group Modeling Strategies	42
2.3. CBR	47
2.4. Evaluation in Group Recommendation Systems	50
2.4.1. Evaluation with User Studies	51
2.4.2. Evaluation with Synthetic Groups	55
2.5. Research Problems	61
CHAPTER 3: Group Testing Framework	64
3.1. Group Testing Framework	65
3.1.1. Group Model Descriptors	65
3.1.2. Group Threshold Matrix	66
3.1.3. Group Formation	68
3.2. Group Dataset	69
CHAPTER 4: Evaluation in Group Recommender Systems	76
4.1. Motivation	76
4.2. Group Characteristics and Structure	78
4.2.1. Hypothesis	78
4.2.2. Evaluation Setup	78

	xi
4.3. Results	81
4.4. Summary	85
4.5. Tradeoffs in Evaluation	88
4.5.1. Hypothesis	89
4.6. Dataset and Evaluation Process	89
4.7. Results and Discussion	89
4.7.1. Relationship between the AGPMs	90
4.7.2. Relationship between the RGPMS	91
4.7.3. Summary	95
CHAPTER 5: RECOMMENDATION TECHNIQUE FOR GROUP- BASED RECOMMENDER SYSTEMS	97
5.1. Memory-Based Collaborative Filtering In Group-Based Recom- mender Systems	98
5.1.1. Motivation	98
5.1.2. Rating Normalization for Prediction Calculation in Memory-Based Collaborative Filtering	99
5.1.3. Results	101
5.2. Neighborhood Selection In Memory-Based Group Recommender Systems	104
5.2.1. Hypothesis	106
5.2.2. Methodology	106
5.2.3. Evaluation Process	107
5.2.4. Results	107
5.2.5. Expert Evaluation	114

5.2.6.	Hypothesis	117
5.2.7.	Evaluation Metric	117
5.2.8.	Results	118
5.2.9.	Discussion	120
CHAPTER 6: GROUP MODELING IN GROUP-BASED RECOMMENDER SYSTEMS		122
6.1.	A Hybrid Approach to Group Recommender Systems	122
6.1.1.	Motivation	122
6.1.2.	A Hybrid Group Recommendation Strategy	123
6.1.3.	Results	125
6.2.	Disagreement Modeling in Group Aggregation for Group-Based Recommender Systems	129
6.2.1.	Motivation	129
6.2.2.	Disagreement Models	130
6.2.3.	Hypothesis	131
6.2.4.	Methodology	132
CHAPTER 7: Case-Based Reasoning Approach in Group Recommender Systems		137
7.1.	Motivation	137
7.2.	Exploring Whole-Group Case-Based Reasoning	139
7.2.1.	Hypotheses	140
7.3.	Case-Based Group Recommender	141
7.3.1.	Active Group to Case Similarity Metric	141
7.3.2.	Case Retrieval	142

	xiii
7.3.3. Adaptation for Recommendation	142
7.4. Experimental Setup	144
7.4.1. Accuracy Measurement	144
7.4.2. Case Base	145
7.5. Results	146
7.6. Summary	151
CHAPTER 8: Conclusion	153
8.1. Dissertation Contributions	153
8.2. Limitations and Future Work	160
REFERENCES	163

LIST OF FIGURES

FIGURE 1: Recommender system elements.	7
FIGURE 2: RentMe recommendation critique.	8
FIGURE 3: Amazon recommendation.	12
FIGURE 4: iTunes recommendation.	13
FIGURE 5: Collaborative filtering rating matrix.	14
FIGURE 6: Group recommendation strategy 1.	22
FIGURE 7: Group recommendation strategy 2.	23
FIGURE 8: Group Testing Framework	66
FIGURE 9: RMSE - High degree of similarity.	82
FIGURE 10: MAE - High degree of similarity.	82
FIGURE 11: RMSE - Medium degree of similarity.	84
FIGURE 12: MAE - Medium degree of similarity.	84
FIGURE 13: RMSE - Low degree of similarity.	85
FIGURE 14: MAE - Low degree of similarity.	85
FIGURE 15: RMSE for Avg_AGPM	91
FIGURE 16: RMSE for LM_AGPM	92
FIGURE 17: RMSE for MH_AGPM	92
FIGURE 20: Differential neighborhood selection	105
FIGURE 21: RMSE over all evaluated groups and test items	108
FIGURE 22: RMSE over all evaluated groups and test items with <i>Group_Neighborhood</i> size > 0	109
FIGURE 23: Overall Hybrid	126

FIGURE 24: Hybrid comparison for groups with high inner similarity levels	128
FIGURE 25: Hybrid comparison for groups with medium inner similarity levels	128
FIGURE 26: Hybrid comparison for groups with low inner similarity levels	129
FIGURE 27: Disagreement 80 20 weights	135
FIGURE 28: Disagreement 20 80 weights	135
FIGURE 29: RMSE for groups across the different inner similarity levels	149

LIST OF TABLES

TABLE 1: Degrees of group similarity	70
TABLE 2: Similarity statistics for test data set	71
TABLE 3: Group threshold matrix statistics	71
TABLE 4: Number of test items across group sizes and similarity levels	75
TABLE 5: Correlations between the RGPMs using the different AGPMs for groups with low similarity levels	93
TABLE 6: Correlations between the RGPMs using the different AGPMs for groups with medium similarity levels	93
TABLE 7: Correlations between the RGPMs using the different AGPMs for groups with high similarity levels	94
TABLE 8: RMSE values across the different group sizes	109
TABLE 9: RMSE values across the different sizes for groups and test items with <i>Group_Neighborhood</i> size > 0	110
TABLE 10: RMSE with respect to inner-group similarity level with prediction aggregation	111
TABLE 11: RMSE with respect to inner-group similarity level with prediction aggregation for groups and test items with <i>Group_Neighborhood</i> size > 0	111
TABLE 12: RMSE with respect to inner-group similarity level with Profile Merging	112
TABLE 13: RMSE with respect to inner-group similarity level with Profile Merging for groups and test items with <i>Group_Neighborhood</i> size > 0	113
TABLE 14: Wining strategies with respect to inner-group similarity level and group size based on RMSE	114

TABLE 15: Wining strategies based on RMSE with respect to inner-group similarity level with Profile Merging for groups and test items with <i>Group_Neighborhood</i> size > 0	114
TABLE 16: Success@3 with Profile Merging	118
TABLE 17: Success@3 with Recommendation Aggregation	119
TABLE 18: Success@3 with Profile Merging (PM) and Recommendation Aggregation (RA) for 100 groups	120
TABLE 19: Number of case-based groups for the different similarity level and size groups	147
TABLE 20: Number of case-based items for the different similarity level and size groups	147
TABLE 21: Success@n rates for our CBR approaches and [60]. * indicates value estimated from figure.	150

CHAPTER 1: INTRODUCTION

Recommender systems are tools that assist users in finding items of interest by helping them navigate through large and complex information spaces [65]. They are intelligent information systems that use machine learning and information filtering techniques to assist the users in finding choices they otherwise would likely have missed. Recommender systems have been applied in many domains. Common examples include recommending movies [54], music [71], books [45], news [40], and web pages [43, 4]. Suggestions for books on Amazon, or movies on Netflix, are commercial examples of such systems. In 2006, Netflix released a data set containing 100 million anonymous movie ratings and challenged the data mining, machine learning, and computer science communities to develop systems that could beat the prediction accuracy of its baseline recommendation on this test data set. The contest created a buzz within the recommender systems field. Until this point, the only publicly available data for use in recommender systems research was orders of magnitude smaller. According to the contest website¹, more than 48,000 teams from 182 different countries had downloaded the data. This has increased the attention on such systems and their research.

Recommender systems have traditionally focused on the individual user as a target for personalized information filtering. As the field of recommender systems has grown,

¹www.netflixprize.com

increasing attention is being focused on the need for systems that provide and tailor recommendations to a group of users rather than single users [38, 5]. For example, a group of friends trying to find a movie to watch together, might use such a system. 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 [57, 30, 70]; finding a restaurant for dinner with work colleagues, family, or friends [49]; finding a dish to cook that will satisfy the whole group [6], the book that a book club should read next, the travel destination for the next family vacation [53, 2, 37], or the songs to play at any social event or at any shared public space [73, 3, 17, 20, 50]. This dissertation is centered around group-based recommender systems exploring how to increase the performance of such systems in terms of accuracy.

Research in group-based recommender systems has been focused on approaches to model the group for recommendation and two main approaches have been proposed: aggregating preferences and aggregating recommendations [38]. In the aggregating preferences approach, the individual group members' preferences are combined into a group preference model. In the aggregating recommendations approach, group modeling is applied by combining recommendations produced for the individual group members into a group recommendation. Group modeling strategies are inspired by Social Choice Theory and focus on modeling the achievement of consensus among

the group [46]. Variations to these strategies that take into account personality and social interactions of the group members have been implemented [63][26]. In our research, we consider research issues with group-based recommender systems along three different dimensions: (1) the group recommendation technique employed to generate recommendations, (2) the group modeling for generating group recommendations, and (3) the group evaluation metrics and procedures used for assessment of the recommendations.

Group-based recommenders have adopted recommendation techniques that have been verified for single-user recommender systems. Research along this dimension focused on adapting these techniques for use in group-based recommenders with little attention given to the fact that these recommendations are generated for a group rather than individuals. We believe that accounting for the group structure and characteristics, as part of the recommendation technique, will result in higher quality recommendations for the group. We evaluate incorporating the group context in the recommendation technique using a finer-grained weighting schema to neighbors of the group members based on the group context.

Recent work in group-based recommender systems focused on evaluating the group modeling strategy performance with respect to group size and cohesiveness. We believe that accounting for the group structure and characteristics in the choice of an aggregation strategy, as well as combining the recommendations generated by the two group modeling approaches, will result in higher quality recommendations for the group.

The availability of public data sets for evaluating recommender systems played

a significant role in the advancement of this field (e.g., Netflix prize, MovieLens²). When it comes to group-based recommender systems such data is not readily available. Generating group-based data requires extra overhead in recruiting the groups together and getting them to cooperate and interact towards a common goal at the same time. Recently, researchers have been utilizing synthetic groups, generated from single-user data sets, to evaluate various approaches to group recommendations. The problem with this approach lies in the scope of establishing the ground truth for these synthesized groups. Previous work that adopted this approach have mainly evaluated the group recommendations against the individual group members' recommendations. We have been exploring ways to utilize the publicly available data sets in the evaluation of group-based recommenders focusing on identifying groups with natural agreement to alleviate the problem with establishing ground truth. We have developed a group testing framework and utilized it to evaluate several foundational techniques employed to generate a recommendation for a group of users [56]. The framework scans the individual-based recommendation data set and creates synthesized groups based on giving constraints and group characteristics as well as identifying some ideal test points for the generated groups.

In this dissertation we extend the evaluation of foundational techniques employed in group-based systems, utilizing this group testing framework and introducing new group modeling and recommendation techniques for group recommenders. We evaluated the tradeoffs in evaluations that measure the performance of the system in terms of the satisfaction of the group in a large-scale evaluation. We evaluated rating

²www.movielens.org

normalization on prediction accuracy for the group. We proposed a neighborhood selection approach as well as a neighbor weighting model based on the group context. We also explored group modeling strategies, used to generate personalized recommendations to a group of users, that combine group modeling approaches using a hybrid model combining profile merging and prediction aggregation taking into account the group structure and characteristics. We evaluated a group modeling approach based on disagreement as well as a group model based on case-based reasoning. The next sections of this chapter introduce recommender systems, techniques and evaluation metrics with a more detailed overview of the techniques relevant to the scope of this dissertation.

1.1 Elements of Recommender Systems

Konstan [39] defines recommender systems as, “Recommender systems help individuals manage a potentially overwhelming set of choices by suggesting specific information, products, or people to those individuals based on the systems’ knowledge of the individual’s preferences and/or current need, and the collected knowledge of preferences within the larger community of system users.” They are commonly used by web sites that suggest products to their users. Suggestions can be based on popularity, demographics of the user, geographic location/time of the user or an analysis of the past behavior of the customer as a prediction for future behavior. These suggestions are personalized, because they help the site adapt itself to each user. Some systems offer personalization without recommendations. For example, some webpages customize the background based on the geographic location of the web request (e.g. Google). On the other hand, some systems offer recommendations

without personalization such as recommendations of best sellers or featured products. Our research focuses on systems that offer personalized recommendations.

In [69] a taxonomy for E-commerce recommender applications is defined, that separates their attributes into three categories: functional I/O, recommendation method, and other design issues. Montaner et al. [55] proposed a taxonomy of recommender systems that is focused on the user profile generation and maintenance, and includes the following five classifications: the profile representation technique, the technique used to generate the initial profile, the source of the relevance feedback which represents the user interests, the profile learning technique, and the profile adaptation technique.

Based on the dimensions of group-based recommenders we identified earlier we re-model these categorizations of recommender systems focusing along three different dimensions: personalization, filtering technique, and recommendation. Figure 1 illustrates these dimensions and outlines how they relate to one another. Personalization relies on a user model to tailor the system behavior to the user's needs and preferences. The filtering technique involves the methods of identifying items of interest to the user. The recommendation dimension depends on information about the target model in order to produce the recommendation.

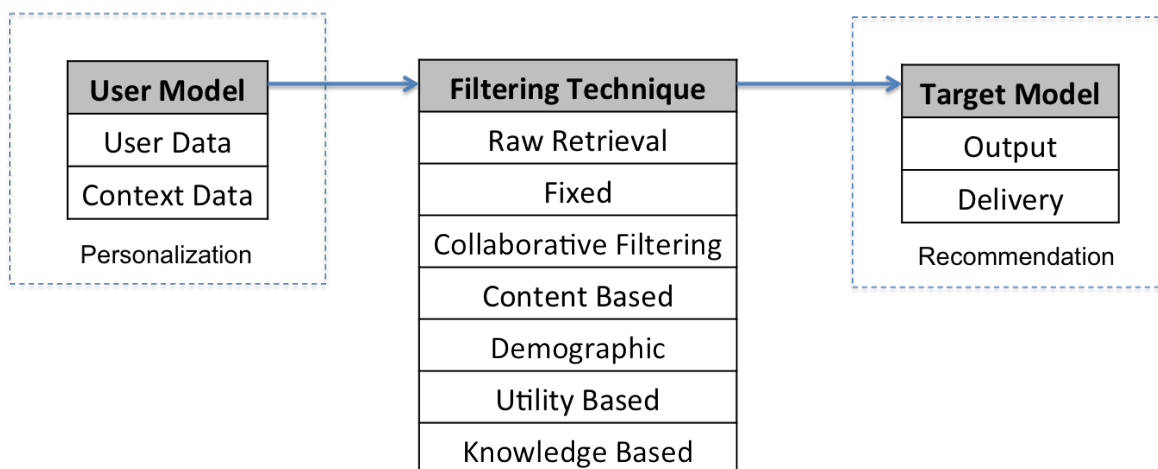


Figure 1: Recommender systems create a user model, based on user and context data, which is utilized by the recommendation technique to identify relevant items and presented to the user based on the target model defined by the system.

1.1.1 Personalization

Personalization consists of a user model that is defined based on user and context information that is relevant to the domain of recommendation. This information comes in two forms, either implicit or explicit user preferences [27]. For the user data, implicit preferences are inferred from the user's behavior without the user's awareness of their use in the recommendation processes. They are usually site specific such as navigation and purchase history. The context data is also inferred from the user's behavior such as item popularity based on site specific or external users' purchase history. Explicit preferences are intentionally provided by the user with the purpose of informing the recommender application of their preferences. These are usually in the form of item ratings or item attributes. The user data contains information the user specifically provided to the system, while the context data contains information pertaining to the domain of recommendation provided by other system users or administrators. Context data includes different types of information, each type defining

a certain aspect of context, such as time, location (e.g., theater), companions (e.g., for seeing a movie), purpose of a purchase, etc.[65].

Variants to implicit and explicit preferences are preferences identified through conversational-based recommender systems. In such systems the user is guided through a sequence of recommendation cycles in which one or more items are recommended based on some evolving user model [16]. During each cycle the user is given the opportunity to provide feedback in order to help guide the system closer to the direction of their desired item. These systems are also referred to as critique-based recommender systems. Figure 2 shows an example of such interactions in the RentMe [16] system.

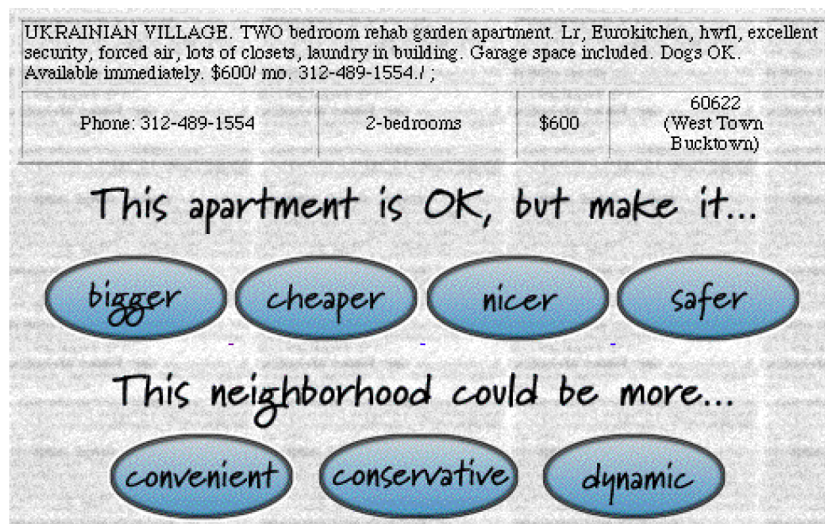


Figure 2: Critiquing an apartment during interaction with the RentMe system [16].

1.1.2 Filtering Technique

The filtering technique refers to the methodology used to identify items that match the user model. These methods can be in their basic form a database lookup to a more granular approach that does the matching based on more complex filtering algorithms. Based on [69, 14, 55] we identify seven approaches: Query, Fixed, Demographic, Col-

laborative Filtering, Content-based, Utility-based and Knowledge-based. The Query, Fixed, and Demographic approaches are the more general of these approaches and offer less personalization. The Utility-based and Knowledge-based approaches are based on knowledge about how a particular object satisfies the user needs by encoding expert domain knowledge. Details on Query and Fixed can be found in [69] Demographic details can be found in [55], Utility-based and Knowledge-based details can be found in [14] and Collaborative Filtering and Content-based details can be found in [69, 55].

The most widely adopted approaches are Collaborative Filtering and Content-based. Collaborative Filtering recommends items based on correlations between users or items. For example, if two users rate items similarly, or have similar behaviors (e.g., buying, watching, listening), the system infers that they will rate or act on other items similarly. Content-based techniques recommend items by identifying them based on attributes, keywords and descriptions of items. For example, knowing the genre information for movies, and knowing that a user liked “X-Men” and “Iron Man,” the system infers that the user will like “Batman Forever,” since its genre classifications: action and adventure, comic book and superhero movies, are similar to the other two movies. The research scope in this dissertation is focused on the Collaborative Filtering approach.

Each filtering technique has its limitations and drawbacks. The query and fixed approach have more of a *general* feel than a *personalized* one, but are useful in situations where the user model is based on limited user and context data. Demographic techniques are similar to collaborative ones in that they are based on correlations

between users, but use different data. Users are categorized based on their personal attributes and recommendations are made under the assumption that users in the same category have the same tastes or preferences. The main advantage of this technique is that it does not require a history of user ratings needed by collaborative and content-based techniques.

Collaborative Filtering (CF) is the most common, most widely implemented, and most mature of these approaches [41]. They aggregate ratings of items, recognize similarities between users based on these ratings, and generate new recommendations based on user-to-user comparisons. CF systems work best in situations where the density of user preferences is relatively high across a more constant space of items and suffer when this is not the case. Given the foundational role of Collaborative Filtering in the research of recommender systems and the availability of public data sets for use in evaluation, we adopt the CF approach as the filtering technique. The following sections give a more detailed overview of Collaborative Filtering systems. Section 1.2 gives a more detailed overview of this approach.

Recommender systems can utilize two or more filtering techniques to overcome limitations of any individual one and gain better performance [15]. Systems that function in this way are often referred to as hybrid recommender systems. One simple approach is by combining the two predictions generated for an item by Content-based and Collaborative Filtering methods using a weighted average. Several hybridization approaches have been identified and employed in recommendation systems [14]. The most common approaches are Weighted, Switching and Mixed. In the Weighted approach, the scores (or votes) of several recommendation techniques are combined

together to produce a single recommendation. In the Switching approach, the system switches between recommendation techniques depending on the current situation while, in the Mixed approach, the recommendations from several different recommenders are presented at the same time.

1.1.3 Recommendation

The recommendation dimension consists of a target model that defines how to select and present items identified through the filtering technique. Two aspects need to be considered here; the form the recommendation output needs to take and how it will be delivered to the user. Some outputs can be in the form of a suggestion of an item, prediction of user preference for an item, or a ranked list of items. Delivery of recommendations has taken several forms such as being directly communicated to the user via email for example, or creating awareness of the availability of such recommendations to the user and waiting for the user to actively request them before delivery. For example, Figure 3 depicts how Amazon notifies the user of the availability of recommendations personalized for them.

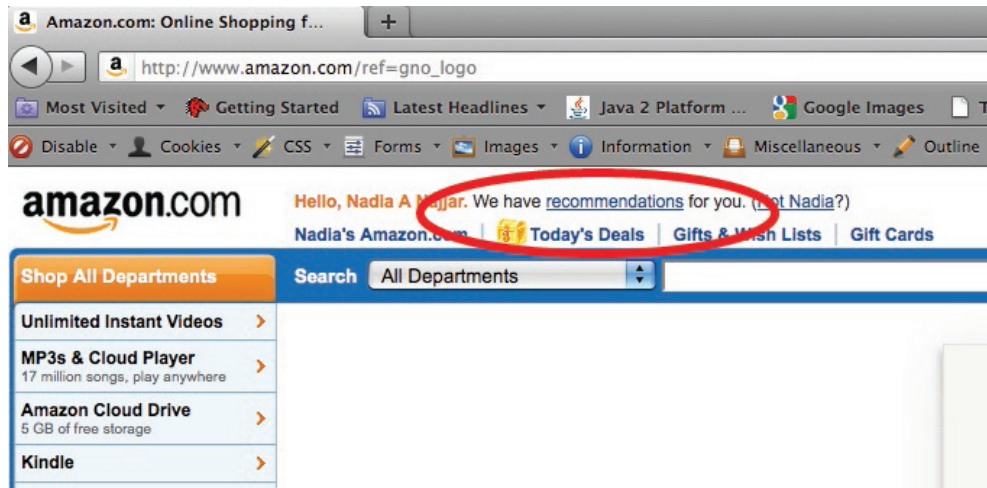
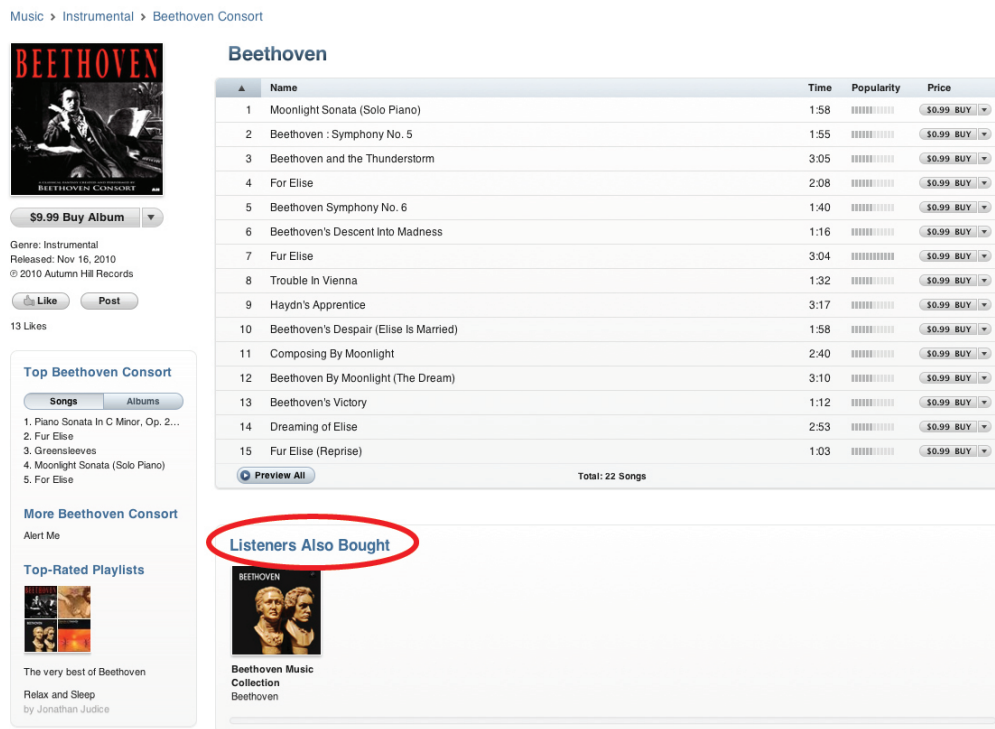


Figure 3: Amazon's recommendation system makes the user aware of the availability of recommendations personalized for them.

On the other hand, some systems might passively introduce recommendations as the user navigates through their site (e.g., people who viewed this item also viewed these items). Figure 4 depicts how the iTunes recommendation system suggests new items to the user based on their current context of navigation.

Music > Instrumental > Beethoven Consort



Beethoven

#	Name	Time	Popularity	Price
1	Moonlight Sonata (Solo Piano)	1:58	██████████	\$0.99 BUY
2	Beethoven : Symphony No. 5	1:55	██████████	\$0.99 BUY
3	Beethoven and the Thunderstorm	3:05	██████████	\$0.99 BUY
4	For Elise	2:08	██████████	\$0.99 BUY
5	Beethoven Symphony No. 6	1:40	██████████	\$0.99 BUY
6	Beethoven's Descent Into Madness	1:16	██████████	\$0.99 BUY
7	Fur Elise	3:04	██████████	\$0.99 BUY
8	Trouble In Vienna	1:32	██████████	\$0.99 BUY
9	Haydn's Apprenice	3:17	██████████	\$0.99 BUY
10	Beethoven's Despair (Elise Is Married)	1:58	██████████	\$0.99 BUY
11	Composing By Moonlight	2:40	██████████	\$0.99 BUY
12	Beethoven By Moonlight (The Dream)	3:10	██████████	\$0.99 BUY
13	Beethoven's Victory	1:12	██████████	\$0.99 BUY
14	Dreaming of Elise	2:53	██████████	\$0.99 BUY
15	Fur Elise (Reprise)	1:03	██████████	\$0.99 BUY

Total: 22 Songs

Listeners Also Bought

Beethoven Music Collection
Beethoven

Figure 4: iTunes music recommendation system introduces new items to the user based on the current context.

1.2 Collaborative Filtering

CF systems are built on the main idea that “people who agreed in the past are likely to agree in the future” [64]. They were first introduced in the context of recommender systems by Tapestry, a system that was designed to filter and recommend documents extracted from newsgroups to a collection of users. They define collaborative filtering as “people collaborate to help one another perform filtering by recording their reactions to documents they read” [28]. These reactions can be explicit, in the form of a preference rating, or implicit actions taken on items that can be interpreted as preferences. These actions can include click-through, viewing, adding to wish list/cart, purchasing, or returning an item. CF systems work by collecting user feedback on items in a given domain and exploiting similarities in preference behavior among users

to identify items for recommendation. The users of a collaborative filtering system share their analytical judgments and opinions regarding each item that they consume so that other users of the system can better decide which items to consume [33].

The core of a CF system is a user-item ratings matrix of size $m \times n$, where m is number of users and n is the number of items. Each user in the matrix has a set of items which the user has rated, or for which their preferences have been inferred, and a set of items where preferences need to be predicted. CF systems face many challenges where techniques used need to be able to deal with highly sparse data (sparsity), to scale with the increasing numbers of users and items (scalability), and to make satisfactory recommendations in a short period of time (speed).

		<i>Items</i>					
		<i>1</i>	<i>2</i>	<i>...</i>	<i>i</i>	<i>...</i>	<i>n</i>
<i>1</i>	:	5	3		1	2	
<i>2</i>	:		2				4
<i>u</i>	:			5			
<i>u</i>	:	3	4		?	1	
<i>u</i>	:					4	
<i>m</i>	:			3	2		

Figure 5: User ratings matrix, where each cell corresponds to the rating of user u for item i . The CF task is to predict the missing rating r_{ui} for an active user.

CF methods are divided into two main categories; memory-based and model-based. The Memory-based [9, 64] CF approach is commonly referred to as Neighborhood-based and relies on the fact that each person belongs in a larger group of similarly behaving individuals. As a result, items (e.g., products, movies, books, etc.) frequently purchased/liked by the various members of the group can be used to form a basis for recommended items. This approach utilizes the entire ratings database to generate a prediction by using statistical techniques to find the neighbors.

The second approach is known as model-based [9], where the CF task is viewed from a probabilistic perspective and involves building a predictive model based on the ratings data. This model is then trained using the available data, and later used to predict ratings of users for new items. Several techniques have been used to build these models such as Bayesian Network (BN) [9], Clustering [9][29], Association Rules Latent Semantic Analysis [35], Latent Dirichlet Allocation [8], Maximum Entropy [74], Boltzmann Machines [66], Support Vector Machines [31], and Singular Value Decomposition [7][68][41].

The model-based approach potentially offers the benefits of both scalability and speed. It offers scalability, since models resulting from model-based algorithms are much smaller than the actual data set, and speed, since the model is smaller to exploit and can usually be constructed offline. Recommendations produced by simple modeling techniques (e.g., clustering, classification) are often lower in their prediction accuracy than ones produced using a memory-based approach [65]. In contrast, a memory-based approach tends to produce recommendations with higher prediction accuracy, but suffers serious scalability problems as the complexity of computing each recommendation grows linearly with the number of users and items.

1.2.1 Neighborhood-based CF

The Neighborhood-based method can be outlined in the following steps [33]:

1. Assign a weight to all users with respect to their similarity to the active user.
2. Select the k users that have the highest similarity to the active user. This forms the “*neighborhood*.”
3. Calculate a prediction for items based on a weighted combination of the iden-

tified neighbors' ratings.

The GroupLens system was first introduced as an automated collaborative filtering system using a neighborhood-based algorithm [64]. GroupLens provided personalized predictions for Usenet news articles. The GroupLens system employed the Pearson Coefficient to weigh user similarity, used all available correlated neighbors, and computed a final prediction by performing a weighted average of deviations from the neighbor's mean.

In the first step of this approach, we need to measure the similarity between the user seeking the recommendation and all the other users of the system. The most common algorithm used to calculate similarity weighting is the Pearson Correlation Coefficient (PCC). Ringo [71] expanded upon the original GroupLens algorithm and reported better prediction accuracy by computing similarity weights using a variant of the PCC, the Constrained Pearson Coefficient, which uses the midpoint of the rating scale instead of average rating. Other methods for similarity computations are Spearman Rank Correlation, which is similar to PCC, except that the ratings are ranks; and Kendall's τ correlation, similar to the Spearman Rank Correlation, but instead of using ranks themselves, only the relative ranks are used to calculate the correlation. Herlocker et al. [33] reported better results using PCC rather than Spearman Rank Correlation difference.

Herlocker et al. [33] also suggests devaluing similarity weights that are based on a small number of co-rated items between any two users reflecting the amount of trust to be placed in a correlation with a neighbor. The more overlap in ratings between two

users the more trust can be placed on the computed correlation as a representative of the true correlation between the two users. They refer to this strategy as *significance weighting* and it is realized by applying a linear drop-off to correlations that were based on less than a certain threshold of co-rated items. Their evaluation revealed that significance thresholds of 25 or more do improve the accuracy of the system.

After assigning similarity weights to the users of the systems, the next step is to select the users that will form the active user's neighborhood and be used in the computation of a prediction. Since most commercial CF systems handle millions of users, considering every user as a neighbor might be infeasible for real-time performance. Furthermore, many users of the system might not have similar tastes to the active user, so using them as predictors will only increase the error of the prediction. Two approaches can be applied to limit the number of neighbors considered; similarity weight threshold and best-n-neighbors. Herlocker et. al. reported that weight thresholding only made matters worse, decreasing both the coverage and the accuracy of the system, and found that neighborhoods of 20 to 50 neighbors are reasonable, providing enough neighbors to average out extremes.

Once the neighborhood has been selected, the ratings from those neighbors are combined to compute a prediction. Since every user has his own internal model of what a particular rating scale means, and rate on their own distribution, there's a need for some sort of transformation so that user's ratings are in the same space. This can be achieved by normalizing each predicted rating based on the user's actual rating habits. The approach taken by GroupLens was to compute the average deviation of a neighbor's rating from that neighbor's mean rating, where the mean rating

is taken over all items that the neighbor has rated. The average deviation from the mean computed across all neighbors is converted into the active user's rating distribution by adding it to the active user's mean rating. Herlocker et. al. evaluated an extension to the GroupLens algorithm by accounting for the differences in spread between users' rating distributions by converting ratings to z-scores, and computing a weighted average of the z-scores. They [33] reported that performing rating normalization produces an obvious benefit over a non-normalized approach and that the z-score normalization performed slightly better than the deviation from mean.

1.2.2 Model-based CF

Model-based techniques provide recommendations by estimating parameters of statistical models for user ratings. A Model-based method can be outline in the following steps:

1. Developing a model of user ratings.
2. Computing the expected value of a user prediction, given his/her ratings on other items.

Early approaches mapped CF as a classification or a clustering problem. Breese et al.[9] presented two model-based algorithms; a clustering model, and a Bayesian network model. The first algorithm follows a probabilistic approach in which the users are clustered and the conditional probability distribution of different items in the cluster is estimated. The probability that the active user belongs to a particular cluster, given the set of items, is then estimated from the clustering solution and the probability distribution of items in the cluster. The clustering solution for this tech-

nique is computed using the expectation maximization (EM) principle. The second algorithm is based on Bayesian network models where each item in the database is modeled as a node having states corresponding to the rating of that item. The learning problem consists of building a network on these nodes such that each node has a set of parent nodes that are the best predictors for the child's rating. They presented a detailed comparison of these two model-based approaches with the user-based approach and showed that Bayesian network models outperformed the clustering model and the user-based scheme.

More recently attention has focused on latent factor and matrix factorization models as state-of the art techniques for model based CF [41]. This approach assumes that the similarity between users and items is simultaneously induced by some hidden characteristics of the users and items in the system. For example, the rating that a user gives to a movie might be assumed to depend on few implicit factors such as the user's taste across various movie genres. Matrix factorization techniques are a class of widely successful latent factor models where users and items are simultaneously represented as unknown feature vectors (column vectors) along k latent dimensions. These feature vectors are learned so that inner products of these vectors approximate the known preference ratings with respect to some loss measure. For a given user the resulting dot product measures the extent of interest the user has in items that are high on the corresponding factors.

This model is closely related to singular value decomposition (SVD), a well-established technique for identifying latent semantic factors in information retrieval. Applying SVD in the collaborative filtering domain requires factoring the user-item rating ma-

trix. This often raises difficulties due to the high portion of missing values caused by sparseness in the user-item ratings matrix and conventional SVD is undefined when knowledge about the matrix is incomplete. Utilizing only the relatively few known entries in a sparse user-item matrix is highly prone to over-fitting. Over-fitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Earlier systems relied on imputation to fill in missing ratings and make the rating matrix dense. Since imputation can be very expensive as it significantly increases the amount of data and an inaccurate imputation might distort the data considerably, recent work suggested modeling directly the observed ratings only, while avoiding overfitting through a regularized model. Koren [41] suggested modeling directly the observed ratings only, while avoiding overfitting through a regularized model. Despite this recent emergence of models based on matrix factorization techniques for CF systems, memory-based models remain foundational and are widespread implemented due to their simplicity, explainability, efficiency, stability and their ability to produce serendipitous recommendations compared to model-based approaches [23].

1.3 Evaluation Metrics for Recommender Systems

When it comes to evaluating recommender systems, several properties of the system need to be considered for system success. The *Recommender Systems Handbook* [65] outlines different properties that might need to be considered in evaluating the success of a recommendation task.

The vast amount of research in recommender systems is based on the assumption that a system that provides more accurate predictions will be preferred by the user.

For example, a system that predicts a rating for an item to a user to be 4, on a 5-scale rating, is better than a system that predicts the same item preference to be 3 if the user actual preference is 5. Since the focus in this dissertation research explores the recommendation algorithm, we will be using the predication accuracy metric to evaluate the success of an approach.

Prediction accuracy metrics in recommender systems can be divided into three broad classes:

- Measuring the accuracy of ratings predictions: this is applicable in systems that seek to predict the rating a user would give to an item. Root Mean Square Error (RMSE) and Mean Average Error (MAE) are the most used metrics.
- Measuring the accuracy of usage predictions: this metric applies for systems that try to recommend items to users that they may use (e.g., Netflix’s movie recommender predicting that the user will add these movies to the queue). Metrics used are Precision, Recall and Area Under the ROC Curve (AUC).
- Measuring the accuracy of rankings of items: for systems that order items according to the user’s preferences (ranked list). Metrics used to measure this property depend on whether we are trying to determine the correct order of a set of items for each user, and measure how close a system comes to this correct order, or we are trying to measure the utility of the system’s ranking to a user. In the former, the Normalized Distance-based Performance Measure (NDPM), Spearman’s ρ or Kendall’s τ metrics can be used while, in the latter one, the R-Score and Normalized Cumulative Discounted Gain (NDCG) metrics are commonly used.

1.4 Group-Based Recommender Systems

1.4.1 Recommendations to Groups

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 [38].

The first strategy merges the individual profiles of the group members into one group representative profile (Figure 6).



Figure 6: Recommender systems merge user models based on an aggregation algorithm to create a group user model. The group user model is utilized by the filtering technique to generate group recommendations that fit the target model.

The second strategy merges the recommendation lists or predictions computed for each group member into one recommendation list presented to the group (Figure 7).

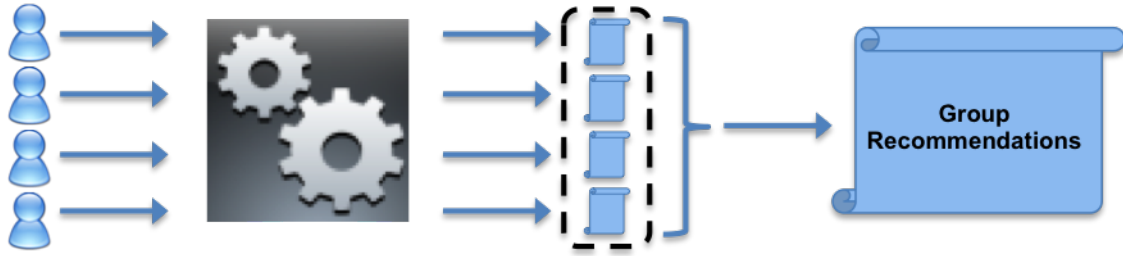


Figure 7: Recommender systems generate recommendations for every member of a group of size n . Recommendations are merged based on an aggregation algorithm to generate group recommendations that fit the group target model.

Both strategies utilize recommendation approaches validated for individual users, leaving the aggregation strategy as a distinguishing area of study applicable for group-based recommenders.

1.4.2 Group Modeling

When it comes to modeling a group 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.

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 [47] tried to find out what strategies people actually use as a group. She evaluated eleven strategies inspired from Social Choice Theory:

1. Plurality Voting: This approach selects the alternative with the most votes .
2. Average: Uses the statistical mean of the individual ratings.
3. Multiplicative: Ratings are multiplied. The larger the product, the more influential the preference
4. Borda Count: Counts points from items' rankings in the individuals' preference lists, with the bottom item getting 0 points, next one up getting one point, etc. When an individual has multiple preferences with the same weight, the averaged sum of their hypothetical count is equally distributed.
5. Copeland Rule: This strategy sorts the preferences according to their Copeland index: the difference between the number of times a preference beats the rest of the preferences (has higher weights) and the number of times it does not.
6. Approval Voting: A threshold is considered for the preferences' weights - only those weights greater than or equal to the threshold value are taken into account for the profile combination.
7. Least Misery Strategy: Takes the minimum of individual ratings. Thus, a group is as satisfied as its least satisfied member.
8. Most Pleasure Strategy: Takes the maximum of individual ratings. The higher the weight, the more influential the preference is for the group.
9. Average Without Misery Strategy: Averages individual ratings, excluding items with individual ratings below a certain threshold

10. Fairness Strategy: The top preferences from all the users of the group are considered. Items are ranked as if individuals are choosing them in turn.
11. Most respected person: Uses the rating of the most respected individual (dictatorship).

Variations of these approaches have been suggested and evaluated, such as a weighted, linear combination depending on social relationships among the group members.

When the prediction model for a group recommender system is a ranked list, the problem can be viewed as the computation of a consensus ranking of the alternatives, given the individual ranking preferences or predictions of the group members. Dwork et. al. [24] identified the following rank aggregation methods:

1. Borda's method: A "positional" method, which assigns a score corresponding to the positions in which a candidate appears within each voter's ranked list of preferences, and the candidates are sorted by their total score.
2. Spearman footrule distance: Measures the distance between ranked lists as the sum, over all elements of the lists, of the absolute difference between the ranks of each element in the individual lists
3. Markov chain methods: The states of the chain correspond to the elements in the lists to be ranked. The transition probabilities depend in some particular way on the given (partial) lists, and the Markov chain ordering is the aggregated ordering.

1.4.3 Group-based Evaluations

Group recommendation presents significant challenges in evolving best practice approaches to group modeling, but even moreso in both dataset collection for testing and developing principled evaluation approaches across groups of users. Early research provided more limited, illustrative evaluations for group recommender approaches (e.g., [50, 57, 49]), but recent work has been exploring more comprehensive evaluative techniques (e.g., [5, 18, 1]).

Broadly, evaluations have been conducted either, via live user studies or, via synthetic data set 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 [5], we adopt the approach of generating synthetic groups for larger scale evaluation.

1.5 Research Scope and Contributions

The scope of this research deals with group recommender systems where the user model is based on users' ratings of items relevant to the context of the domain of recommendation, and implements a collaborative filtering approach to filter items and produce predictions for items, or ranked lists of recommended items, for a group of users seeking a recommendation for items to consume as a group rather than individuals. We identify gaps in previous research in this context along the dimensions of the group model, the recommendation technique and the evaluation of such systems and evaluate several hypothesis with the objective of increasing the prediction accuracy.

A major gap in this research area relates to the difficulty of evaluating the performance of group recommendations with respect to prediction accuracy. Researchers have been employing synthesized groups created from single-user Collaborative Filtering systems to evaluate their proposed approaches to recommendations for groups. A major issue in this approach pertains to the fact that the actual group preference is unknown introducing a problem in establishing the ground truth, used for evaluation, as part of creating the training and testing data sets. Previous work adopting this approach evaluated the group recommendations independently against each group member by comparing the predicted ratings with the ratings observed in the test set of the individual group members. In this dissertation, we design a group-based testing framework that identifies groups from single-user user data sets based on given parameters for group structure and characteristics with ideal test points that reflect group consensus. This establishes a new approach to evaluate group recommenda-

tions as part of a large-scale analysis. We show the feasibility of this approach and compare the performance of some of the most widely implemented group modeling approaches using this evaluation technique to previous research adopting this approach to evaluation. Chapter 3 details the evaluation framework and Chapter 4 includes the experiments in this dimension.

Previous research in group-based recommender systems has mainly adopted filtering techniques validated for single-user recommender systems with little attention focused on adapting these techniques to account for the context of group as a first-class element in the recommendation. Along this dimension we first evaluate the performance of rating normalization on the prediction accuracy of the most common group modeling strategies in group-based recommender systems. We show that rating normalization results in higher prediction accuracy for the groups when employing an Average group model but not for group models based on Least Misery or Most Happiness. We also propose and assess new approaches to recommendation techniques that account for the fact that these recommendations are generated for a group rather than individual users. Using a nearest-neighbor recommendation technique we account for the group context by giving special consideration to common neighbors among group members in the prediction calculation rather than basing the neighborhood selection on the individual group members. We show that this approach increased the prediction accuracy for the group specially for groups with high inner similarity within the group members. Chapter 5 details the evaluations along the dimension of the recommendation technique.

Research in this context has focused on evaluating the strategies used to aggregate

the individual user models or the individual target models they have not explored the performance of integrating both aggregation approaches as part of the group modeling dimension. In this dissertation, we evaluate the integration of recommendations generated by the two main approaches to group modeling and show that combining the profile merging approach with the most happiness group model results in predictions with higher accuracy for the group. We also examine the integration of other elements in the group model such as a disagreement component that is based on item similarity and the known preferences of the group members for such items. We show that the applicability of this approach suffers in the identification step of similar items to the item targeted for evaluation with the constraint of a known preference for all the group members for such item. These evaluations are detailed in Chapter 6.

Case-Based Reasoning (CBR) approaches have been used in recommender systems as a part of the recommendation technique and group-based systems that have adopted this approach was also focused on the recommendation with little attention focusing on adopting such approaches as a part of the group model. We employ a group model that is based on a CBR point-of-view rather than an aggregation only perspective. We employ the group model of the retrieved case as a first class element in the recommendation. Our results showed that a CBR approach to group modeling increased the prediction accuracy for the groups where the case retrieval is based on the group context and group-to-group similarity. Chapter 7 details our CBR model and the evaluation results.

The hypothesis along these dimensions are outlined in chapters 4-7 and are numbered sequentially. The contributions of this dissertation are:

1. An evaluation framework for group-based recommender systems that allows the creation of synthesized groups with varying group contexts. The framework creates the groups and identifies ideal test point for creating training and testing datasets for the groups.
2. A tradeoff comparison in the choice of the baseline for evaluation using synthesized groups. We show that the choice of group model as the actual group preference affects the evaluation results and introduces another variable that needs to be accounted for and minimize its effect.
3. Rating normalization effect on prediction accuracy in memory-based collaborative filtering group-based recommender systems. We show which group models and group contexts benefit from rating normalization.
4. A neighborhood selection model based on the group context with a finer-grained weighting scheme. We show that neighborhood selection strategy that accounts for the group context and utilizes it in neighbor weighing increased the prediction accuracy for the group.
5. A hybrid group model that combines profile merging and recommendation aggregation and show which group aggregation models and group contexts benefit implementing this approach.
6. A disagreement model based on item similarity to the item considered for recommendation. We investigate the applicability for this approach and show that its feasibility is hindered with the constraint of identifying similar items rated by all the members of the group.
7. A case-based reasoning group model that is based on group context and group-

to-group similarity. We show the applicability of this model and the benefits realized for the group in terms of prediction accuracy and group contexts.

CHAPTER 2: GROUP RECOMMENDER SYSTEMS BACKGROUND

In Chapter 1 we introduced recommender systems and overview their categorization as well as group-based recommenders. In this chapter we overview related work to group recommender systems along the dimensions we outlined in the previous chapter and identify the open research questions we target in this dissertation.

Analyzing group-based recommender systems we identified three primary dimensions and include open research questions. Exploring the group recommendation space involves evaluation across a variety of such contexts. The context includes choices made about:

1. The underlying recommendation strategy (e.g., content-based, collaborative memory-based or model-based)
2. Group modeling for making recommendations — the group model used to model the group preference as a part of the recommendation (Recommendation Group Preference Model (RGPM)) (e.g., least misery)
3. Evaluation of the recommendations:
 - Determining actual group preferences for evaluative comparison to system recommendations — the group model used to model the group preference as a part of the evaluation (Actual Group Preference Model (AGPM)) (e.g., choice aggregation)
 - Choices about metrics for assessment (e.g., ranking, rating value).

2.1 Recommendation Technique

In Section 1.4.1 we identified that the recommendation techniques adopted in group-based recommender systems utilize recommendation techniques validated for single-user recommendation systems. For example, Demographic-based, Content-based, Collaborative Filtering, and Knowledge-based approaches have been applied in this context. In the following sections we review some of the research that utilized these techniques in a group-based context to help us understand the form of employing them.

2.1.1 Demographic-based Group-Based Recommender Systems

The INTRIGUE [2] travel recommendation system uses social and demographic information about the users to model the group as a set partitioned into a number of homogeneous subgroups where their preferences are separately represented. Another travel recommendation system that utilizes demographic information as well as content descriptions is presented in [25] and applies a recommendation technique that is based on the group members' tastes, demographic classification, and preferences for places they visited before. Another system that utilized demographic and content-based filtering is the FIT [30] TV program recommendation system. FIT recommended TV programs to family members provided their demographic information and their preferred watching times. The user profile was constructed by assigning the user to a relevant stereotype that includes content-based preferences. For any given time, the system calculates a probability of each household member watching TV and uses it to weigh a linear combination of genre preferences for the present users.

2.1.2 Content-Based Group-Based Recommender Systems

Content-based techniques have been implemented in the music, travel, and restaurant group recommendation domains. MusicFX [50] is a group preference arbitration system that adjusts the selection of music playing at a fitness center to best accommodate the preferences of the people actually there at a given time. Individual users provided ratings to different music genres on a 5 point scale. Users identified themselves to the system by swiping their badges. At any given time the ratings of the present users for each genre are added up and sorted in descending order (most popular category is first). A weighted random selection is applied on the top k genres to select radio stations played.

Another group recommendation system that applied content-based filtering in the music domain was introduced in [3]. Poolcasting [3] presented a web radio architecture where listeners could influence the sequence of songs played based on group customization and interaction. Poolcasting is modeled after home-parties, where participants can contribute, with their own records, to the pool of music and can, in turn, control which songs are played. In Poolcasting, any user can share their personal digital music library by adding their songs to the music pool, and can interact via a Web interface to evaluate the songs played and propose new songs to play. These interactions allow the sequence of songs played on each channel to be customized for the current listeners. The actual choice of music played is taken by a Case-Based Reasoning technique [13, 10] that combines knowledge about songs' associations and listeners' preferences. The system models song preferences by combining the user's

rating for a song with how many times the user played that song normalized according to the average listening behavior of that user.

The Flytrap [20] system applies content-based as well as knowledge-based filtering techniques. Flytrap selects music played in a shared space. The system has users preference data gathered from monitoring the users' music listening activities as well as knowledge data of how the different music genres relate to one another, how artists influenced each other and what kinds of transitions between songs people tend to make. Similar to MusicFX, users' presence is automatically detected in the room where the music is played. For each user in the room the system uses a voting mechanism to score songs in the playlist database according to the artist and genre preferences. A high vote is given to a song if it's an artist the user listened to previously. Songs the user has never listened to before receive positive votes if the genre is the same as, or similar by some degree to, music to which they had previously listened. Once the voting has completed, the votes are combined and then normalized into a probability random distribution across the entire database of songs. Songs that get more votes have a higher probability of being played. Songs that get few votes can still be played, but it's less likely.

Another activity people carry out as a group is selecting a venue for a dining experience. Pocket RestaurantFinder[49] is a system that helps users in that selection by recommending restaurants that will appeal to the whole group applying a content-based filtering technique. Users' profiles consist of their preferences to four different restaurant features as a well as a relative importance of each of these four categories of features.

2.1.3 Collaborative Filtering Group-Based Recommender Systems

PolyLens [57] is a group-based recommendation system that applied a user-centric, Collaborative Filtering recommendation approach in the the movie domain. PolyLens was designed to recommend movies for a group of users seeking recommendations to experience as a group. Users provided movie (item) ratings on a 5 point scale. The Nearest-Neighbor algorithm was used to predict ratings for unrated items for each member of the group. Berkovsky et al. [6] implemented a user-based CF approach for generating recipe recommendations based on an aggregated group user model. They evaluated these recommendations against real-life recipe ratings provided by families interacting with an experimental eHealth portal.

Other research implemented a model-based CF approach and evaluated group recommendations in the movie domain [5, 63, 1]. Chen et al. [18] also evaluated group recommendations in the movie domain by implementing a combination of user and item-based CF. Campos et al. [21] proposed a general, Bayesian Network-based model for group recommendation. Their proposed model is considered to be a collaborative Bayesian network-based group recommender system, where group ratings are computed from the past voting patterns of other users with similar tastes. They evaluated their approach in the movie recommendation domain.

From the research we overviewed it becomes apparent that researchers have adopted the individual-user perspective for the recommendation techniques used in group-based systems. They did not adapt the recommendation generation technique to account for the fact that these recommendations are created for a group of users rather

than individuals. Previous work that utilized Content-based or CF approaches to generate recommendations for the group did not consider the group context in generating the individual recommendations when using the recommendation aggregation approach. When we say group context we are referring to context information pertaining to group characteristics such as the size of the group and the similarity/dissimilarity among group members. Similarly, the group context is not considered in the recommendation calculation when generating recommendations for the group’s pseudo user in the case of profile merging. Content-based approaches use descriptions of the items rated to learn a relationship between the ratings of a single user and the description of the items rated while CF approaches use the rating of a set of people on a set of items to make recommendations but both utilize neighborhood approaches in identifying similar items or similar users. In both approaches limited attention has been given to neighborhood weighting using the group context. We believe that accounting for the group context in the recommendation calculation would result in better recommendation for the group in terms of prediction accuracy. We explore incorporating the group context in the recommendation calculation specifically when using a neighborhood-based CF. We utilize the group context when identifying the neighborhoods used in the prediction calculation (Section 5.2).

2.2 Group Modeling

Section 1.4.2 outlines the various strategies that have been applied as the group modeling strategies. This section overviews related work that applies these group modeling strategies as well as variants of these strategies.

2.2.1 Weighted Linear Combination

Most of the systems that were not focused on evaluating the group modeling strategy apply a linear combination or a weighted linear combination as the group aggregation strategy. Earlier group-based recommenders like MusicFX [50], Let's Browse [44], and PocketRestaurant [49] use a linear combination approach since the main research focus was to assess whether such systems might be socially accepted, or they constituted initial prototypes. Later systems like FIT and GAIN applied a weighted linear combination to aggregate individual preferences into a group preference model based on the probability of users being present and accounting for their preferences. In the TV4M system users assign weights to item features then the system selects features to include in the merged group profile based on total distance minimization. Features that meet the minimum distance are selected to be included in the group profile. The group weight for the selected feature is the average of the normalized weights of the individual group members. The Poolcasting [3] system also applies a weighted average as the group model, where the weight associated to each user depends on their satisfaction about the last scheduled songs. The system tries to achieve group satisfaction and guarantee more fairness by assigning a higher weight to the preferences of the listener that was less satisfied with the last played songs so that eventually every user gets to listen to some songs they like.

The main focus of the TRAVEL DECISION FORUM [37] is to help the group members, for each aspect of a travel vacation that the group members are planning, to arrive at a group preference model to which all members have agreed. The system

can be classified as one that recommends specific preferences for the group model rather than a specific item or items. Users have the option to choose the aggregation strategy that is used to merge the individual preferences. The average, median, or Automatic Mechanism Design that maximizes overall group utility and equity can be specified.

McCarthy et al. [51] evaluates how to aggregate individual user models to produce a group model used to generate recommendations in a critiquing-based, case-based recommender. They use the CATS [52] recommender system to evaluate three different aggregation methods used to assess the quality of a case, in the available case base, with respect to its similarity to a critiqued case and compatibility to the group model that contains the union of critiques of all group members. The first strategy is the *Weighted Average Group Model* that leverages a compatibility score for a case in terms of critiques, contained within the individual user model, of the user applying the critique with a compatibility score to the preferences of the other group members (the group minus the critiquing individual). The second strategy is the *Joint Group Model* which measures the compatibility of the candidate case with the current aggregate group model, and the third strategy is the *Average Individual Group Model* which is the average of the compatibility scores for each group member. They evaluated their approach by comparing the quality of the recommendations generated by the three proposed strategies across different sets of groups of four members with varying levels of inter-user similarity (similar, mixed, diverse). They reported higher quality recommendations to the individuals of the group when compared to single-user techniques.

2.2.2 Social Choice Theory

The voting strategy was applied in the Flytrap [20] system. For each user in the room the system uses a voting mechanism to score songs in the playlist database according to the artist and genre preferences. The individual votes are combined and then normalized into a probability random distribution across the entire database of songs. Songs that get more votes have a higher probability of being played. Songs that get few votes can still be played, but it's less likely.

In the PolyLens [3] prototype it was expected that most groups using the system would be small (2 or 3 members). The choice of a social value function was based on the assumption that the group's happiness was the minimum of the individual members' happiness scores. Predictions for items that were not rated by all group members were aggregated based on the Least-Misery strategy.

In Campos et al.'s [21] approach, the BN formalism is used to represent both, the interactions between group members and the processes leading to the final choice or recommendation. Their approach outlines how the most common aggregation strategies (*average, majority, least misery, most happiness*) might be encoded using Bayesian Network (BN) formalism. Results show that when combining uncertain information using BN at a group layer the best option is to take into account the difference between the posterior and prior probability to correct the prior bias, particularly in those situations where the least misery and the average strategies are used to merge individual ratings. Posterior probabilities are obtained by instantiating the ratings previously given by similar users and prior probabilities are obtained

by propagating in the network without evidence of ratings.

Senot el al.[70] attempted to determine the factors that influence the choice of an aggregation strategy for a group of users. They divide group recommendation strategies into three main categories based on the principles adopted for the conciliation of individual preferences regardless of the aggregation approach that each strategy applies. The three categories are majority-based, consensus-based, and borderline strategies. Majority-based strategies use most popular/shared interest categories or preferred items between group members, e.g. *Plurality Voting*, *Borda Count*, *Copeland Rule*, or *Approval Voting*. They allow satisfying the majority of the members of the group, even if the recommendation is extremely unsatisfying for the others. The Consensus-based strategies consider all group member preferences, such as averaging all users' preferences for each item/concept (*Additive*, *Multiplicative Utilitarian Strategy*, *Average without Misery*, *Fairness*). Borderline strategies consider only a subset of preferences, belonging to a subgroup of the group, based on the roles, or other criteria, identified within the group (*Least Misery*, *Most Pleasure*, *Dictatorship*). They compare the group profiles obtained by aggregating individual user profiles, according to representative strategies of the three aggregation categories, to the group profile acquired implicitly by analyzing group consumptions in the TV domain. They evaluated the *Utilitarian*, *Plurality Voting*, *Least Misery*, *Most Pleasure*, and *Dictatorship* strategies.

Baltrunas et al. [5] compares the effectiveness of individual and group recommendation across various group sizes (2,3,4 and 8) with varying degrees of inter-user similarity (random, high). Group recommendation lists are generated using several

aggregation strategies: *Least Misery*, *Average*, *Borda Count*, and *Spearman Footrule*. Garcia et al.'s [25] approach is based on three strategies that differ in how the individual preferences are combined to form a group preference model. The three strategies are: *aggregation*, *intersection* and *incremental intersection*. The *aggregation* strategy calculates the group preference for an item feature as the average value of the preferences of the users in that group for that feature. The *intersection* strategy finds the preferences that are shared by all the members in the group to establish group preferences. The *incremental intersection* is a weighted average of the most voted preferences among the users in the group, that are the preferences shared by the largest possible group of members. When analyzing which of the three techniques obtains the best recommendations resulting in satisfaction for the whole group, better results were reported with the *incremental* strategy.

2.2.3 Extending Social Choice Theory Group Modeling Strategies

Chen et al. [18] designed a system based on the framework of collaborative filtering. They use Genetic Algorithms (GA) to exploit known preferences of subgroups of the active group and predict possible similarities among group members. These similarities were used to weigh member contributions in item predictions. Their approach to predicting group preferences is based on having access to some item ratings for the target group as well as subgroups of the target group and individual group members' preference information. They use an item-based CF approach to identify items similar to the item for prediction. If the group did not provide a rating for these items a user-based CF was used to predict the individual ratings. Subgroup information was exploited using GA to assign weights in combining the individual

users' ratings into a group rating. Then item-based CF was used to calculate the final group rating for the target item.

Recio-Garcia et al.[63] proposed a method of making recommendations to groups based on existing techniques of collaborative filtering and taking into account the group personality composition. Their approach takes into account group member interactions using personal information about the conflict mode behavior of every group member. They evaluated their approach in the movie recommendation domain using three different variations of group recommendation aggregation strategies, *Minimizing Penalization*, *Average Satisfaction*, and *Least Misery*, taking into account the *Conflict Mode Weights* (CMW) of the group members. Another work that was evaluated in the movie recommendation domain is found in [5]. Baltrunas et al. [5] analyzed the effectiveness of group recommendations obtained by aggregating the individual ranked lists of recommendations produced by a collaborative filtering system.

Berkovsky et al. [6] investigated the use of aggregated group data in collaborative filtering recipe recommendations. They implemented four weighting models (*uniform*, *heuristic*, *role-base*, *family-log*) for aggregating individual data into family-based data. The uniform model weighs users uniformly, i.e., weight for every user equals 1. The heuristic model is role-based, where a role refers to a user's function within a family: applicant, partner, or child. A user's weight is defined solely by their role. The role-based model weighs users according to the activity of users in the same role across the entire community. The family-log model weighs users according to their activity in relation to other family members. Extreme case heuristics deal with extremely positive or negative preferences. The least misery heuristic assigns

a weight of 1 to the user who provided an extremely negative recipe rating, and a weight of 0 to the other family members. The most pleasure heuristic assigns a weight of 1 to the user who provided the extremely positive data, and 0 otherwise. They evaluated CF recommendations generated using the aggregated data against real-life recipe ratings provided by families interacting with an experimental eHealth portal. The results showed that the most appropriate family-based recipe recommendation strategy should aggregate individual user models, rather than individual recommendations, and weigh individual users according to their observed activity rather than according to predefined preferences.

Salamó's [67] work continues previous work [53, 52, 51] on group recommendation. They propose two new approaches for reaching consensus on the recommendations made for a group of users, *Statistical Dispersion* and *Individual Content*. For the *Statistical Dispersion* approach, they propose two strategies, *Mean* and *Purity*, based on the measurement of dispersion used in statistics and probability theory. The *Mean* strategy is defined as the sum of each group member's preferences to derive a central tendency of the preference space for the group. The *Purity* strategy measures the percentage of positive preferences among the whole set of preferences, made by the group, that are covered by the product. For the individual content approach they propose three strategies, completeness, logical sufficiency, and group sufficiency. The objective of the completeness strategy is to favor high scores while penalizing big differences between members. The *Logical Sufficiency* is a standard likelihood ratio statistic, which has been applied to measure rule quality of rule induction systems. This measure divides the proportion of positive examples, that are covered by a rule,

by the proportion of negative examples. They define it as the proportion of satisfied preferences of a group member by the proportion of their preferences that are not satisfied. Instead of analyzing the logical sufficiency of an individual member in relation to their satisfied or unsatisfied preferences, the *Group Sufficiency* measures the satisfaction of a member in relation to the satisfied preferences for the rest of the group and to the unsatisfied preferences for the individual in comparison with the whole group.

Amer-Yahia et al. [1] investigates how to efficiently compute group recommendations given a consensus function, especially in the presence of complex disagreement models. Their approach computes group recommendations by aggregating recommendations generated for individuals using a consensus function that combines relevance of the items for a user and disagreement between members of the group. A predicted rating is a weighted summation of the two component scores. User profiles are modeled as an item-rating matrix. They model relevance using existing strategies such as average and least misery. Disagreement is modeled using two alternative methods: *average of pairwise disagreements* or *rating variance*. The recommendation technique is based on Threshold Algorithms, more specifically, *Top-k*. They study optimization opportunities that exploit the dependencies between disagreement and relevance. They aim to reduce the number of disagreement lists that need to be maintained in order to achieve the best runtime performance.

Gartrell et al. [26] propose a group recommendation method that utilizes both social and content interests of group members. Their proposed group consensus function aggregated individual predicted ratings into a group predicted rating based on

factors related to the social ties, expertise, and interest dissimilarity among group members. They quantify these factors as the following group descriptors: social descriptor, expertise descriptor, and dissimilarity descriptor. The social descriptor is devised to measure the social relationship strength of a group. They quantify the social relationship strength of the pairwise member social link and categorize it into five different contact levels based on the average daily contact frequency between two members. The expertise descriptor measures the relative expertise of individual group members in the domain of recommendation. They categorize the expertise of an individual into five levels based on the number of movies that an individual has watched. The dissimilarity descriptor measures the preference difference among the group members using two metrics, *average pairwise dissimilarity* and *variance dissimilarity* [1]. The choice for a group decision strategy (*average, least misery, most happiness*) depends on the social relationship. The group consensus function proposed is a weighted combination of the predicted rating and disagreement depending on the social relationship strength. They also propose and evaluate a Rule-based group recommendation. First they define attributes that are used to construct the associative classification rules. These attributes are social strength, maximum group member rating, minimum group member rating, average group member rating, standard deviation of member ratings, average pairwise preference dissimilarity, average pairwise expertise dissimilarity, minimum expertise, maximum expertise, expert member identifier, and group rating. After defining these attributes, they use FP-growth to identify frequent item sets in the data. Quantitative association rules are generated from these frequent predicate sets. Using the strong association rules mined from the

data, they write classification heuristics that compute predicted group ratings for a movie given the individual group member ratings for that movie.

Group modeling strategies focused on the approach to model the group (profile merging or recommendation aggregation) or extending the baseline aggregation strategies identified in Section 1.4.2 either by using probability theory to assess satisfaction or weighing the individual user models or target models based on social relationships, interactions, or personality types of the group members. They did not examine the performance of a hybrid between of the two modeling approaches or modeling the group based on known group models in the collection of the information available to the system. A question that remains unanswered is how do these two approaches perform if they are used together rather than choosing one. In this dissertation we evaluate a hybrid model that utilizes both approaches. We evaluate the performance of several aggregation strategies in this hybrid model. The ability to exploit a large CF data set allows us to consider information extracted from the data set that can be reflected on the aggregation strategy used for group modeling without the need for any additional user information. Similarity information about items and users, drawn out from the data set, can be utilized in modeling the achievement of consensus among the group. We explore this novel approach as part of our group modeling hypothesis in Chapters 6 and 7.

2.3 CBR

A more limited number of researchers have explored using Case-Based Reasoning to model group decisions [60] or as a part of the recommendation technique [51, 61].

McCarthy et al. [51] looked at aggregating individual user models to produce a

group model that was used to generate recommendations in a critiquing, case-based recommender. They employed the CATS [52] recommender system to evaluate three different aggregation methods to assess the quality of a case, including similarity to a critiqued case and compatibility to the combined group critique model. To evaluate, they used synthetic groups generated from real-user preference data (34 trial subject profiles converted into a critique-based profile). Four-member groups were randomly generated (3 sets of 100 groups each), with varying similarity levels (*similar*, *mixed*, *diverse*). Each test group received three sets of recommendations, with each set containing one recommendation for each group member. Recommended cases were evaluated based on *compatibility* (# shared features) to the known *perfect case* of the individual group members. Group *compatibility* was measured as the average score. They reported an improvement in recommendation quality across the aggregation strategies for the *similar* groups when compared to the individual group member's own case choice, but not for the *diverse* and *mixed* groups, given preference diversity. In this work the CBR perspective was used as part of the recommendation technique in retrieving possible candidate items for the group to critique in a similar fashion to recommendations made to individuals and not adapted for groups.

Recent work from Quijano-Sanchez et al. employed CBR in several aspects of group-based recommendation [61, 60]. In [61] they evaluated a CBR solution to alleviate “cold-start” problems for group members. Cases represent previous movie recommendation events for groups. When a group seeks a recommendation and some group members are considered in cold-start, they find a previous case with users in similar group-roles who are not in cold-start. Ratings are transferred to user(s)

in the active group from corresponding users in the retrieved case, and then the updated profile is used in the recommendation process. In this work as well, the CBR perspective was used but not directly for group-based recommendations, as such. It was used to address the cold start problem for individual group members.

Quijano-Sanchez et al. also applied CBR directly in modeling group decisions for group recommendation [60]. Employing user-user similarity, each user in the active group is aligned with exactly one user in the case group. Similarity is measured by comparing group members on their age, gender, personality, ratings and the degrees of trust between members of each group. Group to group similarity is calculated as an average of these one to one similarities. All mapping combinations between the active group and the case groups are checked and the top n cases and mappings are then used in the recommendation process. Using item-item similarity, they map contributions in choosing the selected item from each group member in the case. Predictions for items are based on accumulating the similarities to the selected items in similar cases, weighted by the degree of similarity to those cases. If group and case are of different sizes virtual users are added to that group. An overall improvement in success rate was reported for their CBR approach on a data set of 100 cases, which were individually crafted with review input by a panel of experts. As far as we are aware, this is the only group recommender work to date that investigates holistic retrieval and reuse of entire previous groups as cases. This served as an inspiration for our own exploration of the space.

Similar to the strategy of adopting recommendation techniques in group-based recommenders, systems that utilized a CBR approach have mainly been used as a

recommendation technique rather than a group modeling strategy. To our knowledge only the work of Quijano-Sanchez et al. [60] explored the use of CBR as a part of the group modeling strategy. In their approach they combine many subtle aspects of the group recommendation process all at once with some requiring imputed value support. Still this leaves some questions unanswered in this context. We raise the questions of how would a less complex technique for case retrieval perform in terms of prediction accuracy in a group-based recommender and rather than a single case match how can a k -nearest neighbor approach increase the accuracy of predictions. We address these questions in Chapter 7.

2.4 Evaluation in Group Recommendation Systems

Two classic approaches to evaluating the accuracy of predictions in a single-user recommender system can be categorized as either conducting a user study or testing data sets, which are utilized by dividing the data set into learning and test sets. Similar approaches have been adopted in evaluating group-based recommenders. To evaluate the prediction accuracy of the system, a user study can be conducted by gathering groups of users and establishing their preferences for items in the data set. These preferences are the ground truth for those groups. The system generates predictions for those items and the error of the prediction accuracy can then be measured. When it comes to group-based recommenders testing data sets are not readily available. Some approaches tried to work around this by creating synthesized groups from the individual user data sets and ran evaluations using these groups. The next two sections overview related work based on the evaluation approach and metrics adopted. Section 2.4.1 focuses on work conducting user studies. Section 2.4.2

covers work that employs synthetic data sets generated from existing single-user data sets (typically MovieLens).

2.4.1 Evaluation with User Studies

Masthoff [46] employed user studies, not to evaluate specific group recommendation techniques, but to determine which group aggregation strategies people actually use (Section 1.4.2). Thirty-nine human subjects were given the same artificial individual ratings of three people for a set of video clips. The individual ratings were chosen in a way that would enable differentiating between the strategies she expected the subjects to use. 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, they were asked why they made that selection. Results indicated that people particularly use the following strategies: Average, Average Without Misery and Least Misery. She also reported more accurate predictions when rating normalization is applied as well as quadratic ratings (ratings are not linear).

PolyLens [57] evaluated their system by conducting a nine-month field trial. They deployed their approach as an experimental feature of MovieLens, a movie recommendation site. They allowed any MovieLens user to create a group and invite users to join the group. Overall they had 338 groups with 819 members and group sizes ranging from 2 to 4 with the majority of the group (76%) being of size 2. They had a few number (12) of groups with a size larger than 4 but they did not provide the details for those groups. They conducted a field trial with the systems and reported evaluation results that were based on qualitative feedback from users and observations of changes in user behavior with the new feature being added. They did not

evaluate their system in terms of prediction accuracy or how satisfied the users were with the quality of recommendations.

Amer-Yahia et al. [1] ran a user study using Amazon’s Mechanical Turk users to evaluate the prediction accuracy of their prediction aggregation strategies. The users were used to form groups and perform judgments on group recommendations. The study recruited 50 users where they were asked to provide their movie preferences on a subset of the MovieLens items. The subset was selected taking into account item diversity and a user’s familiarity of the items. Groups were formed of sizes 3 and 8 to represent small and large groups. They also varied the cohesiveness of the group members (*similar* 0.9, *dissimilar* 0.28, *random* 0.71) for a total of 6 groups. They established an evaluation baseline by generating a recommendation list using four implemented strategies. The resulting lists were 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 a Normalized Discounted Gain (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. They reported that the *Least Misery* strategy resulted in the best performance for both small and large groups of similar users. They attributed this to the fact that as a group activity of similar users, the objective is usually to agree with the person who has the harshest opinion. The aggregation strategies that account for disagreement among the group members performed worse since there was hardly any scope of disagreement among groups of similar users. These strategies performed best

for the dissimilar groups, especially the large ones. For small groups, *Least Misery* worked best, whereas, for large groups, there was no significant difference between the different strategies.

Other work considers social relationships and interactions among group members when aggregating the predictions [26, 18, 63]. 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. [63] described a group recommender system that takes into account the personality types of the group members. Their user study included 70 students that were presented with a list of 50 heterogeneous movies selected from the MovieLens dataset. The students were joined in groups, of size 2,3,4 and 5, to discuss choosing a movie they would be going to see. They were asked to decide which 3 movies (favorite set) from the MovieLens dataset they would watch together. They used a simple evaluation measure that counts how many of the movies in the favorite set are in the k best items retrieved by the recommender. They reported that *Average* and *Least Misery* with personality weighting reflected improvements in the accuracy of the recommendations.

Gartrell et al. used real-world group-based user studies to evaluate the effectiveness of modeling social, expertise, and dissimilarity group descriptors in the prediction aggregation strategy. They recruited 10 groups, comprised of 32 individuals, to participate in the user studies. For each group, individual group members were asked to describe his or her social relationships with other members in the group. The social relationships between two peers mainly contained the following four types of relationships: couple, close friends, acquaintances and first acquaintances. The rela-

tionship strength was sequentially decreasing. They selected 20 movies, out of the top 250 popular movies from IMDB³, across 10 different movie genres, selecting two movies from each genre. They used movie trailers to capture participants' preferences provided as ratings on a scale of 1 to 5 for these 20 movie trailers. In order to capture the group ratings for the movies, the participants were asked to return to their groups and begin discussion about these 20 movie trailers. They evaluated the effect of the group's social and expertise characteristics on the group decision process. They reported that users with higher expertise levels carried a higher weight in the group decision process. On the other hand, they investigated how social relationships can affect a group's decision. Results indicated that a group with a strong social relationship tends to maximize the satisfaction of a user in the group, while a group with a weak social relationship tends to minimize the misery of a user in the group. In evaluating their group consensus approach, that combines the social, expertise, and dissimilarity descriptors, they reported great improvement in terms of overall prediction precision.

Berkovsky and Freyne [6] carried out an evaluation using a data set of explicit ratings for recipes, gathered during a study observing interaction of families with an experimental eHealth site. Evaluation consisted of 170 users where 108 of them belonged to a family group with size ranges between 1 and 4. They implemented a memory-based recommendation approach. Evaluations compared the accuracy of the two group-based recommendation strategies (aggregated models and aggregated predictions), four weighting models (aggregation strategy), and assessed the impact

³<http://www.imdb.com>

of switching hybridization, extreme case heuristics, and group characteristics on the performance of group recommendations. Their recommendation for a group-based recommender system should aggregate individual user models of the group members into group-based models, and weights assigned to individual user models should reflect the observed importance of users.

2.4.2 Evaluation with Synthetic Groups

Related to aggregation strategies used in CF algorithms, Salamó et. al. evaluated several strategies used to aggregate satisfaction of the individual preferences of group members for an item in the case base; *average*, *Least Misery*, *Most Happiness*, *Multiplicative*, and *Borda Count*. They utilized synthetic groups generated from user profiles of critique preferences from a critiquing-based travel recommender system (CATS). They had 34 individual profiles which they used to generate groups of various sizes (3, 4, 6, and 8) with 300 groups forming 3 sets of 100 groups, where each set was made up of members with certain similarity characteristics (*similar*, *diverse* and *mixed*). The *Similar* group-set contained groups of users with a high average pairwise similarity in the range of 0.53 to 0.82. The *Mixed* group-set contained groups of users with mid-range pairwise similarities in the region of 0.4. Finally, the *Diverse* group-set contains groups of users with average pairwise similarities from 0.04 to 0.28. For each test group, they generate recommendations across these strategies. They evaluate the prediction accuracy of a single top recommendation, and a final recommendation list of 5 products, by comparing the average similarity of the recommendation product to the “Perfect Products” of the group members. Their results indicate that Multiplicative, Borda, and Average strategies perform best across all

group sizes.

Baltrunas et al. [5] 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 0.456, random 0.132). They generated 1000 groups for each condition. 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. The group recommendation lists were generated using several aggregation strategies (*Least Misery*, *Average*, *Borda Count*, and *Spearman Footrule*) to merge the individual recommendation lists for the group. They compared the effectiveness of the group recommendations generated using these strategies with that of the individual recommendations as well as a random aggregation strategy. They reported that varying the group size, the variation of the effectiveness of the group recommendations is not large for groups of size 2, 3, and 4. They also noted that when increasing the group size the effectiveness of the group recommendations tends to decrease only for randomly generated groups while for groups with high inner similarity, as the group size increases the effectiveness increases as well (the recommendations for groups with 8 members have the largest effectiveness). They couldn't

assert a clear winner between these strategies since the best performing method in each evaluation depended on the group size and inner group similarity.

In recent work, De Pessimier et al. [22] evaluated the two group-based recommendation strategies for various recommendation algorithms considering the size of the group. Inspired by [5], they conducted an offline evaluation, but considerably smaller in size (30 groups per size). Employing the nDCG metric and evaluating the effectiveness of a ranked list by averaging the individual nDCG of the group members. In their evaluation they showed that the effectiveness of the recommendations decreased as the group size increased. They also reported that the choice of group recommendation technique should depend on the underlying algorithm employed to generate the individual recommendations as they were not able to designate an overall winner.

Using a group recommender for tourist activities Garcia et al. [25] evaluated three methods for aggregating individual preferences (*aggregation, intersection, incremental intersection*) into a group preference profile. Refer to Section 2.2 for a description of these approaches. The dataset used was composed of 60 individual user profiles containing general preferences, demographic data, visited places and the user's degree of satisfaction when visiting these places. They randomly generated synthetic groups with sizes varying between 2,3,4,5 and 6. They did not report on how many groups were generated. They compared the quality of recommendation lists of 10 items generated using these approaches for the group. Quality was measured using the average and standard deviation of the utility of a recommendation list over all the group members. Their results show that the utility on average is similar across all group sizes. They repeated their evaluation using a sparser dataset and results

were similar. Comparing the dispersion (standard deviation) of the utility reveals that the *incremental intersection* approach produces recommendations that result in all members of the group being equally satisfied. The *incremental intersection* approach closely resembles the *plurality voting* aggregation approach, where the group preference profile is made up of the weighted average of the most vote preferences among the group members.

In addition to the Amazon Mechanical Turk user study evaluation, Amer-Yahia et al. [1] also evaluated their approach using synthetic groups generated from the MovieLens data. These synthetic groups were used to measure the efficiency of different strategies centered around a top-k threshold 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 between 3, 5, and 8. They generated 1 group for each evaluation condition (12 total). They reported that disagreement between group members impacted the quality and efficiency and could be exploited to increase the effectiveness of the group recommendations.

Chen et al. [18] also used simulated groups and addressed the sparsity in the 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. Group recommendations are determined by aggregating the

group's rating of similar items through the item-based collaborative filtering algorithm. If group ratings of similar items are not available, the proposed methodology uses the GA to learn the preferences of groups based on the ratings of its subgroups. They reported that the proposed method can provide reasonable and high-quality group recommendations compared to an approach that does not utilize GA.

Focusing on how individual user models can be aggregated to produce a group model used to generate recommendations in a critiquing-based, case-based recommender, McCarthy et al. [51] evaluated three aggregation strategies (Section 2.2) using synthetic groups generated from real-user preference data. The dataset used included 34 preference profiles of trial subjects that were converted into a critique-based profile. Groups containing 4 members were randomly generated to form 3 sets of 100 groups each. Each set of groups was made up of members with certain similarity level (*similar*, *mixed*, *diverse*). The *similar* groups contained users with a high average of pairwise similarity in the range of 0.67 to 0.84. The *mixed* groups had users with a mid-range pairwise similarity in the region of 0.4 while the *diverse* groups contained users with average pairwise similarities less than 0.25. For each test group, three sets of recommendations were generated with each set containing one recommendation for each group member for a total of 4 recommendations. They evaluated the recommended cases based on their *compatibility* (number of shared features) to the known *perfect case* of the individual group members. The group *compatibility* was measured as the average of these compatibility scores. They reported an improvement in the quality of the recommendation across the three aggregation strategies for the *similar* groups when compared to the individual group member's own case choice.

This was not true for the *diverse* and *mixed* groups since the preferences combined involved compromising more diverse preferences.

Utilizing the synthesized groups approach to evaluation and using the single-user data sets for the evaluation of group-based recommenders is not as straight-forward as in the single-user systems. A major question that then must be addressed in evaluating group recommender systems using this approach, is how to establish the actual group preference in order to compare the accuracy of the system predictions. In other words, how do we establish ground truth for the synthesized groups? Previous work has mainly adopted the *average* aggregation strategy to model the actual group preference. Evaluations are done using two approaches. The first approach takes the average of the individual preferences as the actual group preference and evaluating the prediction accuracy of the system against that average. The second approach evaluates predictions against the individual preferences and then averages the evaluation results, whether they were evaluating a predicted rating or a ranked list of recommendations. For example, one approach that evaluated a ranked list of items, the aggregated group ranking, is compared to the individuals' optimal rankings that they derived from the ratings in the test set [5]. In this approach the relative ranking might be correct, but the users might not actually prefer these items. These are some approaches to handle this issue, but it is our intuition that there is still an open issue here and the possibility exists that, as you make complicated models of the actual group preferences, you are moving away from the actual group preferences. We address this issue in Chapter 4.

2.5 Research Problems

As mentioned in Chapter 1, CF had a foundational role in the advancement of single-user recommender systems. CF suffers when the user-rating data is sparse. Evaluations in single-user recommender systems have taken advantage of publicly available CF data sets to analyze various dimensions of this approach in that context. When it comes to group-based recommender systems, such data is not generally available on a significant scale. Evaluations of the CF approach in the context of group-based recommenders are very limited and small-scale. In our work we address the issue of sparsity in the user-rating matrix by utilizing synthesized groups identified in publicly available CF data sets with ideal test points that illustrate group consensus. We are able to provide extensive evaluations of some of the baseline approaches, validated in the single-user context of recommendations, as well as provide some novel extensions that consider the group context in the recommendation filtering phase.

Group modeling strategies that extended the baseline aggregation strategies identified in Section 1.4.2 focused on either probability theory to assess satisfaction or weighing the individual user models or target models based on social relationships, interactions, or personality types of the group members. The ability to exploit a large CF data set allows us to consider information extracted from the data set that can be reflected on the aggregation strategy used for group modeling without the need for any additional user information. Similarity information about items and users, drawn out from the data set, can be utilized in modeling the achievement of consensus among the group. We explore this novel approach as part of our group modeling

hypothesis in Chapter 6.

Utilizing the synthesized groups approach to evaluation and using the single-user data sets for the evaluation of group-based recommenders is not as straight-forward as in the single-user systems. A major question that then must be addressed in evaluating group recommender systems using this approach, is how to establish the actual group preference in order to compare the accuracy of the system predictions. In other words, how do we establish ground truth for the synthesized groups? Previous work has mainly adopted the *average* aggregation strategy to model the actual group preference. Evaluations are done using two approaches. The first approach takes the average of the individual preferences as the actual group preference and evaluating the prediction accuracy of the system against that average. The second approach evaluates predictions against the individual preferences and then averages the evaluation results, whether they were evaluating a predicted rating or a ranked list of recommendations. For example, one approach that evaluated a ranked list of items, the aggregated group ranking, is compared to the individuals' optimal rankings that they derived from the ratings in the test set [5]. In this approach the relative ranking might be correct, but the users might not actually prefer these items. There are some approaches to handle this issue, but it is our intuition that there is still an open issue here and the possibility exists that, as you make complicated models of the actual group preferences, you are moving away from the actual group preferences. We address this issue in Chapter 4.

Another question that needs to be answered in evaluating recommender systems concerns the evaluation metrics used. Section 1.3 details metrics used in evaluating

single-user recommender systems. These metrics are also applicable in evaluating group-based recommenders and we utilize them in our evaluations.

CHAPTER 3: GROUP TESTING FRAMEWORK

To assess the quality of individual user recommendations, researchers commonly utilize offline evaluations that employ readily available substantial data sets (e.g., Netflix prize⁴, MovieLens⁵). This kind of approach can be used to repeatedly conduct large scale evaluations of proposed techniques. However, when it comes to group-based recommender systems such datasets are not readily available. Generating group-based data directly requires extra overhead in recruiting the groups together and getting them to cooperate and interact towards a common goal at the same time. To address scalability in evaluation, researchers have been utilizing synthetic groups, generated from single-user data sets, to evaluate various approaches to group recommendations [67, 5, 1, 25, 18]. The aim here is to develop a standard that has the highest probability of success based on the analysis of the individual characteristics of “real” subjects. Since interactions between individuals are fluid in nature and can always be highly variable having a “real” group versus synthesized can not affect the accuracy of the analysis. We adopt this approach of generating synthetic groups for evaluating our proposed approach to group-based recommendation.

In order to analyze the performance of a group-based recommender system we propose a novel Group Testing Framework that allows for the creation of synthesized

⁴www.netflixprize.com

⁵www.movielens.org

groups which can be used for testing in memory-based CF recommenders. In this chapter we give an overview of our proposed Group Testing Framework and in the following chapters we report on the evaluations we conducted using this framework. We have developed a group testing framework to support the task of evaluating various group recommender approaches. Given the availability and wide use of datasets of recommender systems based on individual user data the framework is used to generate synthetic groups from this type of data. The groups can be parametrized to test different group contexts. This enables exploration of various parameters of group characteristics such as group diversity.

3.1 Group Testing Framework

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. Figure 8 depicts the components of this framework.

3.1.1 Group Model Descriptors

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. We note here that the group model for evaluation could include inter-group constraints (diversity across groups) as well as intra-group constraints (similarity within

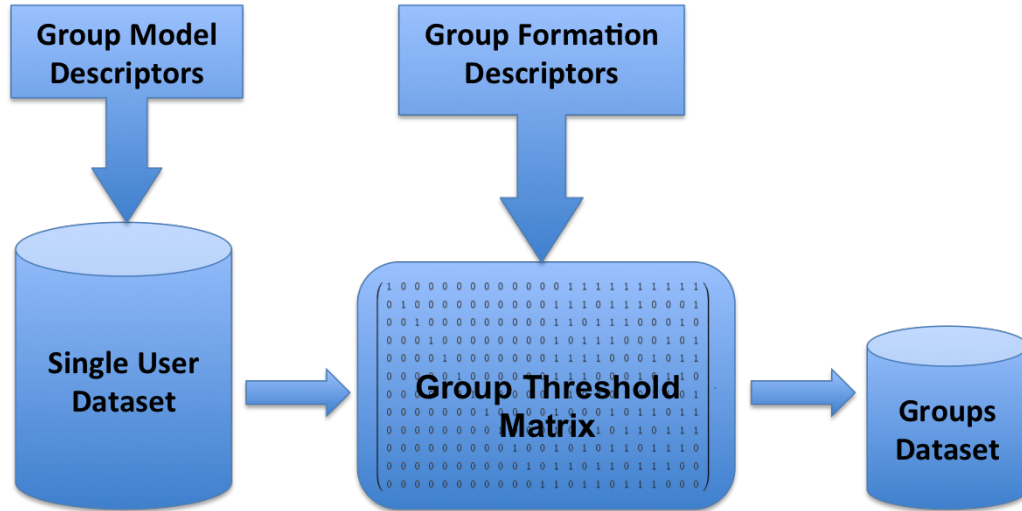


Figure 8: Group Testing Framework Architecture.

groups).

Gartrell et al. [26] 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. We refer to these as *group model descriptors*, and *group formation descriptors*. Some of these group descriptors that can reflect group structure are user-user correlation, and 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. We choose this bit-set notation to effectively exploit all possible combinations of groups. Checking all possible combinations of users can be computationally extensive. For example, given a data set of a 100 users, there are 4950 possible different combinations of groups of two. If we wanted to create groups of five the different possible combinations goes up to 75,287,520. This number exponentially increases as the data set and group sizes increase. The bit-set notation is an extremely efficient approach to model this generate-and-test approach.

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-wise or column-wise operations provides an efficient foundation for a generate-and-test approach to creating candidate groups from pairwise group descriptors. 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.

3.1.3 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. Baseline pairwise *group formation 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. The group formation mechanism applies the set of *group formation 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.

To address the issue of modeling actual group preferences for evaluating system predictions, the framework needs to establish “ground truth”. We realize this by identifying groups where all group members gave at least one co-rated item a rating that satisfies the *AGPM descriptor*. The *AGPM descriptor* can be defined as an exact rating overlap, rating within a certain standard deviation or as a threshold for a group average. In the first set of evaluations, where we utilize this framework, we tune it 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 did not eliminate “universally popular” items, but enough test items are identified that such items did not make a significant difference.

The intuition behind modeling the *AGPM* as an exact rating overlap comes from the fact that the satisfaction of an individual is likely to depend on that of another

individual in the group (emotional contagion), as observed in [48]. Since, individually, group members share the same preference for an item, assuming that their satisfaction as members of a group is equal to their individual satisfaction provides us with a principled evaluation point. This evaluation is also ideal for group recommender systems where the satisfaction of group activity is based on the satisfaction of the members individually. 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 a subset of these identified common item or items for that group.

3.2 Group Dataset

The data we use to ground the testing framework is available from the GroupLens research by University of Minnesota that contain movie ratings made by movie goers. Three datasets of different sizes (100K, 1M, 10M ratings) are available which have 100 thousands ratings from 1,000 users on 1,700 movies, 1 million ratings from 6,000 users on 4,000 movies, and 10 millions ratings with 100 thousands tags from 72,000 users on 10,000 movies respectively. The dataset also contains information about the movies and users' demographics.

We first utilize the 100K ratings dataset. Groups were created based on the different thresholds defined for the group descriptors. By varying the thresholds of the group model 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 was the effect 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 4.2.2.1. To answer this

question, we needed to vary the threshold for the similarity descriptor and the size of the group.

The model descriptors we set are the similarity and number of common rated items. We define this similarity as the inner similarity of the group members calculated as the average similarity between any two users a, b belonging to group G as defined in Equation 1. We defined three similarity levels: high, medium, and low similarity. Table 1 details the similarity thresholds that were defined.

Table 1: Degrees of group similarity

Similarity level	Definition $\forall a, b \in G$
High	$w_{ab} \geq 0.5$
Medium	$0.5 > w_{ab} \geq 0$
Low	$0 > w_{ab}$

To address the amount of trust to be placed in a correlation with a neighbor we set the threshold for the number of common rated items to 5 this establishes a level of significance of the calculated similarity correlations. So we only consider user pairs that have at least 5 common rated items. For this MovieLens dataset used, we have a total of 444,153 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 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

Table 3: Group threshold matrix statistics

		2	3	4	5
High Similarity	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	Total combinations	192,522	30,379,236	3,942,207,750	434,621,369,457
	Valid groups	192,522	17,097,527	875,027,262	241,169,045
	Testable groups	187,436	11,482,472	310,274,656	62,016,716
Low similarity	Total combinations	95,739	7,074,964	421,651,608	21,486,449,569
	Valid groups	95,739	1,641,946	6,184,151	5,869,889
	Testable groups	87,642	470,257	283,676	41,827

The group formation descriptors we set are size, AGPM and the number of test groups for each condition. For the size we varied the group sizes from 2 to 5. For the *AGPM* we set it to an exact overlap and we wanted to have 5000 groups for each size and similarity combination (e.g. 5000 groups of size 2 and with high inner group cohesiveness). To make sure that the thresholds we set for the group descriptors are relevant we first checked their applicability to generate the groups needed. Table 3 reflects the GTM group generation statistics for the underlying dataset used in this

run. The total combinations field indicates 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 3.1.3 and satisfies the *AGPM* (exact overlap). These numbers confirm that as the size of the group increases, the number of combinations the implementation has to check increases significantly. We can also see that the number of possible testable groups is large in comparison to the number of groups used in actual user studies.

Once we have identified the possible groups we needed to generate the testing and training data. A common evaluation approach in this domain is the leave-one-out approach, we adopt this strategy for evaluating the accuracy of a predicted rating. For each randomly selected group a test item is randomly selected from the set of commonly rated items that satisfy the *AGPM* threshold. For each group, one item has been identified as the test item. To generate the training set for each group we take out the rating of that test item from all of the group members' profiles. For example, given a group of $\{user_1, user_2, user_3\}$ and possible test item IDs of $\{50, 100, 315\}$. Assuming that the randomly selected item ID is 50, when calculating the prediction of item 50 for $user_1$, only the ratings of item 50 for users $\{user_1, user_2, user_3\}$ are taken out of the data set to form the training set for that group. Then predictions for that group and item are generated using this training set and so on for the other groups and test items. Overall, for this criteria, the group dataset contains 60,000 group and test item pair we refer to this dataset as Group Dataset 1 (GD1).

Since we are interested in evaluating our recommendation approach for both profile merging and recommendation aggregation, we wanted to ensure that the same training set was used to generate all the predictions for that group. We created a training and testing set for each group based on the test set of the group. We first created the profiles of the pseudo users for each group by merging the individual group members' ratings based on the average aggregation strategy. For each item rated by one or more group member, the rating for the pseudo user would be the average of the ratings based on the number of the group members that rated it.

Once the profiles for the pseudo users were created we can add the pseudo profile to the original data set to include the new pseudo user. This ensured that the same training set was used to generate predictions for the group across all the evaluated techniques, both merging profiles and merging recommendations. To create the training set for each group, we started off with the original MovieLens dataset. We then added the profile of the pseudo user of that group to the dataset. We then took out the ratings of the test items identified for that group from each of the group member's profiles and the pseudo user. In other words, the training set for each group is the original MovieLens dataset plus that group's pseudo user profile minus the ratings for the test item(s) for that group, for each of the group members and the pseudo user of that group.

The second group dataset we generate is based on the 1M MovieLens dataset. We refer to this dataset as Group Dataset 2 (GD2). We define the same group descriptors thresholds as the previous run only changing the *AGPM* and the number of groups generated. We set the *AGPM* to a commonly rated item among the group members

with no restriction on the ratings and we set the number of groups to be generated to 100. For each group criterion (group size / similarity level) we randomly create 100 groups for a total of 1200 unique groups. We placed an additional constraint on group formation that requires a valid group to have at least 3 items that were rated by all of the members of the group. This constraint provides for a minimal group evaluation baseline across those items. From these items we identify test items for each group.

To generate training and testing sets and to deal with the issue of disparity in profile sizes between group members, we employed a training/test set approach to split based on individual profile sizes within groups. For each group, we identified the commonly rated items among the group members. Then we checked if that set is larger than 40% of the smallest group member's profile size. If it was smaller, then those items would be the testing set for that group. If it was larger then we randomly select items from that set, not exceeding 40% of the smallest group member's profile size, to compose the testing set for that group. We do this to ensure that for each group member we have a majority of their original profile as part of the training set with as many test points as possible.

Once the test items for each group were identified, we created a training and testing set for each group in a similar fashion to how the training/testing sets are created in GD1 including the group's pseudo user profile. This ensures that the same training set for each group is used to generate all the predictions for that group. Table 4 outlines the number of test items identified for each group testing category. For the 1200 groups created, we identified 10,543 group/test item pairs. The training set for

Table 4: Number of test items across group sizes and similarity levels

	2	3	4	5
High	1367	493	419	539
Medium	2156	1116	756	496
Low	1896	572	389	344

each group is created by taking the original MovieLens dataset and then removing the ratings of the test items identified for that group from each of the group members' profiles. In other words, the training set for each group is the original MovieLens dataset minus the ratings for the test items for that group, for each of the group members. For each evaluation we conducted we used the same dataset across the different models being evaluated this preserves the assumption that our observations may be considered as random draws from the same probability distribution and are considered Independent and Identically Distributed.

CHAPTER 4: EVALUATION IN GROUP RECOMMENDER SYSTEMS

In order to address the issues we identified in our research of group-based recommender systems our first task was to explore ways of utilizing the publicly available datasets for evaluating CF approaches in this context. The previous chapter detailed the group testing framework that scans the individual-based recommendation dataset and is able to create synthesized groups, based on given constraints and group characteristics, as well as identify some ideal test points for the generated groups. We used the generated group test dataset to evaluate some representative group aggregation strategies (Least Misery, Most Happiness, Average) in aggregating individual predictions into group predictions as well as tradeoffs in evaluation strategies in this context. We implemented a memory-based CF, as described in Section 1.2.1. For evaluating the performance of the group aggregation strategies we measured prediction accuracy, using the MAE and RMSE metrics (Section 1.3), and compared our results to previous work that evaluated these aggregation strategies. For comparing evaluation strategies we analyzed the impact of the choice of a model as the actual group preference on the prediction accuracy using the RMSE metric.

4.1 Motivation

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 [64] 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.

Previous work by [5, 18, 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. [18] used a baseline aggregation as the ground truth while [5] compared 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.

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 initial evaluation approach addresses these issues. We evaluate the performance of three representative group aggregation strategies (average, least misery, and most happiness) [46] 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.

Our initial 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.

4.2 Group Characteristics and Structure

4.2.1 Hypothesis

H1 - A group’s cohesiveness and size has a direct effect on the prediction accuracy of the aggregation strategy applied in aggregating the individual predictions, generated using a memory-based CF, into a group predicted rating.

4.2.2 Evaluation Setup

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 was the effect 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 4.2.2.1.

4.2.2.1 Baseline Collaborative Filtering

Collaborative Filtering played a significant role in the advancement of recommender systems. CF requires a substantial amount of data in order for it to generate qual-

ity recommendations. Availability of data sets in this context contributed to a vast amount of research focused on evaluating and improving this approach for recommendation generation in single-user recommender systems. The difficulties of generating such data for group-based recommender systems have hampered research focusing on evaluating this approach in this context. Utilizing our group testing framework, we are able to lead large scale group-based evaluation of this approach. Given this foundational role of CF, we implement the most prevalent memory-based CF algorithm, the neighborhood-based CF algorithm [33, 64]. 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:

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

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 :

$$p_{ai} = \bar{r}_a + \frac{\sum_{b=1}^n [(r_{bi} - \bar{r}_b) \cdot w_{ab}]}{\sum_{b=1}^n w_{ab}} \quad (2)$$

Herlocker et al. [33] 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 chose 50 as the minimum neighborhood size for each member of the groups we considered for evaluation. Breese et al. [9] 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. This resulted in eliminating groups that had at least one member not having 50, or more similar users with a Pearson correlation higher than 0.5, with at least 5 co-rated items.

4.2.2.2 Group Aggregation Strategies

Previous work in group modeling focused on strategies inspired by Social Choice Theory. For this evaluation we choose three representative strategies: average strategy, least misery, and most happiness. The average strategy has been implemented as a baseline strategy in many group-based recommenders. We choose the Least Misery and Most Happiness since they cover extreme cases in terms of group consensus and satisfaction.

- 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 r_{ai} be the rating of user a for item i , then the group rating for item i is computed as follows:

$$Gr_i = \frac{\sum_{a=1}^n r_{ai}}{n} \quad (3)$$

- Least Misery Strategy: this aggregation strategy is applicable in situations

where the recommender system needs to avoid presenting an item that was highly 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 for any given item among group members and computed as follows:

$$Gr_i = \min_a r_{ai} \quad (4)$$

- 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:

$$Gr_i = \max_a r_{ai} \quad (5)$$

Previous group recommender research has focused on several group aggregation strategies for combining individual predictions. We evaluate three of the most commonly implemented group aggregation strategies outlined here 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 Results

We investigated the relationship between the group's coherence, size, and the aggregation strategy used. For statistical analysis we employed the ANOVA test and the Student's *t-test*. In this evaluation we used ANOVA to compare the different models for each group size and group cohesiveness level. The ANOVA statistical model assures that the differences in the models are significant. We use ANOVA for Repeated Measures with Benferroni adjustment and a *p* value less than 0.5. For this

evaluation we examined the difference between each of the three evaluated models (Average, Least Misery and Most Happiness) for each group size and they were all significant for all the group sizes and group cohesiveness evaluated. We then examine the pair-wise difference between the models using the *t-test* for a more granulated analysis. Figures 10-15 reflect the MAE and RMSE for these evaluated relationships.

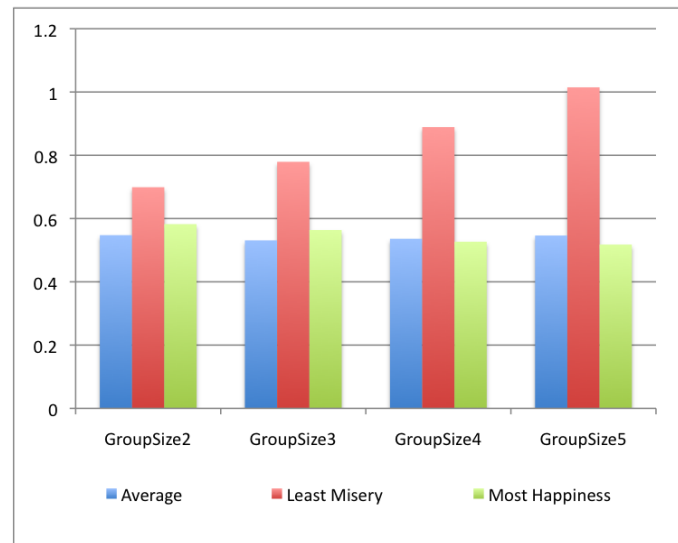


Figure 9: RMSE - High degree of similarity.

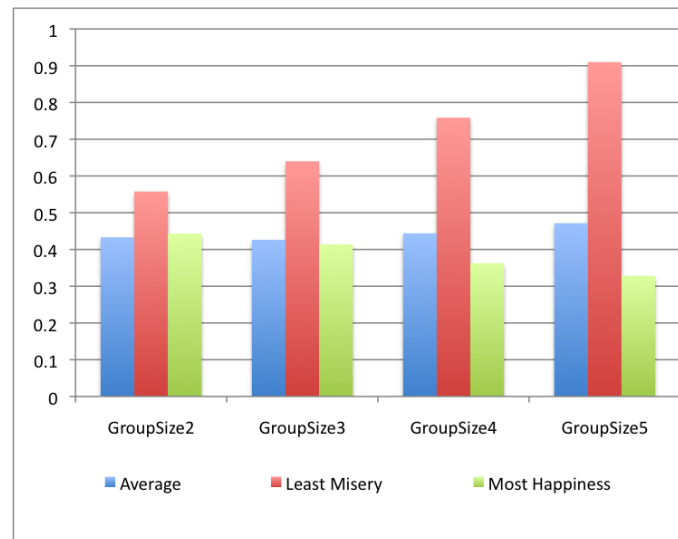


Figure 10: MAE - High degree of similarity.

Examining the graphs for the groups with high similarity levels, Figures 9 and 10 show that average strategy and most happiness perform better than least misery. We conducted a *paired 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, the most happiness strategy performs better than the average strategy ($p < 0.001$). Both least misery and average strategies' performance worsens as the group size grows while the most happiness strategy performance improves as the group size increases from 2 to 4. This indicates that a larger group of highly similar people is as happy as its happiest member.

Figures 11 and 12 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 that of groups with high cohesiveness levels.

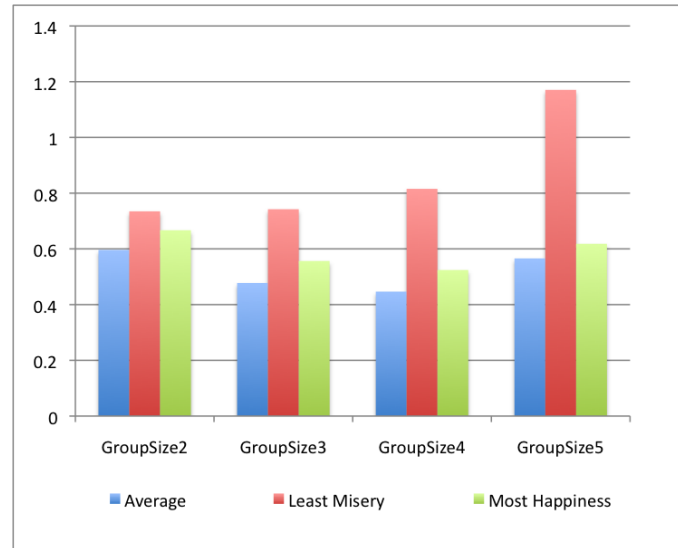


Figure 11: RMSE - Medium degree of similarity.

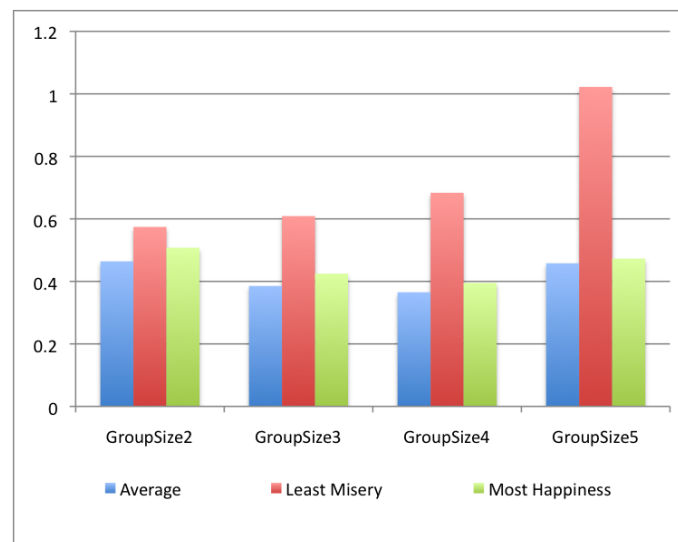


Figure 12: MAE - Medium degree of similarity.

Figures 13 and 14 show the results for the groups with low similarity levels. 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 cohesiveness had a statistically significant p value ($p < 0.001$) compared to both least misery and most happiness strategies.

Unlike groups with high cohesiveness, the most happiness strategy performance starts to worsen and the least misery strategy performance starts to improve as the group size increases, for groups with low cohesiveness.

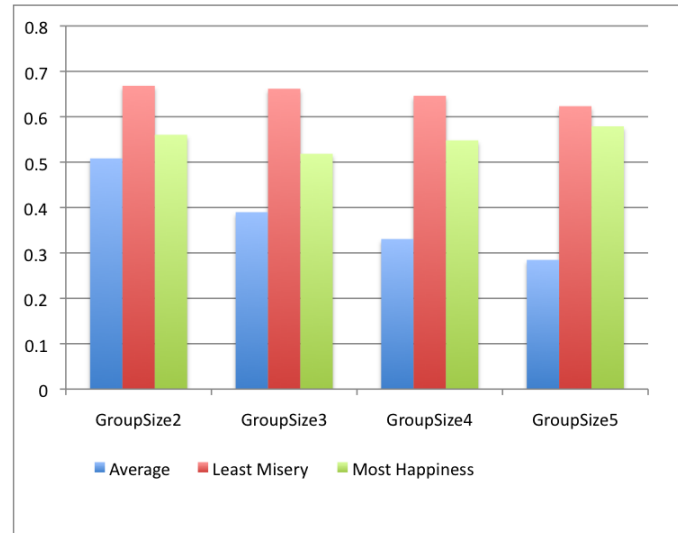


Figure 13: RMSE - Low degree of similarity.

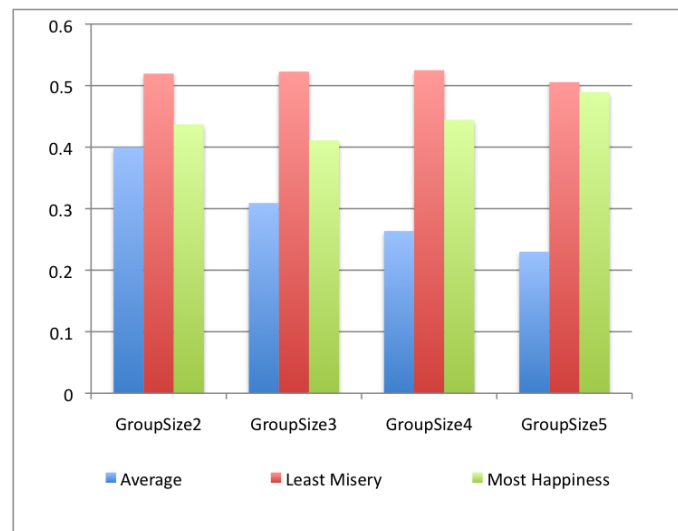


Figure 14: MAE - Low degree of similarity.

4.4 Summary

These evaluation results indicate that, in situations where groups are formed with highly similar members, the most happiness aggregation strategy would be best to

model the RGPM while, for groups with medium to low cohesiveness, the 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, which had 3 members, the average strategy performs best, while for the first-acquaintance, which had one group with 12 members, the least misery strategy performs the best. It is apparent that their results for the couple groups performance is equivalent to our high-cohesiveness groups. The acquaintance groups maps to the medium-cohesiveness groups while the first-acquaintance groups follow the low-cohesiveness groups. Masthoff's 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 cohesiveness levels.

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 our work, we have developed and evaluated a group testing framework that can help address the problem by automating group formation resulting in the generation of groups applicable for testing in this domain. Our work provides novel coverage in the group recommender evaluation space by, (1) considering focus

on traditional memory-based collaborative filtering, and (2) employing precise overlap across individual user ratings for evaluating actual group preference. We evaluated our framework with a foundational Collaborative Filtering neighborhood-based approach, measuring prediction accuracy, and utilizing three representative group prediction, aggregation strategies. Our results show that, for small-sized groups with high-similarity among their members, the average and the most happiness strategies perform the best. For larger size groups, with high-similarity, 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.

This experiment has provided a foundation for evaluation and analysis of group-based recommender systems. We presented this work as a part of the Workshop on Human Decision Making in Recommender Systems held in conjunction with the ACM Conference on Recommender Systems [56].

In evaluating recommendations for synthetic groups, the *AGPM* — the baseline for measuring recommendation accuracy — has typically been modeled as an average rating across group members (e.g., [5]). However, there is comparatively little evidence that real group decisions rely almost solely on an average strategy. For example, [46] and [38] note that different groups employ a variety of different strategies. As a result, evaluations that consider only an average strategy *AGPM* (and the system development that relies on them) may not always provide the best model for group recommender behavior. Recent work by Quijano-Sanchez et. al. [61] utilized human

subjects to establish ground truth for the synthesized groups used in their evaluation of a case-based aggregation model for group recommenders. They based the *AGPM* on a voting scheme of a panel of experts rather than a model based on the individual known preferences of the group members.

In the absence of one true *AGPM*, the meta-issue arises of how to evaluate candidate evaluation models. In order to do so, there are essentially two traditional options: finding a baseline context for comparing *AGPMs*, or relative performance comparison among different evaluation models. Keeping in mind that the dependent variable in such experiments is a component of the evaluation strategy itself. This issue of how to model the *AGPM* in the evaluation of group recommenders using synthesized groups motivated us to investigate alternative approaches for *AGPMs*. Of course changing the evaluation baseline will show differences in outcomes, but we are interested in investigating the shape and extent of such differences as a way to inform the process of evaluation. In the next experiment we examine the scope of potential tradeoffs in the choice of an aggregation model for recommendations for groups when actual group preference may vary. We lay out a survey of evaluation techniques in this context. This can provide group recommender system developers with a better understanding of the implications in choosing a particular *AGPM* baseline.

4.5 Tradeoffs in Evaluation

The purpose of the following sections is to show that when utilizing synthesized groups in evaluation outcomes can vary depending on the underlying model used as the *APGM*. We hypothesize that these changes occur and interested in seeing how they are visible, and examine their impact on conclusions drawn using current

evaluative techniques that do not take them into account.

4.5.1 Hypothesis

H2 - The choice of a model as the actual group preference impacts the accuracy levels of predictions generated for the group. Results will favor the aggregation strategy that is used for both the *RGPM* and the *AGPM*.

4.6 Dataset and Evaluation Process

In this experiment we utilized the same dataset (GD1), evaluation process and setup from the previous experiment with the following modifications:

- To evaluate different models for the *AGPM*, we randomly selected a test item from the set of commonly rated items for each group not limiting it to the item where all group members gave the same ratings. For each group, one item has been identified as the test item we then followed a similar leave-one-out approach using that test item as explained in Section 3.2.
- Since we are interested in the performance of various aggregation models used as the *AGPM* we varied the AGPM between the Average, Least Misery and Most Happiness models.

In the previous experiment the analysis results obtained using the MAE and RMSE metrics were comparable in the following evaluation we analyze and include the results based on the RMSE metric only.

4.7 Results and Discussion

To analyze the impact of the choice of a model as the actual group preference on the accuracy levels of predictions generated for the group, we compare the accuracy levels

of predictions generated using the three different models defined in section 4.2.2.2. Surely changing the baseline will change the accuracy results but we are interested in the size, extent, and trends of such differences across different group conditions.

4.7.1 Relationship between the AGPMs

In this evaluation we used ANOVA to compare the different models for each group size and group cohesiveness level across the various AGPMs. The difference was significant for all the group sizes and group cohesiveness evaluated ($p < 0.5$). Figures 15 to 17 show the RMSE values for the various conditions. To evaluate the difference in RMSE values we use the two-tailed, t-test for statistical significance with a p value of $p < 0.01$. For each group similarity level (high, medium, low) and *AGPM* (LM, MH, Average), we compared the *RGPMs* (LM, MH, Average) to each other. So, for each pair of *AGPM* (Avg/LM, Avg/MH, LM/MH), we compared the various *RGPMs* for each group similarity level and group size (2,3,4,5).

For this evaluation there were 324 conditions (3 *AGPM* pairs x 9 *RGPM* pairs x 4 group sizes x 3 similarity levels). Of those only 4 had a p value > 0.01 . Examining those 4 non-significant relationships further, they are between the LM_AGPM and MH_AGPM. Thus, as expected, there are significant differences in most *RGPM* evaluations, depending on the active *AGPM* baseline. For the LM_AGPM (Figure 16) results indicate that the LM_RGPM performs best across all group inner-similarity levels and the different group sizes. The difference between the *RGPMs* was significant for each group inner-similarity level and each group size. The same results are found for the MH_AGPM (Figure 17) and the Avg_AGPM (Figure 15).

Indeed, we would be surprised, were it not the case, that evaluation outcomes

were biased when the aggregation strategy used for the *RGPM* is also used for the *AGPM*. If we knew which aggregation strategy the group actually used, applying it on the recommendation side, one would expect that it would result in more accurate recommendations. From these results we can conclude that the choice of a baseline to represent the *AGPM* does indeed matter and the commonly adopted average model, as a baseline, does not necessarily provide an ideal baseline for evaluating group-based recommender systems.

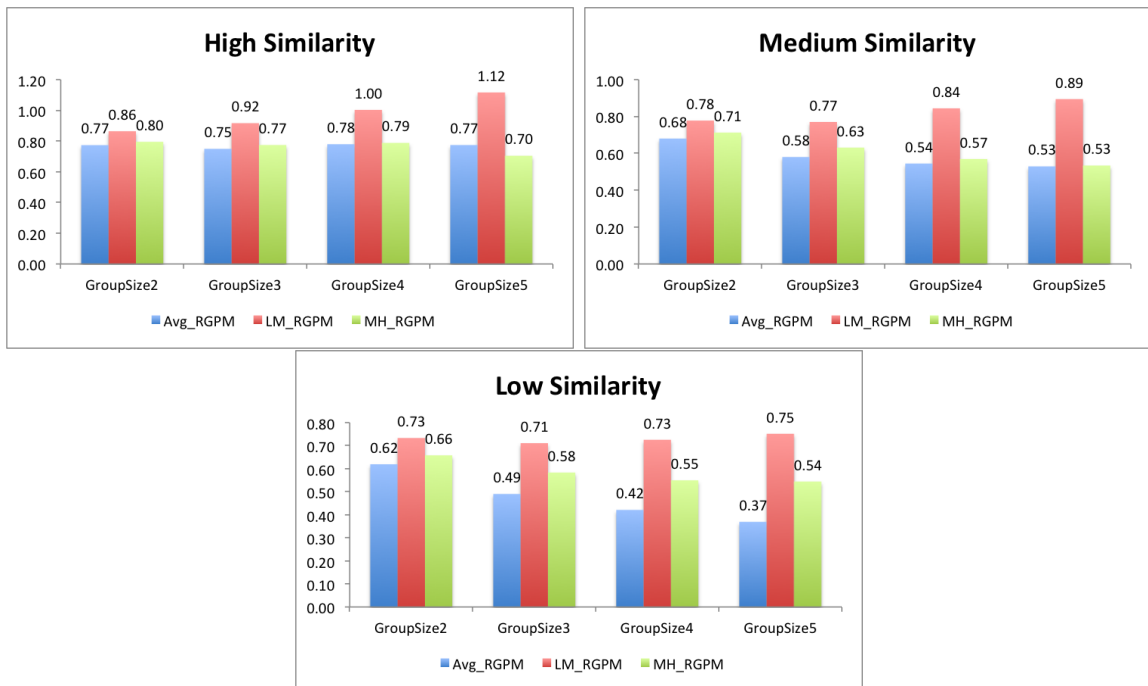


Figure 15: RMSE for Avg_AGPM

4.7.2 Relationship between the RGPMs

In comparing evaluation outcomes, we examine the outcomes across individual conditions and the correlation across all group sizes. Tables 1-3 show the correlation among accuracy values between the three aggregation models when used as the *AGPM* as well as the *RGPM* for the three defined inner group similarity levels.

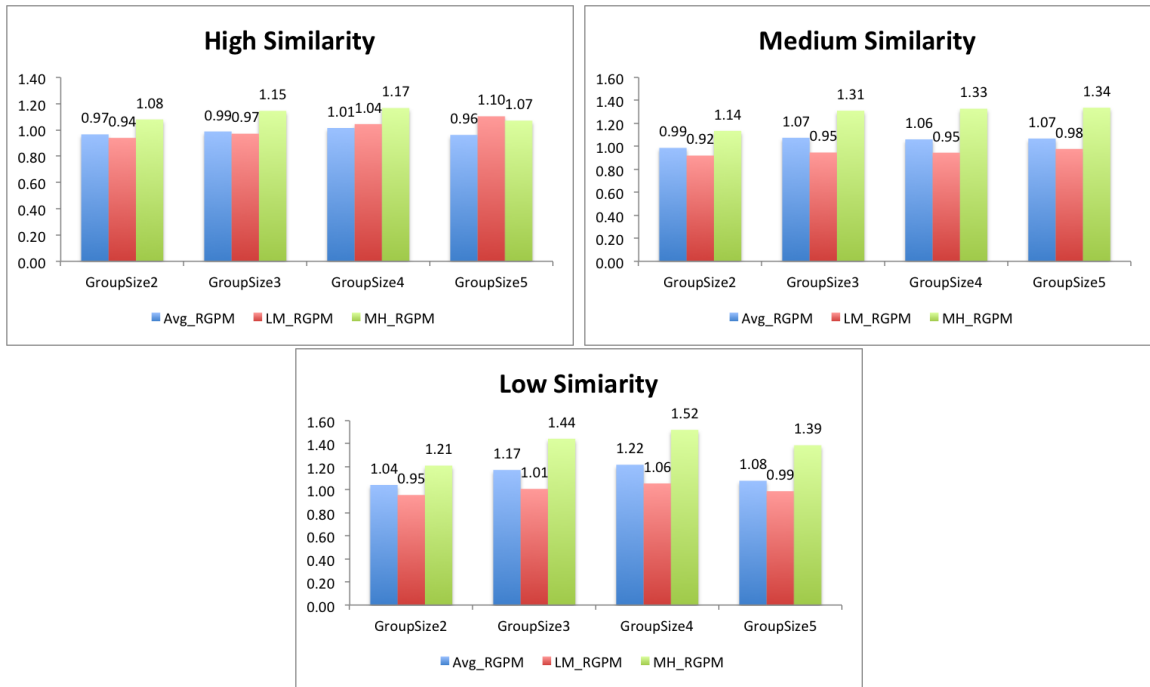


Figure 16: RMSE for LM_AGPM

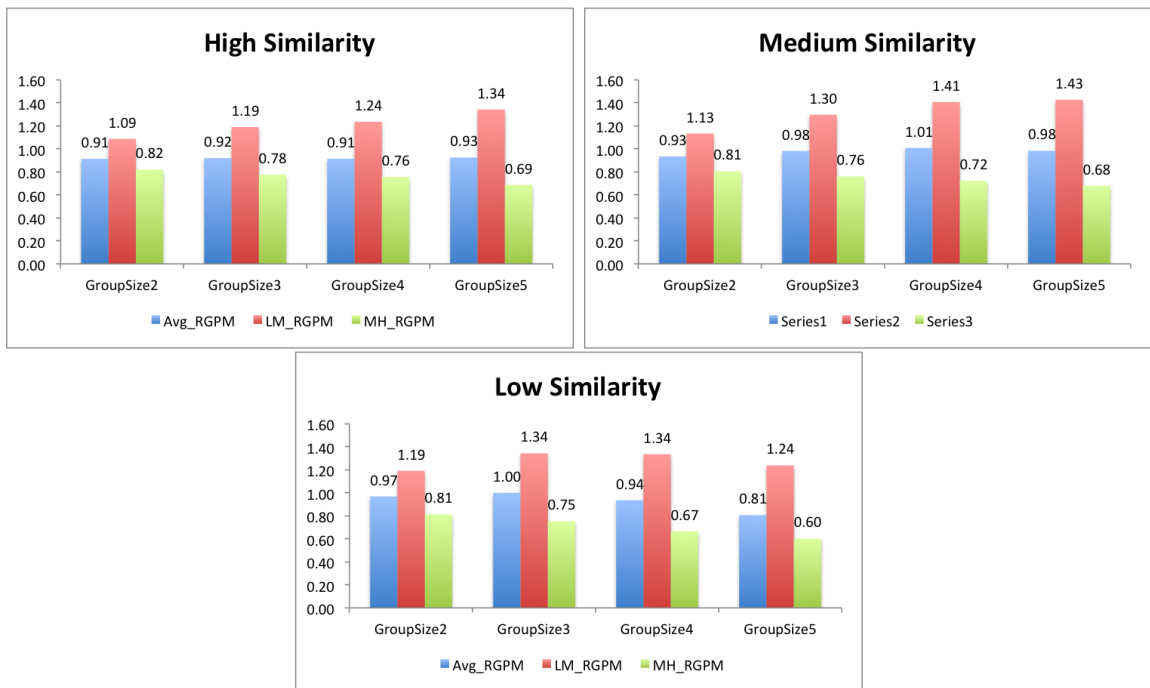


Figure 17: RMSE for MH_AGPM

Examining the correlations between the LM_AGPM and the MH_AGPM, we can see that these two approaches are mostly negatively correlated. The correlation is

Table 5: Correlations between the RGPMs using the different AGPMs for groups with low similarity levels

		Avg_RGPM	LM_RGPM	MH_RGPM
LM – MH	Size 2	-0.1101	-0.0900	-0.1470
	Size 3	-0.0332	-0.0585	-0.0985
	Size 4	0.0216	-0.0403	-0.0557
	Size 5	0.1806	0.0783	-0.0040
LM – AVG	Size 2	0.4320	0.1751	0.7026
	Size 3	0.1738	-0.1626	0.6401
	Size 4	0.1854	-0.1817	0.6181
	Size 5	0.1030	-0.0827	0.5798
MH – AVG	Size 2	0.2761	0.5788	0.0398
	Size 3	0.2303	0.6300	-0.1284
	Size 4	0.1923	0.5853	-0.1401
	Size 5	0.1309	0.4684	-0.0716

Table 6: Correlations between the RGPMs using the different AGPMs for groups with medium similarity levels

		Avg_RGPM	LM_RGPM	MH_RGPM
LM – MH	Size 2	-0.1823	-0.1356	-0.1420
	Size 3	-0.1428	-0.0907	-0.1054
	Size 4	-0.0923	-0.0624	-0.1309
	Size 5	-0.0202	-0.0444	-0.0630
LM – AVG	Size 2	0.5158	0.2703	0.7402
	Size 3	0.2737	0.0087	0.6458
	Size 4	0.1291	-0.0366	0.6374
	Size 5	0.0452	-0.0486	0.5754
MH – AVG	Size 2	0.3134	0.6242	0.1733
	Size 3	0.2730	0.6483	0.0904
	Size 4	0.2471	0.6101	-0.0972
	Size 5	0.1963	0.5836	-0.0536

strongest for the MH_RGPM for groups with a high inner-group similarity level. One explanation for that is that, as the group members are highly similar, they tend to give items ratings that are similar or close. In that case, both aggregation models used as a baseline might yield comparable evaluation results.

When comparing the LM_AGPM to the Avg_AGPM we can also see a similar trend. For groups that have highly similar members the evaluation outcomes, using these

Table 7: Correlations between the RGPMs using the different AGPMs for groups with high similarity levels

		Avg_RGPM	LM_RGPM	MH_RGPM
LM – MH	Size 2	-0.1233	-0.1216	0.0656
	Size 3	-0.2017	-0.0872	0.0535
	Size 4	-0.1425	-0.0657	0.2397
	Size 5	-0.1368	-0.0894	0.2451
LM – AVG	Size 2	0.7159	0.5238	0.8418
	Size 3	0.6006	0.3620	0.8011
	Size 4	0.5751	0.3878	0.7814
	Size 5	0.5544	0.3966	0.7771
MH – AVG	Size 2	0.4186	0.6336	0.4520
	Size 3	0.3168	0.6923	0.3800
	Size 4	0.3487	0.6528	0.6083
	Size 5	0.3007	0.6035	0.5794

two *AGPMs* as a baseline, are highly correlated for all group sizes and *RGPMs*. Since the group members are similar they tend to rate similarly or very close to the group’s average rating for that item.

As the group sizes increase, for groups with medium and low similarity levels, the Avg_RGPM and LM_RGPM can result in different evaluation outcomes. We can see that all the lowest correlations are for the LM_RGPM and appear in groups of medium and low similarity. This can be because, as the group’s inner similarity level decreases, the chances that they rate an item differently increases which makes it further from the average rating of the group members for that item. Another factor is, as we pointed out earlier, the LM_RGPM favors the lowest rating and that might be further away from the average group rating for that same item. This also highlights the bias effect introduced when comparing results with the same aggregation strategies used on the recommendation and evaluation sides. Here we are comparing the LM_RGPM using the LM_AGPM as a baseline.

We attribute this relationship to the fact that these two aggregation strategies are on opposite sides. The LM_AGPM favors the lowest rating while the MH_AGPM favors the highest rating. We can see the conditions where the RGPMs are negatively correlated between the LM_AGPM and the Avg_AGPM. We can see that all the negative correlations are for the LM_RGPM and appear in groups of medium and high similarity. This can be because, as the groups' inner similarity level increases, there is a greater chance that they rate an item more similarly, which makes it closer to the average rating of the group member's for that item.

Another factor is, as we pointed out earlier, the LM_RGPM favors the lowest rating and that might be further away from the average group rating for that same item. This also highlights the bias effect introduced when comparing results with the same aggregation strategies used on the recommendation and evaluation sides. Here we are comparing the LM_RGPM using the LM_AGPM as a baseline. We notice the same trends when comparing either MH or LM to the Avg AGPM. We see the low correlations more in the low similarity groups for the MH_RGPM. This might indicate that, for low similarity groups, using the MH_RGPM might not result in satisfactory recommendations to all the group members. Here the bias effect is also highlighted since the MH_AGPM is one of the baselines used here.

4.7.3 Summary

In this experiment we analyzed different choices of an aggregation strategy to model the actual group preference when evaluating the accuracy performance of a group-based recommender system using synthesized groups. We compared the results of using the Avg_AGPM, LM_AGPM and MH_AGPM as a baseline to evaluate the

performance of three representative aggregation strategies as the RGPMs using synthesized groups with various degrees of inner similarity and size. Results show that the choice of an AGPM, in this context, results in different evaluation outcomes. The choice of an AGPM can also introduce a bias, particularly when the same aggregation strategy is used on the recommendation side.

We aimed to show that modeling the actual group preference does matter when evaluating group-based recommenders, and different models provide different results. Developers of such systems need to consider the tradeoffs when choosing a baseline for evaluation. Here we demonstrated the differences between some of these choices. 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.

CHAPTER 5: RECOMMENDATION TECHNIQUE FOR GROUP-BASED RECOMMENDER SYSTEMS

In our research of group-based recommender systems we identify the following research issues, and outline the primary problems, to include choices made about:

1. The underlying recommendation strategy (e.g., content-based, collaborative memory-based, or model-based)
2. Group modeling for making recommendations — RGPM (e.g., least misery)
3. Evaluation of the recommendations:
 - 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).

We have formulated representative hypotheses to target the issues we identified in group-based recommenders. In this chapter and the following chapters we propose to investigate novel approaches along these contexts. For each issue, we will introduce the motivation that led us to ask these questions. The questions are formalized as hypotheses. The methods, techniques, tasks, and tools needed to evaluate each hypothesis are highlighted in the “Methodology” (Sections 5.1.2.2, 5.2.2, 6.1.2.2, 6.2.4) and “Evaluation Process” (Sections 5.2.3, 6.1.2.3, 6.2.4.1) sections.

5.1 Memory-Based Collaborative Filtering In Group-Based Recommender Systems

5.1.1 Motivation

Previous research in CF utilized the availability of large, single-user data sets to evaluate approaches used in this technique and to assess their validity and success. Since group-based data sets are not readily available, evaluations have been limited to a smaller scale. Utilizing our implementation of the Group Testing Framework defined in Section 3 we are able to provide a baseline evaluation for some of the design choices for Group Neighborhood-Based Collaborative Filtering Algorithms.

In section 1.2.1, a neighborhood-based CF is divided into three components; similarity computation, neighbor selection, and rating combination. Herlocker et al. [33] provided a baseline evaluation of the most widely used approaches along those components for single-user, Neighborhood-Based Collaborative Filtering recommender systems. They found that a gain in prediction accuracy was achieved by using Pearson correlation as a similarity measure, a significance weight to devalue correlations with small numbers of co-rated items and the use of the deviation-from-mean approach to normalize ratings. We believe that special consideration along the components of a Neighborhood-Based CF, that accounts for the group structure and members, will result in a gain in prediction accuracy.

Masthoff [46] applied normalization in measuring satisfaction for a group after viewing a series of items (TV shows). She normalized the individual satisfaction scores for each group member before aggregating them into one group satisfaction score.

She reported an improvement in the final satisfaction score. Herlocker's evaluation compared normalizing ratings, when it comes to combining the individual ratings, to form a prediction for an item. Deviation-from-mean averaging (equation 2) was shown to increase prediction accuracy significantly over a normal weighted average, while z-score averaging provided no significant improvements over deviation-from-mean.

On the same lines, we believe that accounting for the fact that users rate on different scales by normalizing the individual ratings will result in a gain in prediction accuracy for recommendations generated for groups. This is hypothesized in Section 5.1.2. For the the neighborhood selection component we believe accounting for a group neighborhood in calculating the individual predicted ratings will also result in a gain in prediction accuracy. For any group member, when calculating a predicted rating for an item by weighting neighbors ratings, a gain in prediction accuracy is realized by weighting users that belong to the neighborhoods of all group members more than the users belonging only to the individual group member neighborhood. This is hypothesized in Section 5.2.1

5.1.2 Rating Normalization for Prediction Calculation in Memory-Based

Collaborative Filtering

5.1.2.1 Hypothesis

H3 - Normalizing the individual ratings when computing a final prediction by performing a weighted average of deviations from the neighbor's mean or accounting for the differences in spread between users' rating distributions by converting ratings to z-scores, will result in a higher prediction accuracy in the final group item prediction when compared to a non normalized, memory-based CF approach.

5.1.2.2 Methodology

To evaluate this hypothesis, we evaluated the prediction accuracy of a predicted group rating for a test item generated by equation 2 as the CF algorithm to a baseline, non-normalized average approach defined as follows:

$$p_{ai} = \frac{\sum_{b=1}^n (r_{bi} \cdot w_{ab})}{\sum_{b=1}^n w_{ab}} \quad (6)$$

Where p_{ai} is the prediction for the active user a for item i . n is the number of neighbors, r_{bi} is user b rating of item i , and w_{ab} is the similarity weight between users a and b as defined by the Pearson correlation coefficient, equation 1. We will also evaluate the performance of a z-score approach to normalizing ratings, as defined in equation 7 in comparison to the baseline non-normalized approach as well as the deviation-from-mean approach.

$$p_{ai} = \bar{r}_a + \sigma_a \frac{\sum_{b=1}^n [(r_{bi} - \bar{r}_b) \cdot w_{ab}]}{\sum_{b=1}^n w_{ab}} \quad (7)$$

Where p_{ai} is the prediction for the active user a for item i . n is the number of neighbors, r_{bi} is user b rating of item i , and w_{ab} is the similarity weight between users a and b as defined by the Pearson correlation coefficient, equation 1.

5.1.2.3 Evaluation Setup

1. Define the dataset used in this evaluation: We use the GD2 dataset created as defined in section 3.2.
2. Define the baseline prediction group aggregation strategy: We implement the Average, Least Misery, and Most Happiness aggregation strategies.
3. Generate predictions for each group based on the three different user-user CF

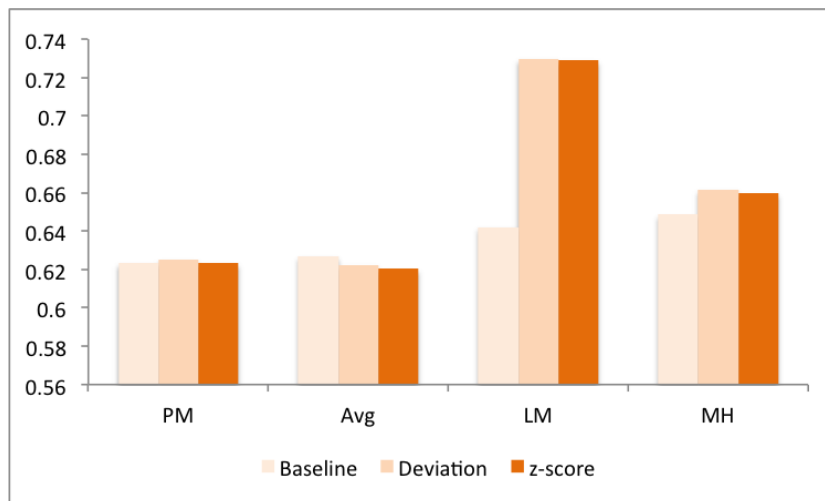


Figure 18: RMSE comparing baseline to normalization with deviation-from-mean and z-score for 1200 groups

algorithms as outline in Section 5.1.2.2.

4. Generate predictions for the pseudo user of that group.
5. Compare prediction accuracy of the non-normalized baseline to the two normalization approaches (Equation 2, Equation 7) based on the RMSE metric for profile merging and recommendation aggregation techniques.

5.1.3 Results

To compare applying these two normalization approaches in prediction calculation we examine the accuracy of these approaches for the recommendation aggregation techniques using average, least misery and most happiness strategies as well as the profile merging technique using the average strategy. We first examine the performance of normalization with considering the group modeling approach. This included all the groups in the dataset (1200 groups/10543 items) with out considering group characteristics. Figure 18 shows the RMSE for these approaches across the group modeling techniques. For the Profile Merging (PM) model there was no advantage to

applying normalization to the ratings when calculating the prediction for the group members. Even though both normalization approaches had a higher RMSE value it was not significant ($p < 0.05$) for both the ANOVA across the three techniques and the pair-wise using the *t-test* and this difference was negligible in value (0.001) with the z-score slightly better than deviation.

For the recommendation technique the performance of the normalization highly depended on the group modeling strategy. When using the average model the normalization approaches were better than the baseline with no difference between them. Overall the z-score approach performed best.

For both the Least Misery (LM) and Most Happiness (MH) these results were conflicting. Normalization resulted in lower accuracy predictions (higher RMSE). This effect was much more noticeable for the LM model. For the MH model even though the difference in RMSE value was not big it was significant.

Next we evaluated normalization with respect to group characteristics across the different group models. The characteristics we considered are the group's inner cohesiveness and size. Figure 19 show the winning approach across each inner cohesiveness level and size of the groups. Figure 19 depicts this comparison by highlighting the winning strategy for each group category tested and reflecting the significance of these results. For the significance testing we used ANOVA and two-tailed, paired T-Test with $p < 0.01$. Upon examination of the first group model (Average), across the different group inner cohesiveness levels and group sizes we can see that normalization using the z-score resulted in higher accuracy predictions when compared to the baseline and the deviation-from-mean approaches for groups with a low inner cohesiveness

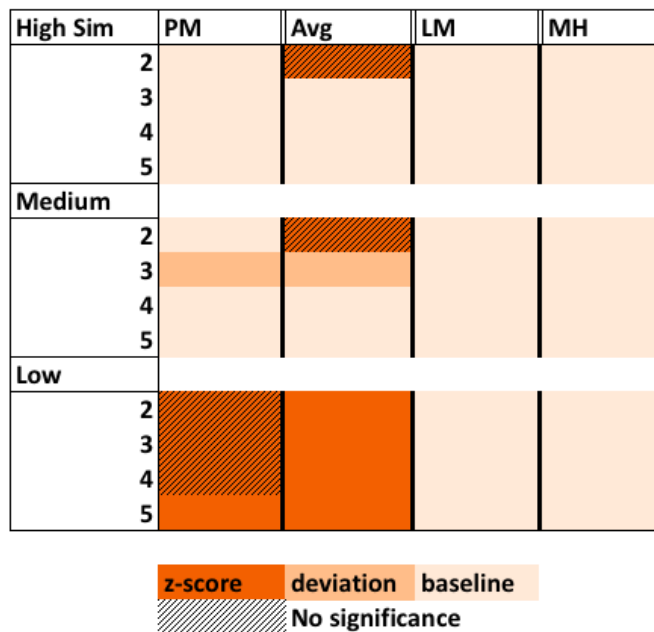


Figure 19: Comparing baseline prediction to normalized prediction using deviation-from-mean and z-score, across group inner cohesiveness and size, in terms of prediction accuracy using RMSE

level. The baseline approach had better results for groups with high and medium inner cohesiveness levels. For groups of high or medium inner cohesiveness and of size 2 the z-score had better but not significant results. Similarly, groups with medium inner cohesiveness and of size 3 had better results with the deviation-from-mean but small in value and not significant. So if we determine the inner cohesiveness level of the group to be of a low level and we are using the average group model when aggregating predictions using normalizing ratings would yield a higher accuracy predictions when compared to a non normalized approach.

Results of the LM and MH approach for each group characteristic we evaluated were consistent with the overall results where the non normalized approach outperformed the normalized approaches.

The last group model we examined in this evaluation was the PM model which

had similar results across the group characteristics as the Average model, where normalization was mostly effective for groups with lower inner cohesiveness levels regardless of the group's size.

Overall, from this evaluation, we conclude that using normalization in the prediction calculation when generating recommendations for a group would result in higher accuracy predictions when using the Average group model with profile merging and prediction aggregation. If we are aggregating predictions using the Least Misery or the Most Happiness group model, a non-normalized approach would result in higher accuracy predictions.

5.2 Neighborhood Selection In Memory-Based Group Recommender Systems

The traditional CF approach is commonly referred to as Neighborhood-based and relies on the fact that each person belongs in a larger group of similarly behaving individuals. For example, items (e.g., products, movies, books, etc.) frequently purchased/liked by the various members of the group can be used to form a basis for recommended items. Similarly, users that appear in more than one of the group members' neighborhoods might be more valuable as a basis for the group recommendation.

Neighborhood-based Collaborative Filtering for single-user recommendation identifies neighbors of the target user and item pair. Extending this to the group-based context, we focus on neighbors of the group as a whole rather than of individual members.

In this evaluation we explored the effect on prediction accuracy if special consideration is given to the neighbors of the group members and they are shared by one or more of the group members.

Figure 20 depicts the neighborhoods we explore in this evaluation and their effect on prediction accuracy in group-based recommendation.

Given an item for which the system needs to predict the rating for the group, for each group member we find the $topN$ similar users that rated the item. This forms a set of *Neighborhoods* for that item across all the group members (Figure 20a). We then considered the intersection of these neighborhoods. The neighbors that are present across all the individual group members' *Neighborhoods* form the group's intersect neighborhood for that item, referred to as the *Intersect_Neighborhood* (Figure 20c). For each group member the users that are present in their *Neighborhood*, and not in the *Intersect_Neighborhood*, form what we refer to as the *User_Neighborhood* (Figure 20b).

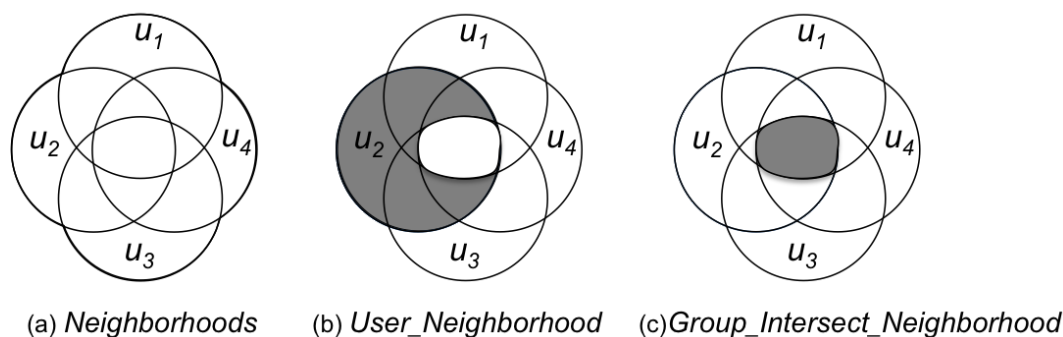


Figure 20: Identified neighborhood in the group context

This enabled us to distinguish between types of neighboring users in making predictions. In this analysis, we investigated higher weighting for *Intersect_Neighborhood* users. We used the deviation-from-mean approach to calculate a predicted rating for group members, where the users that are present in the *Intersect_Neighborhood* are assigned a higher weight than the users that are present in the *User_Neighborhood*.

We refer to this approach as the *Group Intersect Neighborhood Selection* model.

We evaluated the performance of this approach for both main types of group recommendation techniques (merging profiles, merging recommendations). For the merging profiles technique, we computed predictions for the group based on a pseudo-user created by merging the profiles of the group members. For the prediction aggregation approach a prediction is calculated for each group member and then these individual predictions are aggregated into a final group prediction using a group aggregation model. We evaluated the three group aggregation models outlined in Section 4.2.2.2.

5.2.1 Hypothesis

H4 - The prediction accuracy will increase if additional weight is assigned to the neighbors that are in common to all group members when used to calculate a predicted rating using the deviation from mean approach as defined in Equation 8 when compared to the baseline predicted rating calculated using Equation 2.

5.2.2 Methodology

To evaluate this hypothesis we evaluated the prediction accuracy of a predicted group rating for a test item generated by a baseline CF approach, as defined in Section 4.2.2.1, to a group weighted neighborhood approach defined as follows:

1. For every group member, a , create CF neighborhood for item i , N_{ai} .
2. Find the intersect neighborhood for all group members for item i , $IntersectN_{Gi}$.
3. For every group member, a , define two neighborhoods: `group_intersect_neighborhood` $IntersectN_{Gi}$ and `user_neighborhood` $UserN_{ai}$ for item i , where $UserN_{ai}$ is equal to $(N_{ai} - IntersectN_{Gi})$.

4. For every group member, a , compute a predicted rating p_{ai} for item, i , as follows:

$$p_{ai} = \bar{r}_a + \sigma_a \frac{\sum_{b=1}^n \left[\left(\frac{r_{bi} - \bar{r}_b}{\sigma_a} \right) \cdot w_{ab} \right] \cdot \alpha}{\sum_{b=1}^n w_{ab}} \quad (8)$$

where $\alpha = 2$ if $b \in IntersectN_{Gi}$ and $\alpha = 1$ if $b \in UserN_{Gi}$

5. Aggregate the individual group members' predictions into a final group prediction for item i .

5.2.3 Evaluation Process

1. Define the dataset used in this evaluation. We use the GD2 dataset created as defined in section 3.2
2. Define the baseline prediction group aggregation strategy: we implement the Average, Least Misery, and Most Happiness aggregation strategies.
3. Generate predictions for each group and the group's pseudo user based on the baseline CF approach.
4. Generate predictions for each group and the group's pseudo user based on the group neighborhood CF approach
5. Compare prediction accuracy of the group neighborhood CF approach to the baseline average approach (RMSE).

5.2.4 Results

We first considered the overall RMSE performance across all group sizes and similarity levels (Figure 21). An ANOVA showed the the difference between the eight models is significant ($p < 0.05$). We found that the average baseline aggregation performs better than any of the baseline aggregating recommendations approaches as

well as applying the *Group Neighborhood Selection* model whether we are aggregating profiles or aggregating recommendations.

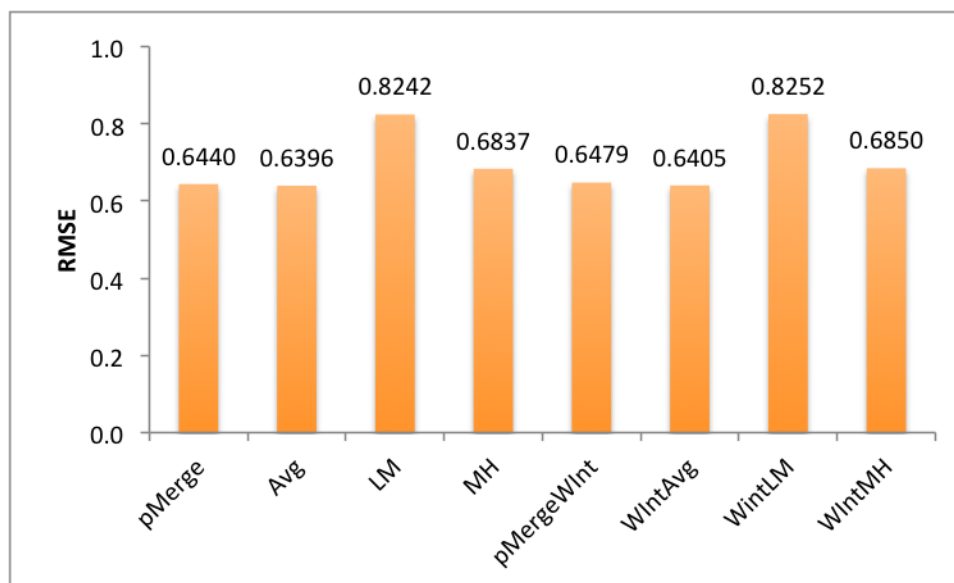


Figure 21: RMSE over all evaluated groups and test items

Given the evaluation approach, it is possible for a subset of the randomly created groups to have a *Group_Neighborhood* size of zero. To examine the potential impact of our model, we specifically considered groups and test items where the *Group_Neighborhood* size is greater than zero. In our results, there were 775 unique groups and 4836 group/item pairs for testing. Figure 22 depicts the results still showing that the average baseline provides higher accuracy recommendations ($p < 0.001$ for *t-test* and $p < 0.05$ for ANOVA with all models) than the other implemented approaches.

5.2.4.1 Group Size

To examine how the group neighborhood model performs with respect to the size of the group we combined the evaluated groups based on size. We first examined the

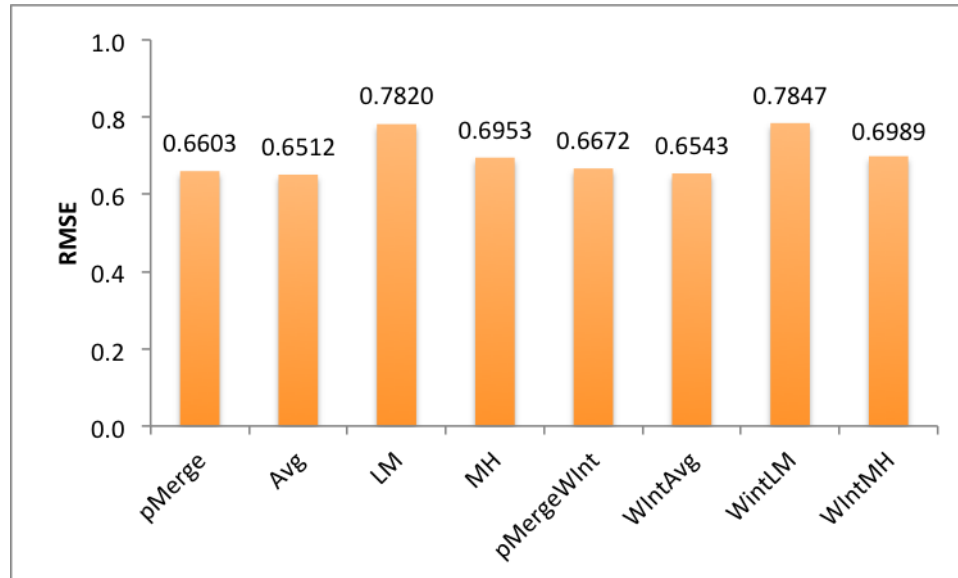


Figure 22: RMSE over all evaluated groups and test items with *Group_Neighborhood* size > 0

results for the aggregating predictions recommendation technique. Table 8 shows the RMSE values for the different evaluated models across group sizes 2-5. Results show that for groups of sizes 2 and 3 the average group model (Avg) performed best while the group neighborhood selection model performed better for groups of sizes 4 and 5. For groups of size 4 and size 5 combining the group neighborhood selection model with the average aggregation yielded the best performance in terms of prediction accuracy.

	Avg	LM	MH	wIntAvg	wIntLM	wIntMH
Size 2	0.69405	0.79391	0.717	0.69558	0.79567	0.71946
Size 3	0.5905	0.82459	0.66772	0.5908	0.82471	0.66819
Size 4	0.54716	0.84672	0.63618	0.5460	0.84608	0.63559
Size 5	0.52167	0.93668	0.58564	0.52085	0.93701	0.58349

Table 8: RMSE values across the different group sizes

To directly analyze the performance of the *Group Neighborhood Selection* model with respect to group size, we once again examined only the groups/test pairs where

it was applicable (*Group_Neighborhood* size > 0). Table 9 shows the RMSE for each model across the various group sizes evaluated. These results are in line with the non-filtered results shown in table 8 and show that the average prediction aggregation model performed best for groups of size 2 and 3. Combining the group neighborhood selection model with average aggregation model was best for groups of size 4 and size 5.

	Avg	LM	MH	WIntAvg	WIntLM	WIntMH
Size 2	0.6719	0.7716	0.6981	0.6761	0.7751	0.7032
Size 3	0.5852	0.7929	0.6784	0.5859	0.7931	0.6794
Size 4	0.5620	0.8614	0.6890	0.5573	0.8579	0.6860
Size 5	0.5298	0.9089	0.7391	0.5216	0.9123	0.7216

Table 9: RMSE values across the different sizes for groups and test items with *Group_Neighborhood* size > 0

From these results we can conclude that, as the group size increases, the group neighborhood selection model results in higher prediction accuracy.

5.2.4.2 Group Cohesiveness

We then examined the performance of the evaluated models with the respect to inner-group similarity.

Prediction Aggregation

We first analyze the results for the prediction aggregation approach. Table 10 shows the RMSE values for the evaluated models with respect to inner-group similarity levels. For groups with high levels of inner-similarity the *Group Neighborhood Selection* model performs better than the other approaches. While the baseline average aggregation model performed better for groups with medium to low inner-similarity levels. The difference between the baseline Average and the *Group Neighborhood Selection*

model was not significant for groups with medium similarity and significant for groups with low similarity levels.

	Avg	LM	MH	wIntAvg	wIntLM	wIntMH
High	0.75063	0.94227	0.75894	0.74548	0.93983	0.75360
Medium	0.59812	0.80642	0.64528	0.60002	0.80662	0.64846
Low	0.52917	0.66920	0.63056	0.53927	0.67764	0.63939

Table 10: RMSE with respect to inner-group similarity level with prediction aggregation

We then considered and evaluated only the groups and test items where there was a *Group_Neighborhood* with respect to inner group similarity. Table 11 shows the RMSE values for those groups that have test items with *Group_Neighborhood* size > 0 . Again, results correspond to the non-filtered results where groups with high levels of inner-similarity had higher accuracy predictions with the *Group Neighborhood Selection* model in comparison to the other approaches. The baseline average aggregation model performed better for groups with medium to low inner-similarity levels. The difference between the baseline (Average) and the *Group Neighborhood Selection* model was not significant for groups with medium similarity and significant for groups with low similarity levels.

	Avg	LM	MH	wIntAvg	wIntLM	wIntMH
High	0.7516	0.8875	0.7871	0.7450	0.8846	0.7799
Medium	0.6334	0.7832	0.6753	0.6361	0.7835	0.6799
Low	0.5826	0.6807	0.6394	0.5965	0.6932	0.6525

Table 11: RMSE with respect to inner-group similarity level with prediction aggregation for groups and test items with *Group_Neighborhood* size > 0

Profile Merging

Next, we examined the performance of the evaluated models with respect to inner

group-similarity when using the profile merging recommendation technique. Table 12 includes the RMSE value for the baseline profile merging model and *Group Neighborhood Selection* model with profile merging. These results show that the *Group Neighborhood Selection* method performs best for high similarity groups when used with the aggregating profiles group recommendation technique. This result was not significant ($p > 0.05$) when compared to the baseline profile merging approach. For groups with medium and low inner group similarity levels the baseline profile merging approach performed best. For groups with medium inner similarity level the difference was not significant when compared to the *Group Neighborhood Selection* model. These results were significant for the groups with low inner similarity level.

	Avg_PM	wIntAvg_PM
High	0.74493	0.74411
Medium	0.60507	0.60887
Low	0.54771	0.56017

Table 12: RMSE with respect to inner-group similarity level with Profile Merging

Table 13 filters the results to include only groups and test items with *Group_Neighborhood* size > 0 . These results correspond to the results from Table 12 where the *Group Neighborhood Selection* method performs better for high similarity groups when used with the aggregating profiles group recommendation technique. This result was not significant ($p > 0.05$) when compared to the baseline profile merging approach. For groups with medium and low inner group similarity levels, the baseline profile merging approach performed best. For groups with medium inner similarity level the difference was not significant when compared to the *Group Neighborhood Selection* model. These results were significant for the groups with low inner similarity level.

	Avg_PM	wIntAvg_PM
High	0.75785	0.75689
Medium	0.64335	0.64883
Low	0.59345	0.61075

Table 13: RMSE with respect to inner-group similarity level with Profile Merging for groups and test items with *Group_Neighborhood* size > 0

5.2.4.3 Group Size and Cohesiveness

Another aspect we examined is the combined effect of the group size and the inner group similarity on the prediction accuracy using the proposed approach and the baseline approaches. Table 14 shows the best performing strategy based on the RMSE calculated for each group size and group inner similarity combination we evaluated. The difference between the models evaluated was significant for both ANOVA and *t - test*. For groups with high inner similarity level we noticed that the *Group Neighborhood Selection* model yields higher accuracy predictions when compared to the baseline models for groups of sizes 2, 4 and 5. The baseline average model had a lower RMSE value when compared the to Average *Group Neighborhood Selection* model, but the difference was not significant. As the group size increases the Most Happiness model performs better than the Average and the Least Misery models. This performance carries over when applying the *Group Neighborhood Selection*. Our results show that as the group size increases for groups with high similarity levels among the group members, combining the Most Happiness model with the *Group Neighborhood Selection* results in predictions with higher accuracy for the group.

Focusing on the *Group Neighborhood Selection* model, Table 15 shows the best performing approach for these filtered results. Once again these results go along with

↓ Similarity → Size	2	3	4	5
High	WIntAvg	Avg	WIntAvg	WIntMH
Medium	Avg	Avg	WIntAvg	WIntAvg
Low	Avg	Avg	WIntAvg	Avg

Table 14: Wining strategies with respect to inner-group similarity level and group size based on RMSE

Similarity Size	2	3	4	5
High	WIntAvg	Avg	WIntAvg	WIntMH
Medium	Avg	Avg	WIntAvg	WIntAvg
Low	Avg	Avg	WIntAvg	Avg

Table 15: Wining strategies based on RMSE with respect to inner-group similarity level with Profile Merging for groups and test items with *Group_Neighborhood* size > 0

the results shown in Table 14 indicating that an advantage is gained for the group, in terms of prediction accuracy, when special consideration is given to neighbors that are common to all the group members. This approach is mostly advantageous as the group size and inner group similarity increases. An ANOVA tests showed that these results were significant for all categories ($p < 0.05$) except for groups of size 5 with low similarity levels.

5.2.5 Expert Evaluation

In this evaluation, we use a dataset we obtained from [60]. A common evaluation approach of group-based systems is to simulate groups from individual datasets [22, 5]. This dataset utilizes this approach but, rather than basing the group preference on a model of the individual preferences, they employed human experts to evaluate the group’s preferences and produce a group decision on which to base and ground the evaluation. Following is a detailed explanation of this dataset.

The baseline dataset used is the MovieLens 1M dataset. Building an effective CF

recommendation system requires sufficient data. This data set provides the basis for that. This dataset contains 1 million ratings, on a scale of 1 to 5, for 6040 users and 3952 movies. Each user has at least 20 ratings. The dataset also gives a small amount of demographic information about each user. In particular, they use the user's gender and age range (under 18, 18-24, 25-34, and so on).

Quijano-Sanchez et al. created 100 groups from this dataset. Group members were chosen at random from all users, but subject to the following restrictions:

1. In a group, users are distinct (but a user may be in more than one group).
2. In a group, they ensure that all the users are in the same age range.
3. In a group, they ensure that there are at least 15 movies which are co-rated by all members of the group. These 15 movies will be the test items for the group.

They conducted a Facebook poll in which they asked respondents to tell them, for the last five times that they went to the movie theatre as a group, how large the group was. There were 105 respondents that reported the group size for 525 events. They used the frequencies from this distribution to create 100 groups. The break down of the groups is as follows: 50 groups of size 2, 18 of size 3, 16 of size 4, 7 of size 5, 5 of size 6, and 4 where they set the size to be 7.

To establish ground truth to be used as the baseline for the evaluation, they used four human experts who were given all the information about a group's members and the candidate movies (test items), including the actual ratings by the members of the group for the items in their test set. The experts were asked to decide on which of the movies the group would be most likely to settle. Each expert evaluated 50 cases,

hence each of the 100 groups was evaluated by two experts (not always the same two). Experts were asked to give an ordered list of three movies from the test set on which they thought the members of the group would agree. They combined the experts' judgements into a single, final ordered list of size three.

Since we are interested in evaluating our recommendation approach for both profile merging and recommendation aggregation, we wanted to ensure that the same training set was used to generate all the predictions for that group. We created a training and testing set for each group based on the test set of the group. We first created the profiles of the pseudo users for each group by merging the individual group members' ratings based on the average aggregation strategy. For each item rated by one or more group member, the rating for the pseudo user would be the average of the ratings based on the number of the group members that rated it.

Once the profiles for the pseudo users were created we added the pseudo profile to the original data set to include the new pseudo user. This ensured that the same training set was used to generate predictions for the group across all the evaluated techniques, both merging profiles and merging recommendations. To create the training set for each group, we started off with the original MovieLens dataset. We then added the profile of the pseudo user of that group to the dataset. We then took out the ratings of the test items identified for that group from each of the group member's profiles and the pseudo user. In other words, the training set for each group is the original MovieLens dataset plus that group's pseudo user profile minus the ratings for the test items for that group, for each of the group members and the pseudo user of that group.

We explored outcomes of prediction accuracy for profile merging and recommendation aggregation using the Average group aggregation strategy. We analyzed the results across the various group sizes. We made a comparison between the baseline nearest neighborhood recommendation technique and the Group Neighborhood Selection techniques as outlined in the previous section. We compared recommendation rankings based on prediction to the item’s ranking provided by the experts using the evaluation metric outline in the following section.

5.2.6 Hypothesis

H5 - Evaluation outcomes for the Weighted Intersect Neighborhood Selection model utilizing the dataset with real users will be in line with the results obtained using the synthesized groups dataset.

5.2.7 Evaluation Metric

To evaluate the performance of the implemented recommendation techniques we compared the recommended list of items to the actual preferences list. A variant of this strategy, $success@n$, was employed in [60] to measure the rate of having at least one recommended item in the top n positions of the actual preferences list. For example, given an ordered set of recommended items $recList$ of size n and an ordered set of the actual preferences $actList$ of the same size, $success@3$ would return 1 if at least one of the items in the top 3 positions of $recList$ appeared in the top 3 positions of $actList$, and 0 otherwise. We used the $success@n=3$ metric in this evaluation. For each recommendation technique we measured the $success@3$ for each group using each expert’s list as the benchmark for evaluation. For an overall success rate we averaged the results across the different group sizes and then averaged the results from each

expert.

5.2.8 Results

Profile Merging

We first analyze our results for the Profile Merging recommendation technique. Table 16 includes the success rate for the baseline and the WtIntGN Model across the different group sizes. For groups of size 2 the Weighted Intersect Group Neighborhood outperform the baseline with a success rate of 87.5%. Similarly, for groups of size 3 the WtIntGN has a higher success rate than the average baseline model scoring 83.3% versus a score of 77.8% for the baseline. For the other group sizes there was no change in the performance in terms of the success rate.

	Baseline	WtIntGN
Size 2	0.7917	0.875
Size 3	0.7778	0.8333
Size 4	1	1
Size 5	0.8571	0.8571
Size 6	1	1
Size 7	0.75	0.75

Table 16: Success@3 comparing the baseline to the weighted group neighborhood selection using the profile merging technique for groups of size 2-7.

Recommendation Aggregation

Next we analyze the results for recommendation aggregation recommendation technique. Table 17 includes the success rate for the baseline and the WtIntGN Model across the different group sizes. For groups of size 2 the Weighted Intersect Group Neighborhood outperform the baseline with a success rate of 81.25%. Similarly, for groups of size 3 the WtIntGN had a higher success rate than the average baseline

model scoring 89.9% versus a score of 83.3 % for the baseline. For the other group sizes there was no change in the performance in terms of the success rate.

	Baseline	WtIntGN
Size 2	0.7708	0.8125
Size 3	0.8333	0.8889
Size 4	0.9375	0.9375
Size 5	1	1
Size 6	1	1
Size 7	0.75	0.75

Table 17: Success@3 comparing the baseline to the weighted group neighborhood selection using the recommendation aggregation technique for groups of size 2-7

Across the Board

We also wanted to examine the performance of the model over all the groups. Table 18 includes the success values for all the groups across the various neighborhood models for both Profile Merging and Recommendation Aggregation. From these results we can see that the Weighted and Frequency Intersect Group Neighborhood models performs best. With a success rates of 88.76% and 86.73% for the Profile Merging and Recommendation Aggregation respectively.

Results show similar patterns to the reported results spanning the group sizes. For the Profile Merging approach and the Recommendation Aggregation, the WtIntGN model had a higher success rate compared to the Baseline model. We recall from the dataset details that the majority of the groups fall in the size 2 category (50 groups of size 2, 18, 16, 7, 5, 4 for sizes 3, 4, 5, 6, and 7 respectively). We perceive the results obtained for this group size as more effective and indicative of the difference in performance of the evaluated models. For either group recommendation technique, assigning a higher weight to a neighbor, if they are shared with more than one group

	Baseline	WtIntGN
PM	0.83673	0.88776
RA	0.83673]	0.86735

Table 18: Success@3 with Profile Merging (PM) and Recommendation Aggregation (RA) for 100 groups

member, increased the success rate when compared to a baseline neighborhood approach. In the baseline neighborhood approach all neighbors are considered to have the same influence in the group context and neighbors are only weighted by their similarity to the individual group members.

5.2.9 Discussion

In this experiment, we carried out further exploration in the space of neighborhood identification in group-based recommendations when employing a Collaborative Filtering recommendation technique. We identified a neighborhood model given a group context as well as a weighting scheme incorporating these neighborhoods. We have evaluated our approach using a success metric for a list of recommended items. We reported results for different group sizes (2-7) and group recommendation strategies (profile merging, recommendation aggregation).

For the profile merging strategy, accounting for a higher influence of users that are neighbors of more than one group member resulted in an increase in success rate when generating a list of three recommended items. Our results shows a 6% increase in success rate, compared to the baseline neighborhood CF.

Similarly, for the recommendation aggregation approach, accounting for a higher influence of users that are neighbors of all the group members resulted in an increase in success rate (4%).

Given that this evaluation is based on “real” user judgment, we believe that our approach has significant grounds in extending these results to real groups, not just synthesized groups.

CHAPTER 6: GROUP MODELING IN GROUP-BASED RECOMMENDER SYSTEMS

6.1 A Hybrid Approach to Group Recommender Systems

6.1.1 Motivation

This study focused on examining a hybrid approach in generating recommendations for groups. Burke's [14] research in hybrid recommender systems highlights the fact that no one recommendation technology or strategy is likely to be optimal in any given recommendation scenario. As a result, considerable attention has been paid to the prospect of developing hybrid recommendation strategies that combine individual approaches such as content-based and collaborative filtering techniques [15]. In fact, in the Netflix Prize, the winning team has used a hybrid approach combining more than 100 models in their final solution [41].

When it comes to group recommendations, limited hybrid approaches have been considered. Berkovsky et al. [6] evaluated a switching hybridization strategy, which selects a recommendation strategy to apply according to the user data density. They apply density thresholds for switching between the general, aggregated models, aggregated predictions, and personalized group recommendation techniques. The switching strategy was discovered to be superior to all the individual strategies across both of the accuracy metrics and obtained extremely high coverage of recommendations.

In Garcia et al.[25], their group travel recommendation system used a hybrid rec-

ommendation technique to combine demographic, content-based recommendation and preference-based filtering. Their strategy applies a mixed hybrid approach to combine the recommendation lists generated by the three recommendation techniques to produce a final recommendation list. Their evaluation compared two preference aggregation strategies (aggregation, intersection), that used a hybrid recommendation technique, to a preference aggregation strategy (incremental intersection) that produced one list of recommended items.

Since hybrid approaches have resulted in better performance with fewer drawbacks than any individual approach, we believe that a hybrid approach to the group recommendation techniques would result in higher prediction accuracy. Our initial results revealed insight on the performance of the aggregation strategy (average, least misery, most happiness) when compared to group characteristics (size, cohesiveness). On the same line, we believe that a hybrid approach to the aggregation strategy, depending on group characteristics, would result in higher prediction accuracy. We examine applying a weighted, hybrid approach to the recommendation strategy (aggregated preferences and aggregated predictions) as well as a switching hybrid approach, based on group characteristics, to the aggregation strategy (average, least misery,...).

6.1.2 A Hybrid Group Recommendation Strategy

6.1.2.1 Hypothesis

H6 - A weighted hybridization of predictions, generated by aggregating the individual preferences of the group members into one group preference profile, with predictions generated by aggregating the predictions generated for each group member, will result in a higher prediction accuracy for the group recommendation.

6.1.2.2 Methodology

To evaluate the hypothesis we will evaluate the prediction accuracy generated by the hybrid approach against the prediction accuracy generated by the single approaches: preference aggregation and predication aggregation. Prediction accuracy metrics, as discussed in Section 1.3, will be used. Hypothesis 6 can be formalized as follows:

Let Gr_{iA} equal the predicted group rating for item i computed by approach

A

Let Gr_{iB} equal the predicted rating for item i computed by approach B

Then, the final group predicted rating Gr_i is computed by employing a weighted hybrid approach.

$$Gr_i = \alpha Gr_{iA} + \beta Gr_{iB} \quad (9)$$

where $\alpha + \beta = 1$

6.1.2.3 Evaluation Process

1. Define the dataset used in this evaluation: we use the GD2 dataset created as defined in Section 3.2
2. Define the baseline prediction group aggregation strategy: we implement the Average, Least Misery, Most Happiness aggregation strategies.
3. Generate predictions for each group recommendation strategy (preference aggregation, predication aggregation).

4. Generate predictions applying a weighted hybrid (average) as in Equation 9.
5. Compare prediction accuracy of the weighted hybrid to the single group recommendation techniques.
6. Vary the weight distribution between α and β .
7. Generate predictions based on the different weight distributions.
8. Compare prediction accuracy of the generated predictions to find ideal threshold values (RMSE, MAE).

6.1.3 Results

We varied the weight distribution between α and β as follows:

$$\alpha = 0.5, \beta = 0.5$$

$$\alpha = 0.7, \beta = 0.3$$

$$\alpha = 0.3, \beta = 0.7$$

We compared the prediction accuracy generated using the Profile Merging (PM), Average Recommendation Aggregation (Avg), Least Misery Recommendation Aggregation (LM) and Most Happiness Recommendation Aggregation (MH), Hybrid with Average (HyAvg), Hybrid with Least Misery (HyLM) and Hybrid with Most Happiness (HyMH). Where in the hybrid approaches the final prediction for a group was calculated using Equation 9 and the values of α and β as defined above.

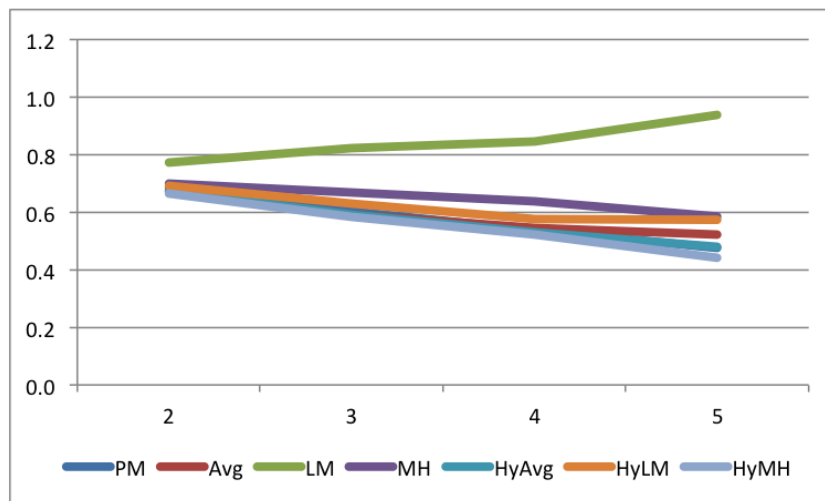


Figure 23: RMSE comparing PM, Avg, LM, MH, HyAvg, HyLM and HyMH for groups of size 2, 3, 4, and 5

Results showed that the hybrid approach resulted in higher accuracy predictions for the group in terms of RMSE calculated when compared to the baseline profile merging approach or the recommendation aggregation approach for all the combinations of α and β . The best results were achieved with the 0.7/0.3 weights with the higher weight assigned to the the Profile Merging approach. In particular the HyMH had the lowest RMSE making it the best performance approach in terms of accuracy prediction. Figure 23 shows the RMSE with respect to group size. An ANOVA test for all the approaches across the group categories showed that the difference is significant $p < 0.05$. Examining the difference between the winning hybrid approach and the best performing approach among the non-hybrids using a two-tailed paired TTest for each group size, all were significant with $p < 0.01$ except for size 3 where there was no difference between the HyMH and the Avg approaches.

Examining the overall RMSE for all the groups the HyMH had the best result in comparison to the PM and Avg with a $p < 0.001$. Comparing the HyMH to PM

there was a 3% increase and a 2.6% increase compared to the Avg approach. All approaches except the ones based on avoiding misery (LM, HyLM) had an increase in accuracy as the group size increased. What was interesting is that despite that the MH approach did not outperform the PM for any of the evaluated group sizes it outperformed all the other approaches when combined with the PM in the HyMH approach.

Next we examined the results with respect to both group cohesiveness and size. Figures 24-26 show the results for groups with high, medium and low inner group similarity respectively. We detected similar trends in performance as the results we observed in the overall analysis for the high and medium inner similarity levels where the HyMH had the highest accuracy and that accuracy increased as the group size increased. For the groups with lower similarity levels the Avg approach was best for all the group sizes (Figure 26). Looking at the HyMH (Figure 24) performance to compare it to the overall results and its performance for other group similarity levels we can see that as the group sizes increase its performance is lower than HyAvg and HyLM. Examining the non hybrid technique we can see that the approach based on LM has lower prediction accuracy compared to the MH and the PM approaches and both HyAvg and HyLm has better performance than the PM. Combining the PM with LM or Avg is better than PM but not better than Avg.

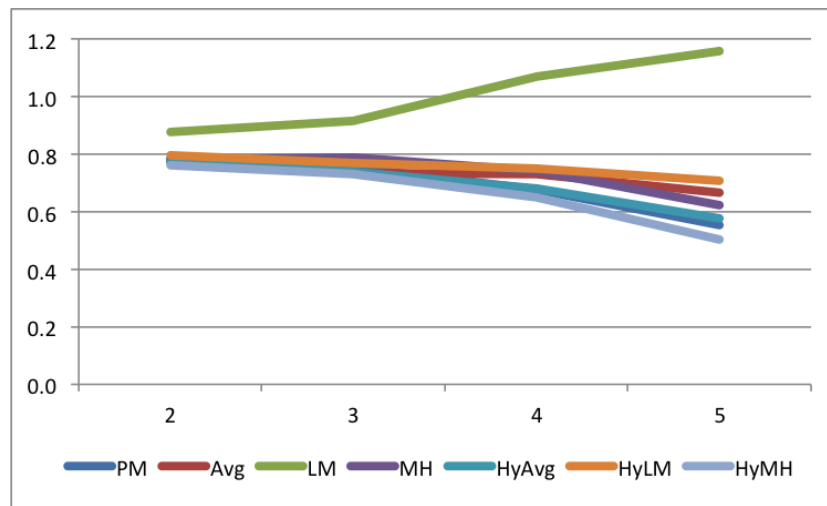


Figure 24: RMSE comparing PM, Avg, LM, MH, HyAvg, HyLM and HyMH for groups of size 2, 3, 4, and 5 and high level inner similarity among group members.

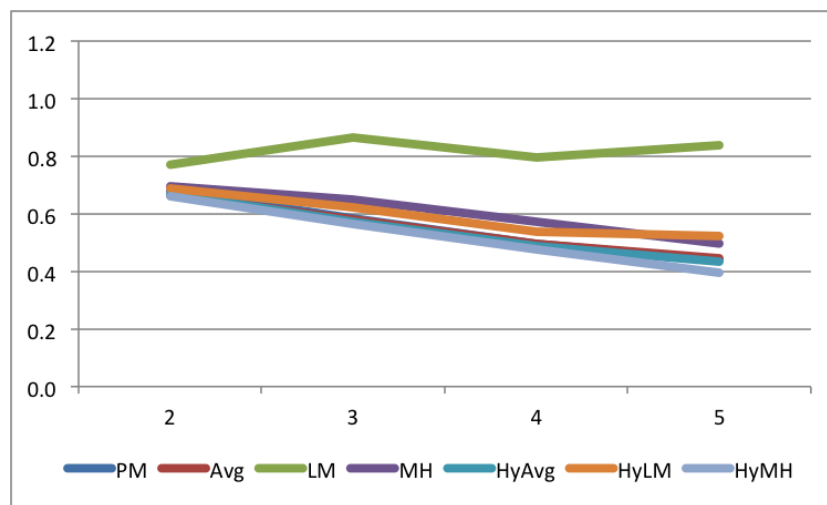


Figure 25: RMSE comparing PM, Avg, LM, MH, HyAvg, HyLM and HyMH for groups of size 2, 3, 4, and 5 and medium level inner similarity among group members.

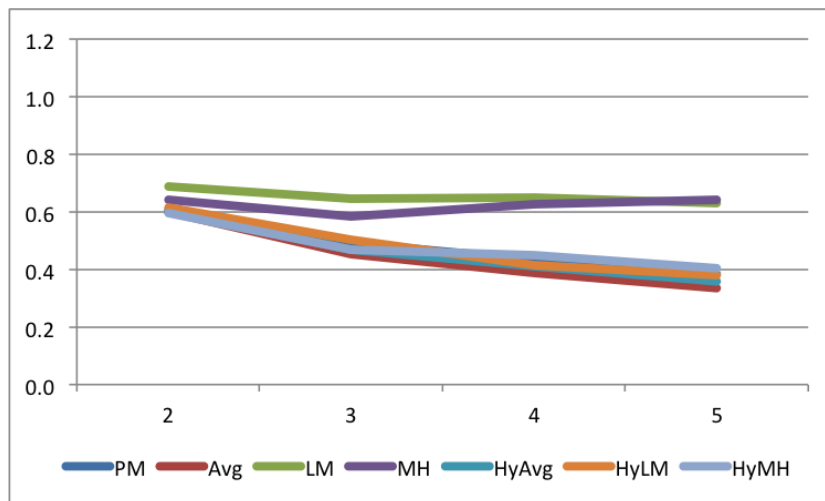


Figure 26: RMSE comparing PM, Avg, LM, MH, HyAvg, HyLM and HyMH for groups of size 2, 3, 4, and 5 and low level inner similarity among group members.

6.2 Disagreement Modeling in Group Aggregation for Group-Based Recommender Systems

6.2.1 Motivation

We pointed out that the main distinguishing feature of group recommender systems, when compared to individual recommender systems, is their need for a mechanism to aggregate individual preferences or aggregate the generated recommendations. For this reason, major research efforts focused on defining aggregation strategies to model real group interaction and achieve consensus for any given group task or activity. Section 1.4.2 overviews the baseline strategies that have been proposed, and in Section 2.2 we highlighted the major work that has been proposed as variations of these baseline strategies to achieve better group modeling. When it comes to modeling actual groups in reaching agreement for an item to consume two factors come to play. Either the group agrees on the item, or there's disagreement among the members and the need for a consensus arises. We believe that accounting for this disagreement as a

part of the aggregation strategy will result in a higher prediction accuracy for the group recommendation. In [1, 26] disagreement is accounted for, in aggregating the predicted individual ratings for an item, by combining the average of the individual predicted ratings with a disagreement score by applying a weighted linear combination. The disagreement score is calculated as the average of pair-wise predicted rating differences for the item among group members, or as the mathematical variance of the predicted ratings for the item among group members. In [63] they use the type of personality trait of dealing with conflicts, of each member, to weight the influence of their ratings in the aggregation model.

Conflict is a natural part of interactions with others. In most groups the different group members have different expectations and desires that usually appear to be incompatible, leading to “conflict situations” and the need for conflict resolution arise. We hypothesize that modeling the disagreement using ratings provided by the group members for an item that is similar to the item for prediction will result in higher prediction accuracy for the group recommendation. We adopt Amer-Yahia et al. formalization for group disagreement. Disagreement is modeled in two ways, average of pairwise disagreement and score variance. We base the model of resolution of disagreement on known preferences of similar items to the target item.

6.2.2 Disagreement Models

We formalize the disagreement models as follows:

Baseline group disagreement models based on the average and variance of a predicted rating among group members as defined in [1]:

1. Average pairwise disagreement coefficient

$$dis(G_i) = \left(\frac{2}{|G|(|G| - 1)}\right) * \sum_1^n (p_{ui} - p_{vi}) \quad (10)$$

for every u and $v \in G$ and $u \neq v$

2. Disagreement variance coefficient

$$dis(G_i) = \left(\frac{2}{|G|(|G| - 1)}\right) * \sum_1^n (p_{ui} - \hat{p}_{Gi})) \quad (11)$$

where \hat{p}_G is the predicted group rating for item i calculated as the average of the predicted rating for each member for that item.

Our hypothesis bases the disagreement coefficient on the known rating of an item that is rated by group members rather than the predicted rating for the test item. What we need to define is the item-to-item similarity metric and threshold.

We make use of the Pearson Correlation to calculate the similarity between all the items in our database and we set the threshold for the item-to-item similarity to be

0.1. The final group prediction can be formalized:

$$p_{Gi} = (w_1 * \hat{p}_{Gi}) + (w_2 * (1 - dis(G_i))) \quad (12)$$

6.2.3 Hypothesis

H7 - Predictions generated by accounting for the difference between the aggregated prediction for the group and the ratings provided by each member for a similar item will result in higher prediction accuracy for the group.

6.2.4 Methodology

To evaluate the hypothesis, we will evaluate the prediction accuracy generated by a baseline prediction aggregation strategy to the prediction accuracy generated by accounting for disagreement among group members over a similar item in the data set. Prediction accuracy metrics, as discussed in Section 1.3, will be used.

6.2.4.1 Evaluation Process

1. Define the dataset used in this evaluation: We use the GD2 dataset created as defined in Section 3.2
2. Define the baseline prediction group aggregation strategy: We implement the Average, Least Misery, Most Happiness aggregation strategies.
3. Define the disagreement model (average, variance): We implement both average (Equation 10 and variance disagreement models (Equation 11)
4. Define a threshold for the item similarity in order for an item to be considered in the disagreement calculation: We defined a threshold of 0.5 and 0.1 for the item-to-item similarity level measured using the Pearson Correlation Coefficient as defined in Equation 1.
5. Generate predictions for each group by aggregating the individual predictions into a group prediction employing the baseline strategies.
6. Generate predictions for each group by aggregating the individual predictions into a group prediction employing the disagreement model.
7. Compare prediction accuracy of the prediction with disagreement to the baseline group aggregation strategy: We compare accuracy using the RMSE metric.

6.2.4.2 Results

The first question we needed to answer in order to evaluate this hypothesis with the data set we selected was to find out the number of test-item/group pairs where there would be a match for the test item in the commonly rated items for that group. We set the item-to-item similarity for the first run to be 0.5. We consider this threshold to represent a highly similar item. For all the test groups (1200) and test items (10500) in this data set there was no match. We attributed this to the fact that the way the test/training set was created by using the commonly rated items among the group members as the test items. So in most groups all of those were considered test items not leaving a commonly rated item in the training set to be considered for the disagreement coefficient calculation. To overcome this we then considered the entire MovieLens 1M data set to look for the similar items rather than the specific group training set. Since the baseline predictions were already generated using each group's training set we can look for a commonly rated item in the original data set for that group without affecting the original predictions. Surprisingly there was still no matching test items and commonly rated items for any group that satisfied the item-to-item similarity threshold.

We next thought of lowering the similarity threshold to check at what threshold we would have a match. We lowered the threshold to 0.1 with no matches. Having a lower similarity than 0.1 would mean that we would be considering items with no to low similarity levels with the test item which was not what we were set to investigate. Since the applicability of a model goes in line with its validity and the disagreement

model we proposed was never valid in this data set, hypothesis H6 is rejected.

Even though we were not able to validate our hypothesis we got results for the disagreement models proposed by Amer-Yahia et al. [1]. Figures 27 and 28 show the RMSE results for Average baseline strategy, the Average with Average pairwise disagreement and the Average with Disagreement Variance. We varied the weights between 80/20 and 20/80 for Equation 12 and Figures 27 and 28 show these results respectively. In both cases the Average baseline approach has a lower RMSE resulting in higher accuracy predictions than the models incorporating disagreement. In their work [1] they reported that the predictions that incorporated disagreement were more beneficial to groups with lower similarity levels among the group members using the nDCG metric to evaluate the quality of a ranked list. We did not see this in terms of prediction accuracy measured using RMSE. We saw a similar trend to the figures below that include all of the groups in our data set for any group size and similarity levels we evaluated. Even though their evaluation was based on real users the final group judgement was still based on the individual responses of each group member and overall they had one group for each category they evaluated.

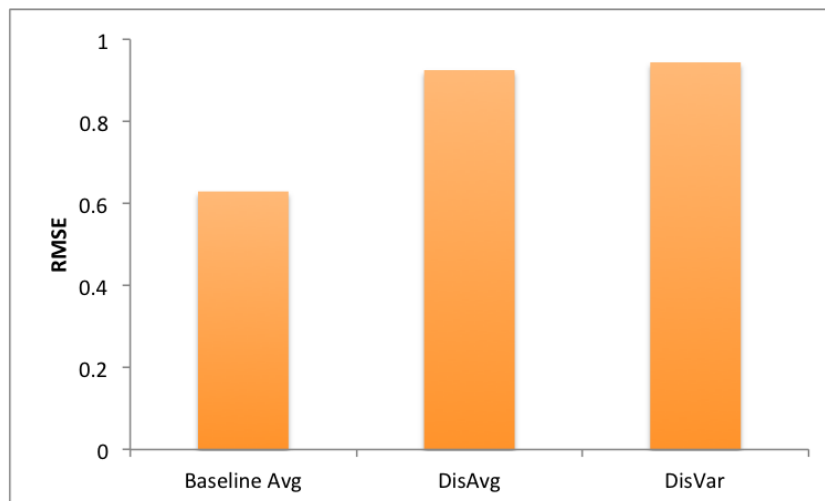


Figure 27: Disagreement 80 20 weights

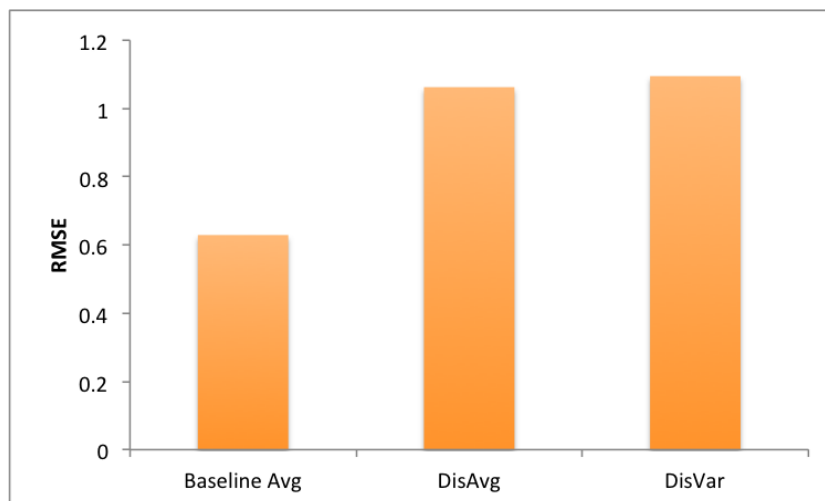


Figure 28: Disagreement 20 80 weights

6.2.4.3 Summary

Even though we were not able to validate our hypothesis using the data set we had available we were able to compare previously proposed disagreement models in a comparatively large evaluation. We implemented a different evaluation metric than the metric used previously. Our results show that these disagreement models did not improve the accuracy of predicted ratings for a group. On the contrary, the

prediction accuracy was negatively affected when disagreement was accounted for based on the predictions generated for the test item for each group member. Group size and similarity level did not affect this observation.

We also found out that was not possible to find a similar item rated by all the group members in the data set we employed. Models depending on such an item might not be feasible in the recommendation process for a group.

At this point we are thinking of the group context as the basis of the recommendation technique. Despite our findings in this evaluation we still believe that utilizing the group context as an initial point of reference rather than the individual members for the recommendation technique would be more beneficial for the group by producing prediction with higher accuracy. The question that remains unanswered here is how can we model this group context?

In this evaluation we explored utilizing disagreement models and item-item similarity in modeling the group context but found that such models might not be applicable in the context of the group dataset we employed in our evaluation. In the next chapter we explore modeling the group context based on a group-to-group similarity perspective and metric. We take advantage of other, previous group contexts as a first-class element for the recommendation technique before we employ the group aggregation strategy to generate a prediction for an item for any given group. We retrieve and reuse a whole, previous experience of another group which is considered to be a CBR perspective on group-based recommendation that we formalize and evaluate in the next chapter.

CHAPTER 7: CASE-BASED REASONING APPROACH IN GROUP RECOMMENDER SYSTEMS

7.1 Motivation

Case-Based Reasoning (CBR) has a long history as a methodology for building recommender systems [11]. In general, case-based recommendation embodies a content-based or knowledge-based approach, with a semantically rich representation of users and/or items (e.g., [51]). However, a number of researchers have also taken a case-based perspective on ratings-based profile representations as employed in Collaborative Filtering recommendation [12, 32, 59, 58, 62], and we adopt this perspective.

Though there are a number of such examples, group recommendation has received comparatively little attention overall. Recent research has largely focused on making straightforward individual recommendations for each group member and then aggregating the results. But this considers only the group context of the query or problem specification (on final aggregation), and does not take advantage of other, previous group contexts as a first-class element of the knowledge base. Retrieving and reusing the previous experience of another group — taken as a whole — is a natural application for CBR. Case-based approaches to group recommendation have appeared, but are quite few. To our knowledge, only the seminal work by Quijano-Sanchez et al. [60] has examined retrieval of entire groups as cases, as opposed to the aggregation of individual retrievals.

To help place our investigation in perspective, we here provide a more extensive consideration of [60] as related work in motivating our study. For example, their approach combines many subtle aspects of the group recommendation process all at once (some requiring imputed value support). Since there was not an ablation analysis, this naturally raises questions about the relative contributions of different aspects. Differently from [60], our technique is less complex overall, which enables a clearer understanding of the contribution provided by baseline case retrieval. The evaluation metric (Success@1, etc.) reported in [60] has not been commonly used, and they did not report on significance, which makes comparisons to other approaches more challenging. Here we evaluate with a more widely employed metric, Root Mean Squared Error (RMSE) with reported significance. Their study case-base included only groups of same-age-range for its group members, whereas we examine explicit conditions for intra-group coherence.

The case-base is both a strength and limitation of the previous study. Data sets with real group decisions are difficult to create and not readily available, so an expert crafted case-base helps to address validity of the group decision data. At the same time, manual review / creation limits the scope of experimental studies, which has commonly been addressed by generating synthetic groups on a larger scale (e.g., [5]). And so, we expand upon their initial exploration of this interesting space with our own studies. While we look at a general accuracy comparison with [60], the main point here is to better understand the nature and effectiveness of CBR approaches that retrieve and reuse whole previous groups for group recommendation. In particular, to help establish baseline effectiveness for straightforward CBR techniques, as

a context for understanding more complex approaches. This includes a streamlined CBR approach, and limiting imputed values for a clearer baseline understanding. It also includes fixing the group size between the active group and the cases considered in the recommendation process eliminating the need to use virtual users. We examine different conditions than [60], such as adaptation from explicit top-1 vs. top-N perspectives, as well as explicit experimental conditions for group homogeneity. Our experiments examine a larger overall case-base size of 1200 groups in comparison to the 100 groups for the cases used in [60].

In this evaluation we carry out further exploration of the space, investigating the effectiveness of case-based reasoning approaches to retrieve and adapt whole previous groups as a basis for recommendation. This chapter presents our foundational approach for case-based group recommendation and evaluation results that investigate a variety of conditions for group size and homogeneity. Section 7.2 introduces our research question. Section 7.3 formalizes the recommendation techniques investigated in our group-based recommender, and experimental outcomes are discussed in Section 7.5. Our results show that foundational group-to-group approaches can outperform individual-to-group recommendations across a wide range of group contexts.

7.2 Exploring Whole-Group Case-Based Reasoning

Our main research question in this experiment is to examine under what conditions will taking advantage of existing group contexts in the knowledge base — in addition to the group context of the target / active group (as query) — improve group recommender performance. That is, if we take a CBR perspective on group recommendation, retrieving whole previous groups as the starting point for predictions

instead of directly aggregating on individuals, when will there be an overall benefit for the system? To investigate this issue, we integrate a foundational CBR component (Section 7.3) with a common group recommender technique, evaluating across a variety of conditions. In this study, we focus on understanding how much traction is really possible with a straightforward whole-group retrieval CBR approach. And so we embrace the limits of the process — limiting retrieval to groups of the same size, allowing for retrieval failure, and so on.

The group recommender component provides a standard aggregation of individual recommendations for group members. The case-based reasoning component retrieves similar groups from the case-base and adapts retrieved group preferences to the target group query. For this study, there is no guarantee of complete coverage by the case base, and so the system acts as a “switching” hybrid recommender [14]: if the CBR component can not make a recommendation with sufficient confidence, then the baseline aggregation method is used. This is similar in spirit to integrations of CBR with generative planners in case-based planning [19, 72]. The standard aggregation technique also serves as the baseline for system evaluation, comparing the CBR approach (with switching as needed) to using only the baseline for each query. Our evaluation examines three main hypotheses.

7.2.1 Hypotheses

H8 - The foundational CBR approach will be able to respond to a substantial number of queries across a range of conditions.

H9 - The hybrid CBR approach will provide significantly better accuracy than the baseline.

H10 - Accuracy results for the foundational CBR approach will show similar trends to those reported in [60]. Significance testing is considered for $p < 0.01$.

7.3 Case-Based Group Recommender

Our case-based group recommender employs ratings-based user profile data as the foundation of the case-base. We formalize our approach as follows. Given a matrix of users (U) and items (I) a case is represented as $\{G_{cb}, I_{cb}\}$. G_{cb} is a group of users, u_{cb} , of size n where $u_{cb} \in U$. I_{cb} is a set of items i_{cb} where i_{cb} is an item rated by all the users in G_{cb} and $i_{cb} \in I$. The active group that is seeking the recommendation is represented as G_a .

7.3.1 Active Group to Case Similarity Metric

In order to retrieve previous group-cases that are relevant to the active (query) group, we define a similarity metric that considers the correlation between each user in the active group and every user in the case group. Similarity between G_a and G_{cb} is measured by considering the *cartesian product* of these two sets. Which is the set of all ordered pairs (u_a, u_{cb}) where $u_a \in G_a$ and $u_{cb} \in G_{cb}$. Let this set be represented as $CProd_G$ and contain the correlations that need to be calculated to measure the similarity between the active group and a case from the case base. Since we only consider cases with the same size as the active group's size ($|G_a| = |G_{cb}|$) then the size of the set $CProd_G$ is equal to $|G_a|^2$. This is the number of correlations that we need to consider to calculate the group to case similarity. In the next step, we calculate the correlation for all elements of $CProd_G$ using Equation 1. The resulting correlations form the group-to-case correlation set $PCorr_G$ where only the possible correlations are stored. The final case similarity can then be calculated as the average

of the $PCorr_G$ set. We note here that the similarity metric we use is based only on the user's ratings, differently from [60] where the similarity, in addition to ratings, includes trust, age, gender and personality.

$$GG_{sim} = \frac{\sum_{g=1}^n PCorr_{Gg}}{|PCorr_{Gg}|} \quad (13)$$

7.3.2 Case Retrieval

Using the defined similarity metric, cases are retrieved using the following constraints. First, only cases of the same size as the active (query) group are considered. Second, only cases that meet a defined similarity threshold θ are considered. Subject to these constraints, the top-N set of identified cases is represented as GG_{CB} . For this study, we do not consider a limit on the number of cases that meet the specified constraints, and we examine prediction strategies based on the single best case and on the entire retrieved set. In the circumstance that no case meets the selection criteria, the null set is returned to indicate failure, which serves as a trigger for the hybrid switching mechanism.

7.3.3 Adaptation for Recommendation

In order to make a prediction on an item rating for the active (query) group, we adapt the retrieved case(s) on the item in question. We also consider several different adaptation variants: either the most similar retrieved case or a set of retrieved cases, as well as scaling. Formally, we examine the items in I_{cb} and the item for which we are calculating a prediction. Let us refer to that item as the active item i_a . Since we are trying to model the group decision for a certain item and attempting to minimize

variables in this approach we choose to base our model on cases where the active item considered for recommendation is present in the item set for the case-base group ($i_{cb} == i_a$). Applying this condition to the set of possible cases, GG_{CB} is reduced to \widehat{GG}_{CB} .

We consider two main adaptation variants. The first adapts only the single, most-similar group-case that contains i_a . The second adapts the entire set of retrieved cases that contain i_a (\widehat{GG}_{CB}). Within each of these two variants, we consider whether group ratings should be normalized between the active group and the retrieved case(s).

7.3.3.1 Adaptation — Best Case / Multi-Case

In the first variant, we consider only the case with the highest similarity to the active group. We refer to this approach as *CBR_single*. Here the case group rating's are used as the basis for the prediction. The prediction for the active group is modeled as the average of the individual rating for the active item i_a given by the members of the case group G_{cb} formalized as:

$$pG_a i_a = \frac{\sum_{u_{cb}=1}^n r_{u_{cb}}}{|G_{cb}|} \quad (14)$$

Where $r_{u_{cb}}$ is the rating for i_a by user u_{cb} and $u_{cb} \in G_{cb}$.

In the second variant, we consider the entire set of retrieved cases. We calculate a prediction for each participating case, as in equation 14, and average the results, formalized as:

$$pG_a i_a = \frac{\sum \left(\frac{\sum_{u_{cb}=1}^n r_{u_{cb}}}{|G_{cb}|} \right)}{|\widehat{GG}_{CB}|} \quad (15)$$

7.3.3.2 Adaptation Scaling

We also consider a scaling condition for each variant, which normalizes the rating scale between the active group and the retrieved case(s). This is formalized as:

$$\hat{p}G_a i_a = \overline{G_a} + (\sigma_{G_a} * \frac{\sum ((\frac{pG_a i_a - \bar{u}_{cb}}{\sigma_{u_{cb}}}) * GG_{sim})}{(GG_{sim} * |G_a|)}) \quad (16)$$

where σ_{G_a} is the average standard deviation of ratings for the users in G_a , $\sigma_{u_{cb}}$ is the standard deviation of ratings for the users in G_{cb} and $\overline{G_a}$ is defined as:

$$\overline{G_a} = \frac{\sum \bar{u}_a}{|G_a|} \quad (17)$$

where \bar{u}_a is the average rating for a user u_a in G_a . In other words, $\overline{G_a}$ is the average of the average ratings for the users in G_a . The value $pG_a i_a$ is calculated using either CBR_single or CBR_multi.

7.4 Experimental Setup

Recommendation Parameters

As a basis for evaluation we employ the common neighborhood-based CF algorithm as outline in Section 4.2.2.1. We set the neighborhood size to 50 based on the recommendations of [33].

7.4.1 Accuracy Measurement

We measure the accuracy of a predicted rating computed for a group across different test conditions using root-mean-square error (RMSE) [34]. To measure the differences between values predicted by a model and the actual values, we compare the group-predicted rating calculated for the test items, using the aggregation approaches

described (Average, Least Misery, Most Happiness), to a model of the actual rating (average) across the different group sizes and inner-group similarity levels.

Another evaluation strategy is to compare a recommended list of items to an actual preferences list. A variant of this strategy, *success@n*, was employed in [60] to measure the rate of having at least one recommended item in the top n positions of the actual preferences list. For example, given an ordered set of recommended items *recList* of size n and an ordered set of the actual preferences *actList* of the same size, *success@3* would return 1 if at least one of the items in the top 3 positions of *recList* appears in the top 3 positions of *actList* and 0 otherwise. We use the *success@n=1,2,3* metric to make a general comparison of our results to the approach in [60].

7.4.2 Case Base

In this experiment we utilize the GD2 dataset defined in Section 3.2. Each group in this set is considered as a case that can be used as the basis in the prediction generation for the target group.

7.4.2.1 Case Retrieval

To retrieve cases that may be used to model group recommendations, we define the threshold θ for group to case similarity (GG_{CB}) with a value of 0.5. This similarity threshold ensures a highly similar group is used as the case-based group. Taking each group at a time as the active group (G_a), we calculate the similarity between that group and all the other groups. The results would be the case groups (G_{cb}) that have the same size as the active group regardless of the groups' inner cohesiveness. Cases that have a group-to-case similarity level higher than the threshold with the active group are then considered for the prediction calculation phase. We note that

similarity threshold dependency could also be analyzed to provide a perspective on results for different levels. In this study, we select a representative level to understand initial results, leaving threshold dependency analysis for future work.

We explore outcomes of prediction accuracy for recommendation aggregation using the most commonly used group modeling strategies (Average, Least Misery, Most Happiness) as outlined in Section 4.2.2.2. We examine these outcomes with respect to group size and inner group similarity. We contrast this approach, using the baseline neighborhood approach to the *Case-based* model approach, by comparing the prediction accuracy of the predicted group rating of each test item to the average of the actual ratings of the individual group members for that test item.

7.5 Results

For each created group and test item we calculate a predicted rating using the baseline group recommender (Section 4.2.2.1) and the case-based group recommender (Section 7.3).

The first hypothesis (H6) we examine is that the CBR approach will be able to respond to a substantial number of queries. Given the constraints, we would consider a baseline 10% response rate for the generated groups to provide an indication of reasonable traction. To some degree, this is a function of the case-base generation model, but it is critical to understand the context of accuracy. It is useful to know for the given similarity threshold in the experiments, the degree to which the approach is able to respond, as this provides context for interpreting the accuracy results. We also note that the model has some influence on intra-group similarity, but not on inter-group similarity, within the generated case-base. To test this hypothesis, we

Table 19: Number of case-based groups for the different similarity level and size groups

	High	Medium	Low
Size 2	63	81	81
Size 3	40	36	9
Size 4	52	11	1
Size 5	90	8	1

Table 20: Number of case-based items for the different similarity level and size groups

	High	Medium	Low
Size 2	181	326	298
Size 3	68	51	14
Size 4	108	14	1
Size 5	367	9	2

inspect the number of items and groups where the case-based approach is applied for each group category. Tables 18 and 19 show these numbers for the different groups. We compare these numbers to Table 4, which represents the overall number of testing items for each of the 100 groups evaluated in each category. This shows the categories in which the CBR approach was best able to respond to prediction queries. We can see that the CBR approach was most applicable for groups with high inner similarity levels.

For these groups the CBR approach was overall in effect for 25% of the test items in 60% of the groups. We also notice that as the group size increased, for the highly similar groups, these numbers increased. For example, in groups of size 5 with high similarity among the group members, the CBR approach was applied for 68% of the test items across 90% of the groups. Thus, if the group is highly cohesive and the group members are more similar to each other, then the cases that are used are more likely to be of high cohesiveness as well. This, in turn, increases the possibility for those cases to have more commonly rated items within the case group, which in turn, increases the likelihood of the active group's test items being shared with the case groups. This is not the case for groups with low inner similarity levels, where these numbers go down as the similarity level for the group decreases and the size

increases. For example, in groups of size 4 and 5 with low similarity level, the CBR approach responded for 1 and 2 test items respectively in only 1 group. Overall, for the representative similarity threshold, the foundational CBR approach was applicable for a substantial number of test queries (H1 accepted), and the applicability of the approach increases as the size and inner group similarity increases.

Our second hypothesis (H7) is that the hybrid CBR approach will provide significantly better accuracy than the baseline. To examine this, we first compare the results of the two case-based approaches, adaptation and adaption scaling. The performance level between the two case-based variations themselves was not significantly different for that we present and discuss only the adaption scaling approach (as representative of both CBR variants) to the baseline approaches. Figure 29 shows the RMSE for the baseline group aggregation models and both scenarios of the case-based approach (CBR_{single} , CBR_{multi}) across the various evaluated group conditions. An ANOVA test for all the models across the group categories showed that the difference is significant ($p < 0.05$). From these figures we can see that the case-based group recommender outperforms the baseline recommender for groups with high and medium inner group similarity. Between the two case-based approaches the CBR_{multi} approach is a slightly better technique than the CBR_{single} , but not significantly so.

In comparison to the baseline, the CBR approach results in significantly ($p < 0.01$) higher accuracy predictions for groups with high inner similarity levels (top graph). The performance of this approach also increases as the group size increases for all similarity levels. For groups with low similarity levels no difference is reported between the performance of the CBR group recommender and the average baseline

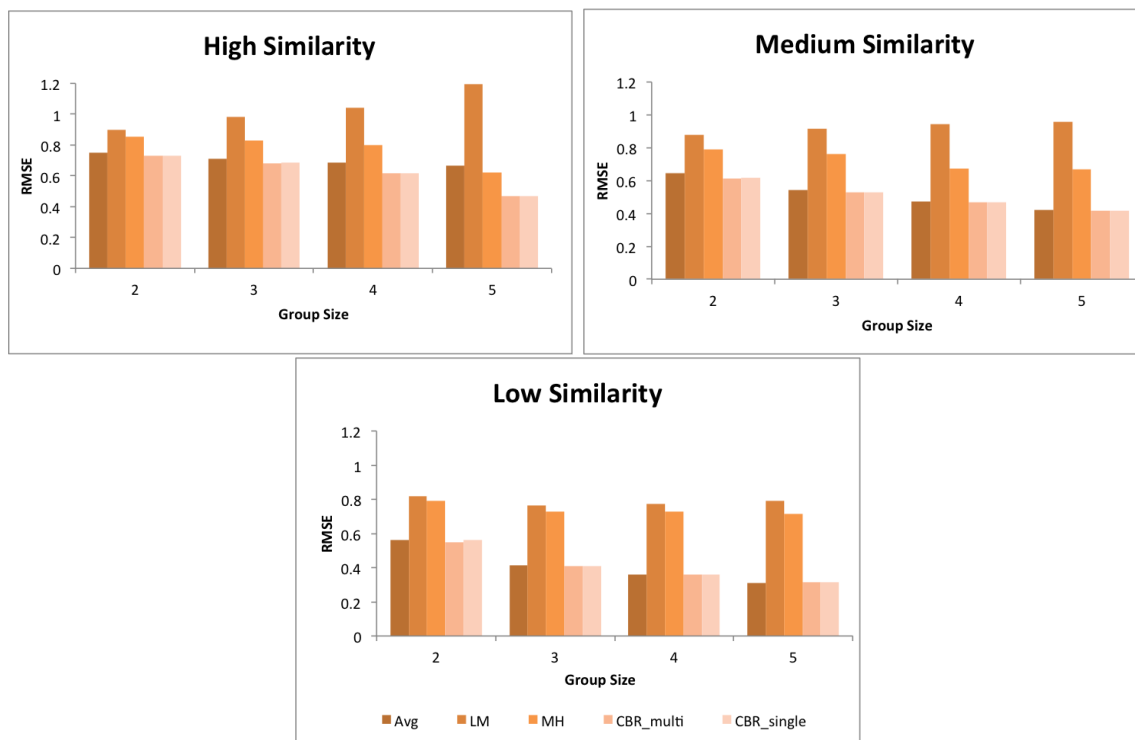


Figure 29: RMSE for groups across the different inner similarity levels

recommender. In this context, the CBR component largely failed to find matching cases. In that circumstance, the group recommender switches to the baseline leading to similar results. We also note for this category (low similarity) of groups of size 2, where 15% of the test items used the case-based recommender in 81 groups, the CBR recommender reported better results. Overall, these results indicate that a significant improvement in accuracy can be achieved by using a case-based group recommender (H2 accepted), where similar cases are considered as part of the prediction calculation.

Our third hypothesis (H8) is that our results would show accuracy trends for a foundational CBR approach to be similar to those reported in [60]. We were very grateful to receive the dataset from [60] for a direct comparison. The details of this dataset is in Section 5.2.5. Given the limited size of this dataset (100 groups/cases)

and the case-to-case size and similarity constraints there was no match in the case retrieval for any group that could be used as the basis for the recommendation. This led to a complete and direct comparison being beyond the scope of the current work, given the overhead for full reimplementing and testing of their approach. However, we are able to make a general comparison, using their *success@n* evaluation metric as a basis. They reported *success@n* evaluation ranking values for $n=1,2,3$. Since it is a ranking metric, we need to be able to predict ratings for more than three items for each group from our case base, and to generate a ranked list of recommendations. To do so, we selected the groups that had a test set with 5 or more items where our case-based approach can be applied to generate predictions for those items. From the 1200 we were able to identify 71 groups that satisfied the condition. For the true ranked list used to compare the predictions, we employed the average rating provided by the group members for those items as the ground truth employing a random selection to resolve equally ranked items. Table 21 shows the success rates at various n . Note

Table 21: Success@n rates for our CBR approaches and [60]. * indicates value estimated from figure.

	SUCCESS @1	SUCCESS @2	SUCCESS @3
CBR_multi	54%	83%	97%
CBR_single	69%	89%	100%
Quijano-Sanchez et al. [60]	12%	61%	92%*

that exact numbers were not reported for all levels in [60], and so some are estimated based on the graphs showing their results (as noted). The percentage success rate results show that our approaches can provide comparable (arguably better in some cases) success rates to those previously reported, and a similar trend (increasing as n

increases) to those reported in [60] (H3 accepted). While this comparison is limited in comparing outcomes across data sets, it serves to corroborate previous results on the potential benefit of CBR approaches in the context of a different case-base and group conditions. Moreover, it indicates that the benefits of a CBR approach may be realized even in a foundational implementation, without substantial demographic or data imputation requirements.

7.6 Summary

In this evaluation, we investigated different foundational CBR strategies for retrieving and reusing whole previous groups as a basis for making group recommendations. Inspired by [60], we carried out further exploration of the space, investigating the effectiveness of case-based reasoning approaches to retrieve and adapt whole previous groups as a basis for recommendation. Results showed that even a straightforward CBR approach can be useful across many group recommendation circumstances, providing a significant performance benefit. In comparison with previous work, we examined different similarity and adaptation metrics, larger case-base size, and multiple additional experimental conditions for group size and cohesiveness across the case-base. We confirmed the potential benefit for integrating whole-group retrieval CBR approaches into group recommendation across different case-base and group conditions. We also demonstrated that the benefits of a CBR approach may be found even in straightforward implementations, showing the potential for a broad range of deployments and investigation in the space. Going forward, we plan on conducting larger evaluations for larger group sizes and case-bases; examining other variations in retrieval and adaptation, such as similarity thresholds; and considering the impacts

of case storage and case-base maintenance.

CHAPTER 8: CONCLUSION

8.1 Dissertation Contributions

In Chapter 1, we outlined the main research question we are trying to answer: how to improve the prediction accuracy in group-based recommender systems employing a memory-based collaborative filtering technique? we answer this question by examine factors that influence the accuracy of recommendations made to groups rather than individuals along the contexts of evaluation, group modeling, and the recommendation technique. We break this down to the following more specific questions:

- How to implement a principled approach to evaluate the prediction accuracy in group-based recommender systems using datasets of individual users' preferences?
- Does the evaluation approach affect the results for the prediction accuracy?
- Does rating normalization increase the prediction accuracy for the group?
- Does incorporating the group-context in the neighborhood selection increase the prediction accuracy?
- Does a hybrid group-based strategy increase the prediction accuracy?
- Does incorporating the group context in the group modeling increase the prediction accuracy?

For evaluation in group-based recommender systems, we designed a testing framework that scans a dataset consisting of individual users' preferences in a collaborative

filtering based system and creates groups of users based on predefined group characteristics. The system can also identify possible test items for these created groups. This framework provided us with a principled evaluation approach to group recommenders using synthetic groups. One major issue we identified in evaluating group-based recommender systems using synthesized groups is the absence of a concrete actual group preference (ground truth). Our initial evaluation focused on ways of modeling the ground truth for the groups in our dataset. We analyzed outcomes of group prediction accuracy based on an actual group preference model (AGPM) of an exact rating overlap where, for each group, we identified at least one test item where all the group members had the exact preference (rating value).

Despite the recent trend towards model-based CF methods, with leading methods based on matrix factorization, e.g., [36, 42], traditional memory-based (or neighborhood-based) approaches are still widespread due to their simplicity, explainability, and effectiveness [23]. Our initial evaluation goal was testing 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 hypothesized that aggregation results will support previous research for the aggregation strategies tested. In doing so, we investigated the relationship between the group's coherence, size, and the aggregation strategy used. Our results, using the 5000 synthesized groups for each category (60,000 total groups), coincide with the results reported by Gartrell [26] using real subjects.

In the absence of one true AGPM, the meta-issue arises of how to evaluate candidate evaluation models. In order to do so, there are essentially two traditional options: finding a baseline context for comparing AGPMs, or relative performance comparison among different evaluation models, keeping in mind that the dependent variable in such experiments is a component of the evaluation strategy itself. This issue of how to model the AGPM in the evaluation of group recommenders using synthesized groups motivated us to investigate alternative approaches for AGPMs. Of course, changing the evaluation baseline will show differences in outcomes, but we are interested in investigating the shape and extent of such differences as a way to inform the process of evaluation. In the second experiment, we examined the scope of potential tradeoffs in the choice of an aggregation model for recommendations for groups when actual group preference may vary. We laid out a survey of evaluation techniques in this context in order to provide group recommender system developers with a better understanding of the implications in choosing a particular AGPM baseline when evaluating their systems.

Results showed that the choice of an AGPM, in this context, results in different evaluation outcomes. The choice of an AGPM can also introduce a bias, particularly when the same aggregation strategy is used on the recommendation side. We showed that modeling the actual group preference does matter when evaluating group-based recommenders, and different models provide different results. Developers of such systems need to consider the tradeoffs when choosing a baseline for evaluation.

Next, we examined the recommendation technique aspect in group-based recommenders. Focusing on memory-based models, we compared the effect of rating nor-

malization using deviation-from-mean and z-score approaches on the prediction accuracy for the group. We analyzed predictions based on group size and cohesiveness levels as well as group modeling approach and strategy. We found that the rating normalization is overall better. The z-score approach results in predictions with higher accuracy, for groups with low inner cohesiveness within the group members, when using the average aggregation model with profile merging and recommendation aggregation. Non-normalized ratings are best employed if we are using the “Least Misery” or the “Most Happiness” with recommendation aggregation.

In the next evaluation, we explored the effect on prediction accuracy if special consideration is given to the neighbors of the group members if they are shared by one or more of the group members. We explored outcomes of prediction accuracy for profile merging and recommendation aggregation using the “Average” group aggregation strategy. We analyzed the results across the various group sizes. We made a comparison between the baseline Nearest Neighborhood recommendation technique and the Group Neighborhood Selection techniques as outlined in the previous section. We compared recommendation rankings based on prediction to the item’s ranking provided by the experts using the evaluation metric outline in the following section. Analysis showed that an advantage is gained for the group, in terms of prediction accuracy, when special consideration is given to neighbors that are in common to all the group members. This approach is mostly advantageous as the group size and inner-group similarity increased.

Continuing along the dimension of the recommendation technique, we carried out further evaluations of the neighborhood selection model in the space of neighborhood

identification in group-based recommendations when employing a Collaborative Filtering recommendation technique. We identified a neighborhood model, given a group context, as well as a weighting scheme incorporating these neighborhoods. We've evaluated our approach using a success metric for a list of recommended items rather than prediction accuracy (*Success@n* vs. RMSE). We employed a dataset that contained "real-user" judgement in comparison to a purely synthesized, group-based dataset. We reported results for different group sizes (2-7) and group recommendation strategies (profile merging, recommendation aggregation). We showed that, when applied, the Group Neighborhood Selection model increased the success rate of a recommendation list of size 3 for both profile merging and recommendation aggregation.

Next we move to the context which is considered the main distinguishing feature of group-based systems to individual based ones, the group modeling context. We employed a hybrid model that incorporated both group recommendation strategies. Our initial results revealed insight on the performance of the aggregation strategy (average, least misery, most happiness) when compared to group characteristics (size, cohesiveness). On the same line, a hybrid approach to the aggregation strategy, depending on group characteristics, resulted in higher prediction accuracy. We examined applying a weighted, hybrid approach to the recommendation strategy by combining the aggregated preferences and aggregated predictions models. Results showed that the hybrid approach resulted in higher accuracy predictions for the group in terms of RMSE calculated when compared to the baseline profile merging approach or the recommendation aggregation approach for all the weight distributions we evaluated. The best results were achieved with the 0.7/0.3, weights with the higher weight assigned

to the the Profile Merging approach. Overall, the HyMH (Profile merging and most happiness recommendation aggregation) had the lowest RMSE making it the best performance approach in terms of accuracy prediction. With respect to group characteristics, we detected similar trends in performance as the results we observed in the overall analysis for the high and medium inner similarity levels, where the HyMH had the highest accuracy and that accuracy increased as the group size increased.

The last viewpoint we took with respect to group modeling explored, not only considering the group context in the final aggregation, but also takes advantage of other previous group contexts as a first-class element of the knowledge base. We proposed a recommendation model that employs the group context in retrieving similar items to the target item and utilizing the group model for this item in the prediction calculation. With the current dataset we were not able to evaluate this model since there was no match in the data where this modeling approach was applicable. Thinking of solving the group model for predicting a rating for a group based on the group model of a previously known rating for the group, we found our selves looking at the group model as an application for Cased-based Reasoning. Retrieving and reusing the previous experience of another group, taken as a whole, is a natural application for CBR.

We examined the conditions under which taking advantage of existing group contexts in the knowledge base, in addition to the group context of the target / active group (as query), improves group recommender performance. We explored outcomes of prediction accuracy for recommendation aggregation using the Average, Least Misery, and Most Happiness group models. We examined these outcomes with respect to

group size and inner group similarity. We contrasted this approach, using the baseline neighborhood approach to the *Case-based* model approach, by comparing the prediction accuracy of the predicted group rating of each test item to the average of the actual ratings of the individual group members for that test item. We validated the applicability of this approach by analyzing the group contexts and found that this approach was highly applicable for groups with high inner cohesiveness. In comparison to the baseline, the CBR approach resulted in higher accuracy predictions for groups with high inner similarity levels. The performance of this approach also increases as the group size increases for all similarity levels. Overall, the results indicated that a significant improvement in accuracy can be achieved by using a case-based group recommender, where similar cases are considered as part of the prediction calculation. The last step we did was contrasting our CBR approach to the most similar application of CBR in group-based recommenders perviously published. We indicated that the benefits of a CBR approach may be realized even in a foundational implementation, without substantial demographic or data imputation requirements.

The contributions of this dissertation are:

1. An evaluation framework for group-based recommender systems that allows the creation of synthesized groups with varying group contexts. The framework creates the groups and identifies ideal test point for creating training and testing datasets for the groups.
2. A tradeoff comparison in the choice of the baseline for evaluation using synthesized groups. We show that the choice of group model as the actual group

preference affects the evaluation results and introduces another variable that needs to be accounted for and minimize its effect.

3. Rating normalization effect on prediction accuracy in memory-based collaborative filtering group-based recommender systems. We show which group models and group contexts benefit from rating normalization.
4. A neighborhood selection model based on the group context with a finer-grained weighting scheme. We show that neighborhood selection strategy that accounts for the group context and utilizes it in neighbor weighing increased the prediction accuracy for the group.
5. A hybrid group model that combines profile merging and recommendation aggregation and show which group aggregation models and group contexts benefit implementing this approach.
6. A disagreement model based on item similarity to the item considered for recommendation. We investigate the applicability for this approach and show that its feasibility is hindered with the constraint of identifying similar items rated by all the members of the group.
7. A case-based reasoning group model that is based on group context and group-to-group similarity. We show the applicability of this model and the benefits realized for the group in terms of prediction accuracy and group contexts.

8.2 Limitations and Future Work

Similar to most studies we acknowledge some limitations that we faced in conducting this research. Some of the limitations in terms of evaluation revolve around employing real subjects in a user studies. Given the context of our work, that is fo-

cused on group-based recommenders and collaborative filtering, a user study was not feasible since these approaches require datasets on a larger scale to be able to ground the evaluation. We address this by obtaining a data set from a research group [60] that enabled us to provide a comparison of our evaluation results using our dataset of synthesized groups to the results obtained from a dataset that contained real subjects' judgement. Another limitation is that we evaluate our approaches to group-based recommendations in one domain using the MovieLens dataset. To be able to generalize and extend our results to other domains we would need to obtain datasets from different domains. Additional limitation we faced relates to the ability of comparing our models to previous research. The structure of the datasets and evaluations of previous research contributed to the infeasibility of providing a direct comparison to our models. For example, we evaluate the prediction of accuracy using the RMSE measure while other studies use a different metric so their dataset is structured for the metric they were adopting. Another way to overcome this is to reimplement the approaches of previous research which might not be feasible since some details of the evaluation setup might be ambiguous and reimplementation is beyond the scope of this dissertation.

There are several future studies that can be conducted to address the limitations we faced in this work. Conducting a large scale study with groups of real subjects would help extend our results. Similarly, evaluating our models in different domains would also enable us to generalize our results further. Another future study to extend our results would be to evaluate our models using a larger size dataset of groups than the dataset we employ. We also realize the advantage of sharing our dataset with

the research community to help alleviate some of problems in evaluating group-based recommenders. This in return allows the grounding of evaluations in the group-based context and can lead to a more direct comparisons between various research. We plan on sharing the dataset containing groups with items of exact rating overlap since we see the value of this approach to evaluations using synthesized groups.

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] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40:66–72, March 1997.
- [5] 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.
- [6] 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.
- [7] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 46–54, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [9] 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 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [10] D. Bridge, M. H. Göker, L. McGinty, and B. Smyth. Case-based recommender systems. *Knowl. Eng. Rev.*, 20:315–320, September 2005.
- [11] D. Bridge, M. H. Göker, L. McGinty, and B. Smyth. Case-based recommender systems. *The Knowledge Engineering Review*, 20(3), 2005.

- [12] R. Burke. A case-based reasoning approach to collaborative filtering. In *Proceedings of the 5th European Workshop on Case-Based Reasoning*. 2000.
- [13] R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(Supplement 32):175–186, 2000.
- [14] R. Burke. Hybrid recommender systems: Survey and experiments. *User-Modeling and User-Adapted Interaction*, 12:331–370, 2002. 10.1023/A:1021240730564.
- [15] R. Burke. The adaptive web. chapter Hybrid web recommender systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.
- [16] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *IN PROCEEDINGS OF THE 13TH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 462–468. AAAI Press, 1996.
- [17] D. L. Chao, J. Balthrop, and S. Forrest. Adaptive radio: achieving consensus using negative preferences. In *Proceedings of the 2005 international ACM SIG-GROUP conference on Supporting group work*, GROUP '05, pages 120–123, New York, NY, USA, 2005. ACM.
- [18] 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.
- [19] M. T. Cox, H. Muñoz-Avila, and R. Bergmann. Case-based planning. *The Knowledge Engineering Review*, 20(3), 2005.
- [20] 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.
- [21] L. de Campos, J. Fernández-Luna, J. Huete, and M. Rueda-Morales. Managing uncertainty in group recommending processes. *User Modeling and User-Adapted Interaction*, 19:207–242, 2009. 10.1007/s11257-008-9061-1.
- [22] T. De Pessemier, S. Dooms, and L. Martens. Design and evaluation of a group recommender system. In *Proc of the Sixth ACM RecSys Conference*, 2012.
- [23] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.
- [24] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA, 2001. ACM.

- [25] I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman. A group recommender system for tourist activities. In *Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, EC-Web 2009, pages 26–37, Berlin, Heidelberg, 2009. Springer-Verlag.
- [26] 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.
- [27] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 54–89. Springer Berlin / Heidelberg, 2007.
- [28] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35:61–70, December 1992.
- [29] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. Technical Report UCB/ERL M00/41, EECS Department, University of California, Berkeley, 2000.
- [30] D. Goren-Bar and O. Glinansky. Fit-recommending tv programs to family members. *Computers & Graphics*, 28(2):149 – 156, 2004.
- [31] M. Grčar, D. Mladenič, B. Fortuna, and M. Grobelnik. Data sparsity issues in the collaborative filtering framework. In O. Nasraoui, O. Zaïane, M. Spiliopoulou, B. Mobasher, B. Masand, and P. Yu, editors, *Advances in Web Mining and Web Usage Analysis*, volume 4198 of *Lecture Notes in Computer Science*, pages 58–76. Springer Berlin / Heidelberg, 2006.
- [32] C. Hayes, P. Cunningham, and B. Smyth. A case-based reasoning view of automated collaborative filtering. In *Proceedings of the 4th International Conference on Case-Based Reasoning*. 2001.
- [33] 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.
- [34] J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 2004.
- [35] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, 2001. 10.1023/A:1007617005950.
- [36] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 2004.

- [37] 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.
- [38] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The adaptive web*, pages 596–627. Springer-Verlag, Berlin, Heidelberg, 2007.
- [39] J. A. Konstan. Introduction to recommender systems: Algorithms and evaluation. *ACM Trans. Inf. Syst.*, 22:1–4, January 2004.
- [40] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: applying collaborative filtering to usenet news. *Commun. ACM*, 40:77–87, March 1997.
- [41] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 426–434, New York, NY, USA, 2008. ACM.
- [42] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.
- [43] H. Lieberman. Letizia: an agent that assists web browsing. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, pages 924–929, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [44] H. Lieberman, N. van Dyke, and A. Vivacqua. Let's browse: a collaborative browsing agent. *Knowledge-Based Systems*, 12(8):427 – 431, 1999.
- [45] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, January 2003.
- [46] 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.
- [47] 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.
- [48] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, 16:281–319, September 2006.
- [49] J. F. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. pages 1–10, 2002.

- [50] J. F. McCarthy and T. D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *CSCW*, page 348, 2000.
- [51] K. McCarthy, L. McGinty, and B. Smyth. Case-based group recommendation: Compromising for success. In *Proceedings of the 7th international conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '07, pages 299–313, Berlin, Heidelberg, 2007. Springer-Verlag.
- [52] K. McCarthy, L. McGinty, B. Smyth, and M. Salamó. The needs of the many: A case-based group recommender system. In T. Roth-Berghofer, M. Göker, and H. Güvenir, editors, *Advances in Case-Based Reasoning*, volume 4106 of *Lecture Notes in Computer Science*, pages 196–210. Springer Berlin / Heidelberg, 2006.
- [53] K. McCarthy, M. Salamó, 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.
- [54] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. MovieLens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI '03, pages 263–266, New York, NY, USA, 2003. ACM.
- [55] M. Montaner, B. López, and J. L. De La Rosa. A taxonomy of recommender agents on the internet. *Artif. Intell. Rev.*, 19:285–330, June 2003.
- [56] N. A. Najjar and D. C. Wilson. Evaluating group recommendation strategies in memory-based collaborative filtering. In *Proceedings of the ACM Recommender Systems Conference Workshop on Human Decision Making in Recommender Systems*, RecSys '11. ACM, 2011.
- [57] 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.
- [58] D. O'Sullivan, D. Wilson, and B. Smyth. Improving case-based recommendation: A collaborative filtering approach. In *Proceedings of the 6th European Conference on Case-Based Reasoning*. 2002.
- [59] D. O'Sullivan, D. Wilson, and B. Smyth. Using collaborative filtering data in case-based recommendation. In *Proceedings of the 15th International FLAIRS Conference*. 2002.
- [60] L. Quijano-Sánchez, D. Bridge, B. Díaz-Agudo, and J. Recio-García. Case-based aggregation of preferences for group recommenders. In *Proceedings of the 20th International Conference on Case-Based Reasoning*. 2012.

- [61] L. Quijano-Sánchez, D. Bridge, B. Díaz-Agudo, and J. Recio-García. A case-based solution to the cold-start problem in group recommenders. In *Proceedings of the 20th International Conference on Case-Based Reasoning*. 2012.
- [62] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jimenez-Diaz. Social factors in group recommender systems. *ACM Trans. Intell. Syst. Technol.*, 4(1), 2013.
- [63] 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.
- [64] 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.
- [65] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
- [66] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 791–798, New York, NY, USA, 2007. ACM.
- [67] M. Salamó, K. McCarthy, and B. Smyth. Generating recommendations for consensus negotiation in group personalization services. *Personal and Ubiquitous Computing*, 16(5):597–610, 2012.
- [68] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *IN ACM WEBKDD WORKSHOP*, 2000.
- [69] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5:115–153, January 2001.
- [70] 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.
- [71] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of Conference of Human Factors in Computing Systems*, pages 210–217. ACM Press, 1995.
- [72] L. Spalzzi. A survey on case-based planning. *Artificial Intelligence Review*, 16(1), 2001.

- [73] 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.
- [74] C. L. Zitnick and T. Kanade. Maximum entropy for collaborative filtering. In *UAI'04*, pages 636–636, 2004.

DISSERTATION PUBLICATIONS

- [1] Najjar, N.A. and Wilson, D.C. (2011). Evaluating Group Recommendation Strategies in Memory-Based Collaborative Filtering. In: Proceedings of the ACM Recommender Systems Conference Workshop on Human Decision Making in Recommender Systems.
- [2] Wilson, David C., and Najjar, N.A. (2013) Tradeoffs in Evaluation Strategies for Group Recommender Systems. In Proceeding of the ACM Information Retrieval Conference Workshop on Benchmarking Adaptive Retrieval and Recommender Systems
- [3] Najjar, N.A. and Wilson, D.C. (2014). Differential Neighborhood Selection in Memory-Based Group Recommender Systems. The Twenty-Seventh International Flairs Conference. **Best Student Paper Award.**
- [4] Wilson, David C., and Najjar, N.A. (2014) Exploring the Space of Whole-Group Case Retrieval in Making Group Recommendations. International Conference on Case-based Reasoning.
- [5] Najjar, N.A. and Wilson, D.C. (2015). Power of the Group Neighborhood. To appear in The Twenty-Eighth International Flairs Conference.