RESEARCH ON RECOMMENDATION FOR GROUP USERS

A Thesis in
Information Sciences and Technology
by
Shandan Zhou

© 2013 Shandan Zhou

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2013
The thesis of Shandan Zhou was reviewed and approved* by the following:

Chao-Hisen Chu  
Professor of Information Sciences and Technology  
Thesis Advisor

Dongwon Lee  
Professor of Information Sciences and Technology

Dinghao Wu  
Assistant Professor of Information Sciences and Technology

Madhu C. Reddy  
Associate Professor of Information Sciences and Technology  
Director of Graduate Program in Information Sciences and Technology

*Signatures are on file in the Graduate School
ABSTRACT

Recommendation is an important information paradigm that attempts to suggest information items (e.g., movie, TV programs, or music etc.) or social elements (e.g., people or events) that are likely to be of interest to users. Individual recommendation has received significant attention in the past decade due to its extensive applications in Amazon, Netflix and Yahoo, etc. How to effectively recommend items to a group of users, however, is still an open issue. Especially, for a new designer, there are few accessible implementation guidelines to help him or her to develop a proper system according to their particular application domain and types of target groups. The main object of this thesis is to make a comprehensive literature review in this area, conduct a systematical experiment so as to generate implementation guidelines for new designers and propose a new group recommendation approach from a different perspective in order to advance the discussion in this area.
# TABLE OF CONTENTS

List of Figures .................................................................................................................. vi
List of Tables ...................................................................................................................... vii
Acknowledgements........................................................................................................... viii

Chapter 1 Introduction ....................................................................................................... 1
  1.1 Problem Statement ................................................................................................. 3
  1.2 Research Questions and Contributions ................................................................. 4
  1.3 Organization of the Thesis .................................................................................... 5

Chapter 2 Literature Review .............................................................................................. 7
  2.1 Representative Architectures .................................................................................. 7
  2.2 Aggregation Strategies .......................................................................................... 10
    2.2.1 Average Strategy ............................................................................................. 11
    2.2.2 Median Strategy .............................................................................................. 12
    2.2.3 Least Misery Strategy ...................................................................................... 13
    2.2.4 Group Contexts .............................................................................................. 13
  2.3 Evaluation ............................................................................................................... 14
  2.4 Conclusion .............................................................................................................. 16

Chapter 3 Generic Architecture of Group Recommenders ................................................. 17
  3.1 System Architecture .............................................................................................. 17
  3.2 Individual Recommendation Systems ..................................................................... 20
    3.2.1 Content-based Recommendation .................................................................... 20
    3.2.2 Collaborative filtering Recommendation ...................................................... 22
  3.3 Group Recommender Systems ............................................................................. 23
    3.3.1 Community-based Recommendation ............................................................ 24
    3.3.2 Group-based Recommendation ...................................................................... 25
    3.3.3 Individual-based Recommendation ............................................................... 26
    3.3.4 Comparison .................................................................................................... 29
  3.4 Conclusion .............................................................................................................. 33

Chapter 4 Design of Recommendation Experiments ......................................................... 34
  4.1 Questions ................................................................................................................. 34
  4.2 Preliminaries .......................................................................................................... 35
    4.2.1 Group Similarity ............................................................................................. 35
    4.2.2 The Strength of Social Relationship ............................................................... 38
  4.3 Experiments Design ............................................................................................... 40
4.4 Evaluation Criteria ........................................................................................................40
4.5 Conclusion ..................................................................................................................42

Chapter 5 Experimental Evaluation of Recommendation Approach .......................................43

5.1 Implementation .............................................................................................................43
5.2 Experiments on MovieLens Dataset ..............................................................................43
    5.2.1 Dataset Description ..............................................................................................43
    5.2.2 Experiment Results ...........................................................................................45
5.3 Experiments on Epinoins Dataset .................................................................................53
    5.3.1 Dataset Description ..............................................................................................53
    5.3.2 Experiment Results ...........................................................................................56
5.4 Discussion ....................................................................................................................58
5.5 Conclusion ....................................................................................................................60

Chapter 6 Clustering-based Group Recommendation Approach ...........................................61

6.1 Introduction ..................................................................................................................61
6.2 Clustering-based Approach ..........................................................................................63
    6.2.1 Data PreProcess ..................................................................................................64
    6.2.2 Clustering ..........................................................................................................65
    6.2.3 Item Selection .....................................................................................................67
6.3 Evaluation ....................................................................................................................69
    6.3.1 System Architecture ............................................................................................69
    6.3.2 Experiments .......................................................................................................71
6.4 Discussion ....................................................................................................................74
6.5 Conclusion ....................................................................................................................75

Chapter 7 Conclusion and Future Work ..............................................................................76

References .........................................................................................................................78
LIST OF FIGURES

Figure 3-1. Generic architecture of group recommender system...........................................19
Figure 3-2. Example of community-based recommendation approach ..................................24
Figure 3-3. Example of group-based recommendation approach ..........................................26
Figure 3-4. Example of individual-based aggregating profile approach .................................27
Figure 3-5. Example of individual-based aggregating recommendation lists approach ..........28
Figure 5-1. MovieLens user-user similarity distribution ..........................................................44
Figure 5-2. Random groups using AP approach .................................................................47
Figure 5-3. Random groups using AR approach .................................................................48
Figure 5-4. Comparing AP and AR approach in random groups ..........................................49
Figure 5-5. Group with high inner similarity using AR approach ..........................................51
Figure 5-6. Comparing AP and AR approach in groups with high inner similarity ..........52
Figure 5-7. Comparing effectiveness of random and groups with high inner similarity ......53
Figure 5-8. Distribution of user-pair similarity ......................................................................54
Figure 5-9. Distribution of with-propagation trust-pair similarity ..........................................55
Figure 5-10. Distribution of without-propagation trust-pair similarity ...................................55
Figure 5-11. Distribution of with-propagation trust-pair similarity ........................................56
Figure 5-12. The impact of the strength of the trust in group with high inner similarity ......58
Figure 6-1. Clustering-based Approach ................................................................................64
Figure 6-2. Architecture of clustering-based recommender .....................................................70
LIST OF TABLES

Table 2-1. Summary of group context according to domains* ........................................... 14
Table 3-1. Summary of architectures’ components * .......................................................... 18
Table 5-1. MovieLens data set description ............................................................................. 44
Table 5-2. Average group similarity of created groups .......................................................... 45
Table 5-3. Summary of the General Linear Model analysis .................................................... 46
Table 5-4. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AP approach ................................................................. 47
Table 5-5. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AR approach ................................................................. 48
Table 5-6. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AP approach ................................................................. 50
Table 5-7. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AR approach ................................................................. 50
Table 5-8. Epinions data set description ................................................................................. 54
Table 5-9. Summary of the General Linear Model Analysis .................................................... 56
Table 5-10. Summary of one-way ANOVA analysis for trust strength in random groups ...... 57
Table 5-11. Summary of one-way ANOVA analysis for trust strength in random groups ...... 58
ACKNOWLEDGEMENTS

First and utmost, I want to thank my academic advisor Prof. Chao-Hisen Chu, for his guidance, patience and kindness during the last two years. He was always there to help and support me whenever I had questions whether academically or personally. I would also like to acknowledge my two other committee members, for their time, interest and effort. Many thanks go to the group members for discussing and providing me with useful inputs. I would also like to thank many friends that I made in Penn State, especially Sandy Gleason, their kindness and assistance made my life in Penn State fruitful and happy. Last but not least, I would like to extend my gratitude to my family members for their unconditional encouragement and belief.
Chapter 1

Introduction

In our daily lives, there are many circumstances for which we need to seek suggestions or opinions from other people for the purpose of achieving better value and experiences. A recommender system is designed to generate recommendations as output or to guide users to achieve potential interesting objects among a sufficient number of possible options (Burke 2002). Much work has been done in this area after the appearance of Tapestry (Goldberg et al. 1992), the first mail recommender system. However, most previous work focuses on recommending products or services to individual users, such as recommending books, digital devices, and other products to individuals at Amazon.com (Linden et al. 2003); movies at MoiveLens (Miller et al. 2003) and music at Pandora (2000).

However, humans are social in nature. Many activities have strong social characteristics, such as watching a movie with friends in a movie theater, celebrating a promotion with colleagues in a restaurant, visiting a museum with family members, or reading a book in a book club. Intuitively, it may be inappropriate to consider only one user's preference for a group event. For example, Petrelli et al. (1999) observe that people's visiting behaviors would change when visiting a museum as a group instead of visiting a museum individually. Some traditional individual personalized recommender systems take the presence of other people as a context when processing recommendations (Adomavicius et al. 2005). These may be insufficient to satisfy all group members, since
they primarily focus on initiator's preference under a specified social circumstance. Taking collaborative browsing as an example, the application may not be considered to be a success if the needs and preferences of others are not included (Lieberman et al. 1999).

Group recommender systems have emerged to address this issue. It produces a set or a sequence of recommendations based on a group's interests or guides the group to achieve a set or a sequence of satisfied items through a great deal of options. Compared with individual recommender systems, group recommender systems are still at their early development stage (Masthoff 2011). Some prototypes have been developed in recent years (Ardissono et al. 2002; Chen et al. 2008; McCarthy et al. 2001; McCarthy et al. 2007; McCarthy et al. 2006; O’connor et al. 2001; Yu et al. 2006). However, they are not widely accepted, nor are they used in real world due to the lack of commonly accepted aggregation strategies (Jameson 2004), an appropriate data collection approach, mechanisms for result explanation and negotiation support (Zhou et al. 2012). Shin and Woo mention that users prefer different strategies when they are with different members (2009). However, for a new designer, there are few accessible implementation guidelines to help him or her to develop a right group recommender according to his or her application requirements. Hence, there is an emerging need for a systematic and comprehensive experiment to build a general implementation guideline for new designers. Moreover, a new aggregation approach from a different perspective is also needed to advance the discussion about enhancing or developing the next generation of group recommender systems.
1.1 Problem Statement

As we mentioned above, the development of group recommender systems is still at their early stage. Researchers in this area have not yet reached a consensus on the problem definition.

Depending on application domains, the goal of group recommendations is to either identify a set or a sequence of items that meet the interests and preferences of all group members. For example, Chen et al. (2008) and Amer-Yahia et al. (2009a) focus on recommending top-k items to a group of users through a consensus function. Although their definitions are clear and direct, they do not present detailed information on consensus function, nor do they provide details on the methods of building the final recommendation list. Masthoff and Gatt (2006) focus on recommending a sequence of items to a group. They point out that the consumption sequence of candidate items can also influence the satisfaction of group members.

Since it is rather difficult to model a sequence of items’ consumption, in this paper, we limit our focus on recommending a set of top-k items. Be aware that the value of k does not necessarily need to be a constant; instead, it can be flexible in value. The notion of rating and recommending top-k items for a group of users is called “Top-k Group Recommendation”, which can be formally defined as:

Problem (Top-k Group Recommendation): Give a target group $G$ (the size $|G| \geq 2$); a set of optional items $I$; preference of individual group member $IP$, preference of the group $GP$, preference of communities to which the target group belongs, and a set of group context $GC$. $GC$ can be empty if no related group information is captured. Similarly, $IP$, $GP$, and $CP$ can be empty as well, but at least one of them needs to be
valued. The problem of group recommendation is to build a consensus function \( F(G, IPs, GP, CPs, GCs, I) \) and generate a list of \( R_g \), which satisfies the following requirements:

1. \( 1 \leq R_g \leq k \).
2. \( \forall i \in R_g \), no group members has rated \( i \) before.
3. \( R_g \) is sorted on decreasing results of consensus function, and \( \exists i \in R_g, j \in I - R_g \ s.t. \ F(G, IPs, GP, CPs, GCs, j) \geq F(G, IPs, GP, CPs, GCs, i) \).

Based on this problem formalization, the key for a group recommender system is the formation of the consensus function.

1.2 Research Questions and Contributions

Group recommender systems have been developed for about one decade now. However, there are few accessible implementation guidelines to new designers. For instance, what components are required in a group recommender system and what factors are needed to take into account when selecting appropriate approaches. In particular, the majority of existing group recommendation approaches still focus on merging individual profiles or recommendation lists into a common profile or recommendation list. However, it is still an open research question to build appropriate social value functions that satisfy different types of groups best (O’connor, et al. 2001).

Motivated by above observation and current research status, the main research of this thesis focuses on the following questions:

**RQ1. What components should a generic group recommender system have?**

**RQ2. How to select an appropriate aggregation approach and merging strategies for a specific target group?**
RQ3. How to evaluate the satisfaction of a target group on the recommended items?

To address the above issues, promote discussion and present potential research directions in this field, this thesis focuses on proposing a generic recommender architecture for group recommender systems. We then conduct a systematic and comprehensive experiment to generate generic implementation guidelines from new designers. Finally, the author also designs a new group recommendation approach so as to advance the discussion in this field. The contributions of this study lie in:

- Analyzing selected representative group recommender architectures, and proposing a generic architecture for group recommender systems, consisting of four layers, from information collection, preference learning, recommendation generation, to presentation interface.

- Conducting a systematic and comprehensive experiment on the exiting aggregation approaches, strategies and impact factors, and providing a generic implementation guideline to new designers based on the experiment results.

- Proposing a new group recommendation approach by leveraging a clustering algorithm. Our approach is different from existing approaches which explores a new direction.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 surveys related works in group recommender architectures, aggregation strategies and evaluation methods. It provides an overview of components used in different group recommender architectures. It also presents a comprehensive review of strategies employed in
aggregating individual’s profiles or recommendation lists into a common file or recommendation list. We also summarize the current evaluation criteria that are adopted in group recommender systems. In Chapter 3, based on our problem statement and incorporating the essential components from existing group recommender architectures, we propose a generic group recommender architecture. Further, we design a systematic and comprehensive experiment in Chapter 4 and the corresponding experiment implementation and evaluation results are presented in Chapter 5. After that, in Chapter 6, a new group recommendation approach is introduced and evaluated by comparing with random suggestion and aggregating recommendation lists approach. Finally, we present the conclusion and future work in Chapter 7.
Chapter 2

Literature Review

In this chapter, we surveyed related works in architectures of group recommender systems, aggregation strategies and evaluation criteria.

2.1 Representative Architectures

Group recommender systems have been studied during the last decade. However, due to the space limitation or the focus of their papers, it is quite common that most of them just describe their aggregation strategies without mentioning the whole systems. Here, we surveyed five representative prototypes, PolyLens, CATs, Bernier’s recommender, GRec_OC, and Gartrell’s group recommender architectures.

PolyLens (O’connor, et al. 2001) is a group movie recommender based on MovieLens infrastructure (Miller, et al. 2003). The design of PolyLens considers group formation and evolution, privacy, social value function and explanation. In order to encourage group formation and keep management simple, PolyLens allows users to create groups and limit the membership to invited users. To handle users’ privacy issues, it looks at a user’s ability to control their membership and recommendation data. Regarding the social value function, PolyLens adopts the Least Misery Strategy to merging individual’s recommendation lists due to the small size of group users. PolyLens
discusses the mechanism of displaying recommendation result via both group and individual interface.

McCarthy et al. (2006) implement CATs to assist a group of people in planning a skiing vacation (McCarthy, et al. 2006). The architecture includes interface, profile, and recommendation modules. The interface module is not only used to acquire individuals’ preferences via providing critiquing feedback components, but also used to support group members’ awareness of other members’ preferences, so as to reach the final consensus more efficiently. The profile module records both individual and group profiles. The group profile is elicited from the individual models. Finally, the recommendation module generates candidate items based on both individual preferences and the group preference.

The Bernier’s recommendation process (Bernier et al. 2010) utilizes a “group analyzer” module to infer the characteristics and constrains of the target group. The characteristics utilized in the process include the nature of the relations, the cohesiveness, the social structure, the profile diversity, and the size of the group. The presences of specific types of people (e.g., children, or elder people), the evolution of the group (e.g., the changes of the presented people), effects (e.g., emotion contagion) are considered to be constraints. The recommendation engine then generates the candidate contents by selecting an appropriate strategy based on the result of group analyzer procedure. The system filters the optional resources by group constraints first and then chooses the relevant strategy by adopting group characteristics. The group recommendation strategies are classified into three categories: majority-based strategies (for instance, selecting the most popular items among group members), consensus-based strategies (for instance, averaging all members’ preferences), and borderline strategies (for instance, paying more
attention to the preference of elder people). Another contribution of this system is that it uses the presence community (the target group’s abstraction) to improve the quality of recommendations. According to this presence community, the system can detect some general interests of the target group.

GRec OC (Kim et al. 2010) is a two-phase book recommendation procedure for online groups. The first phase, called group profile-based filtering, includes three steps: (1) a simple additive method is designed to combine multiple user profiles into a group profile, (2) the nearest neighbor algorithm is used to predict ratings of optional books for the target group, and (3) predicted ratings are adopted to select top-k ranking candidate books through the large set of options. The second phase, called individual profile-based filtering, contains two steps: relevance between feature-based profiles of individual members and candidate books are calculated; then, the obtained compatibility scores are processed to filter the candidate book list. Thus, candidate books, which are considered to be disliked by a single user, could be filtered out by the degree of relevance.

Gartrell et al. (2010) propose an architecture for a group by merging the predicted individual ratings. The system uses recommendation approaches adopted in individual recommender systems to predict each group member’s ratings for unviewed movies. The consensus function module then merges these predicted ratings from each group member into a predicted group rating by considering the group characteristics, such as the social relationship, expertise, and dissimilarity of group members. The social relationship, expertise and dissimilarity are measured by the frequency of contact, the percentage of movie watched, and the rating differences between group members or the variance between individual ratings and average rating, respectively. Three common aggregating
strategies–average satisfaction, minimum misery, and maximum satisfaction– are utilized to combine these descriptors.

From the above discussion, we can see that some general concepts and components are presented among all of them, but due to space limitation or the focus of papers, most of them are not able to cover all the aspects of group recommender systems. For instance, Bernier’s and GRec OC architectures do not contain the way of capturing user preference; PolyLens, GRec OC, CATS and Gartrell’s architectures neglect the community preference. Neither CATS nor GRec OC consider different group contexts and corresponding strategies. With the exception of PolyLens and CATS, others architectures pay less attention to the mechanism of explaining recommended items or supporting negotiation between group members.

2.2 Aggregation Strategies

The majority of current group recommender systems employ the individual-based group recommendation approach due to the difficulty of obtaining data in a group level and the inaccuracy of using data in a community level. In this section, we conduct a literature review on aggregation strategies applied in existing group recommender systems. Since one can easily create hundreds of strategies (Masthoff 2004), we present the most widely employed strategies here.

A handful of studies have conducted the research on aggregation strategies, such as Jameson and Smyth (2007), Kay and Niu (2005), and Masthoff (2004). Masthoff (2004) summarizes and evaluates some heuristic aggregation strategies, such as plurality
voting, borda count, most preferred person, and so on, but the existing group recommenders’ combination methods are primarily based on basic mathematic operations, such as the averaging, adding, multiplying, calculating intersection, and union of each group member’s profile (Jameson and Smyth 2007; Kay and Niu 2005). However, which strategy is right for a specific group? The group size, group similarity, and social relationship are the major factors are taken into account by researchers.

2.2.1 Average Strategy

Intuitively, the most common and simplest way to reach a compromise between group members is to apply the Average Strategy (Or called popular strategy). Intrigue utilized a weighted average strategy to generate the group preference (Ardissono, et al. 2002). The major merit of leveraging the Average Strategy is that this mechanism ensures the fairness among members. In particular, it is easy to explain the result and it performs well in homogenous group (Masthoff 2011). FIT (Goren-Bar and Glinansky 2004) utilizes a weighted average strategy to calculate the group’s preference. The weight assigned to each viewer depends on the probabilities of the user being in front of the TV at the give time slot. Shin and Woo (2009) also employ the Average Strategy to calculate the group preference related to the metadata of TV programs, containing a variety of teams such as genres, keywords, actors and subcategories.

However, the drawbacks of the Average Strategy are obvious as well. The most important issue is the manipulation issue (Jameson and Smyth 2007). Take the vector space model for example, if a user wants to prevent items holding specific features from
being selected by the system, he can simply indicate that he extremely dislikes the features even if he really does not care much about these features but simply wants the system to recommend other features that he likes. This manipulation problem is even worse, in case group members share their profiles with other members. Thus, when applying averaging strategy to obtain the group preference, the aggregated preference may be not able to express the true preference of the group as a whole. Other issues of applying the Average Strategy include not necessarily reflecting individual’s preference, static and uniform, no social interactions and influence considered.

2.2.2 Median Strategy

This strategy uses the middle value of all group members’ ratings as the rating of the whole group. In contrast to the previous approach, the major advantage of the Median Strategy is a non-manipulable aggregation mechanism, since it is hard for members to predict whose’ ratings will be the median value of all individual ratings. This strategy is proposed by Jameson (2004), when he identifies the manipulative problem during the process of aggregating preferences. Another benefit of using the Median Strategy is simple as well, which, in turn facilitates the explanation process. In spite of these merits, the mathematical median sometimes is sub-optimal in terms of acceptability and/or equity (Jameson 2004). For example, it ignores a strong negative preference from a group member.
2.2.3 Least Misery Strategy

In contrast to the Median Strategy, the Least Misery Strategy particularly takes the strong negative preference into account. It sets the group’s predicted rating to be the lowest predicted rating among all the members’ ratings. PolyLens (O’connor et al. 2002) uses the Least Misery Strategy by assuming that the happiness of a group depends on the least happy person. Considering the strong negative preference is a merit of the Least Misery Strategy; however, it is also a drawback of the Least Misery Strategy. The minority opinions can dominate the group’s decision. For instance, if one person does not want others to see a particular movie, he or she can simply give a very low rating to that type of movies.

2.2.4 Group Contexts

In general, to select an appropriate aggregation strategy, the contextual information of the target group and environment need to be considered. Group contexts are any information that is relevant to a target group, which could influence group members’ opinions, feelings, and actions in a group social activity. An increasing number of group recommender systems appear to take a variety of group contexts into account, such as members’ personality, expertise, social relationships, similarities, the group size and composition (Berkovsky and Freyne 2010; Bonnefoy et al. 2009; Gartrell, et al. 2010; Masthoff and Gatt 2006; Recio-Garcia et al. 2009). However, acquiring, modeling and applying group contexts remains an open issue in group recommender systems. Table 2-1 summarizes the group contexts that are used based on the recommendation domains.
According to the table, we can observe that similarity, group size, and the social relationship have been widely considered among the surveyed domains. Nevertheless, no standard guidelines have been proposed to guide designers in properly leveraging this contextual information.

Table 2-1. Summary of group context according to domains*

<table>
<thead>
<tr>
<th>Reference</th>
<th>Domain</th>
<th>Group Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Z. Yu et al. 2006)</td>
<td>TV program</td>
<td>Similarity</td>
</tr>
<tr>
<td>(J. Masthoff and Gatt 2006)</td>
<td>TV program</td>
<td>Emotional contagion, conformity, social relationship</td>
</tr>
<tr>
<td>(Bonnefoy et al. 2007)</td>
<td>TV program</td>
<td>Similarity</td>
</tr>
<tr>
<td>(Ardissono et al. 2002)</td>
<td>Tourism</td>
<td>Composition</td>
</tr>
<tr>
<td>(Recio-Garcia et al. 2009)</td>
<td>Movie</td>
<td>Personality</td>
</tr>
<tr>
<td>(J. K Kim et al. 2010)</td>
<td>Book</td>
<td>Size, similarity</td>
</tr>
<tr>
<td>(Berkovsky and Freyne 2010)</td>
<td>Restaurant</td>
<td>Roles, size, similarity</td>
</tr>
<tr>
<td>(Baltrunas, Makcinskas, and Ricci 2010)</td>
<td>General</td>
<td>Size, similarity</td>
</tr>
<tr>
<td>(Amer-Yahia et al. 2009)</td>
<td>General</td>
<td>Social relationship, expertise, similarity</td>
</tr>
<tr>
<td>(Chen, Cheng, and Chuang 2008)</td>
<td>General</td>
<td>Personality</td>
</tr>
</tbody>
</table>

(* General means the system does not indicate a specific domain)

2.3 Evaluation

Although Shani and Gunawardana (2011) summarize a range of criteria employed to evaluate individual recommendation systems, such as user interface, prediction accuracy, coverage, confidence, trust, novelty, serendipity, diversity, utility, risk, robustness, privacy, adaptivity, and scalability, evaluation methods for real group
recommender systems have not been properly addressed yet. The primary evaluation methods applied in the current group recommender systems include user satisfaction, accuracy, and efficiency. User satisfaction is usually measured by conducting empirical user studies and analyzing users’ feedback, such as O’Connor et al. (2001), McCarthy (2000), Goren-Bar et al. (2004), Chao et al. (2005), Bonnefoy et al. (2009), Park et al. (2008), and Yu et al. (2006). The major disadvantage of user studies is their subjective nature. User tasks and questions are often loosely defined; hence, final scores are open to considerable interpretation (Yu, et al. 2006). Accuracy measures the recommendation mechanisms via a couple of objective approaches, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Precision and Recall, Similarity, and other metrics. For instance, Gartrell et al. (2010) and Boratto et al. (2011) both measure the similarity between the predicted group ratings and the actual group ratings by RMSE. Senot et al. (2010) employ the Cosine and Pearson similarity to compare the aggregated group profile and reference profile. Chen et al. (2008) use MAE to evaluate the absolute deviation between a predicted rating and a true rating. Other systems employ precision and recall to evaluate the accuracy of recommendation results (Kim, et al. 2010; Vildjiounaite et al. 2009; Yu, et al. 2006). Efficiency is also used to evaluate the time and space overhead of group recommendation strategies. For instance, Yu’s TV recommender system (2006) tests its efficiency in terms of time and space.
2.4 Conclusion

In this chapter, we conducted a literature review in group recommender system architectures, aggregation strategies, group contexts, and group-based evaluation methods. According to the surveyed papers, we can observe that most systems are not able to cover all the components of group recommender systems. Moreover, although the three aggregation strategies have been commonly employed, there are few accessible implementation guidelines to new designers. Regarding group-based evaluation methods, it remains an open issue in group recommender systems.
Chapter 3

Generic Architecture of Group Recommenders

This chapter presents the generic architecture derived from the collected literatures. We first present the proposed architecture and how data and information exchanges among each layer. Based on this architecture, we classify group recommender systems into three types: the community-based, group-based, and individual-based group recommenders.

3.1 System Architecture

According to our literature review on group recommender architectures, we can see that some general concepts and components are presented among all of them, but due to space limitation or the focus of papers, most of them are not able to cover all the aspects of group recommender systems. The components mentioned in representative architectures are summarized in Table I.

Figure 1 depicts our proposed generic architecture for a group recommender system, which contains four layers. The general concepts of previous architectures are included, and we integrate most modules that would be used during the process of group recommendation. Layer one, the data source layer, contains the main data sources for recommender systems, which may include: sensors, questionnaires, and commonsense or domain experts’ knowledge. Sensors can capture preference data from multiple
perspectives without interrupting users’ attention, while questionnaires and commonsense or domain expert’s knowledge are more explicit. After collecting related data, it would be

Table 3-1. Summary of architectures’ components *

<table>
<thead>
<tr>
<th>Component</th>
<th>PolyLe ns</th>
<th>CATS system</th>
<th>Bernier’s</th>
<th>GRec_OC</th>
<th>Gartrell’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source Layer</td>
<td>Collaborative input</td>
<td>--</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td>X</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Questionnaire</td>
<td>X</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Prior knowledge</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Preference Layer</td>
<td>Individual preferences</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Group preference</td>
<td>--</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Community preference</td>
<td>--</td>
<td>--</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>Recommendation Layer</td>
<td>Recommender Engine</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Strategies</td>
<td>X</td>
<td>--</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Group context</td>
<td>X</td>
<td>--</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Item information</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Interface Layer</td>
<td>Recommended list</td>
<td>X</td>
<td>--</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Explanation</td>
<td>X</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Negotiation</td>
<td>--</td>
<td>X</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

(* - Doesn’t contain; X contains)

formalized and passed into the next layer. Layer two, the preference layer, concerns the process of extracting and developing user preferences. We classify learned preferences into three categories: individual, group and community preference. Layer three, the recommendation layer, uses different levels of preferences from layer two to select appropriate strategies based on group contexts, then generate candidate items from a
large set of options for the target group. Here, group contexts are these pieces of information related to the target group, such as group size, group composition, group members’ relationship, personality, expertise, roles, and so on. Researchers in different domains may choose to use different group contexts when developing group recommender systems. The last layer, the presentation or interface layer, not only presents the candidate items to the group, but also explains why the candidate items are recommended to them. This layer also provides negotiation support mechanism to facilitate group members’ decision making. Additionally, users’ feedback flow from this layer would be injected into the data source layer for a new cycle recommendation. In the thesis, we focus on the work in the Layer 3, recommendation Layer.

Figure 3-1. Generic architecture of group recommender system
3.2 Individual Recommendation Systems

An extensive work has been done in individual recommendation after the appearance of Tapestry (Goldberg, et al. 1992), the first mail recommender system. In this section, the author primarily presents the representative recommendation approaches that are employed in individual recommender systems: the content-based approach and collaborative filtering approach (Balabanović and Shoham 1997). The two major directions of the collaborative filtering approach are neighborhood methods and latent factor models (Koren et al. 2009). Neighborhood methods are focused on calculating the relationship between users or items, while latent factor models, such as Matrix factorization, try to explain ratings by characterizing both items by mining item rating patterns. In the past decade, latent factor models have received increasing attention due to their good scalability and accuracy.

3.2.1 Content-based Recommendation

The content-based approach is derived from the information retrieval research. It recommends items similar to those that a user likes in the past (Adomavicius, et al. 2005; Adomavicius and Tuzhilin 2005; Burke 2002). The key of this approach depends on how the contents of the optional items are described. The basic and simply way to represent an item is the Vector Space Model, in which each item is described with a vector of features’ weights. And the weight indicates how “important” a feature is to the item.

A user’s preference is usually represented in the same model. A utility function is then adopted to evaluate the similarity between a user's profile model and optional items’
profiles model. Usually, the similarity scores are applied to suggest items directly. So far, there are several ways to calculate a similarity score, such as the Cosine similarity, Euclidean distance, Pearson correction, and Jaccard index. This approach has following pros and cons:

**Pros.** Content-based recommendation is able to recommend new items. Because this approach focuses on the content of the item instead of whether the item has been viewed or purchased by other users before. Therefore, the content-based recommendation approach does not have the new item cold-start problem.

**Cons.** The quality of recommended items cannot be guaranteed, since people’s opinions on items are not utilized in this approach. As long as the items contain the major features that a user liked before, it will be recommended to the user. For instance, the content-based methods are not able to distinguish the differences between a well-written article and a badly written one, if they use the same terms (Shardanand and Maes 1995). Other issue is that the performance of the content-based approach highly depends on the extracted features of target domains. For instance, to recommend movies, using the feature of actors may be not as good as using the feature of genre to match the profile of users and movies. Additionally, for some domains, it is difficult to extract features to represent items. Another challenge of the content-based approach is the new user cold-start problem (Adomavicius and Tuzhilin 2005). Because of lacking historical information of the target user, the system cannot make any personalized recommendation to him or her.
3.2.2 Collaborative filtering Recommendation

Unlike the content-based approach, the collaborative filtering (CF) approach generates a recommendation list according to the assumption that people who share similar interests in the experienced items would also have similar interests in those unexperienced items, too. Since this approach only cares about the similarity between users, it neither needs to obtain domain knowledge before the system's deployment nor extracts and describes the features of items (Zanker and Jessenitschnig 2009), which is the main advantage that makes the CF approach the most promising and widely employed method in research and practice (Sarwar et al. 2001). Currently, there are two common CF algorithms, which are the item-based and user-based respectively. For the user-based CF approach, it explores the relationships between users, while the item-based CF approach considers the similarities between items. The user-based CF is the most successful algorithm for constructing a recommender system. However, the computational complexity of this method grows linearly with the size of customers (Karypis 2001). Thus, because of the scalability and data sparsity issue, the item-based algorithm is more popular than the user-based approach in practice to date. The pros and cons of this approach has summarized as following.

**Pros.** It does not need to extract the content information of recommending items. Thus, for those domains that features are difficult to extract, the CF algorithms outperform the content-based algorithms. Besides, the quality of recommended items can be guaranteed.

**Cons.** New user problem is one of the drawback of the CF algorithms, which means the systems are unable to elicit users’ preferences and provide personalized
suggestions without collecting and analyzing some amount of users’ information (Ahn 2008). It has the new item cold-start problem as well. When a new item comes to the system, if no people provide ratings or comments to it, the system cannot recommend this item to users.

### 3.3 Group Recommender Systems

Traditionally, when people are talking about recommender systems, individual-oriented recommender systems are their major focus. However, human beings are social by nature. Recommender systems are needed by groups of people as well, such as recommending tours, movies, TV programs, restaurants, and so on so forth. Basically, most techniques adopted in individual recommender systems can be used in group recommender systems without change; however, the major differences and challenging of group recommender systems is how to acquire the group’s preference and generate the candidate recommendation list, which can maximize the happiness of the whole group members.

We categorize the existing generation approaches into three classes: community-based, group-based, and individual-based generation, depending on which level of preferences used to generate the recommendation list. The following sections will give more detailed information of each generation method.
3.3.1 Community-based Recommendation

A pure community-based strategy may be less interesting from a recommendation perspective, but is outlined here for completeness. In this strategy, the system learns the most popular items among the members of a community, and recommends these popular items to the target group. As a result, an individual's or a group's specific preferences are not considered in this type of systems. Since it only uses the community-level preference, every group or individual gets the same recommendation list as long as they belong to the same community. An example of this mechanism is depicted in Figure 3-2.

**Example 1:** Let $C_1$, $C_2$, $C_3$, and $C_4$ be the four community members; $W$ column be the weight importance of these community members; and $I_1$, $I_2$, $I_3$, and $I_4$ be the four unrated items for the target group. Table “Community-item matrix” describes ratings $C_1$, $C_2$, $C_3$, $C_4$, $I_1$, $I_2$, $I_3$, and $I_4$.

<table>
<thead>
<tr>
<th>Community member weights</th>
<th>Community-item matrix</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 5</td>
<td>I1 2.5 I2 3.5 I3 3.0 I4 2.5</td>
<td>I1 3.0</td>
</tr>
<tr>
<td>C2 2</td>
<td>I2 3.0 I2 5.0 I3 1.5 I4 3.5</td>
<td>I2 4.0</td>
</tr>
<tr>
<td>C3 1</td>
<td>I3 4.5 I2 2.5 I3 3.0 I4 2.5</td>
<td>I3 3.1</td>
</tr>
<tr>
<td>C4 2</td>
<td>I4 3.0 I2 4.5 I3 5.0 I4 3.5</td>
<td>I4 3.0</td>
</tr>
</tbody>
</table>

\[
\text{PreRating}(G_i) = \frac{\sum_{c \in C} w(e) \times r(e,i)}{\sum_{c \in C} w(e)} = \frac{3 \times 2.5 + 2 \times 3.0 + 1 \times 4.5 + 2 \times 2.5}{3 + 2 + 1 + 2} = 3.0
\]

Figure 3-2. Example of community-based recommendation approach

**Example 1:** Let $C_1$, $C_2$, $C_3$, and $C_4$ be the four community members; $W$ column be the weight importance of these community members; and $I_1$, $I_2$, $I_3$, and $I_4$ be the four unrated items for the target group. Table “Community-item matrix” describes ratings $C_1$, $C_2$, $C_3$, $C_4$, $I_1$, $I_2$, $I_3$, and $I_4$. 

\[
\text{PreRating}(G_i) = \frac{\sum_{c \in C} w(e) \times r(e,i)}{\sum_{c \in C} w(e)} = \frac{3 \times 2.5 + 2 \times 3.0 + 1 \times 4.5 + 2 \times 2.5}{3 + 2 + 1 + 2} = 3.0
\]
and \( C_4 \) given to the corresponding items. For example, community member \( C_1 \) gives 2.5 and 3.5 ratings to item \( I_1 \) and \( I_2 \) respectively. After applying the equation, the potential score to \( I_1 \) for the target group is 3.0.

### 3.3.2 Group-based Recommendation

This strategy uses the learned group-level preference as a single pseudo user; thus, generating the common list for the whole group can be conducted with no difference to recommend items to individuals. Furthermore, since this preference is learned from data generated by the target collectively, the group contexts have already been naturally embedded in the data sources. Therefore, group contexts do not need to be considered in this type of strategy. An example of this mechanism leveraging collaborative based approach depicted in Figure 3-3.
Example 2: Let G, U₁, U₂, U₃ be the users in the system; I₁, I₂, I₃, and I₄ be the four items; and I₃, I₄ be the to-be-recommended items. After utilizing Similarity(G,U) equation, we get the similarity values between the target group and all the other users, listed in the “Group neighbors” table. We then employ equation PreRating to generate the potential score to I₃ and I₄ for the target group.

3.3.3 Individual-based Recommendation

Individual-based generation is the most widely used strategies among existing group recommender systems (Baltrunas et al. 2010; Gartrell, et al. 2010; Lieberman, et al. 1999; McCarthy 2002; O’connor, et al. 2001; Yu, et al. 2006). The major reasons, as we discussed in previous section, are that it is much easier to acquire data individually and that many techniques proposed in the individual recommender systems could be applied without much change in group recommender systems. However, the biggest issue when adopting individual-based generation is how to cater to the group's needs based on the obtained individual preferences, since in different kinds of groups, individuals would prefer to reflect their preference differently (Masthoff and Gatt 2006).

Depending upon when the combination method has been applied, individual-based generation could be further divided into two categories: aggregating profiles and aggregating recommendations.
**Aggregating Profile (AP).** For this strategy, the system first merges all group members' individual profiles into a group profile. It then uses a recommendation approach (content-based or collaborative-based) to generate a common recommendation list for the target group based on the merged profile. Many factors may influence how individual profiles should be merged into a single common profile, such as the size of the group, members' roles, expertise, social relationships, similarities, and their interactions. Figure 3-4 harnesses the average strategy to merge individual profiles as an example to demonstrate this process.

![Diagram](image)

Figure 3-4. Example of individual-based aggregating profile approach

**Example 3:** Let M₁, M₂ and M₃ be the target group members; U₁, U₂, U₃ be the outside users in the system; I₁, I₂, I₃, and I₄ be the four items and I₅, I₆ be the to-be-recommended items. We first adopt the “average strategy” to merge members' preferences into group
preference. We then calculate the similarity values between the target group and all other outside users, listed in the “Similarity” table. Finally, we employ equation $PreRating$ to generate the potential score to $I_3$ and $I_4$ for the target group.

**Aggregating Recommendation Lists (AR).** For this strategy, the system first uses a recommendation approach to generate each group member’s recommendation list according to the individual’s profile. It then merges those recommendation lists into a common recommendation list for the target group by considering the target group contexts. Merging individual recommendation lists into a common list via average strategy is demonstrated in Figure 3-5.

**Figure 3-5. Example of individual-based aggregating recommendation lists approach**
**Example 4:** Let $M_1$, $M_2$ and $M_3$ be the target group members; $U_1$, $U_2$, $U_3$ be the outside users in the system; $I_1$, $I_2$, $I_3$, and $I_4$ be the four items and $I_3$, $I_4$ be the to-be-recommended items. We first use the $\text{Similarity}(M,U)$ equation to generate similar neighbors for each group member. We then employ equation $\text{PreRating}$ to calculate the potential score to $I_3$ and $I_4$ from $M_1$, $M_2$ and $M_3$ respectively. Finally, we apply the “average strategy” to merge members' potential scores of $I_3$ to the target group's potential score to $I_3$.

Compared to the AP method, the AR method is much easier to explain because the recommendation result can be directly related to each member (Kim, et al. 2010; O’connor, et al. 2001). Explanations derived from individuals also give more information which would facilitate members to reach a final decision. Furthermore, AR architecture usually offers better flexibility for dynamic groups (Amer-Yahia, et al. 2009a; Gartrell, et al. 2010). However, AR is less likely to generate serendipitous items (Kim, et al. 2010; O’connor, et al. 2001), since merged recommendation lists still lie inside each individual's prediction. To the best of our knowledge, no systematic experiment has been conducted to compare the recommendation results generated by these two methods.

3.3.4 Comparison

The criteria we will choose to compare the three types of strategies are: accuracy, capability of recommending items to new groups, privacy, quality of recommended items, level of explanation, and flexibility.

(1) Accuracy: Accuracy is the fundamental requirement of any recommender system, since it is the main factor that would influence users' acceptance of
the system. So far, most recommender systems are still focusing on improving the accuracy of recommendation.

(2) Capability of recommending items to new groups: Generally, a system needs to know enough information about target users before it can recommend appropriate items. Thus, for a new group, systems usually are not able to recommend items due to the lack of historical or demographic data about the group. However, the ability of the recommending right items to new groups is highly desirable.

(3) Privacy: Recommendation systems inherently have privacy issues, since they entail collecting considerable amounts of data about their users (Kobsa 2007). These issues have been identified as primary impediments to the development of recommendation systems (Teltzrow and Kobsa 2004). The privacy issues are even worse in group recommender systems than in individual ones (Masthoff and Gatt 2006). Therefore, it is important that a recommendation strategy can help to release a user's privacy anxiety and minimize the privacy breaches.

(4) Quality of recommended items: The quality of items varies in a great deal of options. In general, users prefer the higher quality items among a great number of similar ones. Thus, users would appreciate if a recommendation strategy that is able to support quality guaranteed items to enhance user experiences.

(5) Level of explanation: Buchanan and Shortliffe (Buchanan and Shortliffe 1985) suggest that intelligent systems that act as decision guides need to provide explanations for their suggestions so that people are comfortable with those systems. The correct explanation could increase users' satisfaction and convince them to try the recommended items so as to save them time and effort (Tintarev and Masthoff 2007).

(6) Flexibility: The dynamic attribute of groups is another main challenge of designing group recommender systems (Gartrell, et al. 2010). During the process of recommendation, the group members are not always supposed to be
present through the whole recommendation period. Some members may leave, while some new members may join. It would be desirable for recommendation strategies to handle this issue.

Community-based generation strategy is strong in terms of the quality of recommended items, privacy insurance, the ability to recommend items to new groups, and flexibility. Usually, higher quality items among a set of similar items are preferred and consumed by the public. Thus, they are very likely to be recommended to the target group based on the community-level preference. The capability to recommend items to any type of group is possible for the community-based approach, because it does not take much specific information of the target group into consideration. Additionally, since the community-based approach only takes into account the trends of preferences from a community instead of from a group or an individual, users' identification information does not necessarily need to be disclosed. Thus, the potential privacy risk can be minimized to a large extent. However, privacy and accuracy are conflicting goals (Polat and Du 2005). Accuracy is the main drawback of the community-based generation strategy. Since it does not utilize much information from the target group, the recommended items are only those items that are popular among the community, not necessarily the most desirable items among the target group. Similarly, the explanation of recommended items can only be addressed at the community level.

Intuitively, the group-based strategy could be more accurate than the community-based strategy, since it is based on the preference originally deduced from the target group. Furthermore, recommendations based on joint profiles do not require using or sharing personal preferences, thereby potentially benefitting individuals' privacy. The
ability to recommend items to new groups has not been considered by group-based strategy, but it can be addressed by combining it with the community-based approach. The level of explanation can be more precise than that provided by the community-based strategy, since its capability of explaining recommended items can relate to the specific situation of the target group. The quality of recommended items depends on what kinds of recommendation methods are utilized to generate the candidate item list. Generally, the content-based recommendation method does not guarantee the quality of recommended items. Conversely, the collaborative-based recommendation method ensures that candidate items are those items that have already been viewed and recommended by other users with similar tastes. Flexibility seems to be the major issue, because it is necessary to collect new data produced by the new group whenever group members change. This is the main reason why the strategy has not been widely adopted.

The explanation level of the individual-based strategy is much more precise than that of the previous two strategies. The individual-based strategy can explain the recommended items at both the individual and group levels (O’connor, et al. 2001), since it stores not only individual preferences but also merged group profiles. Flexibility is better than that of the group-based strategy, because data generated by individuals can be reused. The accuracy of the recommended items may be not as good as the previous two strategies, because it is difficult to properly incorporate group contexts into the process of individual-based generation. Moreover, this strategy cannot easily solve privacy issues or the ability to recommend items to new groups. The individual-based strategy requires each group member's data so as to learn individual preference. Similarly to group-based
strategy, the quality of recommended items depends on the recommendation algorithm employed.

3.4 Conclusion

In this chapter, we proposed a generic group recommender architecture, which contains four layers: the information collection layer, the preference learning layer, the recommendation layer, and the presentation layer. We then classified users’ preferences into three levels. Based on the collected preference level, we then categorized the recommendation mechanisms into three types: the community-based, group-based and the individual-based recommendation mechanism respectively. After that, we compare these three mechanisms from six perspectives.
Chapter 4

Design of Recommendation Experiments

For a new designer to develop a group recommender system, there are few accessible implementation guidelines. In order to understand the effectiveness of widely used recommendation approaches (AP and AR), aggregations strategies (the Average Strategy, Median Strategy and Least Misery Strategy), and providing some generic guidelines to new designers, we designed a systematic and comprehensive experiment in different application domains. We also explore the impact of the group size, group similarity, and social relationship on the group recommendation result. In this chapter, we first introduce some preliminaries concepts and then present how experiments are designed.

4.1 Questions

For this evaluation, there are many interesting questions that we want to explore. Following are the major questions we want to learn:

- Q1. How the group size influences the effectiveness of an aggregation strategy?
- Q2. How the group similarity influences the effectiveness of an aggregation strategy?
- Q3. How the group social relationship influences the effectiveness of an aggregation strategy?
• Q4. Aggregation profile and aggregation recommendations lists, is there a constant better approach to suggest items to a group?
• Q5. How the aggregation mechanisms, especially the Average, Median and Least Misery strategy, work in different types of groups?

4.2 Preliminaries

As we mentioned, we try to test the influence of the group size, the group similarity and the group’s social relationship. We take these three main factors into account while forming groups. The group size is quite easy to understand. In this section, we will discuss the concepts and methods of measuring group similarity and the strength of social relationship, in particular, the trust relationship.

4.2.1 Group Similarity

Based on the theory of homophily from sociology (McPherson et al. 2001), people who have similar characteristics tend to form ties, and in turn that they are more likely to have opportunities to attend a social activity together. Thus, it is necessary for us to understand how groups with different inner similarity will make their group decisions.

How to measure the similarities between users or items plays a very important role in recommender systems. Various approaches have been employed to compute the similarity (Adomavicius and Tuzhilin 2005). The Pearson Correlation similarity (Sarwar et al. 2001) is the most widely employed correlation-based approach. The definition of the Pearson Correlation similarity is as following:
Here $R_{u,i}$ denotes the rating of user $u$ on item $i$, $\bar{R}_u$ is the average rating from user $u$, and $I$ is the set of items correlated by both user $x$ and $y$.

Although the Pearson Correlation similarity has been widely adopted, this similarity measurement has some drawbacks in recommender systems. First, it does not take into account the number of overlapped items between users (Owen et al. 2011), which is likely to make inaccurate prediction of users’ similarity. For instance, two users that have more than 100 co-rated items, even if they do not provide similar ratings to these items, are likely to be more similar than two users who have only co-rated two items. Moreover, when a user provides constant rating to all the viewed items, this correlation is undefined.

In contrast with the Pearson Correlation similarity, the Tanimoto coefficient, also known as the Jaccard index, measures the overlapped items between users. The definition is as following:

$$\text{TanimotoCoef}(x,y) = \frac{|I_x \cap I_y|}{|I_x \cup I_y|}$$ (4-2)

Here $I_x$ denotes all the items rated by user $x$. However, the issue of this measurement is that the preference values are not considered. The Loglikelihood-based similarity is similar to the Tanimoto coefficient in terms of taking the number of overlapped items into account. Furthermore, the Loglikelihood-based similarity generally outperforms the Tanimoto coefficient (Owen, et al. 2011). For more detailed information of how the Loglikelihood-based similarity is calculated, please refer to Dunning’s work.
In our experiments, we will employ the Pearson Correlation to compute the similarity between users. And in future work, we plan to explore the Loglikelihood and other similarity measurements as well.

In group recommender systems, no consensus has been reached for the definition of group similarity yet. Some researchers set the group similarity to be the lowest user-to-user similarity, For instance, Baltrunas, et al. (2010). Other researchers also set the average pairwise similarity to be the group similarity, such as (McCarthy, et al. 2007). In our work, we also set the group similarity to be the average pairwise similarity. The definition is as following:

$$GroupSim = \frac{2\sum_{x=1}^{m}\sum_{y=x+1}^{m} sim(x,y)}{m(m-1)}$$  \hspace{1cm} (4-3)

Here, \(m\) is the number of group size, \(sim(x,y)\) is the similarity between user \(x\) and \(y\). We can see that the group similarity is the average similarity among all pairs of users in the group.

Depending on the value of group similarity, we classify groups into two types: **heterogeneous group and homogenous group**. If the group similarity is lower than a predefined threshold, we call it a heterogeneous group. Similarity, if the group similarity’s value is greater than a threshold, we call it a homogenous group. Generally, in practice, heterogeneous groups are likely to be formed naturally by group members themselves because of their social relationships. For instances, friends in college may like to have a dinner together in weekends, a family would like to watch a movie together and colleagues may like to celebrate a promotion jointly. The major motivation of social activities for this type of groups is generally to enhance the social connections. Thus, the
preferences of group members in this type of groups are more likely to be randomly different from or even conflict with others. On the other hand, the homogenous group is usually built because of their common interests. For instances, people, who like travel, are very likely to plan a trip together; people, who like the same brand of clothes, would like to purchase the same brand of clothes together so as to enjoy a great discount.

4.2.2 The Strength of Social Relationship

One of the key differences between group recommenders and individual recommenders is the existing social relationships among group members. As we mentioned before, members’ behaviors are likely to be influenced by the opinions of other members. In a group, there are different kind of relationships, such as role relationship, familiarity relationship, trust relationship, and so forth. In this work, we focus on the trust relationship among group members.

Leveraging trust relationship to suggest items has attract an increasing attention in individual recommenders, e.g., Massa et al. (2004), O'Donovan et al. (2005) and Walter (2008). It is important to mention that trust relationship can be represented as a directed graph, which means user A trusts user B does not necessary mean that user B trust user A as well. Moreover, in some works, such as Walter, et al. (2008) and Guha et al. (2004), they assume that the trust relationship is able to propagate over the network. For instance, user A trusts user B and user B trusts user C, then the system will think user A should have a somewhat trust to user C. On the other hand, some researchers think it is better to assume that trust is not transitive, especially from the network security perspective
(Christianson and Harbison 1997). In this work, we use two methods to measure the strength of trust, without-propagation trust and transitive trust approaches.

Regarding without-propagation trust, we define the strength of trust between two users as equation (4-4).

\[
trust(x,y) = \begin{cases} 
1, & \text{there is direct trust from } x \text{ to } y \\
0, & \text{otherwise} 
\end{cases}
\] (4-4)

For transitive trust, we adopt the “friend-of-friend” ideas to find the possible paths that propagate trust between two users. Like “friend of friend”, we only take one hop paths into account. Intuitively, the more one hop paths, the stronger the trust is. Thus, in this thesis, we define transitive trust as following:

\[
trust(x,y) = \begin{cases} 
1, & \text{there is direct trust from } x \text{ to } y \\
1 - e^{-n}, & \text{otherwise, } n \text{ is the number of paths} 
\end{cases}
\] (4-5)

After obtaining all the pairwise strength of trusts, we need to define the strength of a group’s trust. First, we set a group, in which every member trusts each other, to have the strongest trust relationship and the value equals to 1; then, groups without trust relationships between any two members have the lowest strength of trust and the value is set to 0. Based on these principles, we defined the strength of group trust as following equation:

\[
GroupTrust = \frac{\sum_{x=1}^{m} \sum_{y=1}^{m} trust(x,y)}{m(m-1)}
\] (4-6)

Here, \( m \) is the group size.
4.3 Experiments Design

In order to understand how the group size, group similarity and social relationship influence the effectiveness of three common aggregation strategies in different domain applications, we will use two popular datasets, Epinions (Massa and Avesani 2006) and MovieLens, to conduct the experiments.

**Experiments Testing Group Size.** To test the impact of the group size, we generate artificial groups of users with size 2, 3, 5 and 8. Groups of size 2 represent small groups, while groups with 8 members represent relative large groups. Groups with size 3 and 5 are used to explore the changing trend of these aggregation strategies.

**Experiments Testing Group Similarity.** To compare the effectiveness of three aggregation strategies in different groups in terms of similarity, we design to conduct experiments for random groups and groups with higher inner similarity.

**Experiments Testing Group Relationship.** In order to understand how the trust relationship influence the effectiveness of recommendation results and the performance of three aggregations strategies, we take the strength of groups’ trust relationship into account when forming test cases.

**Experiments Testing Recommendation Approaches.** AP and AR are the two widely employed recommendation approaches in group recommender systems. We also like to compare the performance these two approaches.

4.4 Evaluation Criteria

One of the major challenges of group recommendations is the difficulty of evaluating the effectiveness. As we surveyed in chapter 2, the primary evaluation
methods applied in the current group recommender systems include user satisfaction and accuracy. In general, user satisfaction consists of interviewing real users. And this approach can be conducted by either acquiring users’ evaluations individually and then integrate, or obtaining collective evaluation of the group (Baltrunas, et al. 2010). However, this evaluation by interviewing users can be performed on a very limited set of cases.

In terms of evaluating accuracy, normally it was processed without interviewing users. Moreover, groups are created from the users of a traditional individual recommender system. In this case, group recommendations are evaluated independently, as in the traditional single user case, by comparing the predicted ratings with the actual ratings in the test set of group members.

In this work, in order to test the merging strategies comprehensively, we select to evaluate the work in terms of accuracy. We focus on the group recommendations that are generated to simultaneously satisfy all the group members. In this situation, we do not need to collect the joint group evaluation for the recommended results. Thus, we can use the most popular dataset, e.g., Epinions and MovieLens. In this work, we adopt RMSE, defined as equation (4-7), to measure the effectiveness of recommendation results.

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - a_i)^2}{n}}
\]  

(4-7)

Here, \(p_i\) is the predicted rating value for a target group, \(a_i\) is the actual rating value of an item for a group members.
4.5 Conclusion

In this section, we introduce the concept of group similarity, the strength of trust relationship. We then present our plans of conducting experiments to evaluate the impact of three crucial factors: the group size, group similarity, and group trust relationship. Additionally, the performance of two common group recommendation approaches, AP and AR, will be compared in our experiment as well.
Chapter 5

Experimental Evaluation of Recommendation Approach

Based on our designed experiments in chapter 4, we will focus on implementing, analyzing, and discussing the evaluation results on two popular data sets, MovieLens and Epinions, in this chapter.

5.1 Implementation

To develop a rapid prototype and conduct experiments on two popular data sets, we leverage the open source scalable machine learning libraries, Apache Mahout (Mahout). It provides a set of algorithms and users can build a customized recommender by selecting and combining appropriate approaches. In our experiments, we use the generic user-based recommender approach. The number of neighborhoods is set to 64. Based on this individual recommender, we implement the aggregation strategies using JAVA programing language.

5.2 Experiments on MovieLens Dataset

5.2.1 Dataset Description

The MovieLens dataset is the most popular data set utilized to evaluate effectiveness of a new recommendation method, especially in individual recommender systems. In this work, we use MovieLens 1M data set, which contains almost 1 million
ratings from 6000 users on 4000 movies. The data set includes three files, rating, user and movie files. For the rating file, it contains not only numeric rating in the range of 1 to 5, but also time stamps. For the user file, it includes users’ basic background information as well, such as gender, age, occupation, and location zip-code. Similarly, the movie file contains movie genres and titles information.

In order to increase the chance of acquiring movies viewed by all the group members, we select movies that have been watched by at least 10 users and users who have watched at least 25 movies. Table 5-1 presents the final data been selected in our experiments.

<table>
<thead>
<tr>
<th># users</th>
<th>#movies</th>
<th>#rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>5623</td>
<td>3260</td>
<td>989412</td>
</tr>
</tbody>
</table>

To understand similarity among users, we looked into the distribution of all user-user similarity pairs of selected data. Figure 5-1 shows the result.

Figure 5-1. MovieLens user-user similarity distribution
Based on the above figures, we can observe that 60 percentage of user-pair similarity are greater than 0.3. Therefore, in our experiments, we set 0.4 to be the threshold to distinguish groups with higher inner similarity or random groups.

5.2.2 Experiment Results

To conduct the experiment, the whole data related to group members is divided into training and testing part. The training part is used to learn members’ preferences and the testing part is employed to evaluate the accuracy of predicted ratings compared with the actual ratings. In order to increase the number of overlap items in testing part among group members, we randomly chose 60% of the total ratings from the user profile for training and the remaining 40% for testing. For each group size, we created 500 target groups. Table 5-2 presents the group inner similarity of two types of target groups.

<table>
<thead>
<tr>
<th>Group type</th>
<th>Group Size</th>
<th>Group Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random group</td>
<td>2</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.14</td>
</tr>
<tr>
<td>Group with high inner similarity</td>
<td>2</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.78</td>
</tr>
</tbody>
</table>
Experiment 1. In this experiment, we aim to evaluate the impact of the recommendation approaches (AP/AR), group similarities, group size, and the aggregation strategies on recommendation effectiveness. Because we consider more than two variables, we employed the General Linear Model to analyze the variance. Table 5-3 summarized the result.

Table 5-3. Summary of the General Linear Model analysis

<table>
<thead>
<tr>
<th>Factor</th>
<th>DF</th>
<th>Seq SS</th>
<th>Adj MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP/AR</td>
<td>1</td>
<td>178.678</td>
<td>178.678</td>
<td>7288.21</td>
<td>0.00</td>
</tr>
<tr>
<td>Random/Similar</td>
<td>1</td>
<td>0.802</td>
<td>0.802</td>
<td>32.71</td>
<td>0.00</td>
</tr>
<tr>
<td>Size</td>
<td>3</td>
<td>5.706</td>
<td>1.902</td>
<td>77.59</td>
<td>0.00</td>
</tr>
<tr>
<td>Strategy</td>
<td>2</td>
<td>35.085</td>
<td>17.542</td>
<td>715.54</td>
<td>0.00</td>
</tr>
<tr>
<td>AP/AR * Random/Similar</td>
<td>1</td>
<td>1.451</td>
<td>1.451</td>
<td>59.19</td>
<td>0.00</td>
</tr>
<tr>
<td>AP/AR * Size</td>
<td>3</td>
<td>4.339</td>
<td>1.446</td>
<td>59.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AP/AR * Strategy</td>
<td>2</td>
<td>30.181</td>
<td>15.090</td>
<td>615.53</td>
<td>0.00</td>
</tr>
<tr>
<td>Random/Similar * Size</td>
<td>3</td>
<td>3.057</td>
<td>1.019</td>
<td>41.57</td>
<td>0.00</td>
</tr>
<tr>
<td>Random/Similar*Strategy</td>
<td>2</td>
<td>0.799</td>
<td>0.400</td>
<td>16.30</td>
<td>0.00</td>
</tr>
<tr>
<td>Sizes *strategy</td>
<td>6</td>
<td>10.035</td>
<td>1.673</td>
<td>68.22</td>
<td>0.00</td>
</tr>
</tbody>
</table>

According to the information presented in Table 5-3, we can see that all the factors have significant influence on the effectiveness of the recommendation results. In order to further understand how each aggregation strategy works in different types of groups, which recommendation approach performs better and how the similarity between group members influence the effectiveness, we conduct more detailed variance analysis.

Experiment 2. In this experiment, we aim to understand how aggregation strategies work in random groups with different sizes. We adopt the AP approach and use the one-way ANOVA analysis. The result is presented in Table 5-4 and Figure 5-2.
Table 5-4. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AP approach

<table>
<thead>
<tr>
<th>Group size</th>
<th>Factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Strategy</td>
<td>2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>Strategy</td>
<td>2</td>
<td>0.009</td>
<td>0.004</td>
<td>0.16</td>
<td>0.86</td>
</tr>
<tr>
<td>5</td>
<td>Strategy</td>
<td>2</td>
<td>0.143</td>
<td>0.071</td>
<td>5.14</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>Strategy</td>
<td>2</td>
<td>0.220</td>
<td>0.110</td>
<td>10.43</td>
<td>0.00</td>
</tr>
</tbody>
</table>

From Table 5-4, we can observe that aggregation strategies have significant impact on the recommendation effectiveness when the group size is greater than 3. Moreover, based on the result showed in Figure 5-2, the performance of the Average Strategy is better than other two strategies for random groups.

Similarly, we use the same synthesized groups to evaluate the AR approach. Table 5-5 and Figure 5-3 present the result of AR approach. According to the displayed information, we can see that the impact of the group size and the aggregation strategies on effectiveness of AR is obvious as well. Similarly, based on Figure 5-3, we can observe that the Average Strategy and Median Strategy work much better than the Least Misery Strategy.
Table 5-5. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AR approach

<table>
<thead>
<tr>
<th>Group size</th>
<th>Factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Strategy</td>
<td>2</td>
<td>1.737</td>
<td>0.868</td>
<td>25.51</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>Strategy</td>
<td>2</td>
<td>3.818</td>
<td>1.909</td>
<td>76.45</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>Strategy</td>
<td>2</td>
<td>14.683</td>
<td>7.342</td>
<td>492.16</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>Strategy</td>
<td>2</td>
<td>32.511</td>
<td>16.256</td>
<td>1461.84</td>
<td>0.00</td>
</tr>
</tbody>
</table>

![Boxplot of random group size=2](a)

![Boxplot of random group size=3](b)

![Boxplot of random group size=5](c)

![Boxplot of random group size=8](d)

Figure 5-3. Random groups using AR approach

**Experiment 3.** We further compare the effectiveness of AP and AR approaches in random groups. Since both the Average Strategy performs better in both approaches, we only compare the effectiveness of these two approaches leveraging the Average Strategy. Figure 5-4 shows the result.
Figure 5-4. Comparing AP and AR approach in random groups

Comparing the result of AP and AR, we can see that AR is much better than AP. We can see the average RMSEs of AP is about 1.2; while for AR is about 1.0. One of the reasons may be due to the information lost at the earlier stage, which may be propagated and increased at the following steps.

**Experiment 4.** In this experiment, we aim to understand how aggregation strategies work in groups with high inner similarity. Table 5-7, 5-8 and Figure 5-5 show the experiment result. We can observe that the performances of aggregation strategies do not have much difference in groups with high inner similarity when utilizing the AP approach. However, the aggregation strategies have obvious impact on the recommendation effectiveness when using AR approach. Based on the information
presented in Figure 5-5, we can observe that the Average Strategy and Median Strategy work much better than the Least Misery Strategy.

Table 5-6. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AP approach

<table>
<thead>
<tr>
<th>Group size</th>
<th>Factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Strategy</td>
<td>2</td>
<td>0.003</td>
<td>0.001</td>
<td>0.04</td>
<td>0.956</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td>0.010</td>
<td>0.005</td>
<td>0.18</td>
<td>0.833</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2</td>
<td>0.026</td>
<td>0.013</td>
<td>0.50</td>
<td>0.604</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>2</td>
<td>0.0459</td>
<td>0.036</td>
<td>0.87</td>
<td>0.421</td>
</tr>
</tbody>
</table>

Table 5-7. Summary of one-way ANOVA analysis for aggregation strategies in random groups using AR approach

<table>
<thead>
<tr>
<th>Group size</th>
<th>Factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Strategy</td>
<td>2</td>
<td>1.320</td>
<td>0.660</td>
<td>16.88</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td>2.721</td>
<td>1.360</td>
<td>56.27</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2</td>
<td>9.267</td>
<td>4.634</td>
<td>253.27</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>2</td>
<td>17.548</td>
<td>8.774</td>
<td>527.40</td>
<td>0.000</td>
</tr>
</tbody>
</table>

**Experiment 5.** We also compare the effectiveness of AP and AR approaches in groups with high inner similarity. Since the Average Strategy performs better in AR approach and there are no obvious differences between different aggregation strategies when using AP approach, we only compare the effectiveness of these two approaches leveraging the Average Strategy. Figure 5-6 shows the result. We can see that the performance of AR approach is much better than the performance of AP approach.
**Experiment 6.** In this experiment, we aim to evaluate the effectiveness of group recommendation influenced by group inner similarity. We set the recommendation approach to be AR and the aggregation strategy to be the Average Strategy. Figure 5-7 shows the result. We can observe that for groups with size equals to 2 or 3, the effectiveness of recommending items to groups with high inner similarity is better than recommending items to random groups. For groups with size equals to 5, there is no much difference between them. However for groups with size equals to 8, the result is unexpected. The effectiveness of recommending items to random groups is better than recommending items to groups with higher inner similarity. Generally, we think that recommendation movies to homogenous group (group with high inner similarity) would
be more effective than recommending movies to heterogeneous groups, since it is less likely that there are preference conflicts between homogenous group members. However, in this experiment, we noticed that the similarity does not have obvious impact on the recommendation effectiveness when the group is quite large. It may be because of the similarity measurement, which is not able to accurately capture the similarity between users.

![Boxplot of group with high inner similarity, size=2](image1)

![Boxplot of group with high inner similarity, size=3](image2)

![Boxplot of group with high inner similarity, size=5](image3)

![Boxplot of group with high inner similarity, size=8](image4)

Figure 5-6. Comparing AP and AR approach in groups with high inner similarity

At this moment, it is hard for us to consider the social relationship between group members, since this kind of information has not been provided in MovieLens data set. In the Epinions data set, we will consider the impact of this element.
Figure 5-7. Comparing effectiveness of random and groups with high inner similarity

5.3 Experiments on Epinoins Dataset

5.3.1 Dataset Description

The Epinions is a consumers’ opinions site where users can review items and express their web of trust. Regarding items, users are able to not only give ratings to them from 1 to 5, but also provide comments on them. Regarding trust, users can indicate whose reviews and ratings they have consistently found to be valuable. Users also can make a block list of reviewers who they think their reviews are inaccurate and not valuable.
Similar to processing MovieLens dataset, we decide to only consider items have been reviewed by at least 10 users. For users, we only take those users who have provided at least 25 reviews into account. After these preprocess, the final data is summarized in Table 5-8.

<table>
<thead>
<tr>
<th># users</th>
<th>#items</th>
<th>#rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>4105</td>
<td>11622</td>
<td>215044</td>
</tr>
</tbody>
</table>

Figure 5-8 shows the user-pair similarity. We can see that an interesting phenomenon of Epinions data set. Most of user pairs have either dissimilar preferences (e.g., lower than -0.9), or quite similar preferences (e.g., higher than 0.9). In our experiment, we set the threshold to be 0.7 to distinguish groups with higher inner similarity.

Moreover, we also look at the distribution of user-pair trust strength, especially the strength of with-propagation trust. Due to the significant number of user pairs, we sample
158,000 pairs of users and calculate the strength of trust relationship between them. Figure 5-9 presented the result. We set 0.6 to be the threshold to distinguish users who have strength trust relationship or not.

![Distribution of user-pair trust strength](image)

*Figure 5-9. Distribution of with-propagation trust-pair similarity*

In general, users who have trust relationships are more likely to have similar preferences (Granovetter 1983). We also look into the similarity between

![Without-Propogation trust user-pair similarity distribution](image)

*Figure 5-10. Distribution of without-propagation trust-pair similarity*

users who have strong trust relationships. Figure 5-10 and 5-11 show the similarity distribution of user pairs have strong with-propagation and without-propagation trust
relationship respectively. According to these two figures, there is no obvious connections between trust and similarity.

![With-propagation trust user-pair similarity distribution](image)

Figure 5-11. Distribution of with-propagation trust-pair similarity

5.3.2 Experiment Results

Similarly, the whole data related to group members is divided into training and testing part. Again, 60% of the whole ratings is assigned to be the training data and the remaining for testing. For each group size, we create 500 target groups.

<table>
<thead>
<tr>
<th>Factor</th>
<th>DF</th>
<th>Seq SS</th>
<th>Adj MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random/Similar</td>
<td>1</td>
<td>1.559</td>
<td>1.419</td>
<td>25.22</td>
<td>0.00</td>
</tr>
<tr>
<td>Trust Definition</td>
<td>1</td>
<td>0.276</td>
<td>0.303</td>
<td>5.39</td>
<td>0.02</td>
</tr>
<tr>
<td>Strength of Trust</td>
<td>3</td>
<td>2.051</td>
<td>2.375</td>
<td>42.21</td>
<td>0.00</td>
</tr>
<tr>
<td>Size</td>
<td>1</td>
<td>2.184</td>
<td>2.018</td>
<td>11.96</td>
<td>0.00</td>
</tr>
<tr>
<td>Random/Similar*Trust Definition</td>
<td>1</td>
<td>0.000</td>
<td>0.001</td>
<td>0.02</td>
<td>0.89</td>
</tr>
<tr>
<td>Random/Similar*Strength of Trust</td>
<td>3</td>
<td>0.079</td>
<td>0.114</td>
<td>2.02</td>
<td>0.15</td>
</tr>
<tr>
<td>Random/Similar*Size</td>
<td>1</td>
<td>0.614</td>
<td>0.603</td>
<td>3.57</td>
<td>0.01</td>
</tr>
<tr>
<td>Trust Definition*Strength of Trust</td>
<td>3</td>
<td>0.170</td>
<td>0.163</td>
<td>2.91</td>
<td>0.08</td>
</tr>
<tr>
<td>Trust Definition *Size</td>
<td>3</td>
<td>0.182</td>
<td>0.182</td>
<td>1.08</td>
<td>0.35</td>
</tr>
<tr>
<td>Strength of Trust*Size</td>
<td>3</td>
<td>0.017</td>
<td>0.017</td>
<td>0.10</td>
<td>0.96</td>
</tr>
</tbody>
</table>
**Experiment 1.** We aim to test the impact of the trust types, trust strengths, group size, and the group similarity on recommendation effectiveness. We use the General Linear Model to analyze the variance as well, Table 5-9 summarized the result. We can observe that all the inspected factors have significant influences on the recommendation effectiveness. There is little difference between two definitions of trusts, however. Furthermore, the interactions between those factors do not have much influence to the recommendation effectiveness. Thus, for the following experiments, we only focus on the with-propagation trust.

**Experiment 2.** In this experiment, we aim to evaluate how the strength of trust works in random groups with different sizes. The one-way ANOVA analysis result is presented in Table 5-10. According to the result, we can see that the strength of the trust does not have much influences on the recommendation effectiveness when the target group is randomly formed.

<table>
<thead>
<tr>
<th>Group size</th>
<th>Factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Trust</td>
<td>1</td>
<td>0.110</td>
<td>0.110</td>
<td>0.86</td>
<td>0.355</td>
</tr>
<tr>
<td></td>
<td>Strength</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td>0.038</td>
<td>0.038</td>
<td>0.60</td>
<td>0.438</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>0.008</td>
<td>0.008</td>
<td>0.22</td>
<td>0.643</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1</td>
<td>0.025</td>
<td>0.025</td>
<td>1.11</td>
<td>0.292</td>
</tr>
</tbody>
</table>

**Experiment 3.** In this experiment, we aim to understand how the strength of trust works in groups with high inner similarity. The one-way ANOVA analysis result is presented in Table 5-11 and Figure 5-12. Based on the result, we can see that the trust strength have significant influences on the recommendation effectiveness when the size of target group
is 3 or 8. In particular, we can observe that recommending items to groups with strong trust relationship is more effective than recommending items to group without trust relationships.

Table 5-11. Summary of one-way ANOVA analysis for trust strength in random groups

<table>
<thead>
<tr>
<th>Group size</th>
<th>Factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td>0.062</td>
<td>0.062</td>
<td>0.79</td>
<td>0.376</td>
</tr>
<tr>
<td>3</td>
<td>Trust</td>
<td>1</td>
<td>0.625</td>
<td>0.0625</td>
<td>9.93</td>
<td>0.002</td>
</tr>
<tr>
<td>5</td>
<td>Strength</td>
<td>1</td>
<td>0.094</td>
<td>0.094</td>
<td>2.57</td>
<td>0.109</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1</td>
<td>0.131</td>
<td>0.131</td>
<td>4.66</td>
<td>0.031</td>
</tr>
</tbody>
</table>

Figure 5-12. The impact of the strength of the trust in group with high inner similarity

5.4 Discussion

After conducting experiments on two datasets from different application domains, we find that all the factors have impact on the effectiveness of recommendation results. Firstly, we find that in general, AR approach is better than AP approach. One reason may be the information lost at the earlier stage, which is propagated and increased at the following steps. Secondly, the effectiveness of recommending items to groups with high
inner similarity does not always greater than recommending items to random groups. According to our experiments, this is only true when the target group is small. This conflicts with our normal assumption. Usually, we think that groups with higher similarity will be more likely to have similar opinions on recommended items. However, these may be related to the approach that we used to measure the similarity between users. As we mentioned before, we employed the Pearson Correlation to measure users’ similarities. We discussed the drawbacks of this method before. Although the Pearson Correlation commonly used in recommender systems, it is not necessary a good choice in any situation. Thirdly, after exploring the impact of trust relationship, we do not find obvious differences when target groups are randomly formed. However, for group with high inner similarity, the effectiveness of recommending items to groups with strong trust relationships is better than the effectiveness of recommending items to groups without trust relationships. Finally, regarding the three different aggregation strategies, there is no one strategy that constantly outperforms other strategies. When AP approach is used, the Average Strategy works slightly between than other strategies, especially for large groups. The Average Strategy performs better than other strategies when employing AR approach. The major reason may be due to the way we formed the group. Since we do not collect the real interaction between those group members, the fairness is the only criterion that we can take into account. And the Average Strategy is a simple and effective way to guarantee the fairness among all the group members.
5.5 Conclusion

In this chapter, we evaluated the three basic aggregation strategies in two data sets from different application domains. Especially, we conducted experiments to understand the impact of the group size, group inner similarity, and groups’ social relationship on the effectiveness of recommendation results.
Chapter 6

Clustering-based Group Recommendation Approach

Based on the survey and summarization of current group recommender systems, individual-based generation using collaborative filtering approach is the most widely used method among existing group recommender systems. The biggest issue of adopting individual-based generation is how to cater to the group’s needs based on the obtained individual preferences, especially how to merge individual’s preferences or recommendation lists into a common profile or list. In this chapter, in order to avoid the difficulty of the aggregating process, we propose a clustering-based recommendation generation approach. It divides items into several clusters according to their similarities. Further, we remove those clusters that contain items which are disliked by group members. We also describe a strategy for suggesting the appropriate items for the target groups from the remaining clusters. To illustrate the effectiveness of our approach, we describe a preliminary implementation of our approach with experiments on public dataset, MovieLens.

6.1 Introduction

The majority of the current group recommender systems utilize the individual-based recommendation generation approach, for instance Baltrunas, et al. (2010); Gartrell, et al. (2010); Lieberman, et al. (1999); McCarthy (2002); O’connor, et al. (2001); Yu, et al. (2006). However, as we discussed before, the biggest issue for
individual-based group recommendation approach is the difficulty of merging individual’s profiles or recommendations lists into a common profile or list, especially, when complicated social relationship is taken into account.

In this chapter, we propose a novel solution for group recommendation based on the idea of clustering. As a branch of machine learning, clustering has been applied in a wide variety of applications, e.g., pattern recognition, image processing, and text mining. In order to address the data sparsity and scalability issue of recommender systems, some researchers have stood to leverage clustering techniques, for instance Ungar and Foster (1998); Xue et al. (2005). In our approach, clusters are not only used to assemble similar items together, but they also help us to identify items disliked by the group members. After removing clusters containing disliked items, we further select clusters which include the most liked items among group members. We finally employ a policy to generate the final recommendation list for the target group, such as the most popular or liked items in a cluster. This approach provides several advantages. Firstly, thanks to removing clusters containing disliked items, this approach prevents one single member from getting extremely unhappy about the recommended items. This is consistent with the goal of the Least Misery Strategy. Secondly, a designer does not have to struggle with the incorporation of group contexts into an aggregation strategy, since leveraging group contexts remains an open issue. Third, this approach leads to suggest high-quality items. A high-quality item means a representative item among a set of similar items, which has been most consumed or rated with the highest score. In general, users prefer higher-quality items among a list of similar items. In addition, since this approach harnesses the community preference to some extent, it inherits the advantages of community-based
recommendation generation approach as well (e.g., able to recommend items to groups with a few preferences, and able to protect members’ privacy).

Thus, in this chapter, we focus on illustrating our clustering-based group recommendation approach. In order to understand the effectiveness of this proposed approach, we implement this approach by leveraging hadoop Map/Reduce framework and algorithms provided by Mahout. We then used MovieLens data set to evaluate this approach.

6.2 Clustering-based Approach

Figure 6-1 shows the framework of our proposed approach. This approach includes four major steps. First, it needs to represent the collected data in a specific way so that the data can be used in a clustering algorithm. Generally, the interaction data between users and items will be represented in a Vector Space Model. For some data, the process of feature selection is also needed. Feature selection means to identify those features that can accurately represent users’ preferences. Second, we use clustering engine to separate items’ vectors into different groups based on a distance measurement. After this step, similar items will be grouped into a same cluster. Third, removing step is to recognize those clusters that contain disliked items by the target group members. Since those items in the clusters including negative items are similar to each other, they are also likely to be neglected by the group. Finally, we select items from clusters containing most liked items by the group members. Many strategies could be employed to help select
potential interesting items. For instance, items have been mostly viewed, or items have highest average ratings. In following subsections, we discuss each of four steps in details.

![Clustering-based approach](image)

**Figure 6-1. Clustering-based approach**

### 6.2.1 Data PreProcess

To simplify the illustration of this step, we assume users’ preferences are originally represented by the most common format \((userid, itemid, preference)\). Since we are interested in clustering items, we need to represent each item into in a vector in order to support the next step processing. In our approach, all existing users are used as features of items, ordered by users’ identifications. These features form a dictionary represented as following:

\[
Dictionary = (User_1, User_2, ..., User_n)
\]  

(6-1)

According to the dictionary, each item can be represented as a following vector:

\[
I_i = ((u_1, p_1), (u_2, p_2), ..., (u_n, p_n))
\]  

(6-2)

For each dimension \((user_i, p_i)\), it is the corresponding user, \(ui\), expressed his or her preference on item \(I_i\) with rating of \(p_i\). For instance, if only \(User1\) and \(User4\) provided ratings to \(Item1\) and the ratings are 3 and 5 respectively. The vector of \(Item1\) is \(((1, 3), (4, 5))\). Because the rating data in recommender systems are quite sparse, items represented in the above format could reduce the space cost to a reasonable level.
6.2.2 Clustering

K-Means is a commonly used technique for unsupervised clustering. In our approach, we use K-means clustering to group items into clusters based on who rated them. The process includes three major procedures. First, given the input of item vectors, the clustering approach needs to prepare original centroids for each cluster. As we know, the performance of K-means clustering approach largely depends on how the centroids are selected. Second, it is crucial to employ a proper function to measure the similarity between any two item vectors. Finally, K-means algorithm can start to assign item vectors into different clusters based on the distances to the selected centroids. And this whole process can repeat several times until the predefined parameters are satisfied. The clustering process is described in more detail below.

**Step 1: Select Centroids.**

Generally, researchers employ two ways to generate initial centroids, which are randomly generating and using canopy clustering to identify the initial centroids respectively. For the sake of generality and simplicity, randomly generating initial centroids is widely used. However, users need to decide the number of clusters beforehand, while for many real-world clustering problems, the number of clusters is unknown. One algorithm, called the canopy generation, can estimate the number of clusters. In addition, the canopy algorithm is able to speed up the clustering in the case of large data sets. In our approach, we select the canopy algorithm and cosine distance measurement (more details in the next subsection) to generate the initial centroids. The details of canopy algorithm are as following.
Algorithm 1 Canopy clustering: Given a list of item vectors, and two distance T1, T2, and T1>T2

1: Select any point from this list to form a canopy center.
2: Approximates its distance to all other points in the list.
3: Put all the points which fall within the distance threshold of T1 into a canopy.
4: Remove from the original list of all those points which fall within the threshold of T2. These points are excluded from being the center of and forming new canopies.
5: Repeat from step 1 to 4 until the original list is empty.

One issue about canopy algorithm is to decide the value of parameters T1 and T2. In order to get more accurate clusters, designers need to try different values of T1 and T2 and select the better one for the specific application.

Step 2: Distance Measurement.

To select an appropriate distance measurement is the most important issue for clustering algorithms. From another perspective, this equals to find a method to compute the similarity between items. There are a number of different ways to calculate the similarity between items. Here, we illustrate two approaches. They are cosine and Tanimoto distance measurements.

- Cosine distance measurement: the distance between items is measured by computing the cosine of the angle between two item vectors. Formally, the distance between item \( i \) and \( j \) can be presented as:

\[
Distance(i, j) = 1 - cosine(i, j) = 1 - \frac{i \cdot j}{||i|| ||j||}
\]
• **Tanimoto Distance Measurement**: In this case, the distance between two items $i$ and $j$ is measured by computing the size of intersection divided by the size of union. The equation is shown as following:

$$Distance(i,j) = 1 - \frac{|i \cap j|}{|i \cup j|}$$

**Step 3: K-Means Clustering**

Once we decide how to generate the initial centroids and how to measure the distances between item vectors, the next step is to use the K-Means clustering algorithm to generate clusters. Algorithm 2 shows the detailed steps of K-Means clustering.

**Algorithm 2 K-Means Clustering**: Given a list of item vectors, a distance measurement $d$, and a set of initial centroids.

1: Compute the distance of each vector to all the initial centroids.
2: Assign each vector to the group that has the closest centroid
3: Update the centroids based on the assigned vectors.
4: Repeat from step 1 to 3 until no vectors change their membership.

**6.2.3 Item Selection**

When the above clusters are obtained, the target group members’ information will start to feed into this approach. Intuitively, users would not like to experience similar items that they disliked before. Therefore, in order to prevent the potential dissatisfaction, we first remove those clusters that contain several disliked items from group members. Normally, different group members may have different rating habits. For instance, some members prefer to give high values to their viewed items, while some members prefer to
provide constant low ratings. Thus, we distinguish the threshold for disliked items of each member by taking their average rating values into account. Algorithm 3 describes the procedure of removing those potential disliked items.

Algorithm 3 Removing: Given a list of clusters and a target group $g$.

1: Compute the average rating value of each group member.
2: Collect disliked items of the whole group by comparing the rating values to the obtained average rating values
3: Remove these clusters contain disliked items of the target group.

To some extent, this removing process is to prevent one or more members from experiencing potential disliked items. This similar criterion has been employed in PolyLens as well (O’connor, et al. 2001). On the other hand, this approach is also likely to neglect those items that are extremely liked by some members. For instance, one member likes science fiction movies very much, while another group member hates watching fiction movies. According to this process, the cluster containing fiction movies will be removed.

If we still have multiple clusters left, we then further select those top clusters that include the most liked items of group members. For some situation, we can even adjust the weight of liked items. For instance, for an elder people traveling with his/her family, more weighted can be assigned to this user’s liked items so that the system is able to provide those clusters that are more likely to find items that are similar to the liked items by this elder member.

After obtained the potential clusters which may contain the liked items by the target group, one final step is to select the top relevant items from these clusters. The
relevance of an item in this paper is defined as the average rating value of all ratings from the neighbors of group members. Formally, it is defined and computed with equation (6-3):

\[
relevance(i) = \frac{\sum_{m \in G} \sum_{u \in N(m)} rating(u, i)}{\sum_{m \in G} N(m)} \tag{6-3}
\]

Here, \(N(m)\) denotes the number of neighbors from group member \(m\). \(rating(u, item)\) is the rate that neighbor \(u\) gave to item \(i\). Many strategies can be adopted to generate the top items list in this process. Similar to the methods presented in (Zhou et al. 2009), these strategies include, but are not limited to:

- Iteratively select one item with the highest relevance value from each top cluster, until a maximum of items is reached. The order in which selected items are recommended is the same with the order in which they are selected.

- Select items with the highest relevance value from the top cluster. If there are not enough un-viewed items, items with higher relevance value from the second top cluster are recommended.

6.3 Evaluation

6.3.1 System Architecture

A conceptual level architecture of our approach is shown in Figure 6-2. Web-based user console is an interface between groups and the group recommender system. It not only presents the recommended items to a group, but also captures the feedback from the
group in order to learn users’ preferences more precisely for a new cycle recommendation. We use Hadoop framework to cluster items based on who viewed them. Hadoop is an open source software used for distributed computing that focuses on addressing the scalability issue. Its major components include: Hadoop distributed file system (HDFS) and MapReduce. HDFS is the storage system used by Hadoop cluster; while Hadoop Map/Reduce is a parallel programming model that is used to retrieve the data from the Hadoop cluster. In particular, we leveraged the k-Means clustering algorithm provided by Mahout, an open source machine learning package, to cluster items. Item cluster repository contains the clustered items after applying k-Means clustering algorithm. The recommendation engine recommends top items to the target group according to item clusters and group members’ preferences. As demonstrated in Figure 6-2, a sequence process of removing unlike clusters, obtaining top clusters, and retrieving top items are organized in recommendation engine. We implement a prototype in Java and build a pseudo model Hadoop cluster.

Figure 6-2. Architecture of clustering-based recommender
6.3.2 Experiments

We used the MovieLens data set to verify the feasibility and effectiveness of our proposed approach. After preprocessing the MovieLens data set (items have been rated by at least 10 users and users have provided at least 25 ratings), we get 5623 distinct users, 3260 movies, and 989412 ratings. We conduct 2 major experiments to test the effectiveness of our proposed method comparing with random suggestion and the AR approaches. In order to test the performance of our approach in different sizes of groups, we conduct experiments in groups with size of 2, 3 and 5 respectively.

According to our literature reviews on the evaluation methods employed in current group recommender systems, most systems primarily focus on the accuracy. And researchers prefer to compare the differences between actual and predicted value from target groups. However, for public data set, it is difficult for us to capture the actual value of groups. Thus, in this paper, we utilize group’s satisfaction to evaluate these approaches’ performance. We define groups’ satisfaction as follow:

\[
p(m) = \begin{cases} \frac{\sum_{i\in TopItems} P(item)}{TopItems}, & (P(item) = 1, \text{average}R(m) < \text{predict}R(m, i)) \\ 0, & \text{other} \end{cases}
\]  

(6-4)

\[
GroupSatisfaction(G) = \begin{cases} 1, & \forall m \in G, p(m) > 0.5 \\ 0, & \text{other} \end{cases}
\]  

(6-5)

Here, averageR(m) is the average score of all ratings provided by member m; predictR(m, i) is the predicted rating that member m may provide to item m. p(m) is the percentage of predicted values of recommended items are greater than the average rating of member m. In order to define the group satisfaction, we assume that if more than 50% of the recommended items from a system are greater than the average rating of a user, we
consider this system is effective and will be satisfied by this user. Thus, if all the group members are satisfied with the system simultaneously, we assume that the whole group will satisfy with the group recommender.

Now, we will offer detailed information of the two experiments have been conducted.

**Experiment 1**

In this experiment, we aim to compare the effectiveness of random suggestion, AR with the Average Strategy, and our proposed cluster-based approach, when the target group is randomly formed. We synthesize 300 groups (100 groups, each with 2 members; another 100 groups, each with 3 members, and the remaining 100 groups, each with 5 members). The experiment steps are as follow:

1) Random suggestion

   - Randomly generate 10 items which have not been viewed by any group members;

2) Cluster-based approach

   - According to the cluster-based approach, generate top 10 items have not been viewed by any group members (Top items are these items that have higher average ratings among all users who provided ratings to these items);

3) AR approach

   - Generate recommendation lists for each group member using user-based collaborative filtering algorithm;
   - Select top 10 common items appeared in all group members’ recommendation lists (we employed the Average Strategy to aggregate the lists);
The following are the same steps for all these three methods:

- Predict the preference values of suggested items for each member by employing user-based collaborative filtering algorithm;
- Compare the predicted values with the average rating values of all group members;
- Calculate the group satisfaction

The results of these approaches are shown in Table 6-1. We can observe that our proposed approach is much better than random suggesting items to the target group. However, this method is still not as good as the AR approach using the Average Strategy.

<table>
<thead>
<tr>
<th>Approach name</th>
<th>Percentage of satisfied group (Size 2)</th>
<th>Percentage of satisfied group (Size 3)</th>
<th>Percentage of satisfied group (Size 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Suggestion</td>
<td>17%</td>
<td>9%</td>
<td>2%</td>
</tr>
<tr>
<td>Cluster-based approach</td>
<td>61%</td>
<td>47%</td>
<td>21%</td>
</tr>
<tr>
<td>Aggregating recommendation lists</td>
<td>75%</td>
<td>86%</td>
<td>53%</td>
</tr>
</tbody>
</table>

**Experiment 2**

In this experiment, we aim to compare the effectiveness of random suggestion, AR with the Average Strategy, and our proposed approach, when targets groups have higher inner similarity. Similarly, we create 300 groups. In particular, the first 100 groups, each group contains 2 members; the second 100 groups, each one includes 3 members; and the remaining 100 groups, each one has 5 members. The experiment steps are the same to the previous experiment. The result is presented in Table 6-2. We can observer that our proposed approach is still much better than random suggestion.
Moreover, the performance of our approach is close to that of AR approach when handling groups with higher similarity.

Table 6-2. Results of Experiment 2

<table>
<thead>
<tr>
<th>Approach name</th>
<th>Percentage of satisfied group(Size 2)</th>
<th>Percentage of satisfied group(Size 3)</th>
<th>Percentage of satisfied group(Size 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Suggestion</td>
<td>16%</td>
<td>15%</td>
<td>8%</td>
</tr>
<tr>
<td>Cluster-based approach</td>
<td>61%</td>
<td>55%</td>
<td>46%</td>
</tr>
<tr>
<td>Aggregating recommendation lists</td>
<td>78%</td>
<td>71%</td>
<td>70%</td>
</tr>
</tbody>
</table>

6.4 Discussion

Based on the experiments’ results, we can see that our proposed approach works much better than random suggestion. However, it is still not as good as the AR approach. One of the reasons may be because group members’ information has not been fully used in our approach. For instance, we only consider removing disliked items of the target group members, but recommending items like by all the exiting users. We have not looked deep into group members’ unique preferences. One aspect that we may improve the performance of this approach is to try different similarity measurements or even clustering algorithms. Additionally, we can further evaluate the performance of these three approaches in terms of coverage, confidence, trust, novelty, serendipity, utility, robustness, adaptivity, and scalability (Shani and Gunawardana 2011).

Fortunately, the fundamental idea of our approach is different from the AR approach. Therefore, it is possible that we can combine the advantages of these two approaches, since in general, a hybrid approach works better than the base approaches (Burke 2002).
6.5 Conclusion

In this chapter, we proposed a new group recommendation approach, which is fundamentally different from the existing group recommender methods. The major advantage of this approach is that it does not necessarily need a merging strategy to aggregating individual profiles or recommendation lists into a common profile or recommendation list, since selecting an appropriate merging strategy is still an open issue in group recommender systems. However, the current approach is still not as good as the AR approach. In future, we plan to evaluate the effectiveness of combining our approach with the AR approach.
Chapter 7

Conclusion and Future Work

In this thesis, we have explored the research issues in group recommendation systems. As we stated in the introduction of Chapter 1, the primary objective of this thesis is to provide the generic implementation guidelines to new designers and propose a potential interesting approach so as to advance the discussion in this field.

First, based on the representative group recommender architectures, we proposed a generic architecture so that a new designer has a high level understanding of group recommender systems and understand in where to personalize his or her recommender system. Furthermore, we conducted a comprehensive and systematic experiment to compare the effectiveness of two common recommendation approaches (AP and AR). In particular, we tested the impact of three crucial factors (the group size, group similarity, and group social relationship) on the performance of aggregation strategies, especially the Average, Median and Least Misery strategies.

Besides, in Chapter 6, we proposed a clustering-based group recommendation approach. Aggregating individual’s profiles or ratings into a group profile or rating is a challenge issue due to the complexity of group social interactions. Although current aggregating strategies prefer not to take group contexts into account, they may not be able to cater to the different needs of different types of groups. Our clustering-based approach precludes aggregating process among group members, which focuses on removing those items that would be dislike by the target group, then combining the neighborhoods opinions to obtain the right items for the target group.
In the future, we plan to further investigate and improve the proposed group recommendation approach. First, we can improve the experiments we did in Chapter 4 and 5. For instance, different data sets and similarity measurements can be tested. Second, we consider to combine our clustering-based approach with the AR approach. Since two approaches are based on different ideas, the hybrid approach should inherit the advantages from both basic approaches. Last but not least, although in this thesis we focused on recommending items to satisfy a group of people, understanding the behavior of groups can also help us to predict the potential interests of an individual user. Based on the research of Hwang et al. (2005), peer pressure and group conformity can affect the buying behaviors of individuals. Thus, by understanding the interests of a group of users, we can also derive the potential interest of an individual user.
References


MAHOUT http://mahout.apache.org/.


