FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Enhanced Group Recommender System and Visualization

Wei Wang

A thesis submitted for the Degree of

Doctor of Philosophy

University of Technology Sydney
March, 2016

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

_____

ACKNOWLEDGEMENTS

This thesis represents not only my work at UTS, it is a milestone for almost four years of work at UTS and specifically within the Decision Systems and e-Service Intelligence (DeSI) Lab. This thesis is also the result of many experiences I have encountered at UTS from dozens of remarkable individuals who I also wish to acknowledge.

First and foremost I wish to thank my principal supervisor, Professor Guangquan Zhang, and my co-supervisor, Professor Jie Lu. They have been supportive since the days I began working on DeSI as an undergraduate and have covered all aspects of my PhD study, including research methodology, research topic selection, experiments, academic writing skills and thesis writing, and even the sentence structure and formulas. Their critical comments and suggestions have strengthened my study significantly. Their strict academic attitude and respectful personalities have benefited my PhD study and will be a great treasure throughout my life. Without their excellent supervision and continuous encouragement, this research could not have been finished on time. Thanks to you all for your kind help.

I would like to express my sincerest gratitude to Prof. Dacheng Tao, Dr. Wei Bian and Dr. Tianyi Zhou for their helping me come up with and sharing code of the matrix decomposition topic; to Prof. Maolin Huang and Miss Wenbo Wang for data visualization advice; to Dr. Dianshuang Wu, Dr. Mingsong Mao, for efficient, and most important, friendly assistance in recommender systems and to Ms. Barbara Munday and Ms. Sue Felix for helping me to correct English presentation problems in my publications and this thesis.

The most difficult ones to thank are my family; my mother and father, my wife and son. I appreciate your understanding, your encouragement and enthusiasm, and

your help. I couldn't have done it without your support. Without you nothing in my life would be possible.

Last but not least, I am grateful to the FEIT Travel fund, the Vice-Chancellor's Postgraduate Conference Fund, and the ARC scholarship.

ABSTRACT

Requirement of group recommender systems (GRSs) is experiencing a dramatic growth due to intelligent services being applied more broadly and involved in more and more domains. However, effectivity and interpretability are still two challenges in GRSs. A typical scenario is: a group is formed randomly without active organizing in advance and sufficient negotiation between members before recommending, such as e-shopping and e-tourism. Therefore, deeply modeling the group profile is the first key part to generate recommendations. Moreover, accurately predicting should be a problem under biased and limited information provided by users. The interpretability challenge is that most of GRSs are black boxes for providing no necessary explanation of recommendations but only a list. It is quite important to convince members to make them understand why the specific recommendations are reasonable. Thus, explaining the reason generated recommendations and relationships between members needs to be investigated.

This research aims to handle these two challenges in both theoretical and practical aspects. A novel group recommendation approach is developed and aims to maximize satisfaction within random groups by modeling the group profiles through the analysis of contributed member ratings alone. First, the Contribution Score is defined to numerically measure each member's importance in terms of the sub-rating matrix which makes it practical even when the matrix is highly incomplete and sparse. Second, a local collaborative filtering method is developed to address the biased rating problem caused by severe preference conflicting in random groups. An adaptive average rating calculating model is proposed taking into consideration of the target item by reducing the set to those which are highly relevant to it. By integrating these two models, a Contribution Score-based Group Recommendation (CS-GR) approach is developed to efficiently depict groups. Also, a novel hierarchy graph-based visualization method, based on data visualization techniques, which are

powerful tools to offer intuitive abstractions of concepts, is suggested to offer explanations for users. First a higher level of abstraction of the overall recommender modules, such as group profile modeling and prediction calculating, is presented using a hierarchy graph. To do this, all the entities involved in a group recommender process are summarized and visualized as nodes in the graph and the edges in the graph represent information inherited. Second, the layout provides detailed information for individual members to track their influences in the system by adding pie charts at each single node to show individual influences for all involved members. This enables members to track and compare their influences with others in every single procedure.

This research provides the GRSs effectivity for the biased and sparse information which can be handled to model the group and generate the predictions. The scalability and efficiency are also guaranteed because only rating information is needed and matrix decomposition technique is employed. The visualization is used to provide both overall and detailed explanation for users.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The dramatic growth of information services makes significant advances in the recommender system technology. Recommender systems (RS) (Ricci, Rokach & Shapira 2011)are usually designed for individual users to provide personalized content best fitting their preferences. In recent years, many domains are further extended with a group aspect, such as movie (Quijano-Sanchez, Recio-Garcia & Diaz-Agudo 2011), TV (Sotelo et al. 2009; Yu et al. 2006), music (Baccigalupo & Plaza 2007; McCarthy & Anagnost 1998; Popescu & Pu 2012) and tourism (Ardissono et al. 2003; McCarthy, Salamó, et al. 2006c). One reason is that the users need recommendations, i.e. group members, are not able to specify their preferences explicitly. (Goren-Bar & Glinansky 2004a) introduce a system to filter the TV programs for families. This event requires the system to collect all the individual preferences of family members and model an adaptive one when some of them form a group, viewing together. PocketRestaurantFinder (McCarthy 2002b) is a system that finds a restaurant for dinner for a group of people. Every member presents his/her opinions on some conditions such as distance, price and so on. The system builds a group preference model and evaluates the restaurant according to this model. Another case is that group members are not able to engage for face-to-face interaction. A system proposed by (Zhang, Wang & Feng 2013) helps people planning the events like a party. The interesting part of this system is that it takes the geographic information of attendees into consideration. In (Lorenzi et al. 2008) a multi-agent tourism recommender system for a group of uses is based on the collaboration of two

types of agents: user agent and recommender agent. The user agent stores the preferences of all the members and the recommender agent stores the travel information locally. The recommendations are produced by exchanging information between these two types of agents.

Group recommender systems (GRSs) are then proposed to suggest items for group use when users actively establish a group or form a random group in a specific environment. Another purpose is, like individual recommender systems, to alleviate information overload by suggesting items with up-to-date information. Members in the group specify their personal preference without considering the other members' interests. Systems automatically take all members into consideration and suggest items optimal for the group. Many user-generated content (UGC) which have been utilized in individual RSs are also used in GRS, such as the most common one, ratings; member relationships, social information; context information. Many automatic mechanisms are utilized, for instance, some rating based systems (Jameson, Baldes & Kleinbauer 2004) used average rating to evaluate the satisfaction of a group for an item. Other UGC such as social information is also utilized by many systems (Wang et al. 2012) to discover items via social relationship, with the assumption that items preferred by their friends are also preferred by them. Context information along with the wide use of portable devices can automatically generate items, such as recommend restaurants when it is daytime and recommend hospitals when it is night time. Also most GRS methods are developed using the same input as individual RS, the accuracy of predictions and performance of the systems then are more complex than for individuals.

A prediction for a target item of most neighbour-based collaborative filtering (CF) methods is a linear combination of the estimations from all the user/item neighbours. Therefore, the hidden concept behind predicting method is that the pseudo user's deviations can be approximated by linearly combining the deviations of neighbours over entire domain. However, this combination can be inaccurate when ratings are on sparse domains because global approximation forces the combination focus on dense domains. In this case, the predictions when target items are located in extreme and sparse domain will be inaccurate. Hence, instead of seeking global approximation solution, the local collaborative filtering aims to find locally optimal solution. This

approach calculates predictions by adaptively changing average ratings considering items relevant to the target item. A closer inter-related subset of items, i.e. the potential neighbours of the target item, is identified to calculate local average rating. Therefore, local average rating on subset of items close to the target item can be used in rating prediction to address the biased rating problem.

Another issue is how to identify the neighbours accurately to calculate the predictions. The usual way is to use some similarity measures to evaluate the degree of closeness between two users/items. However, most similarities only simply consider the differences over every dimension of two rating vectors. Incorporating the information entropy is to take the relative relationships of these differences into account for coherence measuring. The information entropy treats all the independent differences as a whole. It captures additional coherent information by measuring relative rating differences comparing to traditional similarities only considering individual absolute rating differences. The proposed similarity model attempts to accurately measure the coherence between two users.

A very important reason for low accuracy is that it is highly dependent on the nature and organization of the groups. To address this issue, many improved methods are proposed, but these analysing methods need additional user knowledge or UGC up to now. Therefore, these methods are limited to extend to new domain. All the user-importance-analysing methods need additional information such as user knowledge or user generated-content or social relationships so far. Separable non-negative matrix factorization (SNMF) is an appropriate technique to analyse representativeness only based on ratings. SNMF guarantees a unique and stable solution for a specific matrix which means the decompositions are not influenced by initial values. SNMF is easy to extend to large groups for its great scalability because only the ratings of members need to be analysed.

Another performance issue is the lack of explanation for most RS. The trackable hierarchy method addresses these problems because it supports system explanation and detailed individual influence data (i.e. group ratings, neighbour similarities and predictions). First a hierarchy graph-based model is developed to build a higher level of abstraction of the system explanation. The model summarizes all involved entities

and graphs them as nodes. The edges of the graph show inherited information. Thus the nodes and the edges combined demonstrate the entire recommender process. The method also illustrates influence data from different members for each node as a pie chart for individual members to track their influence within the system. The individual influences for each member are detailed in each node using pie chart, providing some insight with which to gauge the quality of the recommendations.

1.2 RESEARCH QUESTIONS

This research aims to develop a set of recommendation approaches and related visualization methods for the applications focusing on recommender systems. To summarize, the following research questions will be answered by this research:

Question 1. How to improve the prediction accuracy considering the biased rating patterns?

Question 2. How to identify the users that share the similar rating pattern of the profile considering not only the differences between ratings on single item but the relationships of these differences?

Question 3. How to quantitatively analyse the groups and build the group profiles to represent the entire groups?

Question 4. How to explain the recommender process and the results to members to earn trust?

1.3 RESEARCH OBJECTIVES

This research aims to achieve the following objectives, which are expected to answer the above research questions:

Objective 1. To develop a relevance measure to calculate adaptive average rating.

This objective corresponds to research Question 1. A measure to evaluate two vectors considering missing values and domain range will be proposed. The

relevance between two items will be measured using this measure by comparing two item rating vectors.

Objective 2. To develop a predicting model to deal with biased ratings of the members.

This objective corresponds to research Question 1. A novel recommendation method called local collaborative filtering method will be developed. In this method, the items with similar pattern to the target item will be filtered using the proposed relevance measure and the alternative average rating is calculated over these items. The unknown ratings are calculated by use of the alternative average rating.

Objective 3. To develop a comprehensive similarity measure considering the relative relationships of the differences between ratings.

This objective corresponds to research Question 2. Based on the information entropy theory, a comprehensive similarity measure method considering the relative relationships of the differences between ratings will be developed. The differences over all the common dimensions are treated as a distribution of a random variable, the uncertainty of the distribution is evaluated using information entropy theory. This measure places more emphasis on the entire differences than on the individual ones.

Objective 4. To develop a neighbour identification model to select out highly related users/items.

This objective corresponds to research Question 2. A hybrid similarity measure will be developed. Pearson correlation-based similarity (PCC) will be used to measure the absolute differences over every dimension. Information entropy-driven similarity will be used to measure the relative differences over all dimensions. The neighbours will be selected according to the hybrid similarity. The neighbour-based method will then be used to compute the predictions.

Objective 5. To develop a contribution score measure.

This objective corresponds to research Question 3. To evaluate the importance of the members, a contribution score measure is defined. The measure takes ratings of the members as the input and the representativeness of the members are calculated

using SNMF methods. The results of the CS measure are used to model the group profiles.

Objective 6. To develop a group recommendation method by taking contribution scores of members into consideration.

This objective corresponds to research Question 3. A pseudo user representing the entire group is modelled taking into consideration the contribution score results of all the members. The group profile is designed to maximize the satisfaction of the entire group. The user-based collaborative filtering is then used to generate the recommendations according to the pseudo user's profile.

Objective 7. To develop a visualization method to explain the recommender process.

This objective corresponds to research Question 4. The recommender procedures are often theoretically complex and black boxes for users. It is important for users to trust the system to obtain the necessary explanations. The multi-level hierarchy graph is used to explain key procedures within the entire group/individual recommender systems: profile modeling, neighbour identification and recommendation representation. Numerical values such as similarities, predictions are demonstrated as features of the components: colour, position and width and so on.

Objective 8. To develop a visualization method for members to track their individual influences in recommending.

This objective corresponds to research Question 4. The density-based pie charts are used to illustrate the influences from every group member. Members can compare their influences with others in each single node and track their influences along the entire recommender process. This visualization is important for members to understand the composition and relative representativeness of the group and it gives an insight for the final recommendations why some are suggested but are seen as less relevant for them.

1.4 RESEARCH SIGNIFICANCE

This research work has both theoretical and practical significances in the area of recommender systems.

1.4.1 THEORETICAL SIGNIFICANCE

Theoretically, the research develops a set of recommendation algorithms and solves the four aspects in RSs below.

- **Biased Rating Problem**. This research solves the prediction problem when the ratings of the members are biased. Referring to the research objective 1 and 2, a novel collaborative filtering method is established with the item relevance analysing. It is important to note that either aggregating the deviations from average ratings or aggregating the raw ratings assumes the unknown ratings can be approximated by linear combining all the neighbours. A novel collaborative filtering method is proposed to make this approximation focusing on the target item considering the rating pattern of it.
- **Neighbour Identification**. This research solves selecting neighbours according to the relative relationships of rating differences of two users/items over common dimensions. This problem arises from the research objective 3.
- **Profile Modeling**. This research solves a group modeling problem taking member analysing into consideration. A measure by analysing the members' ratings, called contribution score, can evaluate the importance of them. Referring to the research objective 5, the contribution scores are then used to build the group profile to represent the profile of the entire group. Thus, a method using contribution score-based group profile is raised as a recommendation task to be solved in the research.
- **Visualization Integrating**. This research solves a system explanation problem. Referring to the research objective 7, the data visualization techniques for structural and unstructured data are used to provide necessary explanation including key procedures in the recommender

process. Thus, members can obtain an intuitive understanding of the entire system for the group and are allowed to track their individual influences.

1.4.2 PRACTICAL SIGNIFICANCE

Practically, the research provides guidelines of how to improve the performance of the recommender process by addressing problems embedded in real data of RS.

- Accuracy. The biased and unbalanced data are common in real data of RS. The researches introduce new measures to evaluate the performance of the recommender process and then provide an adaptive method to obtain better results compared to the traditional approximation solution.
- Scalability. The new method can be easily extended because of the simplicity of the required data source. The new methods are designed to only rely on the numerical ratings which is the most common type of UGC in many domains.
- Efficiency. The huge volume of the real data makes the computational cost unacceptable for practical utilization. New methods are designed which are linearly dependent on the length of rating vector profile (individual or group) which makes limited additional computational cost.
- Interpretability. The explanations for the systems are provided to earn trust of the users for the systems. The information about the users and intermediate results are illustrated using intuitive visual ways.

1.5 RESEARCH METHODOLOGY AND PROCESS

The overall research methodology and research process are designed as follows.

1.5.1 RESEARCH METHODOLOGY

Research methodology is the “collections of problem solving methods governed by a set of principles and a common philosophy for solving targeted problems” (Gallupe, 2007). This research belongs to the information system domain, for which various methodologies have been proposed and applied such as case study, field study, design

research, archival research, field experiment, laboratory experiment, and survey and action research (Creswell, 2013; Yin, 2013; Vaishnavi and Kuechler, 2015).

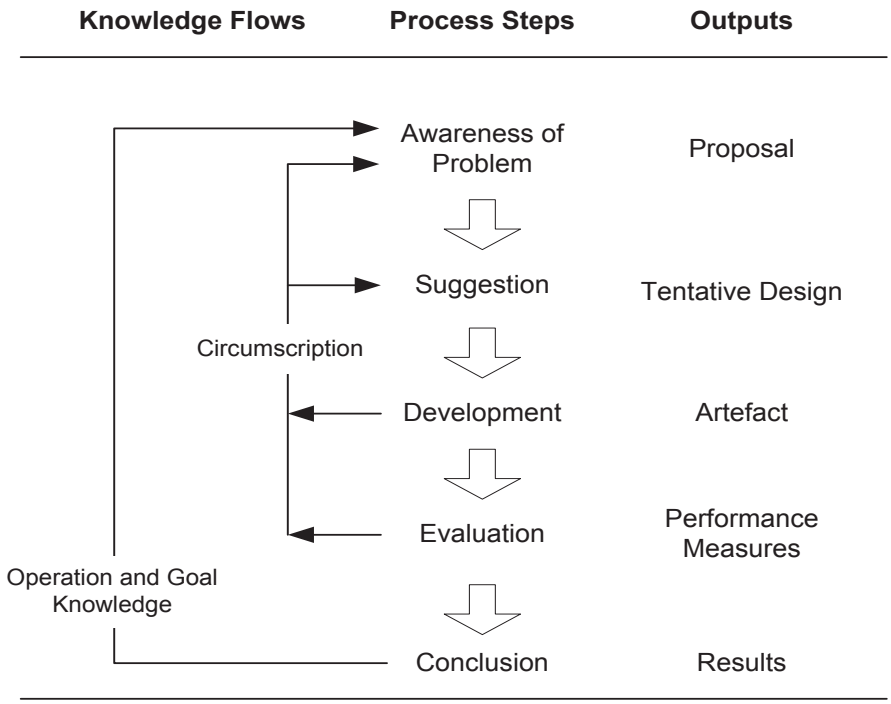


Figure 1-1. Research Methodology

1.5.2 RESEARCH PROCESS

This research was planned according to the methodology of design research. First, the subject of recommender systems was chosen as a very broad research topic of this research. A literature review of previous research in recommender system area and related method incorporating was conducted, and existing literature was retrieved and critically reviewed. The results of the literature review helped to identify specific research questions to be directly addressed in this research. As the research questions grew clearer and more definite, more literature closely related to the research questions was reviewed. Because the existing work in the literature lacks adequate study on some key procedure problems including biased ratings, neighbour selection, profile building and explanation for CF methods abstracted from the research questions/objectives, this research presented some novel models, developed corresponding CF algorithms and recommender systems. Appropriate datasets are

selected to evaluate the proposed set of recommender systems. As indicated in Figure 1-1, evaluation results are fed back to previous steps so that the research outcomes are progressively improved until satisfying results are drawn from evaluations. Finally, writing up the PhD thesis is done at the end of the research.

1.6 THESIS STRUCTURE

This thesis contains seven chapters. Chapter 1 presents the research background, research questions, objectives, significance, research methodology and process, and the basic notations for recommendation problems. Chapter 2 presents the literature relevant to this study, including classical and state-of-the-art recommendation techniques and applications, and related techniques are incorporated in the proposed methods. Chapter 3 proposes a novel recommender system focusing on dealing with biased ratings by incorporating a local approximation into CF techniques. Chapter 4 develops a recommender system based on a hybrid similarity measure considering both absolute rating differences and relative relationships between these differences. Chapter 5 proposes a group recommender system taking user representativeness analysing into consideration. Chapter 6 develops a visualization method for recommender systems especially the method proposed in Chapter 6. Chapter 7 presents the conclusions and further study recommendations. The structure and contents of the thesis are as indicated in Figure 1-2.

1.7 PUBLICATIONS RELATED TO THIS THESIS

Below is a list of the refereed international journal and conference papers associated with my PhD research that have been submitted, accepted and published:

Refereed international journal publications:

1) Wei Wang, Guangquan Zhang, Jie Lu, Member Contribution-based Group Recommender System, Decision Support Systems, vol. 87, pp. 80-93, 2016.

2) Wei Wang, Guangquan Zhang, Jie Lu, Trackable Hierarchy Visualization for Group Recommender Systems, IEEE transactions on Systems. (2nd-round review)

3) Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, Guangquan Zhang. Recommender system application developments: A survey. Decision Support System, vol 74, pp 12–32, 2015.

4) Wei Wang, Guangquan Zhang, Jie Lu, Collaborative filtering with entropy-driven user similarity in recommender systems, International Journal of Intelligent Systems, vol. 30, no. 8, pp. 854-70.

5) Wei Wang, Guangquan Zhang, Jie Lu, A new similarity measure-based collaborative filtering approach for recommender systems, Proceedings of the 8th International Conference on Intelligent Systems and Knowledge Engineering, Springer Berlin Heidelberg, Shenzhen, China, pp. 443-52.

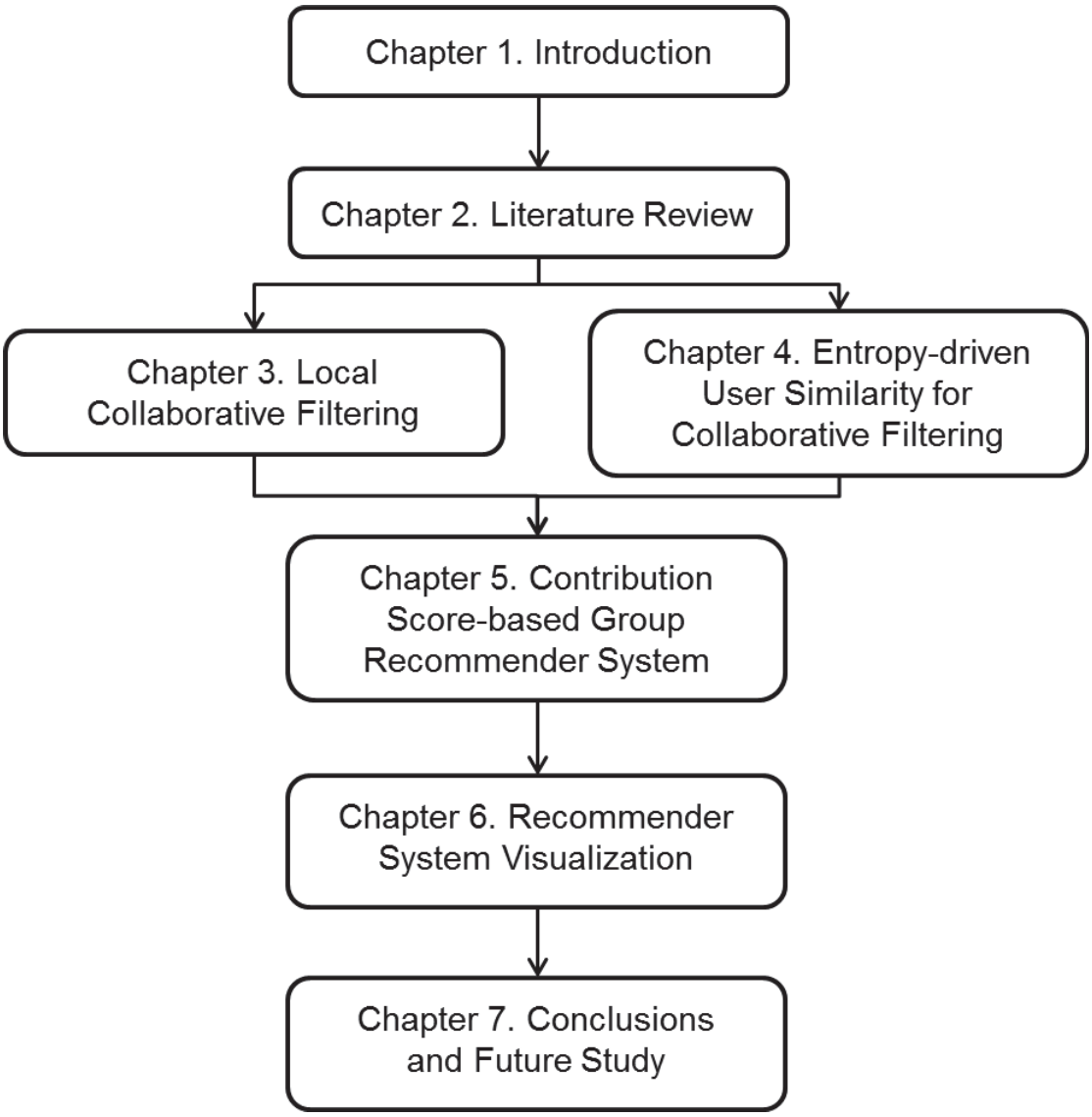


Figure 1-2. The structure and contents of this thesis.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a discussion of relevant work in connection with this research. In Section 2.1, the techniques and applications of individual recommender systems are expatiated. Section 2.2 reviews the group recommender system techniques and applications. Section 2.3 reviews the matrix factorization techniques which are related to this research on the measurement of member representativeness. Section 2.4 reviews the related visualization methods and corresponding employment in recommender systems.

Recommender systems are designed to provide the personalized content that are widely applied in many online services, such as e-commerce, e-government, and e-learning (Chen, Lee & Chen 2005; Linden, Smith & York 2003; Lu et al. 2010). A problem that is often referred to is information overload, where users find it very difficult to obtain information they really require from the massive amount of information available. This limits users' experience when they use these online services. Recommender systems are proposed as one of the most successful techniques available to address this problem by analysing users' information to model users' preferences and target-related information. Both service providers and users benefit from fast and accurate recommendations because they increase revenue and save time.

2.1 BASIC RECOMMENDER SYSTEMS

2.1.1 CONCEPTS

Recommender systems are designed to provide personal generalized items, such as products and schedules, for the particular users (individuals or groups) by predicting a user's interest in an item based on related information about items, users and the interactions between items and users (Bobadilla et al. 2013). RSs are proposed because of the information overload problem in many services in which it is difficult to retrieve the most relevant information from a huge amount of data. The most important feature of an RS is its ability of automaticlly analyzing the users' behaviours and then obtaining their preferences (Amoroso & Reinig 2004).

RSs have been widely used in many service domains in the past two decades (Adomavicius & Tuzhilin 2005). Most techniques used in early RS, such as information retrieval and filtering, are based on key words and demographic information (Goldberg et al. 1992), and since the collaborative filtering technique is used in the mid-1990s, RSs emerged as an independent research area (Adomavicius & Tuzhilin 2005). There have been many recommendation techniques developed since the emergence of recommender systems, including the classic techniques, such as collaborative filtering (CF) (Schafer et al. 2007), content-based (CB) (Pazzani & Billsus 2007) and knowledge-based (KB) (Burke 2000) techniques, and many recently developed advanced recommendation techniques, such as social network-based recommender systems (He & Chu 2010), fuzzy recommender systems (Lu et al. 2013b; Zhang et al. 2013), context aware-based recommender systems (Adomavicius & Tuzhilin 2011) and group recommender systems (Masthoff 2011). RSs have been extended from recommending web-pages, academic articles to a variety of domains including e-commerce (Markellou et al. 2005), e-learning (Lu 2004b), e-government (Lu et al. 2010), and e-tourism (Batet et al. 2012).

In this section, the main recommendation techniques, including traditional methods such as content-based, collaborative filtering based, knowledge-based, and hybrid methods, and recently developed advanced methods, such as computational intelligence based, fuzzy set-based, social network-based, context awareness-based, and group recommendation approaches, will be reviewed.

2.1.2 TECHNIQUES

(1) *Content-based recommendation techniques*

Content-based (CB) recommendation techniques are used when items are described with key words or text features, such as news and academic articles. Some examples of CB recommender systems are WebWatcher (Armstrong et al. 1995) and Websail (Chen et al. 2000). Items are recommended to an active user when she preferred similar items (Pazzani & Billsus 2007). The basic principles of CB recommender systems are: 1) To analyse item descriptions to build profiles with word-value pairs for all the items; 2) To represent the active user's profile by combining all her preferred items' profile; 3) To compare each item's profile with the active user's profile so that only items that have a high degree of similarity with the user profile will be recommended (Pazzani & Billsus 2007).

In CB recommender systems, two types of techniques have been used to generate recommendations. One type is using information retrieval techniques. The relevance, such as the traditional cosine similarity measure, between two items is evaluated and the recommendations are generated heuristically by indexing all the relevance values. The other type is using statistical models and machine learning methods, such as decision tree, naive Bayesian and *k*-nearest neighbours. The typical technique is to learn a classification model to indicate the users' interests from the users' historical data (training data).

The advantages of the CB recommender systems are that they are not sensitive to the new and unpopular items because the content of the items are mostly predefined. It does not need additional information such as the preferences of other users in making recommendations, so it does not suffer from the sparseness problem associated with CF systems. Furthermore, it can be easily explained for users why certain items are recommended by presenting the related contents.

However, one of the main limitations of CB recommender systems is the new user problem. The CB recommendation approach is not able to offer accurate recommendations to a new user since he/she has few rated items. The CB approach also has the overspecialization problem. It can only recommend items to a user according to the preferred items in his/her user profile so it cannot recommend items outside the user's profile. Additionally, in some particular cases, it may not be

desirable for a recommender system to recommend items which are too similar to users, such as different news articles that describe the same event. Another limitation of the CB approach is the item content dependency problem. As the CB approach makes recommendations according to contents of items, it is hard to use CB to recommend items which cannot be represented as keywords, such as images and movies. The CB approach cannot distinguish between the items which are represented by the same set of content features.

(2) *Collaborative-based recommendation techniques*

Collaborative filtering (Resnick et al. 1994a; Sarwar, Karypis, Konstan & Riedl 2001) has become the most widely used method for most recommending systems; it allow users to give numerical valuation for items. According to the nature of the methods, two different types of CF methods, memory-based CF and model-based CF, can be distinguished.

In model-based approaches, the unknown ratings are predicted by establishing a model such as matrix factorization-based models, and its required parameters are learned from the observed UGC. Once the model is established, the training UGC are not stored in memory. Different machine learning algorithms are used to accomplish the model building process such as the Bayesian network (Breese, Heckerman & Kadie 1998b), clustering (Jia, Jin & Liu 2010) and rule-based techniques. These algorithms mainly use a probabilistic approach to compute prediction values for un-rated items (Adomavicius & Tuzhilin 2005; Schafer et al. 2007). Matrix factorization techniques are also widely used in model-based CF methods which are reviewed in detail in Section 2.3.

By contrast, in memory-based approaches, all the training UGC are stored in memory, and predictions are computed by exploration and heuristics. The recommendations are generated by identifying the similar user of the active user or the similar items of the target item. Memory-based CF approaches have been adopted in many practical systems, such as Amazon and Netflix, for their simplicity and high effectiveness. The memory-based CF technique can be divided into user-based and item-based CF approaches (Sarwar, Karypis, Konstan & Reidl 2001). Generally there are two major procedures to make recommendations: identifying “neighbours” and

rating prediction. By representing the profiles as vectors of UGC, often ratings, of the user, a vector relevance measure is then used to identify users/items sharing the similar profile pattern, i.e. "neighbours". Once the neighbours have been selected, an active user's unknown rating is predicted by simply aggregating the neighbours' ratings, and those items that have higher predicated ratings are recommended. The item-based CF approach recommends items to an active user that are similar to the items the user has rated highly in the past. First, the item-based CF approach creates a vector for each item which contains the ratings from all users to the item. For a target item to an active user, it then computes the similarity between the rated items of the active user and the target item. The most similar rated items to the target item are then selected as the item's nearest neighbours. Finally, the prediction for the target item is generated by taking a weighted average of the active user's ratings on the neighbour items. It is found that the item-based algorithms are able to provide the same quality of provided services as the user-based algorithm but with less online computation because the relationships between items are relatively static compared with the relationships between users (Sarwar, Karypis, Konstan & Reidl 2001).

There are a number of similarity measures to evaluate the relevance. Jaccard (Koutrika, Bercovitz & Garcia-Molina 2009) and mean squared difference (MSD) (Cacheda et al. 2011) are two other widely used measures. Compared to PCC-based weighted PCC (WPCC) and sigmoid function-based PCC (SPCC), Jaccard and MSD similarity remove the PCC computation and use a number of common rated items. The basic idea in Jaccard, defined in Equation (2-1), is that two users potentially have similar preferences when they have many co-rated items even though they have rated them differently. MSD similarity, defined in Equation (2-2), only captures the rating difference between two users.

$$Sim_{u,v}^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \tag{2-1}$$

$$Sim_{u,v}^{MSD} = 1 - \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - r_{v,i})^2}{|I_u \cap I_v|} \tag{2-2}$$

where $I_u \cap I_v$ is the set of co-rated items by both users *u* and *v*.

The widely used Pearson correlation-based similarity (PCC) in (Resnick et al. 1994b) and cosine-based similarity (Sarwar, Karypis, Konstan & Reidl 2001) are shown below. The Pearson correlation-based similarity between two users *u* and *v* is calculated by

$$Sim_{u,v}^{PCC} = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r_u}) \times (r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r_u})^2} \times \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r_v})^2}}, \quad (2\text{-}3)$$

where $I_u \cap I_v$ is the set of co-rated items by both users *u* and *v*, $r_{u,i}$ and $r_{v,i}$ represent the ratings of users *u* and *v* on item *i* respectively, and $\bar{r_u}$ and $\bar{r_v}$ represent the average ratings of users *u* and *v* on all of the items in $I_u \cap I_v$. The cosine-based similarity between two items *i* and *j* is calculated by:

$$Sim_{i,j}^{cos} = \frac{\sum_{u \in U_i \cap U_j} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{u \in U_i \cap U_j} r_{u,i}^2} \times \sqrt{\sum_{u \in U_i \cap U_j} r_{u,j}^2}}, \quad (2\text{-}4)$$

where $U_i \cap U_j$ are the users who rated both items *i* and *j*, $r_{u,i}$ and $r_{u,j}$ represent the ratings of user *u* to items *i* and *j* respectively.

There are many similarities which have been proposed to improve the performance to accurately identify the neighbours. Constrained Pearson correlation-based (CPC) similarity between two users *u* and *v* is described in detail as follows:

$$Sim_{u,v}^{CPC} = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - r_{mid}) \times (r_{v,i} - r_{mid})}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - r_{mid})^2} \times \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - r_{mid})^2}}, \quad (2\text{-}5)$$

where $I_u \cap I_v$ is the set of co-rated items by both users *u* and *v*, $r_{u,i}$ and $r_{v,i}$ represent the ratings of users *u* and *v* on item *i* respectively, and r_{mid} represents the mid-point of the rating scale.

The similarity between items can be calculated by adjusted cosine-based similarity measures (Deshpande & Karypis 2004), which are described in detail as follows:

$$Sim_{i,j}^{acos} = \frac{\sum_{u \in U_i \cap U_j} (r_{u,i} - \overline{r_u}) \times (r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{u,i} - \overline{r_u})^2} \times \sqrt{\sum_{u \in U_i \cap U_j} (r_{u,j} - \overline{r_u})^2}}, \quad (2\text{-}6)$$

where $I_u \cap I_v$ are the users who rated both items i and j, $r_{u,i}$ and $r_{u,j}$ represent the ratings of user u to items i and j respectively, and $\overline{r_u}$ represents the average ratings of user u to all of the items.

Breese et al. (Breese, Heckerman & Kadie 1998a) proposed that items with similar ratings should have less impact on determining user similarity than items with different ratings. They suggested using inverse user frequency as the weights of items. Intuitively, if two users rate more items in common, the similarity between them will be more trustworthy. The number of items that have been co-rated by users are taken into account in some improved similarities. WPCC is proposed in (Herlocker et al. 1999) and similarity decreases when the co-rated item number is smaller than a predefined threshold. WPCC is defined as

$$Sim_{u,v}^{WPCC} = \begin{cases} \frac{|I_u \cap I_v|}{T} Sim_{u,v}^{PCC}, & |I_u \cap I_v| < T \\ Sim_{u,v}^{PCC}, & otherwise \end{cases} \quad (2\text{-}7)$$

where I_u and I_v are items that have been rated by u and v respectively, $|I_u \cap I_v|$ represent the number of co-rated items and T is the threshold, which is set to 50 in their work. A similar factor for evaluating the degree of trustworthiness of the similarity is proposed in (Jamali & Ester 2009) which instead uses the ratio in WPCC of a sigmoid function. This approach can weaken the similarity of small common items among users and the equation for this similarity is defined as

$$Sim_{u,v}^{SPCC} = \frac{1}{1 + exp(-\frac{|I_u \cap I_v|}{2})} Sim_{u,v}^{PCC} \quad (2\text{-}8)$$

Lu et al. (Lu et al. 2013a) proposed a similarity measure by incorporating fuzzy set theory to allocate different weighting to rating differences. The absolute ratings are converted to semantic terms and the distance between the vectors is then transformed to the distance between terms. This fuzzy similarity (FS) is defined as

$$Sim_{u,v}^{FS} = \frac{\sum_{i \in I_u \cap I_v} \int_0^1 \frac{1}{2} [(A_{u,i} - \overline{A_u^-}) \times (A_{v,i} - \overline{A_u^-}) + (A_{u,i} - \overline{A_u^+}) \times (A_{v,i} - \overline{A_u^+})] d\lambda}{\sqrt{\sum_{i \in I_u \cap I_v} (\int_0^1 [(A_{u,i} - \overline{A_u^-}) + (A_{u,i} - \overline{A_u^+})])^2} \sqrt{\sum_{i \in I_u \cap I_v} (\int_0^1 [(A_{v,i} - \overline{A_v^-}) + (A_{v,i} - \overline{A_v^+})])^2}} \quad (2\text{-}9)$$

When calculating the similarity between items using the above measures, only users who have rated both items are considered. This can have an impact when items which have received a very small number of ratings express a high level of similarity with other items (Shambour & Lu 2011a). To improve similarity accuracy, the difference between the number of common users who rated both items and the total number of users who rated each item individually was considered, and an enhanced item-based CF approach was presented by combining the adjusted cosine approach with the Jaccard metric as a weighting scheme (Shambour & Lu 2011a). To compute the similarity between users, the Jaccard metric was used as a weighting scheme with the CPC to obtain a weighted CPC measure (Shambour & Lu 2011c). To deal with the disadvantage of the single-rating based approach, multi-criteria collaborative filtering was developed (Shambour & Lu 2010, 2011b).

(3) *Knowledge-based recommendation techniques*

Knowledge-based recommenders will try to characterize users and items based on more domain specific information. This information is usually provided by domain experts or inferred from different available attributes. The user can be asked explicitly to complete his profile by providing the domain information needed. An example of a domain where this is the preferred approach is real estate or cars.

Knowledge-Based (KB) recommendation techniques offer items to users based on knowledge about the users and/or items. Usually, a KB recommender system retains a functional knowledge base that describes how a particular item meets a specific user's requirement, which can be performed based on inferences about the relationship between a user's need and a possible recommendation (Burke 2002).

Case-based reasoning (CBR) technique is a common example of KB recommendation techniques (Smyth 2007). Case-based reasoning systems rely on the idea of using the past problem solving experiences as a primary source to solve the new problem (Aamodt & Plaza 1994). It is represented by a four-step (4Rs) cycle: retrieve, reuse, revise and retain (Aamodt & Plaza 1994). The past problem solutions are stored in a database as cases, each case is typically made up of two parts, the specification part and the solution part. The specification part describes the problem at hand, whereas the solution part describes the solution that is used to solve this problem. A new problem is solved by retrieving a case whose specification is similar to the current problem and then fitting the attained solution to match the current problem. Case-based recommender systems represent items as cases and generate the recommendations by retrieving the most similar cases to the user's query or profile. In these systems, items are described in terms of a well defined set of features (e.g., price, colour, make, etc.) (Smyth 2007). Case-based recommenders borrow heavily from the core concepts of retrieval and similarity in case-based reasoning. The case-based recommender system can be seen as a special type of content based recommender systems. There are two important ways in which case-based recommender systems can be distinguished from other types of content systems: (1) the manner in which products are represented; and (2) the way in which product similarity is assessed (Smyth 2007). Case-based recommender systems rely on more structured representations of item content. In the existing case-based recommender systems, cases are usually represented as fixed predefined feature vectors. There appears to have been no system to deal with hierarchical tree-structured cases yet. The second important distinguishing feature of case-based recommender systems relates to their use of various sophisticated approaches to similarity assessment when it comes to judging which cases to retrieve in response to some user query. Similarity assessment is clearly a key issue for case-based reasoning and case-based recommender systems. The existing similarity measures focus on feature vector represented cases. The similarity measure for tree-structured cases still needs to be researched.

The underlying semantic knowledge associated with users and items has been exploited to generate recommendations in certain types of KB recommender systems called semantic-based recommender systems (Ruiz-Montiel & Aldana-Montes 2009).

The semantic knowledge about items consists of the attributes of the items, the relation between items, and the relation between items and meta-information (Resnik 1995). Taxonomies and ontologies as the major source of semantic information can be taken advantage of in recommender systems, since they provide a means of discovering and classifying new information about the items to recommend, user profiles and even context (Ruiz-Montiel & Aldana-Montes 2009). For example, product taxonomies have been presented in several recommender systems to utilize the relevant semantic information to improve recommendation quality (Albadvi & Shahbazi 2009; Cho & Kim 2004; Hung 2005). In a business environment, product categories are used to evaluate the semantic similarity between businesses (Guo & Lu 2007; Lu et al. 2010; Shambour & Lu 2012). Ontology-based Recommender Systems (Middleton, Roure & Shadbolt 2009) are typical KB recommender systems. An ontology is a conceptualization of a domain into a human-understandable, but machine-readable format consisting of entities, attributes, relationships, and axioms (Guarino & Giaretta 1995). Ontology-based recommender systems classify items using ontological classes, represent user profiles in terms of ontological terms, use ontological inference to improve user profiling, and use ontological knowledge to bootstrap a recommender system and support profile visualization to improve profiling accuracy. The semantic similarity between items can be calculated based on the domain ontology (Al-hassan, Lu & Lu 2011). Ontologies are usually expressed as hierarchical tree structures. However, the items or user profiles are not tree-structured in current research. Cantador (2008) used ontologies to represent the semantics and exploit the semantic relations in recommender systems, and developed a hybrid recommendation approach that merges semantic content-based and collaborative information. The user preferences and item features are modelled as vectors of ontological concepts, but tree structured users or items are not considered. The usage of the semantic information can provide additional explanation about why particular items have or have not been recommended, and provide better recommendation effectiveness than current CF techniques, particularly in cases where little or no rating information is available (Shambour & Lu 2012). In this study, to make accurate recommendations of tree-structured items, the semantic information of tree-structured items or user profiles should be fully considered. A comprehensive semantic similarity measure on tree-structured data will be proposed.

The KB recommender systems have some advantages when compared with CB and CF-based recommender systems. As KB recommender systems exploit deep knowledge about the product/service domain, they are able to support intelligent explanations and product recommendations which are determined by a set of explicitly defined constraints (Felfernig et al. 2006; Felfernig et al. 2008). KB approaches are in the majority of cases applied for recommending complex products and services such as consumer goods, technical equipment, or financial services (Felfernig et al. 2008). A KB recommender system has no cold-start problem as a new user can get recommendations based on a simple knowledge of his/her interests. A KB recommender system generates recommendations by computing the similarities between the existing cases and the user's request, so it does not require the user to rate or purchase many items in order to generate good recommendations. KB recommender systems still have some limitations (Burke 2002; Leung 2009). For instance, a KB recommender system requires extensive effort to acquire and maintain the knowledge, and to retain the information about items and users for making recommendations. It also requires more user-generated content and involvement from an active user in order to make an appropriate recommendation for the user.

(4) *Hybrid recommendation techniques*

To achieve higher performance and overcome the drawbacks of traditional recommendation techniques, a hybrid recommendation technique that combines the best features of two or more recommendation techniques into one hybrid technique has been proposed (Burke 2007). According to Burke (2007), there are seven basic hybridization mechanisms of combinations used in recommender systems to build hybrids:

1) weighted: scores of each of the recommendation approaches are combined numerically to produce a single prediction (Mobasher, Jin & Zhou 2004);

2) mixed: results from different recommendation approaches are presented together, either in a single presentation or combined in separate lists (Smyth & Cotter 2000a);

3) switching: one of the recommendation approaches is selected to make the prediction when certain criteria are met using decision criteria (Billsus & Pazzani 2000);

4) feature combination: a single prediction algorithm is provided with features from different recommendation approaches (Basu, Hirsh & Cohen 1998);

5) feature augmentation: the output from one recommendation approach is fed to another (Melville, Mooney & Nagarajan 2002);

6) cascade: one recommendation approach refines the recommendations produced by another (Burke 2002);

7) meta-level: the entire model produced by one recommendation approach is utilized by another (Pazzani 1999).

The most common practice in the existing hybrid recommendation techniques is to combine the CF recommendation techniques with the other recommendation techniques in an attempt to avoid cold-start, sparseness and/or scalability problems (Adomavicius & Tuzhilin 2005; Bellogin et al. 2013).

2.1.3 RECOMMENDER APPLICATIONS

(1) *E-business*

Many recommender systems have been developed for e-business applications. In general, some systems focus on recommendations generated to individual customers, which are business-to-consumer (B2C) systems, while others aim to provide recommendations about products and services to business users, which are business-to-business (B2B) systems. In this study, e-business recommender systems refer to recommender systems for B2B applications. E-commerce/e-shopping recommender systems refer to recommender systems for B2C applications. In this section, B2B (e-business) recommender systems are reviewed. The e-commerce/e-shopping recommender systems will be reviewed in the next section.

To help catalog administrators in B2B marketplaces maintain up-to-date product databases, an ontology-based product-recommender system was presented (Lee et al.

2006), in which keyword-based, ontology and Bayesian belief network techniques are used to generate recommendations. To help business users select trusted online auction sellers, a recommender system was designed (Wang & Chiu 2008) in which trading relationships are used to calculate the level of recommendations. Recommender systems were also applied in digital ecosystems where agents negotiate services on behalf of a number of small companies (De la Rosa et al. 2011). To build stable digital business ecosystems by means of improved collective intelligence, a model of negotiation-style dynamics from the point of view of computational ecology was introduced in [84], which inspires an ecosystem monitor and a novel negotiation-style recommender. To help private bankers provide suitable investment portfolios to their clients, a multi-investment recommender system PB-ADVISOR was presented (Gonzalez-Carrasco et al. 2012). The system used both semantic technologies and fuzzy logic to improve recommendation quality. The semantic characterization of the investments and their characteristics enable the private banker to recommend a wide spectrum of products with very diverse characteristics. The relations between investments and investors are defined by means of fuzzy rules that represent expert advisor knowledge. The results obtained have shown that the system is able to offer recommendations comparable with those from experts in the field.

Customer relationship management is very important for the telecom industry. To support telecom companies in recommending suitable products and services to their business and individual customers, a telecom recommender system has been developed (Zhang et al. 2013). Zhang et al. (2013) designed and implemented a personalized recommendation approach and a software system called fuzzy-based telecom product recommender system (FTCP-RS). The FTCP-RSs can generate the service plan and package recommendations for a customer and can also give recommendation explanations. To deal with sparsity problems and improve prediction accuracy, particularly in handling customer data uncertainty and fully using business knowledge in the recommendation process, the proposed approach integrates item-based CF (IBCF) and user-based CF (UBCF) with fuzzy set techniques and a KB method (business rules). The implemented system has undergone preliminary testing in a telecom company and has achieved excellent performance.

This study has found that in e-business recommender systems, the KB approaches, such as ontology and semantic techniques, are widely integrated with CF and CB recommendation methods. The main reason for this is that e-businesses have a high need for domain knowledge to assist their recommendations.

(2) *E-government*

Electronic government (e-government) refers to the use of the Internet and other information and communication technologies to support governments in providing improved information and services to citizens and businesses. The rapid growth of e-government has caused information overload, leaving businesses and citizens unable to make effective choices from the range of information to which they are exposed. Increases in this information overload could clearly hamper the effectiveness of e-government services, and difficulties in locating the right information for the right users will increasingly impact on the loyalty of users. Recommender systems can overcome this problem and have been adopted in e-government applications (Guo & Lu 2007; Terán & Meier 2010).

To support citizens in their access to personalized and adapted services supplied by public administration offices, a multi-agent system was presented by De Meo et al. (De Meo, Quattrone & Ursino 2008). The proposed system identifies and suggests the most interesting services for a user by considering both the user's profile and the profile of the device being used. To assist voters to make decisions in the e-election process, a recommender system was proposed (Terán & Meier 2010), which uses fuzzy clustering methods and provides information about candidates close to voters' preferences. To provide personalized exercises to patients with low back pain problems and to offer recommendations for their prevention, a recommender system called TPLUFIB-WEB was presented in (Esteban et al. 2014). The system can be used in any place and at any time, yielding savings in travel and staffing costs. It is very user-friendly, designed for individuals with minimal skills and using fuzzy linguistic modeling to improve the representation of user preferences and facilitate user-system interactions. TPLUFIB-WEB satisfies the Web quality standards proposed by the Health On the Net Foundation (HON), Official College of Physicians

of Barcelona, and Health Quality Agency of the Andalusian Regional Government, endorsing the health information provided and warranting the trust of users.

In G2B services, many items from a business perspective are one-time items, such as events, which typically receive ratings only after they have ended. Traditional CF techniques cannot recommend these kinds of items due to the sparse rating data. To handle this problem, Guo and Lu (2007) proposed a new approach which handles an attribute-considered recommendation issue by integrating the semantic similarity techniques with the traditional item-based CF. A recommender system called Smart Trade Exhibition Finder (STEF), which suggests suitable trade exhibitions to businesses, has been developed. To flexibly reflect the graded/uncertain information in the G2B domain, Cornelis et al. (2005) modeled user and item similarities as fuzzy relations. They also proposed a novel hybrid CF-CB approach whose rationale is concisely summed up as "recommending future items if they are similar to past items that similar users have liked". A hybrid fuzzy logic-based recommendation framework was then developed (Cornelis et al. 2007) to improve the trade exhibition recommender system for e-government.

(3) *E-learning*

E-learning recommender systems have become increasingly popular in educational institutions since the early 2000s based on the development of traditional e-learning systems. This type of recommender system usually aims to assist learners to choose the courses, subjects and learning materials that interest them, as well as their learning activities (such as in-class lecture or online study group discussion). In more than ten years' accumulated study on this topic, many practicable e-learning recommender systems have been developed.

Zaiane (2002) proposed an approach to build a software agent that uses data mining techniques such as association rule mining to construct a model that represents online user behaviours, and used this model to suggest activities or shortcuts. The suggestions generated assist learners to better navigate online material by finding relevant resources more quickly using the recommended shortcuts. A personalized e-learning material recommender system (PLRS) was proposed in the work of Lu (2004a). Once a learning material database or a learning activity database is created

and a learner's registration information is obtained by the system, the PLRSs uses a computational analysis model to identify an individual's learning requirement and then uses matching rules to generate a recommendation of learning materials (or activities) for the learner. Web usage mining is the process of applying data mining techniques to the discovery of behaviour patterns based on Web click-stream data, which provides information to help understand users' preferences. A recommender system that utilizes Web usage mining to recommend the links in an adaptive Web-based educational system was proposed in (Romero et al. 2009). A Web mining tool and a recommendation engine were developed and applied into the Adaptive Hypermedia for All (AHA) system to help the instructor to carry out the whole Web mining process. In the personalized courseware recommender system (PCRS) continuously developed in (Chen, Duh & Liu 2004) and (Chen & Duh 2008), a fuzzy item response theory (FIRT) is proposed to initially collect a learner's preferences, following which the learner provides a fuzzy response as a percentage of their understanding of the learned courseware. The system framework of (Chen, Duh & Liu 2004) contains both online and off-line modules. The online modules provide the evaluations of a learner's preference and the matching process between learners and courseware. The off-line module provides a courseware management agent to assess the level of difficulty of each course, in support of the matching process. To recommend learning goals and generate learning experiences for learners, a recommendation methodology was defined and a recommender system prototype component developed for integration into a commercial adaptive e-learning system called IWT (Capuano et al. 2014). The recommendation methodology applies a hybrid recommendation approach which consists of three steps: concept mapping, concept utility estimation and upper level learning goals (ULLG) utility estimation. Once the utility of each ULLG is estimated for a learner, the ULLGs with the greater utility can be suggested to the learner.

(4) *E-tourism*

Internet and mobile devices provide tourists with great opportunities to access tourism information, but the dramatic increase in the number of available tourism choices make it difficult for tourists to choose which option they prefer. E-tourism recommender systems are designed to provide suggestions for tourists. Some systems

focus on attractions and destinations, while others offer tour plans that include transportation, restaurants and accommodation.

There are several restaurant recommender systems. Burke et al. (1996), for example, proposed a recommender system called Entrée to recommend restaurants based on KB approaches. The knowledge was collected from users and retrieved by Entrée to find similar choices by refining such search criteria as price and taste. Burke (2002) improved Entrée by incorporating CF into KB, which meant that apart from restaurant features, the assessments of users also became criteria.

As was mentioned before, mobile devices provide opportunities for the development of mobile-based recommender systems. Hung-Wen and Von-Wun (2004) designed a system to suggest restaurants for tourists in Taipei. This system is a CB recommender system which allows users to obtain real time suggestions from a mobile application. CATIS (Pashtan et al. 2003) is a context-aware recommender system which recommends tourist accommodation, restaurants and attractions. The context information (e.g., location and wireless device features) is dynamically collected by a context manager. A collection of Web services provided by an application server is used to gather user context information. The recommendations are generated by combining the user query and the user context information from the application server.

Another restaurant recommender system, REJA (REstaurants of JAén) hybridizes CF and KB approaches (Martinez, Rodriguez & Espinilla 2009). The recommendations can be provided by the CF approach when the system is able to construct a user profile according to the user's ratings. When the system has insufficient information about a user, a case-based reasoning approach is executed.

A personalized sightseeing planning system (PSiS), which is used to aid tourists to find a personalized tour plan in the city of Oporto, Portugal, was developed in (Lucas et al. 2013). To avoid the shortcomings of current recommender systems, such as scalability, sparsity, first-rater and gray sheep problems, a hybrid recommendation approach was proposed. The proposed hybrid recommendation approach employed CF and CB approaches, combined a clustering technique and an associative classification algorithm, and also used fuzzy logic to enhance the quality of

recommendations. SigTur/E-Destination (Moreno et al. 2013) was designed to provide personalized recommendations of tourism activities in the region of Tarragona. To make proper recommendations, the SigTur/E-Destination integrated several types of information and recommendation techniques. The information used in the recommender includes demographic data, details that define the context of the travel, geographical aspects, information provided explicitly by the user and implicit feedback deduced from the interaction of the user with the system. The SigTur/E-Destination employs many recommendation techniques, such as the use of stereotypes (standard tourist segments), CB and CF techniques, and artificial intelligence tools including automatic clustering algorithms, ontology management, and the definition of new similarity measures between users, based on complex aggregation operators.

SMARTMUSEUM, a mobile-based recommender system, presents users with recommendations for sites and objects on those sites on their mobile phones (Ruotsalo et al. 2013). In this system, an ontology-based personalization, annotation, and information filtering framework was developed. The contextual data, whether input by users or captured by the built-in sensors of mobile devices, are mapped to the concepts defined in the ontologies. The filtering framework introduced ontology-based query expansion for triples, feature balancing, and result clustering, which led to significant improvements in the accuracy of information filtering. iTravel, another mobile-based recommender system, was developed to provide tourists with on-tour attraction recommendation (Yang & Hwang 2013). In this system, the techniques of CF and mobile peer-to-peer communication were combined. To utilize the information of other tourists with similar interests in mobile tourism, three data exchange methods for users to exchange their ratings of attractions they had visited were proposed.

Moleskiing (Avesani, Massa & Tiella 2005) is a website for assisting community users to plan their skiing activities. This recommender system allows users to share their opinions and experiences of particular sites as well as the trust degrees for specific users. People who are going skiing can exploit the snow condition information to personalize a safe route. DIETORECS (Fesenmaier et al. 2003) is a case-based reasoning (CBR) recommender system which creates a complete plan for tourists. Users can utilize the system in different ways according to their experience.

The experienced user can make detailed preferences for attractions, while the less-experienced user can simply make a list of attractions of interest. An on-board recommender system for drivers called MASTROCARONTE (Console et al. 2003) utilizes KB approaches to recommend attractions, restaurants, and hotels. It utilizes context information to suggest appropriate items to drivers such as restaurants at meal times or nearby fuel stations when fuel is exhausted.

The SPETA system (García-Crespo et al. 2009) uses the knowledge of a user's current location, preferences, and the history of past locations to recommend the services that tourists expect from a human tour guide. It combines social networks, Semantic Web, and context-awareness in pervasive systems to improve tourists' experiences. It offers a personalized guide, and solves the problem of tourism service disintegration in respect of searching, finding and presenting personalized services by means of semantic, geo-location, and social technologies. Traveller (Schiaffino & Amandi 2009) is proposed to provide package holidays and tours. It builds an agent which combines CF with CB and demographic recommendation approaches.

(5) *E-resource*

The e-resources mentioned here refer to content such as videos, music and documents which is uploaded by users. Some recommender system users share sources to the Internet so that other users can access the resources that interest them. This section focuses on several typical applications of recommender systems in resource services: Tag, TV program, webpage, document, video and movie recommendation.

Tags are arbitrary words specified by users to label and manage the resources that are uploaded to the Internet. Users want tags to be personalized and convenient to enable the easy sharing of resources, but it is often difficult for users to select appropriate tags from the wide range of possibilities. Tag recommender systems thus become increasingly important for making tag selection easy and personalized.

Folksonomies, which contain tag recommender systems, are Web-based systems that allow users to upload their resources (e.g., documents, pictures), and to label them with tags. Folksonomies can be seen as three-part systems comprised of

resources, users and tags. Zheng and Li (2011) implemented a folksonomy recommender system based on CF. They exploited the tag and time effects in the recommendation procedure. Instead of utilizing the rating matrix in traditional CF, they built matrixes based on tag and time relations. Three strategies, tag-weight, time-weight and mixed, are used to calculate the similarities based on corresponding matrixes. The recommendations are predicted by neighbours who are identified based on new similarities.

Another tag recommendation approach, FolkRank, was proposed in (Hotho et al. 2006; Jäschke et al. 2007), in which the tags are recommended by calculating the distance from the uploaded resource. Gemmell et al. (2009) suggested that CF, especially item-based CF, could be incorporated into the traditional graph-based approach to augment the performance in FolkRank. The item-based CF identifies the relevant resources by tags that are common to the user. The final recommendations are predicted by the linear combination of graph-based and CF approaches.

TV programs can be seen as a special type of resource released by broadcasters. A large increase in the number of TV channels and programs has been seen in recent years due to the growth of interactive and two-way TV. Even with an electronic program guide, it is difficult for viewers to find interesting programs from the hundreds or thousands of options. A program recommender system (PRS) is required to help viewers to choose programs that interest them.

Content information for TV programs can be described by features (e.g., genre, actor), so the CB method is commonly used; where the TV mode allows the user to give feedback (e.g., ratings), the CF method is well applied. In PTV (http://www.ptv.ie), proposed by Smyth and Cotter (2000b), viewers rate programs through a Web system to specify their preferences. After the system has collected explicit data from viewers, both CF and CB methods are used to find similar programs according to the ratings given by viewers and the program information.

With the development of smart TV sets, users are allowed to give ratings on TV, which has resulted in TV-based recommender systems. TiVo (Ali & Stam 2004) allows viewers to rate programs using the remote control, and CF is utilized to suggest suitable programs. The implicit feedback, such as whether the program is

being recorded, is taken into account in addition to the explicit ratings from viewers. Requiring users to respond to programs is tedious and raises privacy issues, so some systems try to collect the required data in the background. User preferences are built using program attributes such as program title, genre, subgenre, channel, and actors in (Bjelica 2010). TV programs are recommended by comparing the features of the past viewing set with current programs. Hyeong-Joon and Kwang-Seok (2011) proposed a novel similarity method that applies raw moment-based similarity (RMS) which is then used in memory-based CF approaches to address such shortcomings as cold start and high calculating cost. The application queveo.tv, developed by Barragáns-Martínez et al. (2010), combines the CB approach and item-based CF approach to address the problems of gray sheep, cold start and first rating. The dimensionality reduction technique, singular value decomposition, is incorporated to solve sparsity and scalability problems.

Suggesting webpages, documents and news is a traditional area for recommender systems because such resources grow rapidly. In most instances, textual content such as news, emails, documents and webpages is described as a list of keywords, which can be extracted from historic data, URLs and search engines, and many recommender systems are designed on the basis of analysing keywords. Probabilistic models such as the information retrieval technique are common in this area. Contextual resources are transformed to a vector, with each element representing a keyword which takes frequency and location (title or plain text) into account. The recommendations are generated by retrieving resources that are similar to the user patterns. For example, AMALTHAEA (Moukas 1997; Moukas & Maes 1998) draws keywords from URLs by examining the hotlist and browsing history, and investigates the interest shown by users by information retrieval (IR). CB approaches are adopted in ifWeb to measure the similarity between pages (Asnicar & Tasso 1997). CF is feasible if systems can collect information about whether users evaluate the content by ratings. For instance, News Dude (Billsus & Pazzani 2000), a news recommender system, uses CF to model users' short term interests. Other examples of systems in which CF is used are the joke recommender system Eigentaste (Goldberg et al. 2001) and the Usenet news recommender system GroupLens (Konstan et al. 1997; Resnick et al. 1994c). In addition to News Dude (Billsus & Pazzani 2000), which builds long-

term preferences through Bayesian methods, other model-based systems have also been proposed, such as Foxtrot (Middleton, Shadbolt & Roure 2004b), which uses k-nearest classification. Graph-based clustering was adopted in WinPUM (Jalali et al. 2010), in which the authors transformed websites into graphs and classified user navigation patterns according to users' session information. Recently, Nguyen et al. (2013) suggested that by integrating ontology and semantic knowledge, which are used to analyze session data, the system could navigate more accurately. In Eigentaste (Goldberg et al. 2001), principal component analysis is adopted to deduce the dimension of a keywords matrix to accelerate the process of user clustering and the computation of recommendations.

Apart from the keywords taken from the textual content, implicit and explicit feedback from users is also taken into account. Lifestyle Finder (Krulwich 1997) uses demographic information to model the user and provide webpage recommendations. ACR News Vectors (Mobasher, Cooley & Srivastava 2000) are built based on implicit feedback and viewing frequency for webpages. The clustering model is then trained and pages are recommended by a CB approach on related clusters. In ArgueNet (Chesnevar & Maguitman 2004), another webpage recommender system, users are allowed to address such criteria as the trustworthiness of websites. The user preferences are modeled by keywords along with these criteria to generate personalized recommendations.

With the extensive usage of mobile devices in recent years, a particularly rapid growth in movie, video and music resources has taken place. However, users experience frustration when searching for content that interests them on mobile devices. To solve the problem, many movie recommender systems, such as PocketLens (Miller, Konstan & Riedl 2004) and CinemaScreen (Salter & Antonopoulos 2006), and music recommender systems such as Flycasting (Hauver & French 2001), Smart Radio (Hayes & Cunningham 2001), RACOFI (Lemire & Boley 2003) and Foafing the Music (Celma & Serra 2008), have been developed. Because most of these systems allow users to rate resources, CF recommendation approaches are commonly used in these recommender systems. In some systems, such as Flycasting (Hauver & French 2001), users cannot rate music directly, so this system first translates historical listening information into ratings and then carries out CF. To

address the cold start and sparsity problems of CF approaches, CB approaches are incorporated in some systems. For example, Melville et al. (2002) utilized CB to overcome the sparsity and first-rate problem. CinemaScreen (Salter & Antonopoulos 2006) also used a CB method to solve cold start problems in movie recommendation. One feature of movie and music recommender systems is that it is not easy to obtain the content and navigation history from multimedia resources. These resources contain such features as artist and genre, and how to extract the underlying correlations is an important issue in this area. Model-based approaches like semantic analysis and social network are also integrated into CF. RACOFI (Lemire & Boley 2003) utilizes Semantic Web techniques. Foafing the Music (Celma & Serra 2008) maintains friend of friend profiles which work in a similar way to social networks. CoFoSIM (Lee, Cho & Kim 2010), a mobile music recommender system, utilizes multi-criteria decision-making (MCDM) techniques to analyze the implicit feedback and partial listening records, and aggregates them into a composed preference. An interesting aspect of music recommender systems is that some systems use implicit feedback to augment or replace the explicit ratings from users. For example, both CoFoSIM (Lee, Cho & Kim 2010) and Smart Radio (Hayes & Cunningham 2001) use the listening history to infer their user ratings.

2.2 GROUP RECOMMENDER SYSTEMS

2.2.1 CONCEPTS

GRSs provide recommendations to groups taking into account the preferences of the members of the groups and GRSs have been developed for a variety of domains, such as leaning resources recommendation (Dwivedi & Bharadwaj 2015). Many GRSs and related methods are reviewed in (Kompan & Bielikova 2014; Pessemier, Dooms & Martens 2013) . All the GRSs can be categorised into two classes in terms of application scenarios: active and passive groups. Some scenarios allow users to actively announce that they are in a specific group, while in others, users are passively allocated to a group. For example, members in a reading group actively form the group and then obtain book recommendations for all members. On the other hand, when people passively become a group as a result of attending a music show,

recommendations for other music shows cannot be determined simply on the basis of that single attendance.

A group recommender system is defined as R, and it provides generalized items, such as books or music, for users. The system then determines all the members in the group and makes recommendations for them as a single entity after the group has been formed. All the items are denoted in R as I and all the users as U. A group G, $G \in U$, is a collection of the users gathered actively (e.g. people who choose the same reading group) or passively (e.g. people who attend a show) while their preferences or profiles are collected by R. The group recommender system can be represented as three tuples $\langle R, G, S \rangle$ that select a number of items S of which $S \in I$ matches as many preferences of G as possible.

2.2.2 TECHNIQUES

Many GRSs are reviewed in (Lu et al. 2015) and, generally, most existing recommendation approaches in GRSs can be classified into two categories, as illustrated in Figure 2-1:

- aggregating individual profiles, in which the profile of a pseudo user is modeled by aggregating individual members' preferences to represent the preferences of the whole group, and the pseudo user's profile is then used to generate group recommendations;
- aggregating individual recommendations, in which individual members' recommendations are generated independently and group recommendations are produced by aggregating individual recommendations (Baltrunas, Makcinskas & Ricci 2010).

These two categories of approach are compared in (Berkovsky & Freyne 2010), and it is suggested that the former approach is slightly better than the latter. The challenge of the pseudo user approach is that group members may not always share the same preferences, and preference conflicts may occur when a group profile is modeled to represent the preference of all the group members. In general, many strategies are required to alleviate and minimize the dissatisfaction caused by preference conflicts.

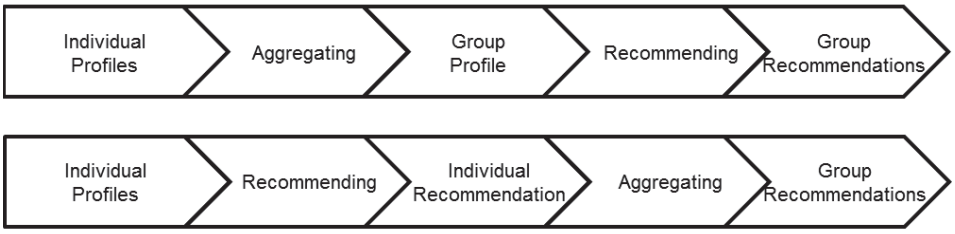


Figure 2-1. The two basic approaches to making group recommendations. The top approach aggregates individual preferences and the bottom approach aggregates individual recommendations.

One of the most important procedures is the aggregating method. Either aggregating profiles or aggregating recommendations need an aggregating method to combine individual UGCs. Many strategies have been employed as aggregating methods, most of which are summarized in (Masthoff 2011). If all the individual profiles/recommendations are treated as multi-dimensional data over $\mathbb{R}$ with dimensionality I, these strategies are methods of how to select a point $\vec{r}$ in $\mathbb{R}$ to represent the entire group. The strategies are summarised and they then can be categorised into four types, to emphasise different aspects of observed values:

- Consensus-based: on each dimension, to select the average or weighted average of all the observed values. This value can be different from all the observed values.
    - Fairness
    - Average
- Majority-based: the value on each dimension is selected from the most common among the observed values.
    - Plurality Voting
- Borderline: the extreme value is selected on each dimension
    - Least Misery
    - Most Pleasure
- Dictatorship: only the value from a specific member is selected on every dimension.
    - Most Respected Person

Of these four categories, majority-based strategies are often used to aggregate individual recommendations, while the other three categories are used to aggregate individual preferences to build a group profile. As previously mentioned, two aspects are of key concern in modeling the group profile. The first is the common interest of the group and the second is the disappointment caused by preference conflicts; these two aspects drive the basic design principles for generating group recommendations, maximizing satisfaction and minimizing disappointment. Consensus-based, Majority-based and Dictatorship strategies are widely used to maximize satisfaction; for example, the average strategy in (Gorla et al. 2013), and the variation on average strategy used in (Chen & Cheng 2010) to aggregate rankings. Borderline strategies, such as the least misery strategy in (O'Connor et al. 2001), are used to minimize disappointment. A combined strategy called "average without misery" has been proposed (McCarthy & Anagnost 1998; Quijano-Sánchez et al. 2012b) which balances the two principles by taking both aspects into consideration. However, this strategy needs to determine a threshold that will explicitly exclude members who do not meet requirements. Let $Profile_G$ is the corresponding profile for a group G. Strategies can be summarized as

$$Profile_G = \sum_{u \in G} \vec{\omega}_u \, Profile_u \tag{2-10}$$

where $\vec{\omega}_u$ is the weight vector for u and different $\vec{\omega}_u$ leading to different strategies. When $\vec{\omega}_u = {}^{1}/_{|G|}$, Equation (2-10) becomes an average strategy. When only one member's weight vector's elements are equal to 1 and other members' are zero vectors, Equation (2-10) becomes a dictatorship strategy.

Many improvements have been proposed by providing complex methods to calculate the weights and the corresponding group profile. Most of these improvements need extra UGC, such as social relations tags and expert opinions.

Social relationships and personalities are used describe the characteristic and relationships among the group members in (Ye, Liu & Lee 2012) to identify the most representativeness measures. The basic idea behind this is that users tend to purchase those products that are preferred by the user's social contacts. The group recommendation methods were proposed by (Gartrell et al. 2010a; Quijano-Sánchez

et al. 2013) which combines both the social and content interests of the group members. Moreover, some enhanced methods are proposed to improve building the group profile. The genetic algorithm is used in (Chen, Cheng & Chuang 2008) to obtain the personalities of members and interactions among them. The ratings of individual members and subgroups of the active group are analysed to calculate the weights and individual rating for the target item. The unknown group ratings are computed using weighted sum of the individual ratings. The genetic algorithm is also used in (Gartrell et al. 2010b). Gartrell et al. define the measures of social tie, expertise and dissimilarity among group members. Accordingly, a generic framework is designed to automatically analyse group characteristics and generates the corresponding group consensus function to predict group preferences.

Group profiles with tags were built by (Amer-Yahia et al. 2009; Liu et al. 2009; Roy et al. 2010). These improvements suffer from the problem that they do not work when the required additional UGC are unavailable, and the problem may be worse when a random group is involved. For example, it is difficult to identify the social relationships between a group of strangers on an airplane, and it is not feasible for passengers to tag their preferences in advance.

Other extra information, such as domain knowledge, is also applied in modeling group profiles. User prototypes for tourism activities were predefined by (Ardissono et al. 2005b; Ardissono et al. 2003) to model the pseudo user profile for a random group, which was demonstrably useful; however, it was also necessary to introduce domain knowledge into the system. A more complex example including domain knowledge is presented in (Vildjiounaite et al. 2009), in which three support vector machines are trained for the different preference aspects of TV viewers. The overall viewing preference is constructed by combining three aspects with case-based reasoning. Note that the dictatorship strategies mentioned all depend on incorporating extra information. In (Berkovsky & Freyne 2010), for example, the family-log model weights users by their number of ratings. All such improvements need additional information to incorporate with ratings. The knowledge-based methods are also used in GRSs. (Quijano-Sánchez, Díaz-Agudo & Recio-García 2014; Quijano-Sánchez, Recio-García & Díaz-Agudo 2010) use additional knowledge from Thomas–Kilmann Conflict Mode Instrument (TKI) test identifying users' ability to handle the conflicts

in decision-making processes. A personality value is then computed with the test results and interactions among members are described accordingly. The group ratings are predicted taking into consideration the relative relationships among members. Cases are used to described the tourism groups in (McCarthy, McGinty, et al. 2006). The case-based preference similarity and compatibility are computed using average strategy and the CBR method is then used to compute the score for the target item by weighted sum. (Quijano-Sánchez et al. 2012a) also use case-based method to analyse the personality among the group members and the raw ratings are transformed into delegation-based rating considering personality. The group profiles are modeled using most pleasure strategy to aggregate the DBR ratings into a single group rating. The scores for target items of an active group are the weighted average from scores of similar cases.

Context information is also used in (Pérez, Cabrerizo & Herrera-Viedma 2010) combining fuzzy preference relations; 2) orderings; 3) utility functions; and 4) multiplicative preference relations for ambient intelligence. Because this prototype incorporates both selection and consensus processes, it allows us to model group decision-making situations. Introduce experts to obtain additional knowledge to build the group profile. (Chen et al. 2014) describe the design process Empatheticons, Emotion Awareness Tools and visualize group members' emotions in GroupFun, a group music recommender. Recommend music. They can specify music preference by rating the songs provided by GroupFun. When multiple participants have annotated the songs with their emotions, GroupFun shows these dynamic emoticons in a continuous and animated way while the music is playing.

Instead of executing the recommendation methods automatically, another way to improve the effectiveness of modeling a group profile is to provide interactive functions for group members to explicitly specify their preferences (Garcia & Sebastia 2014; McCarthy, Salamó, et al. 2006a), but these functions are not always available when a group is formed randomly. CATS system uses DiamondTouch tabletop device critiquing technique to build the group profile.

2.2.3 RECOMMENDER APPLICATIONS

There are some scenarios (e.g., recommending a TV program to a group of people) in which users are unable to specify their preference explicitly, and some scenarios (e.g., taking a tour with others) in which users need to negotiate online to engage in an activity together. In these cases, people need online decision support for a whole group. Traditional recommender systems only make suggestions for individual users, thus group recommender systems (GRS) are proposed to combine and balance the individual expectations of group members to produce satisfying recommendations to the group. There are two main types of GRS: one called an off-line GRSs for a group which has already been formed (a family, for example), and one called an on-line GRSs for a group which needs to be formed by the system. GRSs have been applied in practice for both types. The four main domains are summarized below.

Many GRSs are designed to recommend textual items such as books, documents and webpages. Probabilistic-based models such as information retrieval, Bayes and user-item matrix are utilized to describe items and present the relationship between those items and users. I-SPY is a search engine that recommends resources to communities of likeminded users (Freyne et al. 2004; Smyth & Balfe 2006; Smyth et al. 2004). The system maintains a hit matrix, connecting the users and queries for a community, and updates the matrix when a user visits the search results. The relative resource is re-ranked after the matrix is updated, and other users in the community can obtain more accurate results. Besides utility, other requirements are taken into account in GRS, such as satisfaction and fairness.

GRec_OC is a book recommender system developed by Kim et al. (2010) to validate their approach for an online community. Their intention is to satisfy the small number of group members who are likely to be ignored although the majority of the community is satisfied. They adopt two levels of filtering mechanism, CB and CF-based methods, to generate candidate books by CF according to the group preference, and eliminate candidate books if any member's compatibility score is below the threshold.

To augment browsing activities, Sharon et al. (2003) proposed a mediator, Context Aware Proxy-based System (CAPS), to collect the frequency and dwell time for pages, which works as a proxy for browsers without requiring the user to input data actively. The repositories for a group of collaborative members and ranks for pages are built to augment other members' browsing and searching activity.

Some music recommender systems automatically broadcast music to users without user selection (Baccigalupo & Plaza 2007; McCarthy & Anagnost 1998; Popescu & Pu 2012); these are referred to as radio-based recommender systems. For example, MusicFX (McCarthy & Anagnost 1998) is a GRSs that recommends music to all the people in a gym. Members' preferences are stored in the system, and the recommended music is generated according to personal preferences and played for members without further selection. Flytrap (Crossen, Budzik & Hammond 2002) is another GRSs that selects music to be played in a public room. Instead of collecting personal preferences by asking, Flytrap automatically collects meta information about the music that the user is listening to. Genres and artists are used to build a network with edges between network nodes representing the similarity between them. The playlist is ultimately determined by a voting mechanism, with some constraints predefined by the system. Like a threshold on rating to measure a particular kind of music, the similarity combined related threshold can also be used to measure preference. In (Chao & Forrest 2003), adaptive radio is proposed to broadcast songs to people who share the radio. The system adopts a simple mechanism whereby rejected songs, or other songs which are similar to the rejected ones, will not be played, whereas recommendations will be broadcasted and played automatically.

PolyLens (O'Connor et al. 2002), which supports group creation and management, is extended from MovieLens and is designed for movie recommendation for a relative small group. It considers the nature of the group, the group's formation and evolution, privacy, group recommendation generation and interfaces. PolyLens merges the recommendations generated for individual users by nearest neighbour methods and sorts the merged list according to the lowest ratings ascribed to the movie; it can therefore provide more information to both individuals and the group. jMusicGroupRecommender and jMoviesGroupRecommender proposed in (Christensen & Schiaffino 2011) recommend entertainment content

including music and movies. It combines merging of individual recommendation and multiplicative aggregation. Merging is used to filter candidate items, taking only those with the highest rating for each individual member. Then, an aggregate rating is obtained by multiplying individual ratings for those short listed candidate items. Colombo-Mendoza et al. in (Colombo-Mendoza et al. 2015) implement an ontology-based context system RecomMetz on mobile devices to recommend movie on theatres. The context information includes theatre, movie, show time and crowd factors.

TV program recommendation (TPR) has gained more attention and is important not only for individual personalization but also for group adaptation, such as when family members watch programs together. A challenge in making TPR different from other Web-based group recommender systems is that it is difficult to identify the members in a group because the group could be dynamic, with members able to join and leave the group at any time. In the Family Interactive TV System (FIT) reported in (Goren-Bar & Glinansky 2004b), viewers are modeled according to their stereotypes and the probability of preferred watching time for each type. The programs are recommended according to the combined probability. Yu et al. proposes a TV program recommendation strategy in (Yu et al. 2006) for multiple viewers based on user profile merging; they introduce three alternative strategies to achieve program recommendation for multiple television viewers, the preferences are described by vector of key-value pair which are obtained by a rule-based method. The key is, for example, genre, actor, and keyword about TV programs and the value is weight of the keyword. The group profile is merged using weighted average strategy. Model-based techniques are also utilized in TPR. Vildjiounaite et al. (Vildjiounaite et al. 2009) built a model for a family by supporting vector machine and made suggestions using a KB approach. Sotelo et al. presents an approach in (Sotelo et al. 2009) to content recommendation for families. The relevance between two members in a group is evaluated using semantic ontologies and the group is identified as homogeneous or heterogeneous group. For a homogeneous group, a virtual user is modeled using average without misery strategy and for a heterogeneous group the group is extended to include new members who are similar to the original one; the virtual users are modeled using Average without Misery strategy and the final prediction is an average from all the neighbor groups.

E-tourism, such as (McCarthy, Salamó, et al. 2006a; Noguera et al. 2012) is also an important domain that GRSs are widely developed to recommend attractions, accommodation and restaurants for groups. Pocket Restaurant Finder (McCarthy 2002a) is a GRSs that locates a restaurant for a group of people. Every member presents their opinions, stipulating such conditions as distance, price and so on. This GRS builds a group preference model and evaluates each restaurant according to the model. The final recommendations are produced as a list. CB approaches are mainly used to produce personal preferences. The Collaborative Advisory Travel System (CATS) was proposed in (McCarthy, Salamó, et al. 2006b; Salam, Mccarthy & Smyth 2012) to recommend a plan for ski holidays for a group of friends (see Figure 2-2). Users present their explicit critiques for the features of the plan and negotiate to reach agreement on those critiques, called the group user model. The system produces recommendations according to this model. INTRIGUE (Ardissono et al. 2005a; Ardissono et al. 2003) is a tourism GRS that is also based on aggregating recommendation approaches (see Figure 2-3). The group is first divided into several subgroups according to the demographic information (e.g., number of children). Recommendations are generated for each subgroup and the final result is built by taking into consideration the influence of subgroups (e.g., people with disabilities). Personalized Electronic Tourist Guides (PETs) (Garcia et al. 2009) provides a solution for personalized route generation based on the profile and constraints of a group of tourists. The solution is integrated by three aspects: demographic information, route information and specified preference. With each aspect, a group preference model is constructed and recommendations are added to a candidate list. e-Tourism (Garcia, Sebastia & Onaindia 2011) generates recommendations about personalized tourist tours in the city of Valencia (Spain) for a single person or a group of tourists. In the e-Tourism system, group preferences are elicited from individual preferences through the application of intersection and aggregation mechanisms. Instead of making recommendations that directly match the group preferences, e-Tourism also applies a hybrid recommendation technique by combining demographic, content-based recommendation and likes-based filtering, which ensures that e-Tourism is always able to offer a recommendation, even when the user profile contains very little information. In (Lorenzi et al. 2008), a multi-agent recommender system for tourism was developed based on the cooperation of two types of agent:

user agent and recommender agent. The user agent stores the user preference information and the recommender agent stores the travel information locally. The recommendations are produced by the exchange of information between these two types of agent. For users who want to plan a vacation together but find it difficult to negotiate face to face, the tourism GRS also takes negotiation support into consideration. Jameson et al. (Jameson 2004) proposed a system called Travel Decision Forum that helps groups to plan a vacation using an asynchronous communication mechanism. Users in a group can view and even copy other members' preferences. After the users have reached agreement, the system aggregates individual preferences with the median strategy. McCarthy et al. (2006b) proposed a CB simultaneous collaborative group critiquing recommender system to produce ski holiday suggestions for groups of up to four members. The features predefined by the system for both resorts and accommodation are critiqued by members. All the UGC is aggregated and the recommendations most likely to satisfy the group as a whole are generated.

Figure 2-2. CATS tourism system critiquing interface (McCarthy, Salamó, et al. 2006b)

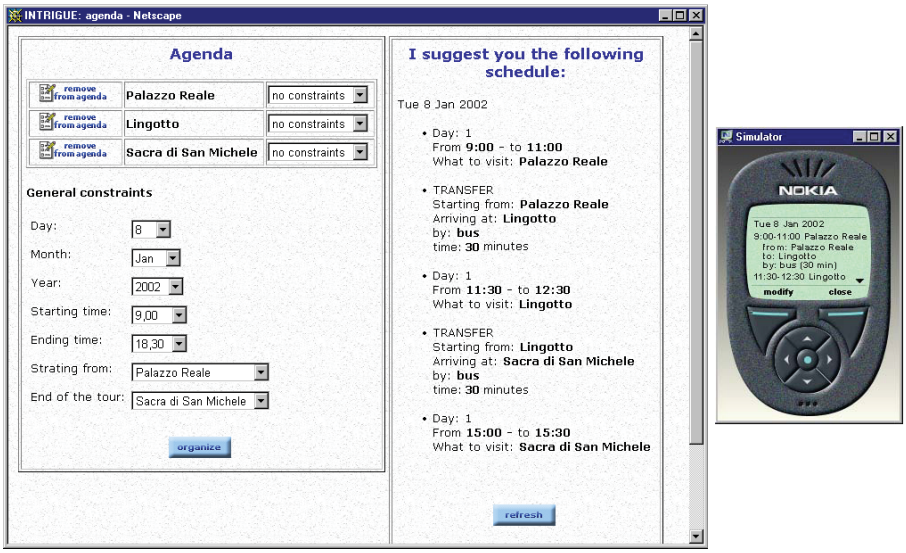


Figure 2-3. INTRIGUE system preference specification interface (Ardissono et al. 2003)

2.3 MATRIX FACTORIZATION

2.3.1 MATRIX FACTORIZATION IN RECOMMENDER SYSTEMS

Matrix factorization (MF) is the most common technique used in model-based CF recommendation methods. The key concept behind MF is the fact that the users' preferences are often based on a combination of features. Taking movies as an example, a user may prefer a specific genre such as action movies or epic movies rather than the designated movie. MF enables systems to analyze and extract those feature combinations, i.e. latent features, from a numerical rating matrix. MF decomposes the matrix into two matrices and the basic expression of MF is

$$R = WH, \tag{2-11}$$

where R is the original matrix, and W and H are low rank matrices that can approximate R by multiplying them, $R \approx R' = WH$. MF provides highly accurate and scalable solutions for recommender systems because after the rating matrix is decomposed, the entire decompositions do not need to be loaded into memory. An unknown rating for a specific active user and target item can be predicted by simply multiplying the corresponding user and item latent vector.

Consider a typical problem in recommender systems: given a n by m matrix R representing a rating matrix, n rows represent n user and m columns represent m items. The number of latent features is k, and then the users and items can be represented as vectors with dimensionality k. The original rating matrix R is then approximated as

$$R^{n\times m} = W^{n\times k} H^{k\times m}. \quad \textbf{(2-12)}$$

$W^{n\times k}$ and $H^{k\times m}$ are the mappings of users and item over latent feature space and can be learned from observed entries when every row in $W^{n\times k}$ represents a user's preference and every column of $H^{k\times m}$ represents an item property. The unknown entries (ratings) can consequently be predicted by multiplying these two matrices.

MF is not new either in individual recommender systems or group recommender systems. In (Sarwar et al. 2000), Sarwar et al. first introduce MF technique, singular value decomposition (SVD) into recommender systems to approximate the original rating matrix by far more low rank matrices. Unfortunately, this method relies on matrix completion and can be very computationally expensive due to a very large rating matrix and extreme sparsity, although it provides advanced storage efficiency in most cases. Hence, many recent works suggest directly approximating the observed ratings only. MF techniques become very popular for their high performance since a number of MF-based models achieve almost 10 percent increase in accuracy at the contest Netflix Prize over exiting methods on Netflix evaluation data set. Biased regularized MF, which is proposed in (Paterek 2007), including regularization when decomposing, is suggested to model independent user base line based on the items that they have rated to avoid explicitly parameterizing each user. In biased regularized incremental simultaneous in (Takács et al. 2009) and the SVD++ model in (Koren, Bell & Volinsky 2009), MF models analyze items, concerning not only the ratings but also additional information item qualities.

In GRSs, there are generally massive missing ratings which cause difficulty in appropriately modeling the group profile. SVD is employed in (Hu, Meng & Wang 2011) to decompose the rating matrix and model the group profile by aggregating the decomposed user profiles. However, this method suffers from the strategy selection

problem when modeling the group profile. It is important to note that no stable solution is guaranteed by traditional matrix factorization because factorization results are significantly affected by initial values and matrix update protocols, therefore it is not possible to build a stable and unique group profile. Another problem is the high computational cost and low quality when faced with high dimensional sparse data, which makes it impractical for real recommender scenarios.

In (Shi, Wu & Lin 2015), the authors design a group recommendation method to build the group profile based on latent space instead of the raw ratings. The implicit profile on latent space can better describe the individual profiles. Therefore, the raw rating matrix is decomposed using the stand matrix factorization and thus obtain the latent individual profiles over latent space. The group profile is built by average aggregating the individuals and predictions are calculated by inter product by group profile and latent item vector.

Another matrix decomposition technique used in (Loveymi & Hamzeh 2015) is to generate a transition matrix by multiplying raw rating matrix and the transition matrix to project the raw ratings into a new space to make users more similar. The recommendations are made using item-based CF and the group recommendations are made by aggregating the individual recommendations.

2.3.2 NON-NEGATIVE MATRIX FACTORIZATION IN RECOMMENDER SYSTEMS

One limitation of MF is that the representations which allow negative entries do not make sense in some applications, such as recommender systems. In general, MF is an unsupervised training process, in where an objective function is defined to model the performance of approximation. The decomposed matrices are iteratively updated according to that function. Since no constraint is set for them when fitting observed entries, MF then allows negative entries, which do not make sense in some applications, appear in the decomposed matrices. However, the non-negative constraint is very important for making latent features which can actually represent preferences of users or properties of items.

Comparing to the traditional MF, non-negative matrix factorization (NMF) incorporating the non-negative constraint is first introduced by Lee et al. in (Lee & Seung 1999, 2000). The predictions are rescaled to cancel out the negative entries during updating of the vectors, and thus only the non-negative entries remain. NMF, therefore, makes more sense than basic MF and the decompositions enhance interpretability in RSs.

$$\begin{aligned} & R = WH \\ & s.t. \quad \begin{aligned} & w \geq 0 \quad \forall w \in W \\ & h \geq 0 \quad \forall h \in H \end{aligned} \end{aligned} \tag{2-13}$$

However, because the data in RSs are often highly sparse, which means the observed ratings are far less than the unknown ones, decompositions lack practicability and reliability for the over-fitting problem. Therefore, regularization methods are used to improve the performance of NMF methods.

Wang et al. hybrid two procedures to learn the decompositions: one is based on Expectation Maximization (EM) and one is based on weighted non-negative matrix factorization (WNMF) in (Wang et al. 2006). The EM procedure converges well and is influenced less by initial values and the WNMF is computationally efficient. In (Chen, Wang & Zhang 2009), a method called orthogonal nonnegative matrix tri-factorization is used to cluster the rows and columns of the rating matrix. The neighbours of user/item are selected according to user-based/item-based clustering and the prediction is the linear combination of user-based and item-based weighted average results.

Kim and Choi, in (Kim & Choi 2009), introduce several fast and scalable techniques to predict unknown ratings. These techniques are well-studied optimization techniques to resolve the WNMF problem. An alternating least squares method and a generalized expectation-maximization method are employed to generate new updating processes.

Luo et al. develop a novel updating process for the NMF-based model incorporating Tikhonov regularizing terms in (Luo et al. 2014) to address the industrial extreme sparsity problem. When updating the element, every feature

involved is treated as minimizing target and the new updating process updates each element according to new objective function.

2.3.3 SEPARABLE NON-NEGATIVE MATRIX FACTORIZATION IN RECOMMENDER SYSTEMS

Despite all these efficiencies, MF and NMF still suffer from some problems. Either traditional MF or NMF starts with some guesses for W and H. Then some learning methods are employed to iteratively improve them by updating methods such as gradient descent. Unfortunately, the performance of decomposition is usually strongly influenced by starting guesses (initial values). Therefore, two limitations influence their performance: a) initial values; b) dimension of latent features.

An MF-based model works by mapping both the items and users into the same latent feature space, training on observed ratings to obtain desired user/item features. The predictions for unknown ratings are generated by the inner products of corresponding user-feature and item-feature vector pairs. Because the performance of predicting is heavily relying on the obtained low-rank user-feature and item-feature matrix, when the original rating-matrix is extreme sparsity, decompositions using MF and NMF techniques is greatly influenced by initial values of user-feature and item-feature vectors. The value of dimension of latent features is set and pre-defined by systems. This value needs to be tested and set for specific different domains and data sets.

SNMF can address these two problems because it guarantees a unique and stable solution. Once the partial rating matrix M_S has been obtained, SNMF (Koren, Bell & Volinsky 2009; Rennie & Srebro 2005) is employed in the CS model to identify the vertices. Compared to traditional matrix factorization techniques that rely heavily on initial values, a stable solution and representativeness degrees can be guaranteed by the SNMF. The SNMF on M_S is defined as

$$M_s = WM_s^B \text{ ,} \qquad \textbf{(2-14)}$$

where ${M_S}^B$ is the basic matrix, W is the weight matrix. The basic matrix ${M_S}^B$ consists of a number of rows from M_S which can be used to recover M_S more accurately than

other rows. Members with profiles in the basic matrix can be seen as representative members. The representative member set for U_s is denoted as ${U_s}^B$. To the best of my knowledge, there is no one using SNMF related techniques in recommender systems up to now.

2.4 DATA VISUALIZATION

2.4.1 CONCEPTS

Data visualization aims to simplify understanding comprehended and abstract data for human beings and the related techniques are widely used in computational engineering and science. It is an overall concept covering both information visualization and scientific visualization.

The difference between information visualization and scientific visualization is that the former focuses on abstract data while the latter focuses on physical data. For example, a rendered 3-D image of Mars according to data collected by a probe is scientific visualization; a pie chart to demonstrate its atmosphere composition is information visualization. Therefore, in RSs, data visualization often specifically refers to information visualization.

According to the nature of data, for information visualization, in (Card, Mackinlay & Shneiderman 1999), Shneiderman categorises them into seven classes:

- 1-dimensional : document lens, code lens, value bars
- 2-dimensional : GIS, images
- 3-dimensional : CAD, architecture
- Multi-dimensional: parallel coordinates
- Tree: folder system, organization chart
- Network: social network, topic net
- Temporal: timelines, project manager

2.4.2 DATA VISUALIZATION IN RECOMMENDER SYSTEMS

One of the most significant problem suffered by RSs – the inadequacy of explanations – means they lack persuasion. Design and employ visualization techniques, hybrid with RSs, can effectively address this issue. Many visualization techniques have been used to provide an instinctive understanding of the system and reveal deep-level relationships in data. Systems that show results as graphs are known to engender more trust, and therefore more loyalty, from its owners and users.

Even simple graphs, like line, bar, scatter chart can improve user understanding. Middleton et al. Figure 2-4 presents a system – Foxtrot proposed in (Middleton, Shadbolt & Roure 2004a) that recommends on-line academic research papers. They used a profile visualization approach to allow users to interact with the system and build the profile in line form with varying time.

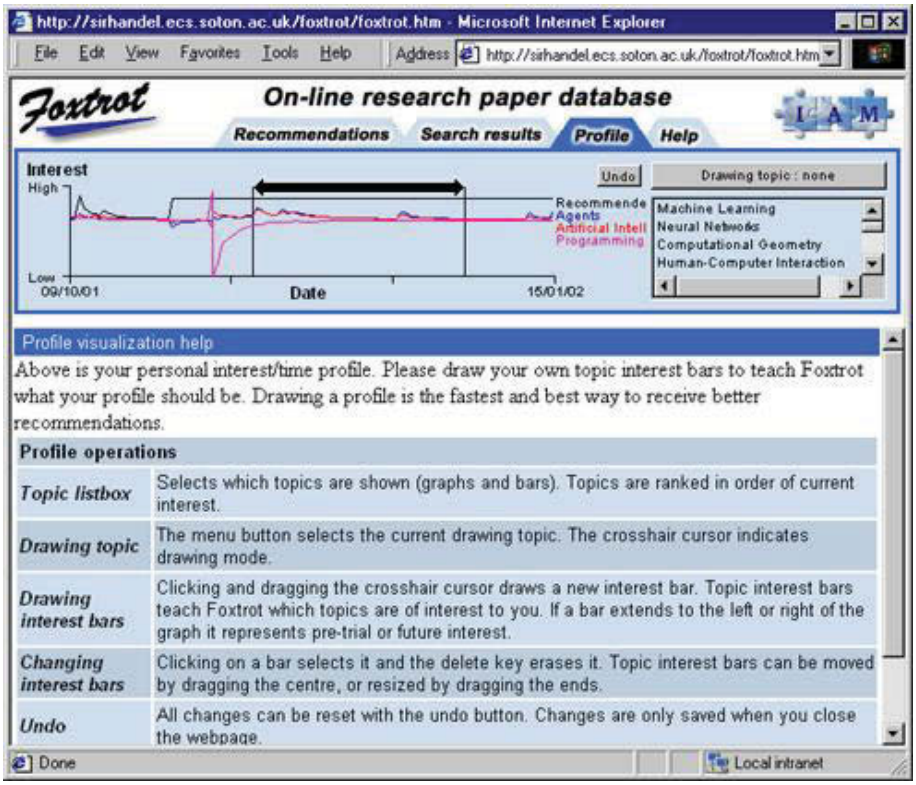


Figure 2-4. Foxtrot system. The profile is illustrated at the top of the page.

Figure 2-5. TasteWeights system for music recommendation.

The profile is represented in ontological terms understandable to the users, when the profile is used to search most related articles. In (Bostandjiev, O'Donovan & Höllerer 2012), TasteWeights system, which is presented in Figure 2-5, recommends items to users and the music is evaluated according to degree of preference in other sources such as Facebook and Twitter. The slides and bar charts are used to demonstrate the number of trusted items and weights for each item with respect to their sources.

Besides the above visualization to represent peer-peer comparing relationship, the graph visualization is suitable when inherited relationships exit. A typical example of this kind of relationship is a product tree as shown in Figure 2-6. A graph G can be represented as $= (N, E)$, where N represents a set of nodes and E represents a set of edges. The edges connect two nodes in N, and are directed from one node to another, if and only if, there is an inherited relationship between two nodes. Hierarchy graphs are therefore well-suited to RSs because the items naturally inherit on users' preferences.

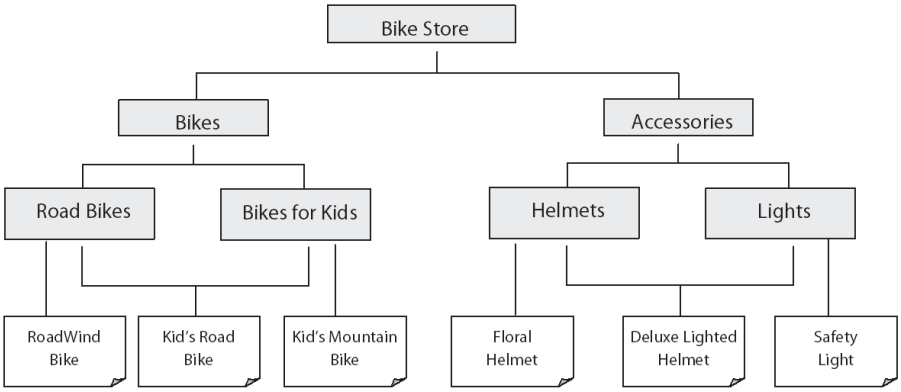


Figure 2-6. An example of product tree.

There are many different layouts to represent a graph. The node-link layout is the simplest (Kaufmann & Wagner 2001; Purchase 1997). It computes positions of each node and draws every edge as a curve. More examples are given in (Battista et al. 1999). Layouts created by tree algorithms and spring algorithms fall within this node-link category. The classic tree was first proposed in (Reingold & Tilford 1981) and has become one of the basic methods for describing data with inherited relationships. It is sometimes the sole focus of a study because of its simplicity and popularity. Classic trees are straightforward and provide clear 2D representations, but many variations (Marriott & Sbarski 2007)on the classic layout , like radial (Eades 1992; Nguyen & Huang 2003; Yee et al. 2001), have been proposed to improve space efficiency and generate compact graphs. The spring layout, first proposed in (Eades 1984), is well-known as a force-directed graph and is also well studied because of its simplicity. Spring layouts are based on a cost function that models graph edges as "springs" with forces between nodes that pull two nodes together or push them further apart. The graph iteratively changes until it becomes stable. A simple example of a spring layout is a social network graph that moves two nodes closer, or further apart, depending on the closeness of their relationship. The final graph is generated after several iterations to adjust relative positions of each node. Some improvements to spring layout focus on different cost function as outlined in (Davidson & Harel 1996; Misue et al. 1995). The most important difference between tree and spring layouts is that the edges in tree layouts do not form a circle, making them inappropriate for neighbour-based RSs. Recommendations for an active user in neighbour-based systems are determined by aggregating the opinions of similar neighbours. In this case, if an item is recommended because it is preferred by two neighbours, then edges

between nodes representing the active user, two neighbours and the recommendation form a circle. Spring layout points the nodes over concerning the cost function and thereby generates a unorganised graph for RSs. For example, a spring layout graph representing members, neighbours and items as nodes, tends to mix and scatter all the nodes and make it difficult for specific member to find those directly related recommendation nodes at once. Therefore, the spring layout can explain relationships between nodes but lack organizing in recommender system.

Heckerman et al. (Heckerman et al. 2001) used a network to show predictive relationships as shown in Figure 2-7. The data are represented as nodes in the network and all the nodes have one or more dependency links to the other nodes that describe the degree of correlation between them. When the degree is less than a user-defined threshold, the link is not displayed in the network.

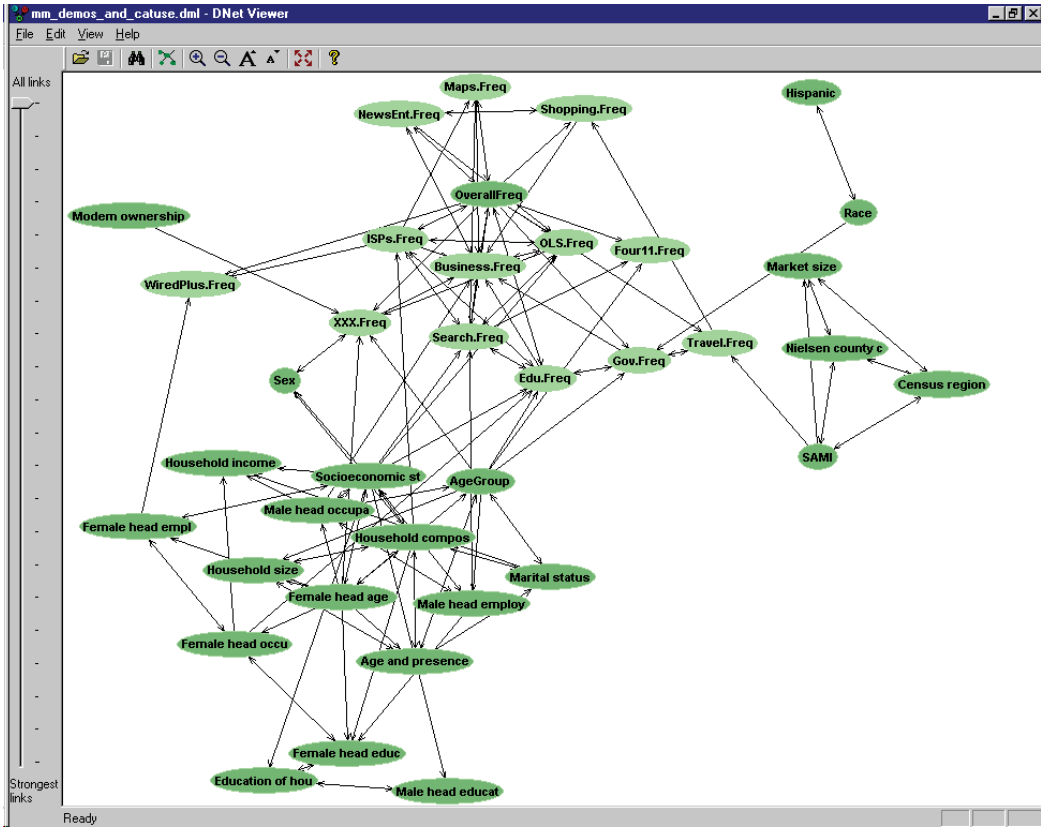


Figure 2-7. A dependency network in (Heckerman et al. 2001) to show the relationships between demographic information and internet-use data.

A CF-based recommender system (O'Donovan et al. 2008) used a network to show the relationships between recommendations, the active user, neighbours and recommendations. Variations of network such as trees are also used to explain

recommender system. A novel explanation technique, based on trees, was presented by Hernando et al. (Hernando et al. 2013). Items are represented as leaves, and the links between them provide reliable recommendation information.

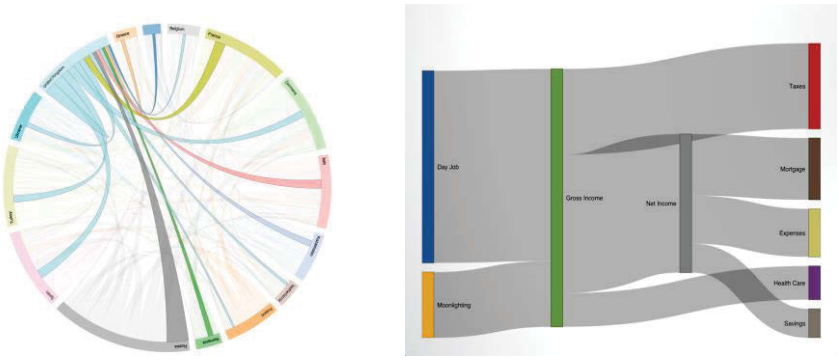


Figure 2-8. An example of chord and Sankey diagram

Chord and Sankey diagrams have also been used to depict quantitative relationships between nodes. Chord diagram, as a weighted directed graph, were proposed in (Stoica et al. 2003); however, when many types of nodes are required – such as users, neighbours and recommendations – they lose readability. Sankey diagrams (Wongsuphasawat & Gotz 2012) clearly show the hierarchy information and are very readable compared to other types of node-link layouts, but the technique is unpredictable and highly likely to produce too small bars. D. Gotz and Z. Wen (Gotz & Wen 2009) used a variation on the tree layout, named fanlens, along with many other basic layouts, such as bar, line and scatter charts, to find interaction patterns and adaptively switch the layout of displaying result for users, which is shown in Figure 2-9. Map-based visualization techniques are used in many geographic location systems, but this constraint narrows the range of applications. In (Gansner et al. 2009), users can gain a clear image illustrating data of selection for TV shows and music based on canonical maps as shown in Figure 2-9. 2D mapping such as SOM is used in (Chen et al. 2013) as shown in Figure 2-10 to map and visualise all the services according to their relationships. In (Hamasaki, Goto & Nakano 2014) and (Hernando et al. 2014), graph is also used to present the relationships between items.

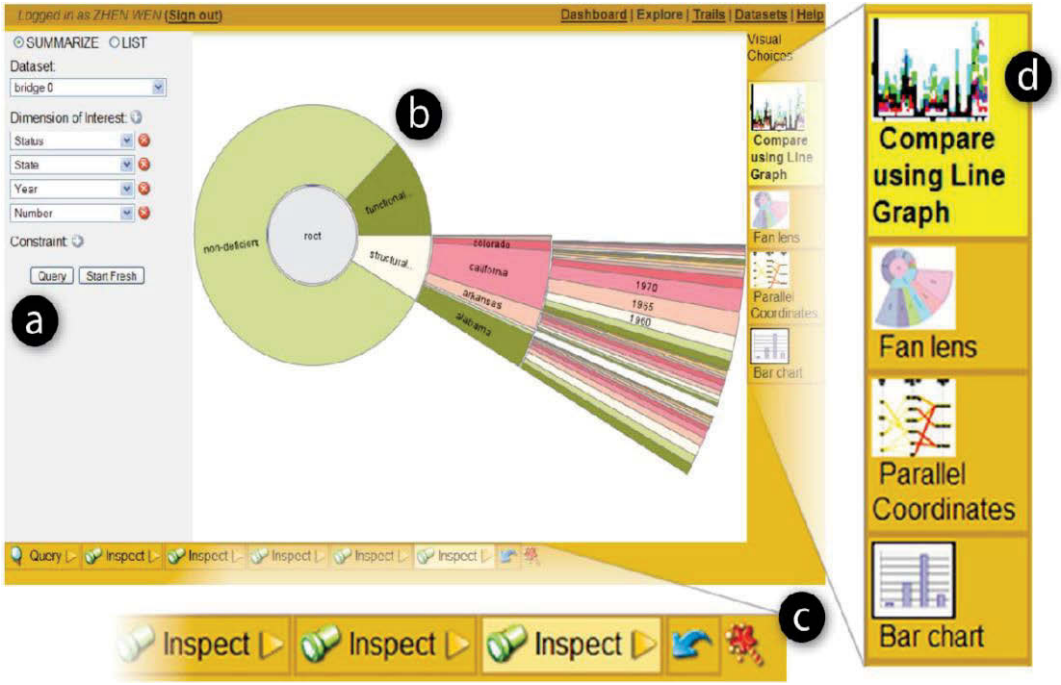


Figure 2-9. Fan lens diagram is presented in the main window when line chart, parallel coordinates and bar chart are shown at the side bar as alternatives.

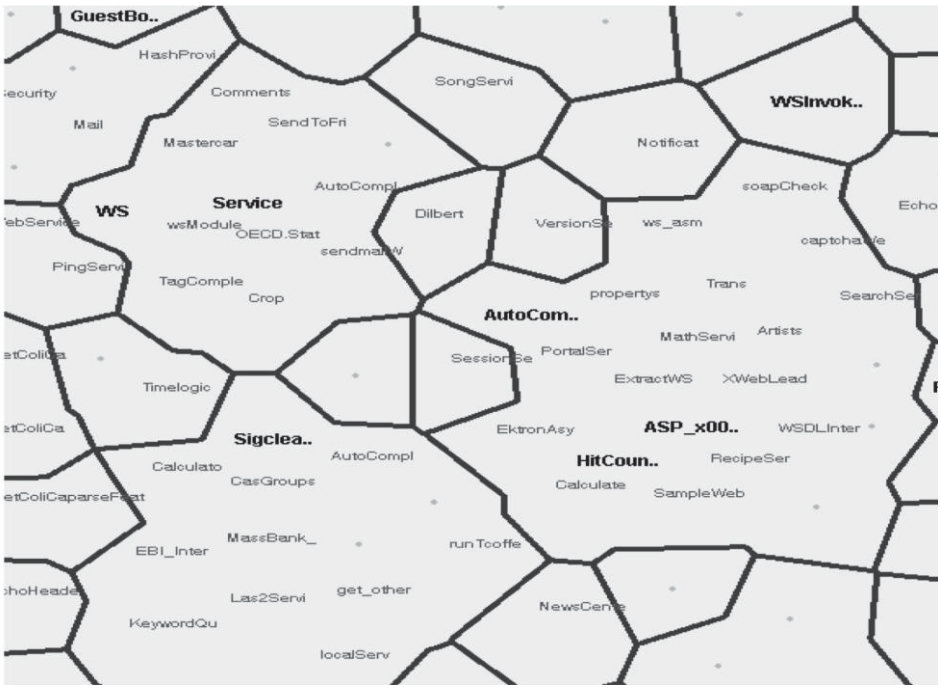
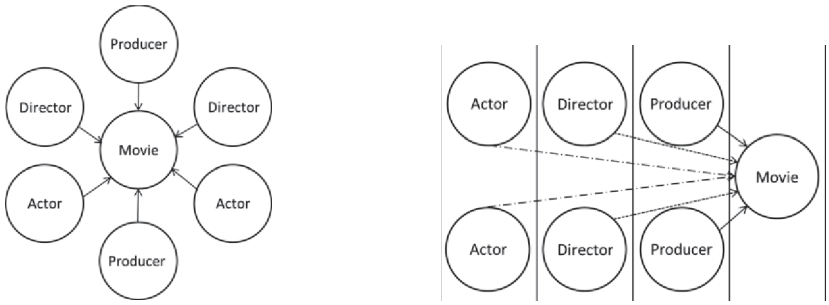


Figure 2-10. SOM visualization.



(a) (b)

Figure 2-11. (a) is an example of hierarchy relationship on a movie. Producer, director and actor are there different types of nodes that a movie inherits from. (b) represents general hierarchy graph and use different level to represent different type of nodes when different line type to represent different edge type.

Hierarchy graphs take components organization one step further when each node and each edge has an indicator to specify a type. For example, in figure (2a), a spring layout graph of a movie would show the directors, producers and actors as three types of roles. The movie inherits its script from the producer, its visualization from the director, and its performance from the actors. Figure (2b) shows the hierarchy graph using different node levels and line types to represent node and edge types. In (Gretarsson et al. 2010), a multi-layer network is used to show individual RSs. Profiles, common friends and recommendations are presented as nodes and are allocated to different levels and depicted using different colors. Unfortunately, nodes in the same level are unordered and lack readability. Furthermore, using both color and level to represent the type of node is redundant.

CHAPTER 3

LOCAL COLLABORATIVE FILTERING APPROACH

3.1 INTRODUCTION

User-based collaborative filtering recommendation methods are widely used in group and individual RSs. In these methods, for GRSs, a pseudo user is first build to represent the preference of the overall group and neighbor users, i.e. share the similar rating pattern with pseudo user, are identified using some similarity measures. The unknown group rating for a target item is predicted by combining all the ratings specified by neighbors for that target item. Therefore, the weighted average method can be used for predicting. However, because users usually specify biased ratings, i.e. they tend to give extreme ratings comparing to the others. Biased ratings make simply using weighted average method unable to predict unknown group ratings accurately. To address this issue, instead of raw observed ratings, deviations from corresponding average rating are widely used for predicting.

Unfortunately, the accuracy of predictions is also influenced by pseudo user's average rating. Actually, the hidden concept behind deviation combining is that the pseudo user's deviations can be approximated by linearly combining the deviations of neighbors over the entire domain. However, this combination can be inaccurate when ratings on sparse domain because global approximation force the combination focus on dense domain. It is important to point out that most users specify biased and sparse tail ratings. In this case, when ratings are far from the average rating, the predicting

accuracy for them is low. It is important to note that the pseudo user's profile is obtained by aggregating individual profiles; this profile (rating distribution) becomes more complex and this issue is more severe when group size becomes larger or the group is formed randomly. This problem can also influence the accuracy when using neighbor-based collaborative filtering methods in individual RSs, because the active user is treated as the same as the pseudo user in GRSs.

It is quite important to note that using some specific distributions to simulate users' rating patterns are not feasible because they are highly related to individual behaviour habits which could be largely different from the others. Also, these kinds of simulations could cause overfitting problem when building a simulation model based on a user's known rating which only covers a small part of the items comparing the whole item space. Another issue is that these kinds of simulations for an individual user are computation consuming.

The rest of this chapter is organised as follows. Section 3.2 introduces a local collaborative filtering approach and its utilization in individual RSs and GRSs is presented in 3.2.1 and 3.2.2 respectively. A case study is shown in Section 3.3 and leave-one-out cross validation is conducted. The results and discussion are also given in Section 3.3. Lastly, the summary is given in Section 3.4.

3.2 LOCAL COLLABORATIVE FILTERING APPROACH

Neighbour-based collaborative filtering utilizes vector relevance measures to identify similar users/items (neighbours) and thereby, generate linear approximation by neighbours for unknown ratings. However, this approximation is calculated on overall user/item space, and because specified ratings are often biased, it is difficult to get a single global approximation matching all the unknown ratings. From the predicting accuracy perspective, it obtains higher accuracy in density area, and larger error in sparse area. Enhanced local collaborative filtering approach is to refine average rating calculating using vector relevance measures, resulting in better approximation considering target item.

To compute local approximation is: for every target item, measuring relevance between target item and items that have been rated by active; identifying neighbours according to relevance results; the local average rating is calculated based on ratings for neighbours of active user; the unknown rating is predicted by weighted average by observed ratings for target item of neighbours. The key of local collaborative filtering is to calculate average rating under the constraint of item relevance evaluation. In dense and reasonable rating region, the average rating can be calculated accurately. In sparse and biased rating region, the average rating is calculated alleviating effects of ratings far from the target item and therefore obtain accurate average rating.

3.2.1 LOCAL AVERAGE RATING ESTIMATION APPROACH

Since unknown ratings in neighbour-based CF experience preference biased problem when predicting the unknown ratings especially those falling on tail because they are predicted by overall profile not taking take target item into consideration. A local average rating estimation is established to refine the profile by remove irrelevant ratings in it. As discussed above, the local average raging first measures item relevance by Manhattan distance, and average rating is calculated on a reduced profile that only contain relevant entries. Evidently, the unknown ratings are predicted using local average ratings instead using overall average ratings directly.

The overall item space is noted as I, $I = \{i_1, i_2, \dots, i_{|I|}\}$ which i_k is an item that belongs to I. The overall user space is noted as U, $U = \{u_1, u_2, \dots, u_{|U|}\}$ which u_k is a user that belongs to U. All the observed ratings are represented as a rating matrix R over I and U. The active user is noted as u^* and target item is i^*. Apparently, because r_{u^*,i^*} is unknown, $i^* \notin I_{u^*}$. Let f be the employed recommendation method. The prediction of unknown rating r_{u^*,i^*} is then can be represented as Equation (3-1)

$$r_{u^*,i^*} = f(u^*, i^*, R) \tag{3-1}$$

Let I_u be all the items that have been rated by user u. Specifically, I_{u^*} will be all the items that have been rated by active user u^*. The overall average rating is mean of all the observed ratings for u^*. The local average rating is to find part of items belongs to I_{u^*} those rating pattern is similar to target item i^*.

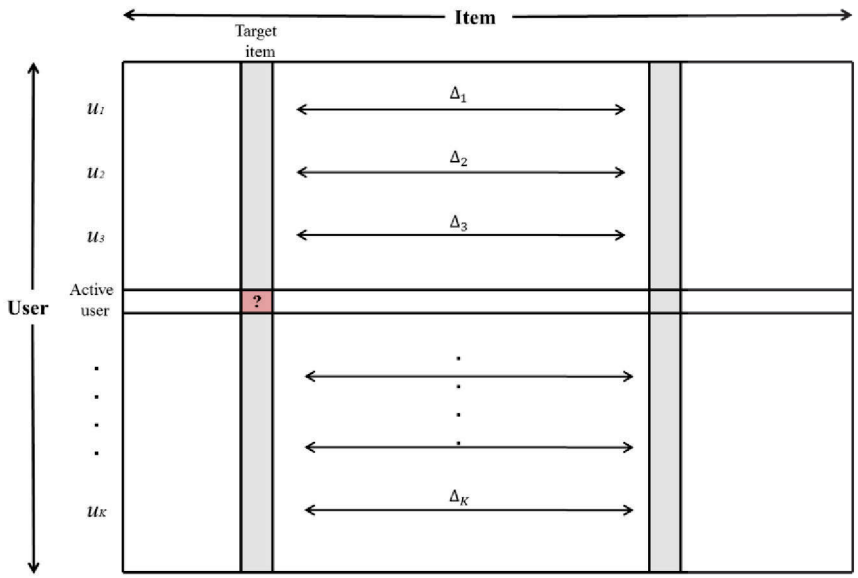


Figure 3-1. Explanation of calculating the local average rating. The relevance between the target item and a non-target item is measured according to distance between two corresponding item rating vectors.

In order to identify all the items relevant to i^*, relevance between two item rating vectors need to be measured, the Manhattan distance-based distance is used to measure the relevance between two users. The same method is employed to measure the relevance of two items.

Because only items that have been rated by active user are needed to be considered, for specific active user, it does not need to compute all the relevance between target item and every non-target items. Given active user u^*, the target item i^* and non-target item i in I_{u^*}, the corresponding known rating vectors are r_{i^*} and r_i. The Manhattan distance-based method is employed to measure the distance between two vectors. Basically, Manhattan distance for two vectors is the sum of the absolute differences on every dimension of the vectors. It is a special Minkowski distance with exponent =1, and is also known as rectilinear distance, L_1 distance or ℓ_1 norm. Manhattan distance make sense here because it does not overweight an absolute difference on single dimension which make the measure would not ignore most the dimensionality.

However, r_{i^*} and r_i are often contain many missing values inRSs. Therefore, they cannot be uses as input of Manhattan distance directly and need to be projected to a space without missing values. Assume there are *K* common users over overall user

space. Then, two vectors are projected to this sub user space. Let two projected vectors are $r_{i^*} = (r_{1,i^*}, r_{2,i^*}, \dots, r_{k,i^*})$ and $r_i = (r_{1,i}, r_{2,i} \cdots r_{k,i})$ $k = 1 \dots K$. The distance between them is $\Delta r_{i^*,i}$.

Let $\Delta r_{i^*,i}^k$ be the absolute rating difference between r_{i^*} and r_i over user k, therefore, $\Delta r_{i^*,i}^k = |r_{k,i^*} - r_{k,i}|$. The Manhattan distance of these two vectors is sum of all the distances that $D_{i^*,i} = ||r_{i^*} - r_i||_1 = \sum_{k=1}^{K} \Delta r_{i^*,i}^k$.

The less distance the two vectors are similar, in extreme situation, when two vectors are same, the distance between them is 0. The potential neighbour items are then can be identified those distances smaller than a predefined threshold. However, one limitation of this threshold is that it is difficult to extend across different systems. For example, one system allows users specify ratings scale of 1 to 5 while another might be a scale of 1 to 100. A rigid threshold 3 could be appropriate for former one but too strict for latter one. Therefore, basic Manhattan distance cannot achieve a unified evaluation for different systems.

To address this issue, differences on single dimension $\Delta r_{i^*,i}^k$ are abstracted according to domain of system ratings before calculating relevance. This abstraction is mainly because available ratings in some systems are finite, such as MovieLens only enable users to specify integer ratings from 1 to 5. Therefore, simply normalizing difference may be excessively precise which cause losing hidden potentials. Therefore, subsection function is defined to describe closeness degree of $\Delta r_{i^*,i}^k$. The difference between upper and lower bounds of available ratings is divided into three levels with respect to the different system. Assuming the upper bound is r_{MAX} and lower bound is r_{MIN}, let S be r_{MAX}-r_{MIN}, and then the three levels are $[0, \frac{1}{3}\mathrm{S})$, $[\frac{1}{3}S, \frac{2}{3}\mathrm{S})$ and $[\frac{2}{3}S, \mathrm{S}]$. The definition of differences mapping function is Equation (3-2):

$$\Delta \tilde{r}_{i^*,i}^k = \begin{cases} 1, & 0 \le \Delta r_i < \dfrac{S}{3} \\ 0.5, & \dfrac{S}{3} \le \Delta r_i < \dfrac{2S}{3} \\ 0, & \dfrac{2S}{3} \le \Delta r_i < S \end{cases} \tag{3-2}$$

Thus the Manhattan distance of r_{i^*} and r_i becomes $D_{i^*,i} = \sum_{k=1}^{K} \Delta \tilde{r}_{i^*,i}^{k}$. The calculation is detailed explained in Figure 3-1. Apparently, the range of $D_{i^*,i}$ is [0,K] depending on the number of common user in r_{i^*} and r_i. For example, r_{i^*} and r_i have 10 common users when r_{i^*} and r_j have 20 common users, in extreme condition, $D_{i^*,i}$ equal to 10 $D_{i^*,j}$ equal 20, but actually these both two differences represent one thing, i.e. relevance reaches maximum. This makes the threshold is difficult to be set. Evidently, this problem is addressed by averaging the Manhattan distance as the final relevance of items i and j considering the number of common users. The equation is presented as follow:

$$Rel_{i^*,i} = \frac{D_{i^*,i}}{K} \tag{3-3}$$

According to Equation (3-4), comparing the overall item space, user may only provide a very small number of available ratings.

A threshold T is set to determine whether items , i and i^* are sufficiently close and whether $r_{u^*,i}$ is taken into consideration to compute the average rating of user u^* for item i^*. The local average rating of active user u^* is calculated by averaging all the ratings relevant to target item i^*, where the relevance is greater than T.

$$\bar{r}'_{u^*} = avg(r_{i^*,i}) \quad if \quad Rel_{i^*,i} \geq T \tag{3-4}$$

One problem with local average rating is over fitting.

Figure 3-2 presents an example for item relevance measuring. It demonstrates that the lastly result of average rating estimation is influenced by density of items close to the target item.

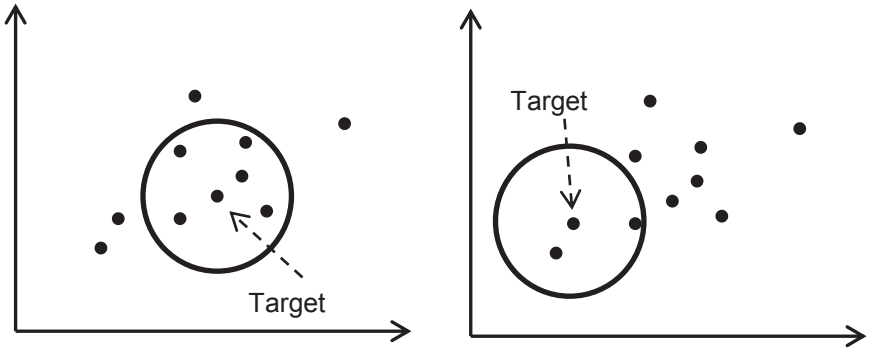


Figure 3-2. A 2D example for item relevance measuring.

When the target item is located in dense region, relevance measure can identify many neighbours which make the local average rating can be estimated accurately. However, when the target item locates in sparse region, influences from some far away items far away from target item are over weighted. Therefore, when the neighbours are limited, the reliability of local average rating is low. Hence, another threshold N for number of neighbours is set to indicate if the estimation is trustworthy. When the number of neighbours is lower than N, the estimation is not reliable and local average rating is equal to global average rating. Let LN be the number of neighbours using item relevance measuring, and GN be all the items have been rated by u^*, the local average rating $\overline{r}'_{u^*}$ is

$$\overline{r}'_{u^*} = \begin{cases} \dfrac{\sum_{i=1}^{LN} r_{u^*,i},}{LN} & LN \geq N \\ \dfrac{\sum_{i=1}^{GN} r_{u^*,i}}{GN} & otherwise \end{cases} \tag{3-5}$$

After obtaining the similarity of the active user and the average rating, the predicted rating can be calculated by using the weighted sum of deviation from the average rating of similar neighbours. The equation to predict the rating of user u for item i^* is shown in Equation (3-6).

$$r_{u^*,i^*} = \overline{r}'_{u^*} + \frac{\sum_{u \in Neighbors} (r_{u,i^*} - \overline{r}_u) \times Sim(u^*, u)}{\sum_{u \in Neighbors} |Sim(u^*, u)|} \tag{3-6}$$

where $\overline{r}'_{u^*}$ denotes the local average rating of active user u obtained by the Manhattan distance-based model, and active user neighbours are selected according to entropy-driven hybrid similarity. User u is one of the neighbour users of active user u^* and $\overline{r_u}$ is the corresponding average rating of u.

Algorithm 3.1 Local collaborative filtering approach in individual RSs

Input: rating matrix M, the active user u^*, target item i^*, parameter for local average rating T and N

Output: r_{u^*,i^*}, the rating prediction for i^* of u^*

[Begin]

$avg_{global} = 0$

$avg_{local} = 0$

$count_{local} = 0$

Get the item set I_{u^*} that has been rated by u^*

For $i \in I_{u^*}$ and $i \neq i^*$ do

$avg_{global} += r_{u^*,i}$

Projecting rating vectors of the target item r_{i^*} and non-target item r_i to common user space $u_1, u_2 \ldots, u_K$

$D = 0$

For $k \in 1 \ldots K$

$$\Delta r^{k}_{i^*,i} = | r_{k,i^*} - r_{k,i} |$$

$$S = r_{MAX} - r_{MIN}$$

$$\Delta \tilde{r}^{k}_{i^*,i} = \begin{cases} 1, & 0 \le \Delta r_i < \frac{S}{3} \\ 0.5, & \frac{S}{3} \le \Delta r_i < \frac{2S}{3} \\ 0, & \frac{2S}{3} \le \Delta r_i < S \end{cases}$$

$$D = D + \Delta \tilde{r}^{k}_{i^*,i}$$

End For

$$D = \frac{D}{K}$$

If $D \le T$

$$avg_{local} = avg_{local} + r_{u^*,i}$$

$$count_{local} = count_{local} + 1$$

End If

End For

Determining the local average rating according to if it is reliable

If $count_{local} \geq N$

$$\bar{r}'_{u^*} = \frac{avg_{local}}{count_{local}}$$

Else

$$\bar{r}'_{u^*} = \frac{avg_{global}}{|I_{u^*}|}$$

End If

Using user-based collaborative filtering to predict unknown group rating for i

$$r_{u^*,i^*} = \bar{r}'_{u^*} + \frac{\sum_{u \in Neighbors} (r_{u,i^*} - \bar{r}_u) \times Sim(u^*, u)}{\sum_{u \in Neighbors} |Sim(u^*, u)|}$$

[End]

3.2.2 GROUP LOCAL AVERAGE RATING ESTIMATION APPROACH

In this section, the global approximation is considered from the perspective of group recommender system. As mentioned in Section 2.2, most group methods using a combined pseudo user to represent the overall group preference and user-based collaborative filtering is then used to predict unknown group ratings. Therefore, the rating distribution is more complex than individuals. In this case, biased and sparse rating issue is more severe, especially when the group size is large or the group is formed randomly.

The first reason is mainly related with the differences of individual rating habit. No matter what strategy or method is employed to model the pseudo user, it is the same to some extent that the final rating distribution is synthesized by all the individual rating distributions. Therefore, the final distribution is highly probably fat tail. For example, a rating fall on tail of one member's distribution even when the

others are not prefer it because of the preference difference between members. Thus, it is more likely to construct a fat tail distribution for a group than a single user.

The second reason is the group need recommendation is passively formed randomly and members have no opportunity to specify their preferences or negotiate a consensus preference. Therefore, individual preference is highly probably conflicted with others. These random groups may be homogenous and have highly conflicting group preferences; for example, recommending music for all the people at a party. The confliction problem make it very difficult to find a compromise for diverse interests, and is more difficult to model, and recommendations are consequently more difficult to produce.

In this section, the local collaborative filtering, which is introduced in last section, is also employed in GRSs. Once the profile of pseudo user of the group is modeled, and the profile is often represented as a rating vector that contain all the items that have been rated by members. The method has three main steps to generate predictions: a) calculate similarities; b) estimate local average ratings considering target item c) calculate unknown group ratings. In the following sections, details of the method are introduced.

Group profile is expressed as the group rating vector over the items that have been rated by group members. Once this has been generated, it is used in this phase to predict unknown group ratings. A similarity measure, PCC similarity, is adopted in the work to identify neighbours close to the group. To minimize the error caused by the fat tail, a Manhattan distance-based measure is applied to compute local average ratings for the group.

Step 1: compute profile similarity between group members and non-group members

Let G be one group and g be the pseudo user modeled represented the overall group. The group profile of G is expressed as the group rating vector over the items that have been rated by group members. Once the profile, $profile_g$, has been obtained, it can be seen as a preference of a pseudo user. It is then possible to compute the similarities between the pseudo user and non-member group users. PCC

similarity, which has been widely used in a number of recommendation systems, is employed for the similarity computation. Let g be the pseudo user and u be a non-group system user. Let I_g and I_u be the item set that has been rated by g and u respectively. The PCC similarity between g and u is computed based on their common ratings as follows:

$$Sim(g,u) = \frac{\sum_{i \in (I_g \cap I_u)} (r_{g,i} - \bar{r}_g)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in (I_g \cap I_u)} (r_{g,i} - \bar{r}_g)^2 \sum_{i \in (I_g \cap I_u)} (r_{u,i} - \bar{r})^2}} \tag{3-7}$$

where $I_g \cap I_u$ is the set of common rated items by both g and u, $r_{g,i}$ and $r_{u,i}$ represent known ratings for item i, $\bar{r_g}$ is the average rating of g and $\bar{r_u}$ is the average rating of u. The similarity $Sim(g,u)$ between two users ranges from -1 to 1, where a large value indicates a higher similarity.

Step 2: compute group local average rating

Using the Manhattan distance-based method introduced in Section 3.2.1 to measure the relevance $Rel_{i^*,i}$ between target item i^* and non-target item i.

A threshold T is set to determine whether items i^* and i are sufficiently close and whether group rating $r_{g,i}$ is taken into consideration to compute the average rating of pseudo user g for item i. The local average rating of g for target item i^* is calculated by averaging all the ratings relevant to i^*, where the relevance is greater than T and reliable count is N, which is defined as Equation (3-8).

$$\bar{r}'_{u^*} = \begin{cases} \dfrac{\sum_{i=1}^{LN} r_{u^*,i}}{LN} & LN \geq N \\ \dfrac{\sum_{i=1}^{GN} r_{u^*,i}}{GN} & otherwise \end{cases} \tag{3-8}$$

where LN is the number of neighbours obtained using local average rating and GN is the neighbour number using global average rating.

After obtaining the similarities and local average ratings of the pseudo user, the unknown group ratings can be predicted. In the work, user-based collaborative filtering is adopted, and unknown group ratings are calculated by the weighted sum of deviations from the average rating of similar neighbours. Let $r_{G,i}$ be the unknown group rating for item *i*, and $i \notin I_G$, and $r_{G,i}$ can be computed by Equation (3-9).

$$r_{g,i^*} = \overline{r}'_g + \frac{\sum_{u \in Neighbors} (r_{u,i^*} - \overline{r}_u) \times Sim(g,u)}{\sum_{u \in Neighbors} |Sim(g,u)|} \tag{3-9}$$

where $\overline{r}'_g$ denotes the local average rating of group *G* obtained by the LCF model, and neighbours close to the pseudo user are selected out according to PCC similarity. User *v* is one of the neighbour users of active user *u* and $\overline{r_v}$ is the corresponding average rating of *v*. The final recommendations are selected as the top-*k* items with the highest predictions.

Algorithm 3.2 summarize these three steps to show how to generate group recommendations given the group profile, and give a detailed description of the local average rating computation.

Algorithm 3.2 Local collaborative filtering approach in GRSs

Input: rating matrix M, the group pseudo user g, target item i^*, parameter for local average rating T and N

Output: r_{g,i^*}, the rating prediction for i^* of u^*

[Begin]

$avg_{global} = 0$

$avg_{local} = 0$

$count_{local} = 0$

Get the item set I_g that has been rated by members

For $i \in I_g$ do

$avg_{global} += r_{g,i}$

Projecting rating vectors of the target item r_{i^*} and non-target item r_i to common user space $u_1, u_2 \ldots, u_K$

$D = 0$

For $k \in 1...K$

$$\Delta r_{i^*,i}^{k} = | r_{k,i^*} - r_{k,i} |$$

$$S = r_{MAX} - r_{MIN}$$

$$\Delta \tilde{r}_{i^*,i}^{k} = \begin{cases} 1, & 0 \le \Delta r_i < \frac{S}{3} \\ 0.5, & \frac{S}{3} \le \Delta r_i < \frac{2S}{3} \\ 0, & \frac{2S}{3} \le \Delta r_i < S \end{cases}$$

$$D = D + \Delta \tilde{r}_{i^*,i}^{k}$$

End For

$$D = \frac{D}{K}$$

If $D \le T$

$$avg_{local} = avg_{local} + r_{g,i}$$

$$count_{local} = count_{local} + 1$$

End If

End For

Determining the local average rating according to if it is reliable

If $count_{local} \ge N$

$$\overline{r}_g' = \frac{avg_{local}}{count_{local}}$$

Else

$$\overline{r}_g' = \frac{avg_{global}}{| I_g |}$$

End If

Using user-based collaborative filtering to predict unknown group rating for i

$$r_{g,i^*} = \overline{r}_g' + \frac{\sum_{u \in Neighbors} (r_{u,i^*} - \overline{r}_u) \times Sim(g,u)}{\sum_{u \in Neighbors} | Sim(g,u) |}$$

[End]

3.3 A CASE STUDY

To test and explain the local collaborative filtering approach, a case including 3 users and 6 items is given. The information about the case is shown in Table 3-1.

Table 3-1. Rating matrix of the case.

	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$	$item_6$
$user_1$	5	4	4	4	2	1
$user_2$	4	1	4	2	4	5
$user_3$	1	5	2	4	4	4

Before illustrating the scenarios, some explanations about this case are given. The ratings are specified from 1 to 5 and, for simplicity, no missing rating. The rating vectors of three users are same, i.e. [5,4,4,4,2,1]. The distribution of ratings is biased and sparse when ratings are extremely high/low. The PCC similarities between them are:

$$sim(user_1, user_2) = -0.50,$$
$$sim(user_1, user_3) = -0.50,$$
$$sim(user_2, user_3) = -0.50.$$

Apparently, these three users are designed to represent three types of preference pattern. It could be more complicated in real data.

3.3.1 LEAVE-ONE-OUT CROSS VALIDATION

The leave-one-out cross validation is executed on designed data set. Due to the fact that there is no missing value in matrix, every rating in it is selected as test sample in leave-one-out cross validation. Considering the matrix has 18 ratings, the validation will execute 18 times, and in each iteration, one rating is selected as testing data, and the remain 17 ratings are training data. To calculate predictions, PCC similarity is used, and the predicting results are compared between using global and local average rating. The threshold of item relevant is 0.5 and reliable threshold is set to 0. The computation results are shown below.

As in the most recent research papers, Mean Absolute Error (MAE) is used to measure recommendation performance:

$$MAE = \frac{\sum_{i=1}^{N} | r_{u,i} - \hat{r}_{u,i} |}{N} \quad (3\text{-}10)$$

In the equation above, r_i is the actual rating of user *u* for item *i* in the test data, $\hat{r}_{u,i}$ is the predicted rating generated by the recommender approach, and *N* is the total number of ratings that need to be predicted in the test data. The smaller the value of MAE, the more precise a recommender approach. The final MAE for an approach is the mean of all MAEs from all folders.

Table 3-2. Prediction results of the case.

	$user_1$		$user_2$		$user_3$	
	Global	Local	Global	Local	Global	Local
$item_1$	2.8353	3.8353	3.1778	3.9778	2.719	1.9190
$item_2$	3.1778	3.9778	2.8353	3.8353	2.8353	3.8353
$item_3$	3.4695	2.9362	3.4695	3.2695	2.8000	2.2000
$item_4$	3.4695	3.2695	2.8000	2.2000	3.4695	2.9362
$item_5$	2.8000	2.2000	3.4695	2.9362	3.4695	3.2695
$item_6$	2.7190	1.9190	2.7190	1.9190	3.1778	3.9778

The MAE results are

Table 3-3. MAE results of the case.

	$user_1$		$user_2$		$user_3$	
	Global	Local	Global	Local	Global	Local
$item_1$	2.1647	1.1647	0.8222	0.0222	1.7190	0.9190
$item_2$	0.8222	0.0222	2.1647	1.1647	2.1647	1.1647
$item_3$	0.5305	1.0638	0.5305	0.7305	0.8000	0.2000
$item_4$	0.5305	0.7305	0.8000	0.2000	0.5305	1.0638
$item_5$	0.8000	0.2000	0.5305	1.0638	0.7305	0.8000
$item_6$	1.7190	0.9190	1.7190	0.9190	0.8222	0.0222

The average MAE for using global average rating is 1.094467 when using local average rating is 0.683367. Apparently, using local average rating to approximate is more accurate than without considering target item.

To test the performance of local collaborative filtering when target rating is extreme and sparse, two scenarios in leave-one-out validation are selected out and calculation details are presented. Let $user_1$ be active user.

- Scenario 1: the target item is $item_1$, an extreme high real rating $rating_{user_1,item_1}$ is treated as unknown rating and the real value is 5.
- Scenario 2: the target item is $item_6$, an extreme low real rating $rating_{user_1,item_6}$ is treated as unknown rating and the real value is 1.

3.3.2 SCENARIO 1

Because of $rating_{user_1,item_1}$ is treated as test data, the profile of $user_1$ becomes $Profile(user_1)$ = [4,4,4,2,1] . The new similarities, $sim(user_1, user_2)$ and $sim(user_1, user_3)$ are

$$sim(user_1, user_2) = -0.75$$

$$sim(user_1, user_3) = -0.16$$

The average rating of $user_1$ is 3; $user_2$ is 3.2; $user_3$ is 3.8. Hence, the prediction is calculated using global average rating is

$$prediction_{global} = 3 - \frac{(4-3.2)\times 0.75 + (1-3.8)\times 0.16}{0.75+0.16} = 2.8353 .$$

The rating vector of target item is [4, 1]. The Manhattan distances between items are

	$item_2$	$item_3$	$item_4$	$item_5$	$item_6$
$user_2$	3	0	2	0	1
$user_3$	4	1	3	3	3

After mapping using subsection function, the distances become

	$item_2$	$item_3$	$item_4$	$item_5$	$item_6$
$user_2$	1	0	0.5	0	0
$user_3$	1	0	1	1	1

Hence, the average relevance between item 2 to 11 and target item are

	$item_2$	$item_3$	$item_4$	$item_5$	$item_6$
$item_1$	1	0	0.75	0.5	0.5

The threshold is set as 0.4.

The local average rating is mean of ratings for item from 3.

$$avg_{local} = \frac{r_{user_1, item_3}}{1} = \frac{4}{1} = 4$$

Thus the prediction using local average rating is

$$prediction_{local} = 4 - \frac{(4-3.2)\times 0.75 + (1-3.8)\times 0.16}{0.75+0.16} = 3.8353 .$$

The MAE results are

	Global	Local
MAE	2.1647	1.1647

3.3.3 SCENARIO 2

Because of $rating_{user_1, item_6}$ is treated as test data, the profile of $user_1$ becomes $Profile(user_1)$ = [5,4,4,4,2] . The new similarities, $sim(user_1, user_2)$ and $sim(user_1, user_3)$ are

$$sim(user_1, user_2) = -0.16$$

$$sim(user_1, user_3) = -0.53$$

The average rating of $user_1$ is 3.8; $user_2$ is 3; $user_3$ is 3.2. Hence, the prediction is calculated using global average rating is

$$prediction_{global} = 3.8 - \frac{(5-3)\times 0.16 + (4-3.2)\times 0.53}{0.16+0.53} = 2.7190$$

The rating vector of target item is [5, 4]. The Manhattan distances between items are

	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$
$user_2$	1	4	1	3	1
$user_3$	3	1	2	0	0

After mapping using subsection function, the distances become

	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$
$user_2$	0	1	0	1	0
$user_3$	1	0	0.5	0	0

Hence, the average relevance between item 2 to 11 and target item are

	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$
$item_6$	0.5	0.5	0.25	0.5	0

The threshold is set as 0.8.

The local average rating is:

$$avg_{local} = \frac{r_{user_1,item_3} + r_{user_1,item5}}{2} = \frac{4+2}{2} = 3$$

Thus the prediction using local average rating is

$$prediction_{local} = 3 - \frac{(5-3)\times 0.16 + (4-3.2)\times 0.53}{0.16+0.53} = 1.9190$$

The MAE results are

	Global	Local
MAE	1.719	0.919

3.3.4 DISCUSSION

From the results of leave-one-out cross validation, the mean MAE result when using traditional collaborative filtering is 1.094467 when using local collaborative filtering is 0.683367. when using global average rating. Prediction results show that the local collaborative filtering approach is better than traditional collaborative filtering. The great improvement between results from two approaches indicates that local collaborative filtering approach cans effectively filtering out irrelevant items compared to traditional collaborative filtering approaches.

Two scenarios are also given when the target ratings are extremely high/low. The calculation details are presented in these scenarios. In these scenarios, items relevance evaluation procedure identifies items with similar rating pattern for target item. As for prediction accuracy, the local collaborative filtering generates more precise results.

The significant of local collaborative filtering is, in RSs, the items are recommended with highest ratings. However, highest ratings means they are difficult to predicted accuracy because they often are far from the average rating of the active user.

3.4 SUMMARY

This chapter outlines the local collaborative filtering method to deal with biased and sparse rating prediction in individual and group RSs. To focus on local approximation, item relevance is first measured to filter out irrelevant items. The relevance model developed in this chapter is utilized to evaluate the pattern similarity between target item and non-target items. In local collaborative filtering method, the predicting deals well with the target ratings on sparse region and far from pseudo user's average rating. The effectiveness of the approach is shown in a case study. Leave-one-out cross validation on a small data set have been conducted, and the results show in the proposed recommendation approach has good performance. Two target ratings are selected and presented to demonstrate the detail of method and shows the method is well-suited in predicting unknown rating on sparse and extreme domain.

CHAPTER 4

ENTROPY-DRIVEN USER SIMILARITY FOR COLLABORATIVE FILTERING

4.1 INTRODUCTION

Neighbor identification is the core of neighbor-based collaborative filtering methods for RSs. Hence, the precision of closeness evaluation, i.e. similarity measurement, is one of the most important parts. In the most memory-based CF, similarity calculation is inherently determined by the ratings of users. For instance, Pearson Correlation Coefficient similarity and cosine coherence measures are two typical similarity measurements that are widely applied in collaborative filtering approaches to compute user similarity (Sarwar, Karypis, Konstan & Riedl 2001). They are calculated either by the absolute rating differences between two users or the absolute differences between the deviations from respective average ratings. However, these similarities are only simply aggregating similarity information obtained from every single dimension.

To illustrate the effect of using absolute difference in PCC, an example is shown in Table 4-1. Two user/item profiles are represented as vectors v_1 and v_2. Obviously, the rating patterns of these vectors are biased. v_1 is normal when v_2 biases to lower ratings. Except for the last dimension, attitudes are actually opposite. When v_1 is the

highest ratings, v_2 is the lowest ratings, and when v_2 is the lowest ratings, v_2 is the highest ratings. Therefore, similarity should be close to 0 using cosine measure and close to -1 when using PCC similarity. However, the cosine similarity between them is 0.75 and the PCC similarity is -0.19. These two similarities are both imprecise. Actually, every user has personal habits when expressing opinions, which means that the ratings are usually centralized around a particular average attitude. The closer the ratings given by two users for one item, the more similar are the two users. It is necessary to additionally consider the relative difference of deviations as well as their absolute difference. If the absolute difference of deviation is large but occurs only rarely on a small number of items, similarity should not be affected significantly. By contrast, if there are always differences on most items, the similarity should be lower, even if these differences are small.

Table 4-1. Two vectors for similarity calculating.

vectors	elements						
v_1	5	5	3	3	1	1	5
v_2	1	1	2	2	3	3	5

Therefore, the enhancement of similarity calculation is the key to improving recommendation performance. Many approaches improve similarity calculation by assigning various weights for these differences (Cacheda et al. 2011; Herlocker et al. 1999; Jamali & Ester 2009; McLaughlin & Herlocker 2004). As mentioned in Chapter 2, most improvements to similarity computation only consider absolute rating differences. Incorporating the information entropy with coherence measuring takes the relative differences between ratings into account. The information entropy treats all the independent differences as a whole. It captures additional coherent information by measuring relative rating differences comparing to traditional similarities only considering individual absolute rating differences. The proposed similarity model attempts to accurately measure the coherence between two users.

The rest of this chapter is organised as follows. An uncertainty-driven user similarity model to improve the accuracy of the similarity computation is introduce is described step by step in Section 4.2. The experiments and results are demonstrated in Section 4.3. Lastly, the summary is given in Section 4.4.

4.2 INFORMATION ENTROPY-BASED COLLABORATIVE FILTERING

4.2.1 FRAMEWORK OF METHOD

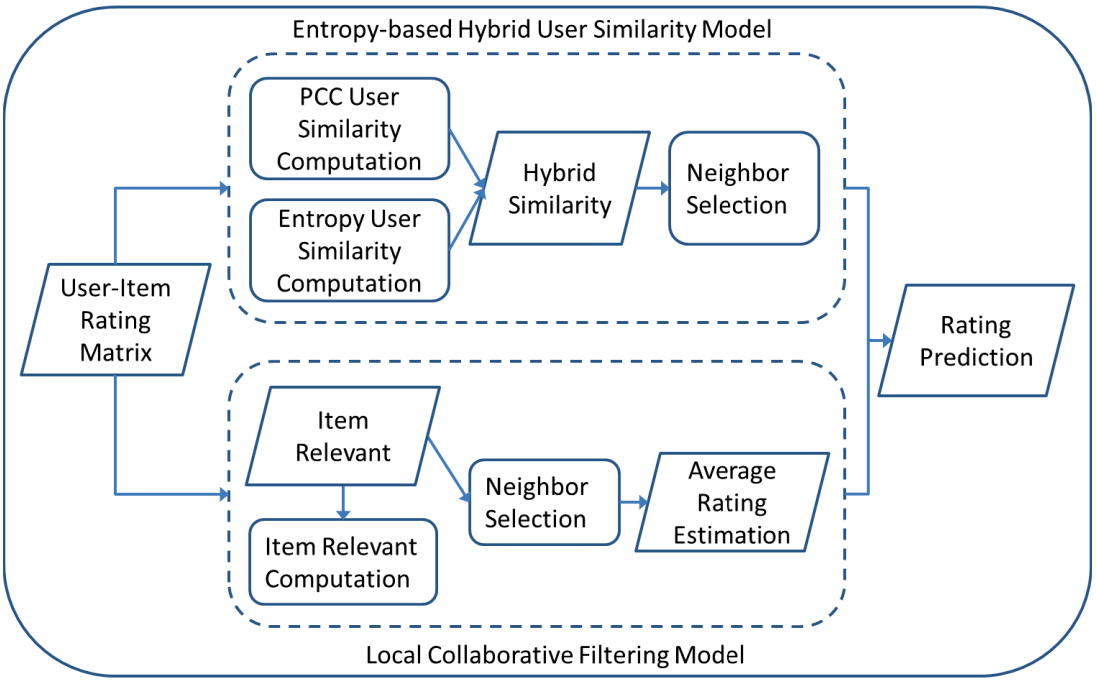


Figure 4-1. Flowchart of the approach.

The entire method is illustrated in Figure 4-1. The rating matrix is the only input of the method. Two models are executed to generate unknown ratings. Entropy-based hybrid user similarity model is to identify neighbours and Manhattan distance item relevant model is to generate local average rating. Combining two models, the unknown ratings are predicted using collaborating filtering technique.

In similarity measuring model, both entropy-based and PCC similarity are considered to evaluate relevance between two users because PCC can measure the relevance based on absolute deviation differences and entropy similarity can measure the relative differences of deviations. The similarity ensemble is then used to select users with higher relevance.

In local average rating estimation model, the relevance between non-target items and target item are calculated. A reduced set of items that has been rated by the active

user are selected out. Hence, the local average rating is calculated based on the reduce set.

Based on user neighbours and local average rating, the prediction is calculated using traditional deviation-based aggregating method.

4.2.2 ENTROPY-DRIVEN USER SIMILARITY MEASURE

In physics, entropy is used to measure the disorder of the objective things. In information theory, obtaining more information leads to uncertainty reducing. In this case, information entropy is a measure for degree of uncertainty. In RSs, similarity measuring is kind of uncertainty measuring over limited observed information. When two users have specified their ratings over the same item, the information (knowledge) can be obtained to reduce the uncertainty between two users.

For two users u and v, rating vectors of them are r_u and r_v. $r_{u,i}$ and $r_{v,i}$ represent two ratings of u and v for item i. If both u and v have specified a rating on same item , i is a common item in r_u and r_v. Let N be the number of the common items of r_u and r_v, then common ratings of u and v are represented as $\{(r_{u,i}, r_{v,i})\}$ and $i \in [1 \cdots N]$. The respective average rating of them is $\overline{r_u}$ and $\overline{r_v}$, and the deviation from average rating is $r_{u,i} - \overline{r_u}$ and $r_{v,i} - \overline{r_v}$ for item i. The absolute difference between two deviations is $\Delta d_i = |(r_{u,i} - \overline{r_u}) - (r_{v,i} - \overline{r_v})|$ and $i \in [1 \cdots N]$. A serial normalized ratio on all co-rating items can be obtained, given Δd represents the sum of all differences that $\Delta d = \sum_{i=1}^{N} \Delta d_i$

$$p_i = \frac{\Delta d_i}{\Delta d} \tag{4-1}$$

By definition of p_i, it can easily obtain:

1) For any p_i, $0 \leq p_i \leq 1$.

2) For all p_i, $\sum_{i=1}^{N} p_i = 1$.

Then p_i can be seen as probability distribution of a random variable over range $[1 \cdots N]$. Furthermore, information entropy can be used to measure the uncertainty of this variable. The information entropy of p_i is defined as

$$H = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = -\sum_{i=1}^{N} p_i \log_2 p_i \quad \textbf{(4-2)}$$

where the base of the logarithmic function is 2. Because denominator cannot be 0, when $p_i = 0$, $log_2 \frac{1}{p_i}$ is 0.

Theorem 4.1. Equation (4-2) gets the maximum when the distribution of probability function is uniform.

Proof. To maximize $-\sum_{i=1}^{N} p_i \log_2 p_i$ under the constraint $\sum_{i=1}^{N} p_i = 1$, rewriting the equation using Lagrange format:

$$L(p) = -\sum_{i=1}^{N} p_i \log_2 p_i - \delta(\sum_{i=1}^{N} p_i - 1) \quad \textbf{(4-3)}$$

Computing partial derivatives of Equation (4-3), then have

$$\frac{\partial L}{\partial p_i} = -\log(ep_i) - \delta = 0 \quad \textbf{(4-4)}$$

Therefore, when the distribution is uniform, $p_i = \frac{1}{N}$, Equation (4-4) get the maximum.

$$H_{max} = \log_2 N$$

This completes the proof.

Based on information entropy, the relevance between two users is defined as

$$Sim_{u,v}^{Entropy} = \begin{cases} 1 - \dfrac{H}{\log_2 N}, & N \geq 2 \\ 0, & otherwise \end{cases} \quad (4\text{-}5)$$

where N is the numbers of items that users u and v rated in common. According to the definition and condition, Entropy similarity has the following property. The greater the uncertainty of this variable, the less similarity there to be.

Theorem 4.2. $Sim_{u,v}^{Entropy}$ is 1 when all the differences between two deviations over every common item are same value.

Proof. Given $\forall i \in [1 \dots N]$, $(r_{u,i} - \overline{r_u}) = (r_{v,i} - \overline{r_v}) \Rightarrow \Delta d_i = 0$.

$$p_i = \frac{\Delta d_i}{\sum \Delta d_i} = 0$$

$$Sim_{u,v}^{Entropy} = 1 - \frac{H}{log_2 N}$$

$$= 1 - \frac{-\sum_{i=1}^{N} p_i \, log_2 \, p_i}{log_2 \, N}$$

$$= 1$$

This completes the proof. Theorem 2 guarantees that, if user *u* and *v* have the same ratings or even same deviations on every common item, the uncertainty between them is 0, hence the similarity is 1.

Theorem 4.3. $Sim_{u,v}^{Entropy}$ is 0 when all the differences between two deviations over every common item are equal.

Proof. Given $\forall i \in [1 \dots N]$, $|(r_{u,i} - \overline{r_u}) - (r_{v,i} - \overline{r_v})| = \Delta$d.

$$p_i = \frac{\Delta d_i}{\sum \Delta d_i} = \frac{1}{N}$$

$$Sim_{u,v}^{Entropy} = 1 - \frac{H}{log_2 N}$$

$$= 1 - \frac{-\sum_{i=1}^{N} \frac{1}{N} log_2 \frac{1}{N}}{log_2 \, N}$$

$$= 0$$

This completes the proof. Theorem 3 shows that, if difference between deviations of user *u* and *v* on every common item are equal, the uncertainty between them is maximum, hence the similarity is 0.

Theorem 4.4. The maximum of $Sim_{u,v}^{Entropy}$ is $1-\frac{log_2(N-n)}{log_2 N}$ if only n deviations over common items are same.

Proof. According Theorem 4.1, the maximum entropy over n probabilities are $\log_2 n$.

$$H_{max} = H_{same} + H_{unsame}$$

$$= 0 + log_2(N-n)$$

$$= log_2(N-n)$$

Therefore, the maximal similarity is

$$Sim_{u,v}^{Entropy} = 1 - \frac{H}{log_2 N}$$

$$= 1 - \frac{log_2(N-n)}{log_2 N}$$

This completes the proof. The concept behind theorem 4 is that, if only a few common items, where n is small, can reduce the uncertainty, the similarity is still low. When the number of similar rating increasing, the similarity becomes high.

Example 4.1. User *u* and *v* have 20 common items. If 10 deviations of them are same, then the maximal similarity between them is $1-\frac{log_2 10}{log_2 20} = 0.23$. If only 5 deviations are same, the maximal similarity is $1-\frac{log_2 15}{log_2 20} = 0.10$.

4.2.3 HYBRID USER SIMILARITY

After the entropy-based similarity is obtained, both PCC and entropy-driven similarity are taken into account. These two similarities are combined to a weighted ensemble defined in Equation (4-6) to calculate the final similarity between two users.

$$Sim_{u,v} = \alpha \times Sim_{u,v}^{PCC} + (1-\alpha) \times Sim_{u,v}^{Entropy} \quad (4\text{-}6)$$

The parameter α determines the extent to which the similarity relies on PCC similarity and entropy similarity. With $\alpha = 0$, it indicates that the prediction depends completely on PCC similarity such as the classical user-based CF approach, and with $\alpha = 1$, it indicates that the neighbours are determined by entropy similarity. A can be determined experimentally using cross-validation.

In this step, those neighbours that are the most similar to the active user are selected out to generate the prediction. Two methods are currently employed in RSs: the top-N method (i.e., a predefined number of users with higher similarity are selected), and the threshold method (i.e., all users with a similarity correlation exceeding a certain threshold are selected). The top-N method is used as recommended by (Herlocker, Konstan & Riedl 2002).

4.2.4 GROUP LOCAL COLLABORATIVE FILTERING

Once the group profile, $profile_G$, has been obtained, it can be seen as a preference of a pseudo user g. It is then possible to compute the unknown group ratings using a local collaborative filtering method.

The adaptive average rating using in LCF method is introduced in Section 3.2.2. Let i be the target item, $\bar{r}'_{g,i}$ be the local average rating for i of g. The prediction of $r_{g,i}$ is

$$r_{g,i} = \bar{r}'_{g,i} + \frac{\sum_{u \in Neighbors} (r_{u,i^*} - \bar{r}_u) \times Sim(g,u)}{\sum_{u \in Neighbors} |Sim(g,u)|} \quad (4\text{-}7)$$

The final recommendations are selected as the top-k items with the highest predictions.

4.3 EXPERIMENTS AND EVALUATION

This section introduces the data sets in the experiments, as well as the evaluation metric and design. The results of the experiments are then presented by comparing state-of-the-art CF approaches with different parameters in the approach.

4.3.1 DATA SET

The experiments are conducted on two data sets to examine the performance of the proposed CF approach. One is from the public data set MovieLens and the other is a set of data collected by the SBS.

A MovieLens data set (http://www.grouplens.org) is used as the benchmark dataset to develop the offline experiments and assess the performance of the proposed approach. MovieLens data sets are related to a movie recommender service and are collected by the GroupLens Research Project at the University of Minnesota. The data sets are publicly available and have been widely used to evaluate RSs, thus this option is chosen because the approach can be compared with other similar approaches. The 100K MovieLens data set is used which contains 1682 movies, 943 users and a total of 100000 ratings on a scale of 1–5 (where "1" = "Awful", "2" = "Fairly bad", "3" = "It's OK", "4" = "Will enjoy", "5" = "Must see"). Each user has rated at least 20 movies. The sparsity level of the MovieLens data set is 93.7% (sparsity level = 1-(100000/ (943×1682)) = 0.937).

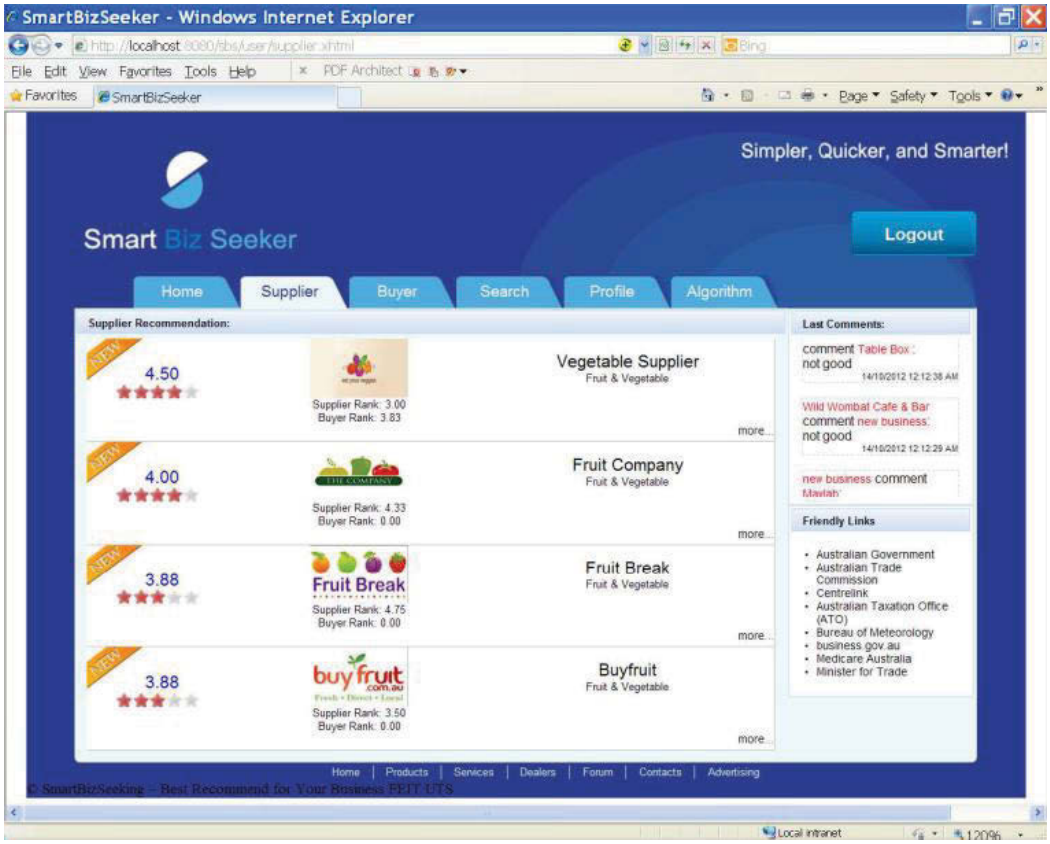


Figure 4-2 A page of SBS system show a list of recommended suppliers.

The SBS data set, shown in Figure 4-2, has been extracted from the SBS system and contains 1602 ratings of 332 businesses from 100 users. The SBS system is designed for business users to enable them to obtain a recommendation list of potential business partners. The data are combined by two kinds of ratings: first, ratings by suppliers who comment on providers, and second, ratings by providers who comment on suppliers. These ratings are not distinguished in the experiments. The ratings are also on a scale from "1" to "5", representing the degree of relatedness, in which "5" represents highly related and 1 represents not related. The sparsity level of the SBS data set is 95.2% (sparsity level=1–(1,602/ (100×332))=0.952).

4.3.2 EXPERIMENT DESIGN

To measure the improvement of the approach, several successful and state-of-the-art CF approaches are implemented and compared the results with ours. The label descriptions used to denote each of these algorithms are shown below.

IBC: the basic item-based CF approach with PCC similarity (Resnick et al. 1994a).

UBC: the basic user-based CF approach with PCC similarity (Sarwar, Karypis, Konstan & Riedl 2001).

SPCC: the user-based CF approach with sigmoid function-based PCC similarity (Jamali & Ester 2009).

FS: the item-based CF approach with fuzzy semantic similarity. (Lu et al. 2013a) argue that hybrid the item category similarity with PCC to calculate similarity. For assuming there are no side information, item category information, in the experiments, this part of the work is not implemented.

FU: the user-based CF approach with fuzzy importance similarity and user relevance rating prediction. The previous work and one parameter, user relevance threshold, is needed to select user neighbours. In the experiments, this parameter is set to 0.2 (Wang, Lu & Zhang 2014).

EU approach: the user-based CF approach with information entropy similarity and local average rating estimation. Instead of assigning a unique label for the approach, parameter values are used as labels in the approach to represent it.

To evaluate the performance of all approaches, a k-folder cross validation is conducted, which is widely used to evaluate the performance of recommender approaches, on two data sets. Five-folder cross validation is adopted and the two data sets are each split into five sub sets. In each round, one of the sub sets is selected out as test data and the rest as training data.

MAE is used to measure recommendation performance:

$$MAE = \frac{\sum_{i=1}^{N} | r_{u,i} - \hat{r}_{u,i} |}{N} , \tag{4-8}$$

where r_i is the actual rating of user u for item i in the test data, $\hat{r}_{u,i}$ is the predicted rating generated by the recommender approach, and N is the total number of ratings that need to be predicted in the test data.

4.3.3 EXPERIMENT RESULT

In this subsection, a comparison of the experiment results is first presented of different approaches. The results of using different parameters are presented in the approach, α to hybrid PCC and entropy-driven similarity and T to estimate active user's local average rating. Lastly, the results between approaches with varying neighbour size are also compared.

Table 4-2. MAE with different approaches

	IBC	UBC	SPCC	FS	FU	EU
MovieLens	0.792	0.7625	0.753	0.743	0.741	0.731
SBS	0.904	0.8878	0.886	0.881	0.866	0.862

Table 4-2 shows the MAE result obtained by IBC, UBC, SPCC, FS, FU, and the approach. The rows entitled with data sets present the results obtained by the approach over the other approaches. As shown in Table 4-2, the approach has a lower rate of error than all the other approaches on both data sets. On MovieLens, the new

approach improves the MAE by 7% compared to the item-based CF approach with PCC similarity, and improves about the MAE 4% compared to the UBC approach. Comparing the SPCC, the new approach improves accuracy by about 3%. Comparing the most recent FS and FU approaches, the EU approach still outperforms them by 1.68% and 1.37% respectively. In contrast, the results on SBS show a greater percentage of error when compared with the same approach on MovieLens. This is mainly because items in MovieLens concern only movies, whereas items in SBS consist of many kinds of products provided by companies. This means that MovieLens data are relatively more coherent than SBS data. The results show that the performances of the IBC, UBC, SPCC and FS approaches, which are only based on absolute ratings, are similar to one another. The EU approach is slightly better than other approaches. An interesting fact is that the result from the previous work, the FU approach, which also considers relative rating differences and combines them with a weight defined in a fuzzy set, shows similar performance to EU. This clearly shows that relative difference is sometimes able to measure item connection more precisely.

Figures from 4-3 to 4-6 show the impact of the parameters used to control the approach in terms of MAE. As discussed above, the parameter α is used in Equation (4-6) to balance the similarities of PCC and entropy, and T is the threshold used to identify target item neighbours for local collaborative filtering. To determine the impact of the two parameters, the experiments varied α from 0.9 to 0.4 and T from 0.6 to 0.1.

As shown in Figure 4-4 and 4-6, entropy-driven similarity slightly improves performance when T is fixed, and estimated local average rating improves performance when α is fixed. Too small T means that only a small number of items can be used to estimate the average rating and too loose T also causes MAE to increase significantly. These results show that MAE is mainly affected by average rating estimation rather than similarity calculation. This can also be observed in Figure 4-3 and 4-5. The results show that T affects performance significantly more than α, and it can further infer that global error is mainly caused by the fat tail rating problem, which means it is more important to deduce prediction error.

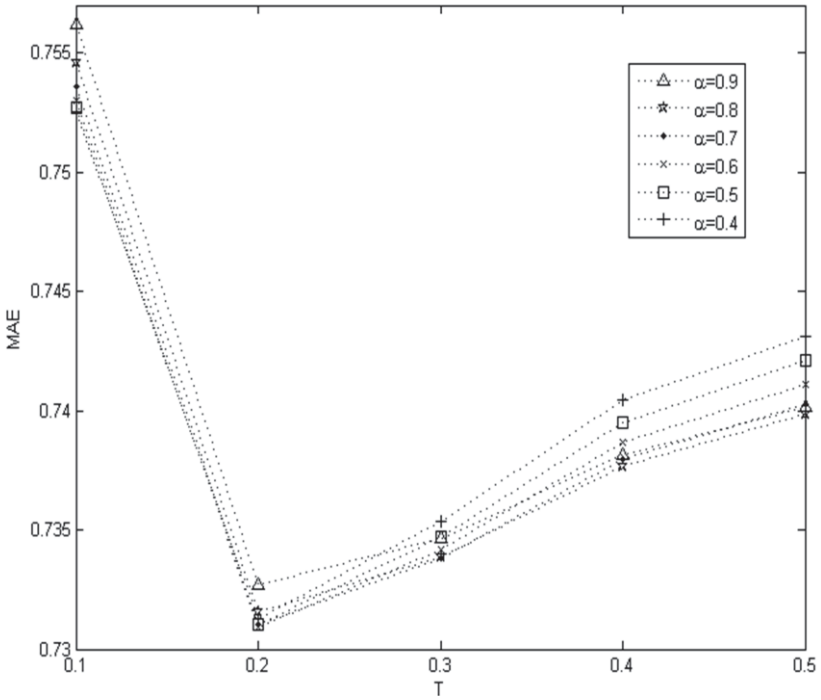


Figure 4-3. MAE results on MovieLens when using different α under given threshold *T*

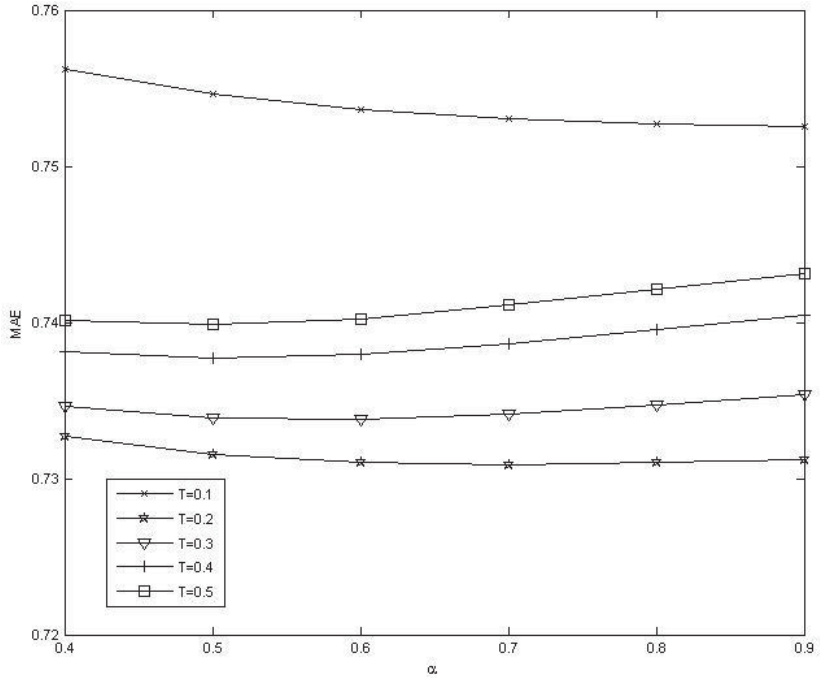


Figure 4-4. MAE results on MovieLens when using different *T* under given threshold α

To exam the sensitivity of performance with different neighbour size, several experiments are performed in which the number of neighbours varied to generate the predicted rating. The MAE results of the proposed approach are also compared with others. Figure 4-7 shows the result when different numbers of neighbours are

considered for predicting target rating on MovieLens, and Figure 4-8 shows the equivalent result on SBS. From Figure 4-7, it can be observed that the size of neighbourhood affects performance. SPCC outperforms IBC and UBC, and the accuracy of prediction is improved as the neighbourhood size increases from 5 to 30. For greater values, the curve flattens. Again, FU and FS outperform SPCC, and the approach outperforms both of them when the neighbour size greater than 50. In addition, it can be seen that the approach results in stable than other methods when there is no neighbour constraint. The reason is that the approach has access to more accurate contributions from each item from which to compute similarity. From Figure 4-8, it can be observed that a lower error rate can be obtained when setting T to 0.5 and great error occurs when setting T to 0.2. This shows that when T is too strictly defined, prediction error cannot be deduced when items are inherently different.

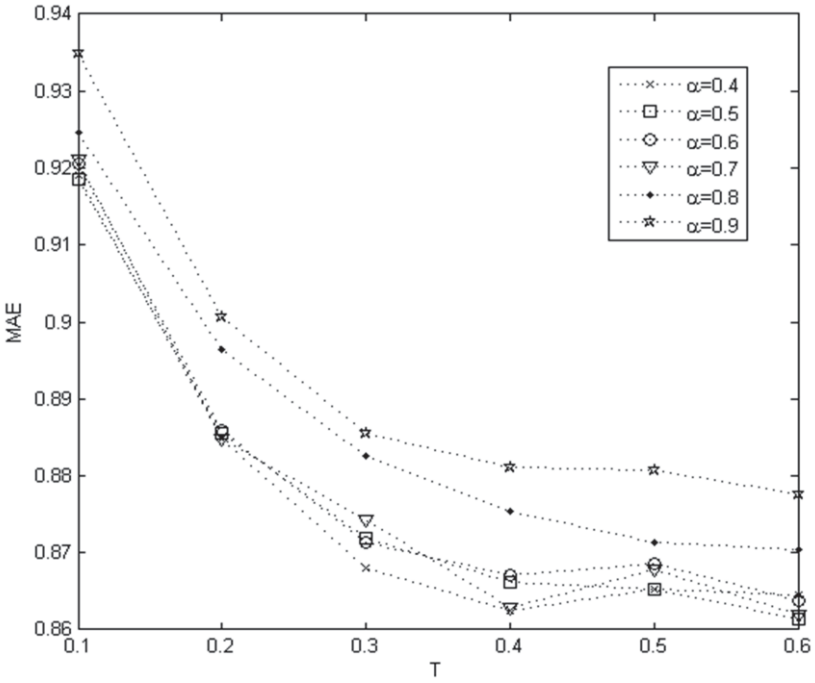


Figure 4-5. MAE results on SBS when using different α under given threshold *T*

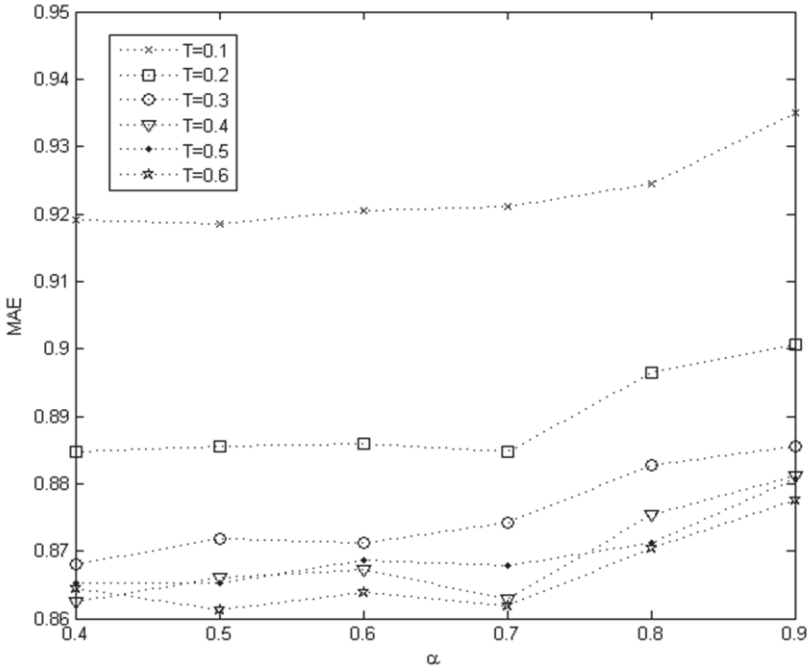


Figure 4-6. MAE results on SBS when using different *T* under given threshold α

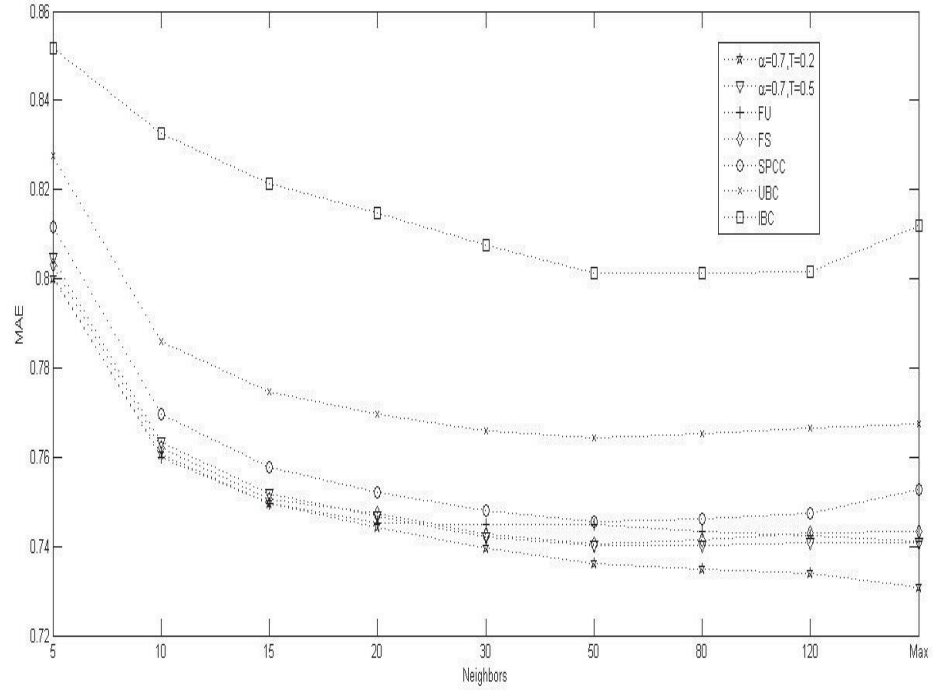


Figure 4-7. Neighbours vs MAE on MovieLens

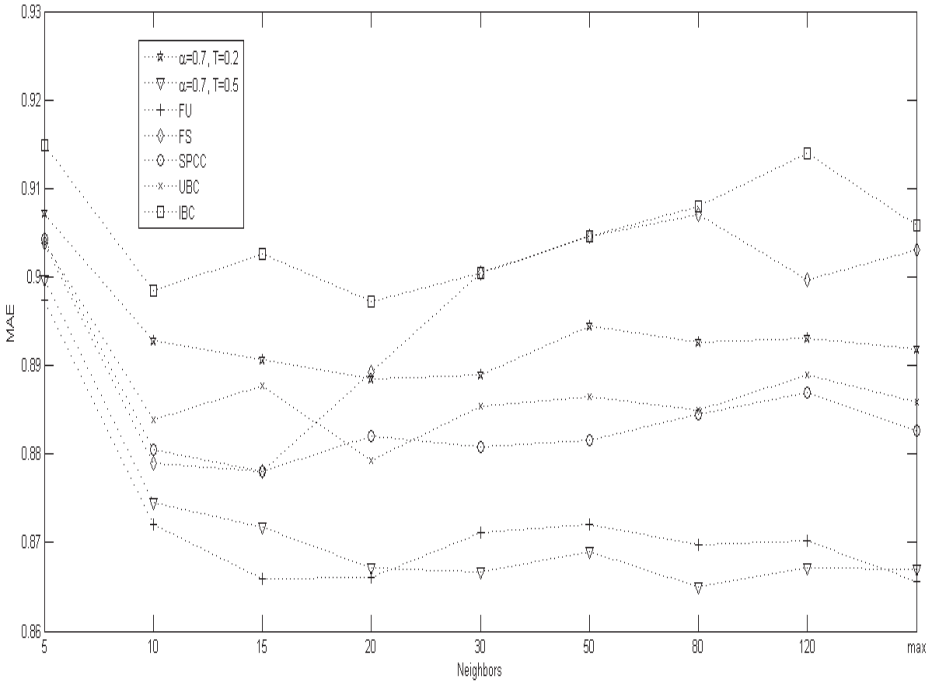


Figure 4-8. Neighbours vs MAE on SBS

4.4 SUMMARY

In this chapter, to improve the performance of CF-based approaches, a novel similarity measure is proposed based on information entropy to analyze the relative difference in ratings. The new similarity measures the degree of uncertainty of differences and make co-ratings with higher difference contribute much to similarity only when the number of these items increased. The new similarity is not sensitive to rare impulse differences but is sensitive to continued differences.

From the experimental results, it can see that the new approach which combines the new similarity measure and local collaborative filtering obtains more accurate prediction of user preference than most other approaches.

CHAPTER 5

CONTRIBUTION SCORE-BASED GROUP RECOMMENDER SYSTEM

5.1 INTRODUCTION

An important issue for GRSs is that most group profiles are constructed without considering representativeness differences between group members. In most cases, a group profile is represented as a rating vector and each entry of it is a group rating which represents a rating for an item given by the group as a whole. Actually, group ratings are calculated by an aggregating method using some strategies, such as average or least misery.

However, regardless of strategies used, all the individual members are treated equally. A typical example is, when group profile is constructed using average strategy, for an item which has been rated by many members, the system assumes them equally important, and the group rating for it is then the mean of all the observed ratings members have specified for it. Another example is, if least misery strategy (a popular border-based strategy) is used, a group rating in profile is determined by the lowest observed rating without considering if the member specified it as representative enough for the whole group. Therefore, ignoring the significance of the representativeness evaluation makes group profile modeling inaccurate. Some methods have been proposed to address this issue by introducing prior knowledge about members, such as social relationships and member behaviour patterns (Quijano-

Sanchez, Recio-Garcia & Diaz-Agudo 2014). Unfortunately, in most cases when groups are formed randomly, the knowledge is unavailable for systems to access.

Moreover, another issue is, in most cases, users only provide extremely limited information and rating matrix thereby is sparse. Actually, users only provide very few ratings comparing to a huge amount of available items and accordingly for every item in the group profile, a large proportion of members do not specified ratings for it. Hence some researchers use matrix completion techniques to fulfil the rating matrix before modeling the group profile. However, this pre-processing is computation consuming and its performance is highly dependent on if the used rating distributions match the members' rating patterns. In addition, many researchers have attempted to solve the member difference and sparsity problem by focusing on building complex individual preferences by introducing additional information, such as social network information, tags or context information, to depict member interaction or personality (Jøsang, Ismail & Boyd 2007; Recio-Garcia et al. 2009; Roy et al. 2010; Wang et al. 2012; Zhang, Wang & Feng 2013). However, there is no generally-accepted additional information available across application domains, and in many scenarios there is no opportunity to access additional information about members in a random group.

Contribution Score is developed to evaluate the representativeness of a member in a group. The evaluation depends on the rating matrix and corresponding decomposition results using SNMF techniques. SNMF equation can be represented as

$$R = WR^B \tag{5-1}$$

where R is n rows rating matrix and each row represents a user profile. SNMF gives two matrixes, W and R^B, that can approximate the original rating matrix R and the basic matrix R^B consists of a number of rows from R which can be used to recover R more accurately than other rows. Members with profiles in the basic matrix can be seen as representative members.

Rating matrix decomposition aims to develop a group recommendation approach which can maximize satisfaction within random groups by modeling preferences through the analysis of contributed member ratings alone. The proposal in this study measures each member's importance in terms of the sub-rating matrix which makes it

practical even when the matrix is highly incomplete and sparse. This approach consists of two main phases: (1) a group profile generator and (2) a recommendations generator. The contribution score is first defined for group members to model the group profile and predictions are calculated using the local collaborative filtering method (LCF) according to this profile. By integrating the CS and LCF models, a Member Contribution-based Group Recommendation (CS-LCF) approach is developed. Lastly, a group recommender system and its application in online tourist groups is presented.

The rest of this chapter is organised as follows. Section 5.2 introduces the definition of "Contribution Score". A Contribution Score-based group recommender system is presented in Section 5.3. Related experiments and results analysis are demonstrated in Section 5.4. A practical group recommender system using the proposed method is presented in Section 5.5. Lastly, the summary is given in Section 5.6.

5.2 CONTRIBUTION SCORE MEASURE FOR MEMBERS

To maximize the satisfaction of group, member differences need be measured and group profile modeling need consider these differences. A notion of a contribution score is developed to measure the importance degree or representative status of members when modeling the pseudo user and depicting his profile.

Definition 5-1. The representative members are those rating vectors can be used to recover the original rating matrix.

A typical rating matrix in RSs is: given a n by m matrix R represents rating matrix, n rows represent n user and m columns represent m items. The SNMF is to learn two low rank matrixes that without negative entries to approximate R and the basic matrix, one of the two matrixes, consist of some rows in R. The equation is

$$R^{n\times m} = W^{n\times k} R^{k\times m} \quad \textbf{(5-2)}$$

where $R^{n\times m}$ is the original rating matrix, $W^{n\times k}$ is the weight matrix and $R^{k\times m}$, is the basic matrix. Remember that $R^{k\times m}$ consists of k rows from R which can be used to

recover R more accurately than other $(n-k)$ rows. Members with profiles in the basic matrix $R^{k\times m}$ can be seen as representative members.

Definition 5-2. Contribution score of a member if he can be identified as a representative member in factorization results.

Let U be overall member set and the corresponding representative member set is denoted as U^*. For each member $u \in U$, the CS of u is defined as

$$CS_u = \begin{cases} 1, & u \in U^*; \\ 0, & otherwise. \end{cases} \quad \textbf{(5-3)}$$

5.3 CONTRIBUTION SCORE-BASED GROUP RECOMMENDATION METHOD

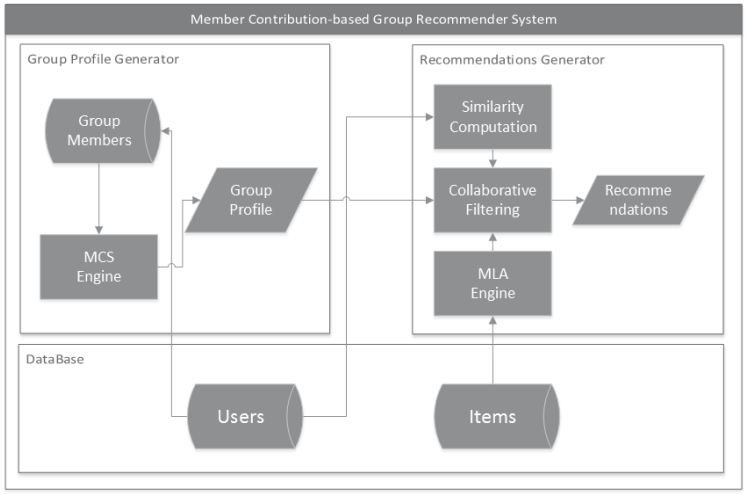


Figure 5-1. Contribution Score-based Recommender System Architecture

Contribution score-based group profile modeling method (Wang, Zhang & Lu 2016) includes the necessary computations to generate a single profile to represent the overall preference. To build the profile for a group, especially a complex random group, the preferences of the most representative members should be considered above others. Unfortunately, these representative members are difficult to identify because of the uncertainty in the system, i.e. the sparsity. To address this problem, a sampling and aggregating architecture is designed over the item space. The rating vectors of users can be perceived as high dimensional data, with each dimension representing one item. For example, suppose a movie recommender system consists

of only four movies. A rating vector of user u is $vector = [r_{u,1}, r_{u,2}, r_{u,3}, r_{u,4}]$ and the dimensions are the ratings of each movie. Instead of considering the vectors over the whole item space, sampling selects a reduced set of items for which members can provide a rating matrix without missing values, after which the representative members can be precisely evaluated on this partial rating matrix. After multiple sampling, the representative members across the global item space can be approximated by aggregating the results from all the samplings. In the work, this sampling and aggregation process is implemented in the CS model. Note that, in the work, no side information is needed. A high level of the group profile compromise equation with respect to CS can then be written as

$$Profile_G = \sum_{u \in G} CS_u \, Profile_u \,. \tag{5-4}$$

5.3.1 CONTRIBUTION SCORE CALCULATION IN SINGLE SAMPLE

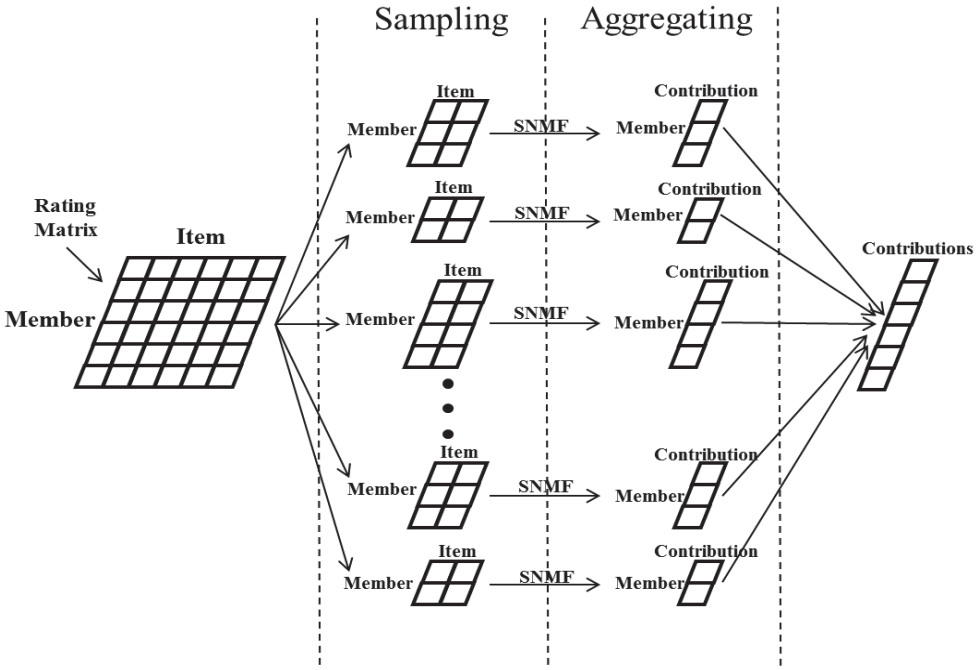


Figure 5-2. Explanation of sampling and aggregating of CS model to compute the contributions of the group members.

Traditionally, the contribution of a member to the group profile is highly correlated with the strategy adopted by the system. For instance, when adopting the least misery strategy, the member who gives the lowest rating for each item is

selected out to build the group profile. When rating matrixes are incomplete, the massive missing values make it difficult to generate the group profile according to a specific strategy. An example of a group is shown in Table 1. The unknown ratings for each item make the generated group profile less reliable. One method of addressing this problem is matrix completion, which predicts missing ratings before building the group profile and introduces new uncertainty to the system. Rather than concerning a specific strategy, CS model focuses on the representative members. A sampling consists of a projection of a rating matrix and corresponding members who have no missing values. When there are no missing values, the representative members can be measured precisely.

A sampling is noted as I_s, and any item that belongs to it is randomly selected out with equal probability. To select out members, which are denoted U_s, corresponding to I_s, filtering is carried out to exclude members who have missing ratings on I_s. After determined I_s and U_s, the partial rating matrix M_s, which has no missing value, can be projected from the original matrix M to items belonging to I_s and users belonging to U_s. M_s is used as the input of the CS model to calculate the representative members in U_s.

Actually, each sampling is a projection of rating matrix and theoretically, to minimize the error of contribution evaluation, the infinite samplings should be considered. However, in practice, it is not necessary to consider all the projections for a rating matrix of a group, it is impossible. The statistic relationship between sampling numbers and real value is shown below. Denote all the possible projections as P, the contribution of a member u is

$$C(u) = \sum_{p \in P} C^p(u)$$

Assume that the contribution of u is a vector variable with some distribution on P, thus according to the theorems of large numbers, if the random samples $c_1, c_2, c_3, \ldots, c_N$ are selected using probability, the expected value can be estimated by average $\hat{c}$

$$E[C(u)] \approx \hat{c} = \frac{1}{N} \sum_{p=1}^{N} C^p(u)$$

Assume the variance of $C^p(u)$ is σ^2 and all the samples are independent and the variance of $\hat{c}$ is:

$$D^2[\hat{c}] = \frac{1}{N}\sum_{p=1}^{N}\sigma^2 = \frac{\sigma^2}{N}$$

Thus the deviation is

$$D[\hat{c}] = \frac{\sigma}{\sqrt{N}}$$

The error can be reduced when more sampling are taking into account and samples size can be roughly estimated. The method, for any group size, ignores the statistical differences between these samplings and only uses a part of them to measure the group members' contributions, i.e. all the item-pair subspaces, to obtain an approximate solution, which makes σ is in proportion to $D[\hat{c}]$.

Intuitively, a member is not representative when his/her preference is highly correlated with and can be represented by the preferences of others. Taking all the members' profiles as data in high dimensional vectors, a finite set of vertices can be selected to define a convex hull and all the other data in the convex hull can be linearly represented. These vertices, i.e. preferences, are more representative than the preferences in the convex hull. This is the motivation for the proposal of the "contribution score" concept to depict the representativeness degree of a member. Taking this point of view, the representativeness measuring problem in the work is converted into the identification of the set of preferences on hull vertices.

Once have obtained the partial rating matrix M_s, SNMF (Koren, Bell & Volinsky 2009; Rennie & Srebro 2005) is employed in the CS model to identify the vertices. Compared to traditional matrix factorization techniques that rely heavily on initial values, a stable solution and representativeness degrees can be guaranteed by the SNMF. The SNMF on M_s is defined as

$$M_s = WM_s^B \tag{5-5}$$

where ${M_s}^B$ is the basic matrix, W is the weight matrix. The basic matrix ${M_s}^B$ consists of a number of rows from M_s which can be used to recover M_s more accurately than

other rows. Members with profiles in the basic matrix can be seen as representative members. The representative member set for U_s is denoted as ${U_s}^B$.

CS measures the representative preferences of each member according to Equation (5-6). For each member $u \in U_s$, the CS of u is defined as

$$CS_u^{I_s} = \begin{cases} 1, & u \in U_s^B; \\ 0, & otherwise. \end{cases} \tag{5-6}$$

5.3.2 GLOBAL MEMBER CONTRIBUTION IN ALL SAMPLES

Note that each sampling selects a portion of items and only selects members who have no missing ratings. Members involved in one sampling may not cover the group. The representativeness degree of a group member over the whole item space can be approximated by aggregating the CS results of all the samplings in which he/she is involved.

Theoretically, all the possible samplings should be considered to evaluate each member's contribution accurately. However, in practice, it is impossible to complete this task within the time limit, considering the infinite projection probability for a rating matrix of a group. To address this issue, a portion of the projections for any group size are selected out as the samplings, i.e. all the item-pair subspaces, to measure CS. For the whole item space I, the ith sampling is $S_i \;\; i \in [1 \ldots d]$. The items and members involved in S_i are I_{S_i} and U_{S_i} respectively, and the CS of each member u, irrespective of whether or not it is involved in G on S_i, can be represented as

$$CS_u^{S_i} = \begin{cases} CS_u^{S_i}, & u \in U_{S_i} \\ 0, & otherwise \end{cases} \tag{5-7}$$

The contribution of member u, $u \in U_{S_i}$, is 1 or 0 depending on whether u can be identified as representative members, and is always 0 when u is not involved. For each group member $u \in G$, overall CS for member u is obtained by aggregating her all CS on all I_{S_i} into a single CS.

$$CS_u = \sum_{i=1}^{d} \frac{2}{n} CS_u^{S_i} \tag{5-8}$$

(Zhou, Bian & Tao 2013), proposed an efficient method to resolve the SNMF problem. Equation (5-9) is the specific SNMF method employed to compute the representative members by the maximum and minimum angles between the 2D random projections of the *n* data points and the horizontal axis in a 2D plane which match the item-pair projection,

$$CS_u^{s_i} = \begin{cases} 1, & u = \arg\max_i (\arctan 2(u\,\beta_{sub_{j1}}^T, u\,\beta_{sub_{j2}}^T)) \\ 0, & otherwise \end{cases}, \tag{5-9}$$

where $\beta_{sub_{j1}}$ and $\beta_{sub_{j2}}$ are two unit vectors of the plane and

$$\arctan 2(y,x) = \begin{cases} \arctan(\frac{y}{x}), & x \geq 0 \\ \arctan(\frac{y}{x}) + \pi, & y \geq 0, x < 0 \\ \arctan(\frac{y}{x}) - \pi, & y < 0, x < 0 \\ \frac{\pi}{2}, & y > 0, x = 0 \\ -\frac{\pi}{2}, & y < 0, x = 0 \end{cases} \tag{5-10}$$

Let I_G be all the items that have been rated at least once by a group member. The number of all possible subspaces is $C_{|I_G|}^2$. In practical systems, users tend to give a small number of ratings and $C_{|I_G|}^2$ should be acceptable. Combined with Equation (5-11), the final CS is defined as

$$CS_u = \sum_{i=1}^{C_{|I_G|}^2} \frac{2}{n} CS_u^{s_i} \tag{5-11}$$

5.3.3 GROUP MODELING USING CONTRIBUTIONS SCORES

Once the CS for each group member has been obtained, it is normalized for further group profile calculation.

$$\omega_u = \frac{CS_u}{\sum_{u \in G} CS_u} \tag{5-12}$$

The group profile $Profile_G$ is represented as a vector and every dimension represents an item only when it has been rated by group members. Let I_G be all the items that have been rated by group members,$item_i \in I_G$ $i \in [1..|I_G|]$, $Profile_G = [r_{G,item_1}, r_{G,item_2}, \dots, r_{G,item_{|I_G|}}]$. For each $item_i$, let U_{item_i} be all the members who have rated $item_i$, then the group rating for $item_i$ is computed as follows:

$$r_{g,item_i} = \sum_{u \in U_{item_i}} \omega_u r_{u,item_i} \tag{5-13}$$

In Algorithm 5.1, a summarization is presented to show how to compute the group profile and give a detailed description of the CS calculation. A numerical case of Algorithm 5.1 is also given in Example 5.1.

Algorithm 5.1 Group profile generator algorithm

Input: rating matrix *M*, Item set *I*, group *G*

Output: group profile of *G*

[Begin]

Set all the members' contribution to zero

$CS_u = 0, u \in G$

For *sub* =1 to $C^2_{|I_G|}$ do

Step 1:

Get all the members involved in U_{sub}

Get items involved in I_{sub}

$$CS_u^{I_{sub}} = \begin{cases} 1, & u = \arg\max_i (\arctan 2(u\,\beta^T_{sub_{j1}}, u\,\beta^T_{sub_{j2}})) \\ 0, & otherwise \end{cases}$$

Step 2:

For Each member in U_{sub} , aggregate CS_u

$$CS_u = CS_u + \frac{|I_{sub}|}{|I_G|} CS_u^{I_{sub}}$$

End for

Step 3:

Normalise $CS_{u \in G}$ to $\omega_{u \in G}$

For item i in I_G

Group rating $r_{g,i} = \sum_{u \in U_i} \omega_i r_{u,i}$

End for

Group profile $Profile_G = [r_{G,i}], i \in I_G$

[End]

Example 5.1: Let $G = \{User_1, User_2, User_3, User_4\}$, $I_G = \{Item_1, Item_2, Item_3, Item_4\}$, as shown in Table 1.

	$Item_1$	$Item_2$	$Item_3$	$Item_4$
$User_1$	5	4	4	?
$User_2$	4	4	?	?
$User_3$	?	?	5	2
$User_4$	3	1	?	3

All the item subspaces are sampled: $\{I_{1,2}, I_{1,3}, I_{1,4}, I_{2,3}, I_{2,4}, I_{3,4}\}$. After calculating CS in the samplings, i.e. step 1, then have $MCS_{user4}^{I_{1,2}}$=1, $MCS_{user1}^{I_{1,3}}$=1, $MCS_{user4}^{I_{1,4}}$=1, $MCS_{user1}^{I_{2,3}}$=1, $MCS_{user4}^{I_{2,4}}$=1and $MCS_{user3}^{I_{3,4}}$=1.

In $I_{1,2}$, only $user4$'s contribution is 1 when $user1$ and $user2$ are 0, shows that not all the members involving in an sampling means they are representative in it.

In step 2, aggregate CS from all the samplings.

$$MCS_{user1} = \frac{2}{4}\Big(MCS_{user1}^{I_{1,2}} + MCS_{user1}^{I_{1,3}} + MCS_{user1}^{I_{1,4}} + MCS_{user1}^{I_{2,3}} + MCS_{user1}^{I_{2,4}} + MCS_{user1}^{I_{3,4}}\Big) = \frac{2}{4}(0+1+0+1+0+0) = 1$$

$$MCS_{user2} = \frac{2}{4}(0+0+0+0+0+0) = 0$$

$$MCS_{user3} = \frac{2}{4}(0+0+0+0+0+1) = \frac{1}{2}$$

$$MCS_{user4} = \frac{2}{4}(1+0+1+0+1+0) = \frac{3}{2}$$

In step 3, contribution is normalized and the group profile is then modeled taking member contributions into consideration.

$$\omega_{user1} = \frac{MCS_{user1}}{MCS_{user1} + MCS_{user2} + MCS_{user3} + MCS_{user4}} = \frac{1}{1 + \frac{1}{2} + \frac{3}{2}} = 1/3$$

$$\omega_{user2} = \frac{0}{1 + \frac{1}{2} + \frac{3}{2}} = 0$$

$$\omega_{user3} = \frac{1/2}{1 + \frac{1}{2} + \frac{3}{2}} = 1/6$$

$$\omega_{user4} = \frac{3/2}{1 + \frac{1}{2} + \frac{3}{2}} = 1/2$$

$$r_{G,item1} = \omega_{user1} r_{user1,item1} + \omega_{user2} r_{user2,item1} + \omega_{user4} r_{user4,item1}$$
$$= \frac{1}{3} \times 5 + 0 \times 4 + \frac{1}{2} \times 5 = 3.17$$

$$r_{G,item2} = \tfrac{1}{3} \times 4 + 0 \times 4 + \tfrac{1}{2} \times 1 = 1.83$$

$$r_{G,item3} = \frac{1}{3} \times 4 + \frac{1}{6} \times 5 = 2.17$$

$$r_{G,item4} = \frac{1}{6} \times 2 + \frac{1}{2} \times 3 = 0.83$$

The output group profile is $Profile_G = [3.17, 1.83, 2.17, 0.83]$.

The group profile is expressed as the group rating vector over the items that have been rated by group members. Once this has been generated, it is used in this phase to predict unknown group ratings. A similarity measure, PCC similarity, is adopted in the work to identify neighbours close to the group. To minimize the error caused by the fat tail, a Manhattan distance-based measure is applied to compute local average ratings for the group. By combining PCC similarities and local average ratings, a user-based individual collaborative filtering approach is applied to predict unknown group ratings.

5.3.4 UNKNOWN GROUP RATINGS CALCULATION USING LOCAL COLLABORATIVE FILTERING METHOD

Once the group profile, $profile_G$, has been obtained, it can be seen as a preference of a pseudo user g. It is then possible to compute the unknown group ratings using local collaborative filtering method.

Neighbours of group can be identified by calculating the similarities between the pseudo user g and non-member group users. In LCF method, PCC similarity, which has been widely used in a number of recommendation systems, is employed for the similarity computation. Let g be the pseudo user and u be a non-group system user. Let I_u be the item set that has been rated by u. The PCC similarity between g and u is computed based on their common ratings as follows:

$$Sim(g,u) = \frac{\sum_{i \in (I_g \cap I_u)} (r_{g,i} - \bar{r}_g)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in (I_g \cap I_u)} (r_{g,i} - \bar{r}_g)^2 \sum_{i \in (I_g \cap I_u)} (r_{u,i} - \bar{r})^2}}, \quad (5\text{-}14)$$

where $I_g \cap I_u$ is the set of common rated items by both g and u, $r_{g,i}$ and $r_{u,i}$ represent known ratings for item i, $\bar{r}_g$ is the average rating of g and $\bar{r}_u$ is the average rating of u. The similarity $Sim(g,u)$ between two users ranges from -1 to 1, where a large value indicates a higher similarity.

The adaptive average rating using in LCF method is introduced in Section 3.2.2. Let i be target item, $\bar{r}'_{g,i}$ be the local average rating for i of g. The prediction of $r_{g,i}$ is

$$r_{g,i} = \bar{r}'_{g,i} + \frac{\sum_{u \in Neighbors} (r_{u,i^*} - \bar{r}_u) \times Sim(g,u)}{\sum_{u \in Neighbors} |Sim(g,u)|} \quad (5\text{-}15)$$

The final recommendations are selected as the top-k items with the highest predictions.

Algorithm 5.2 summarises the steps to show how to generate group recommendations given the group profile, and give a detailed description of the local average rating computation.

Algorithm 5.2 Recommendations generator algorithm

Input: rating matrix M, group profile $Profile_G$, parameter for local average rating T

Output: Recommendations

[Begin]

Get the item set I_G that has been rated by members

For $i \notin I_G$ do

Step 1: similarities between the pseudo user g and non-member user u are computed by PCC similarity

$$Sim(g,u) = \frac{\sum_{i \in (I_g \cap I_u)} (r_{g,i} - \bar{r}_g)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in (I_g \cap I_u)} (r_{g,i} - \bar{r}_g)^2 \sum_{i \in (I_g \cap I_u)} (r_{u,i} - \bar{r})^2}}$$

Step 2: calculate local average rating

$Rel_{i,j}$ is mean normalized Manhattan distance over whole user space.

Local average rating $\overline{r_{G,i}}' = avg(r_{G,j})$ $if\ Rel_{i,j} \geq T$

Step 3: using user-based collaborative filtering to predict unknown group ratings

$$r_{G,i} = \overline{r_{G,i}}' + \frac{\sum_{u \in Neighbors} (r_{u,i} - \overline{r_u}) \times Sim(g,u)}{\sum_{u \in Neighbors} |Sim(g,u)|}$$

End for

Recommendations are selected out with higher group rating

[End]

5.4 EXPERIMENTS AND EVALUATION

In this section, an empirical study of the approach is presented on real datasets. The datasets are introduced and pre-processed in Section 5.4.1. The group generating protocol is described in Section 5.4.2. The metrics employed to evaluate the performance of the proposal are shown in Section 5.4.3and the experiments design is

shown in Section 5.4.4. The comparison between results from the proposal and baseline methods and discussion are presented in Section 5.4.5.

5.4.1 DATASETS AND PRE-PROCESSING

To the best of the knowledge, no benchmark datasets have been designed and implemented to assess the performance of group recommendations. For this reason, MovieLens datasets (http://www.grouplens.org) and the Jester dataset, which are benchmark datasets that can be employed to assess individual recommendation methods, are employed and develop offline experiments. MovieLens datasets contain integer ratings and tags applied to movies by users of an online recommender service and were collected by the GroupLens Research Project at the University of Minnesota. The 100K and 1M version of MovieLens datasets are used to evaluate performance. Another data set is Jester, which asks users to rate jokes. Jester dataset allow the users to provide real number ratings from -10 to 10. In experiments, only users who have rated between 15 and 35 jokes are selected to avoid the group profile covering all the items. The key statistics of these three sets are shown below:

Table 5-1. Features of three test data sets.

Dataset	User	Item	Rating	Sparsity	RatingRange
ML100K	943	1682	100,000	93.7%	1-5
ML1M	6040	3706	1,000,209	95.5%	1-5
Jester	24938	100	616912	75.3%	-10.0-10.0

The datasets are each split into two parts consisting of a training dataset and a test dataset. For items that have been rated by group members but will not be recommended to the group again, a small test set may cause fewer recommended items to be found in the test dataset, leading to poor quality evaluation. Therefore, 50% of the data are randomly selected for the training set and 50% for the test set.

Note that because these datasets are composed of ratings provided by individual users to assess individual recommender system performance, they contain no real group information. Therefore, a group generating protocol and appropriate metric are needed to evaluate the approach.

5.4.2 GROUP GENERATION PROTOCOL

Although the CS-GR method may handle all kind of groups, the experiments focus on random groups which may have a higher level of preference conflict. Two important features affect the nature of groups when they are generated: group size and internal member relevance. The larger a group is, the more difficult it is to model the group profile. Most previous group recommendation approaches have focused on relatively small groups, less than 10 in number, with which it is easy to achieve a compromise opinion. In the experiments, the group size is initially set as 5 and increases it each time by 5 until 30 is reached to assess the feasibility of the proposal for both small and large groups.

Apart from size, internal member relevance is another important feature that can affect the effectiveness of the recommendation approach. The preference conflict can be greater when a group is generated randomly, because in this case, members have no knowledge about other members. Therefore, the experiments are designed focusing random groups. In the experiments, groups are formed by randomly selecting users who have no explicit shared preference relevance, such as people traveling on the same airplane.

5.4.3 METRICS

To evaluate the approach based on a list of recommendations, both Normalized Discounted Cumulative Gain (nDCG) and F measure are adopted. Widely used in information retrieval, nDCG has been adopted by many researchers to measure the performance of group recommender algorithms. It attempts to measure the rank performance between predicted group ratings and real values. *F* measure, which is widely employed in individual RSs, is also employed to evaluate accuracy by considering missing labelling data.

nDCG is more appropriate than RMSE and MAE because it not only considers accuracy but also takes recommendation order into account. Let $l_1 \cdots l_k$ be the recommendation list obtained and u be a user. DCG is defined as

$$DCG_{u,k} = r_{u,l_1} + \sum_{i=2}^{k} \frac{r_{u,l_i}}{log_2(i)} \tag{5-16}$$

and the corresponding nDCG is defined as

$$nDCG_{u,k} = \frac{DCG_{u,k}}{IDCG_{u,k}} \tag{5-17}$$

where IDCG is the optimal possible gain value for user *u* where recommendations are re-ordered in descending order based on their relevant scores in the obtained list. DCG defined in Equation (5-17) measures the accuracy of a list of recommendations that is ordered by score (predicted rating). An item's score will be penalized for logarithmically proportional to the position of each item in the list. nDCG can then be used to measure the performance of the recommendation list. Clearly, given that nDCG ranges from 0 to 1, the higher the nDCG obtained, the better recommendations have been made.

As a result of the lack of ground truth required to assess the recommendations generated for the group, it will calculate the average nDCG value for each of the group members. In the experiments, nDCG is computed on all the items in the test set of the user, sorted according to the ranking computed by the recommendation algorithms. In other words, nDCG is computed on the projection of the recommendation list on the test set of the users. For example, imagine that $rec = \{item_1, item_2, \dots, item_6\}$ is an ordered list of recommendations for a group *G*. User *u* is a member of *G* and the corresponding $r_{u,item_1}$, $r_{u,item_3}$ and $r_{u,item_5}$ in the test set. Hence the nDCG score for *u* is computed only by $\{item_1, item_3, item_5\}$.

F measure is used to evaluate the missing prediction and group rating classification for members. For example, let the threshold for members to accept one item be set to 3. To predict a group rating for an item as 4 if one member rates this item as 2, this item may be recommended whereas should be dropped. In the experiments, "3" is set as the threshold to label whether accept one item for both datasets rating ranging from 1 to 5. True positive (TP) for an item is when all the member ratings for that item are higher than 3 (ignoring members whose rating for this item is unknown) and the group prediction for the item is higher than 3. False negative (FN) for an item is when all the member ratings for that item are higher than

3 (ignoring members whose rating for this item is unknown) and the group prediction for the item is lower than 3. False positive (FP) for an item is when some member ratings for that item are lower than 3 (ignoring members whose rating for this item is unknown) and group prediction for the item is higher than 3. *F* measure is shown in Equation (5-18). *F* ranges from 0 to 1, and similar to nDCG, the higher the F obtained, the more accurate is the group rating prediction.

$$F = 2\frac{TP}{(TP+FN)}\frac{TP}{(TP+FP)} \tag{5-18}$$

5.4.4 EXPERIMENT DESIGN

To measure the improvement of the CS-GR approach, several successful and popular group recommendation approaches are implemented as baselines. Below are the labels and descriptions using to denote each of these baselines.

LM: the group profile is generated using the least misery strategy and basic user-based CF is used to generate group recommendations. The group ratings in the group profile are calculated according to Equation (5-19).

$$rating_{G,i} = Min(rating_{u,i}) \tag{5-19}$$

AVG: the group profile is generated using the average strategy and basic user-based CF is used to generate group recommendations. The group ratings in the group profile are calculated according to Equation (5-20).

$$rating_{G,i} = \frac{1}{|G|}\sum_{u\in G} rating_{u,i} \tag{5-20}$$

AM: the group profile is generated using the average without misery strategy. This method aims to find a compromise between LM and AVG. A threshold is used to filter out items that will cause disappointment for members who have ratings lower than a predefined threshold. In the experiment, this threshold is set to 2. The group ratings in the group profile are calculated according to Equation (5-21). After building the group profile, basic user-based CF is used to generate group recommendations.

$$rating_{G,i} = \frac{1}{|G|}\sum_{u\in G}\{rating_{u,i} \mid \forall\, rating_{u,i} > 2\} \tag{5-21}$$

CS: the group profile is generated using weighted individual preference, and weights are computed by CS for each member. Group recommendations are predicted using global average rating.

CS-GR (or called CS-LCF): the group profile is generated using weighted individual preference, and weights are computed by CS for each member. Group recommendations are predicted using the LCF model. The parameter for identifying neighbour items is set to 0.2.

For each specific group size, 1000 groups are randomly generated, and the average metrics over 1000 groups give the final result. For instance, for a 10-member group, 50% of data are randomly selected as the training data and the rest as the test data. Ten members are randomly selected from users in the test set to form the group, for avoiding a situation in which the selected member's ratings are all in the training set and cannot be measured over the test data. Calculating the metrics and repeat this process 1000 times to obtain the average metrics.

5.4.5 RESULTS AND DISCUSSION

Figure 5-3 to Figure 5-5 show the nDCG results obtained by LM, AVG, AM and the approach. As shown in figures it is clear that the approach, whether local average rating is used or not, consistently outperforms the baseline approaches. On the ML100K dataset, the LM, AVG and AM approaches are close when group size is relatively small, and AM is the best approach when group size increases. CS is 2.4% better than LM when group size is 5. When group size is 30, CS is 2.5% better than AVG, 4% better than LM, and 4.8% better than AM. The CS-GR results on various sized groups show that local average rating significantly improves performance. The CS-GR is 3.2% better than AVG and about 5% better than LM and AM when group size is 5. When group size is 30, the CS-GR is 3.5% better than LM, 5.7% better than AM, and 7.6% better than AM.

On the ML1M dataset, by contrast, CS is 1.4% better than AVG when group size is 5. When group size is 30, CS is 1.5% better than AVG, 1.6% better than AM, and 2.2% better than LM. The CS-GR results on various sized groups show that local average rating significantly improves performance. The CS-GR is 2.7% better than

AVG and AM, and 3.5% better than LM when group size is 5. When group size is 30, CS-LCF is 2% better than AVG, 2.1% better than AM and 2.7% better than LM.

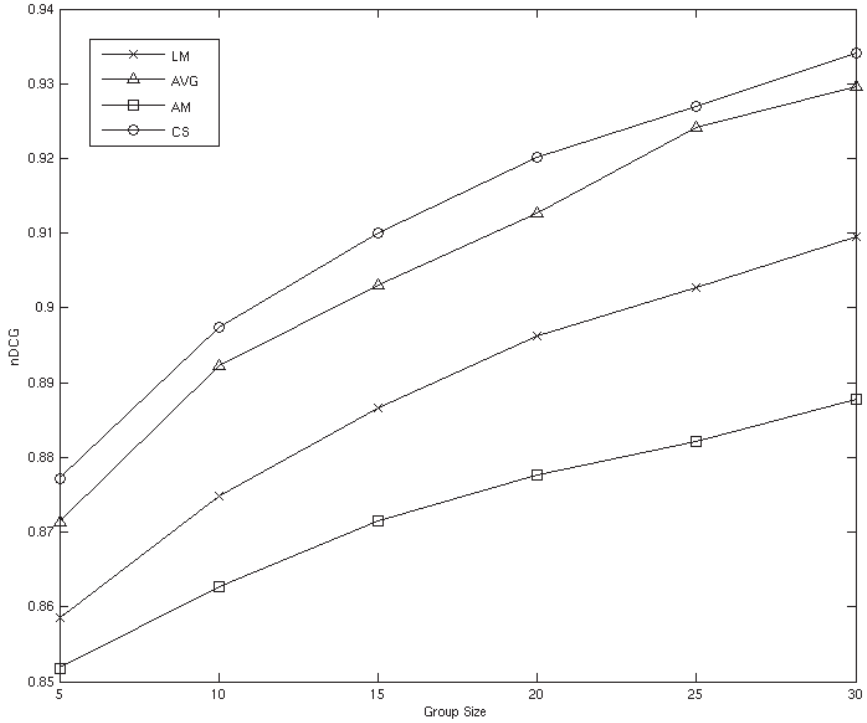


Figure 5-3. nDCG result of MovieLens100K.

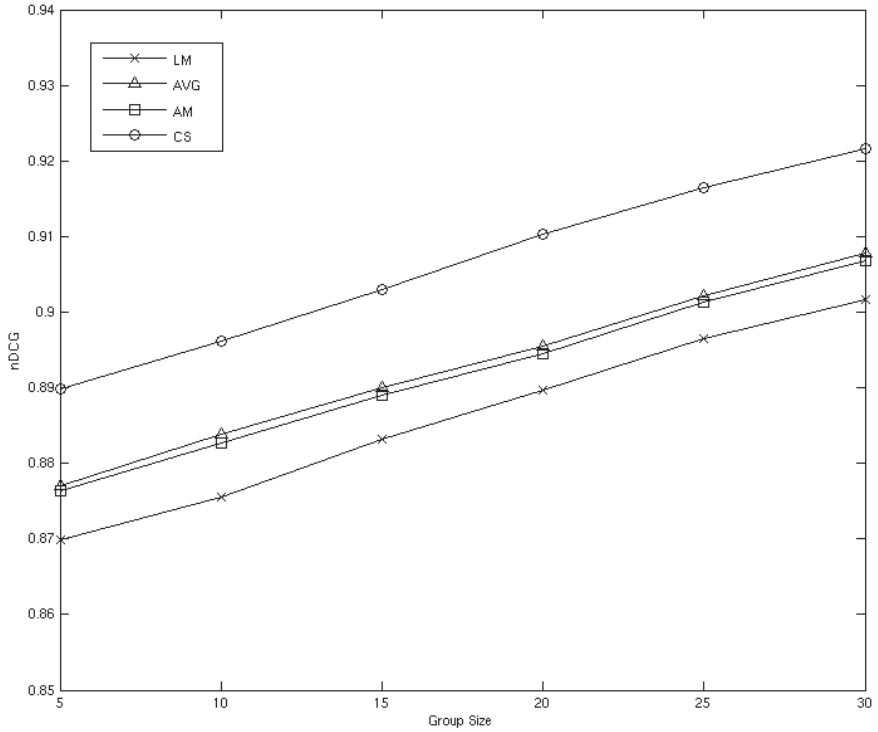


Figure 5-4. nDCG result of MovieLens1M.

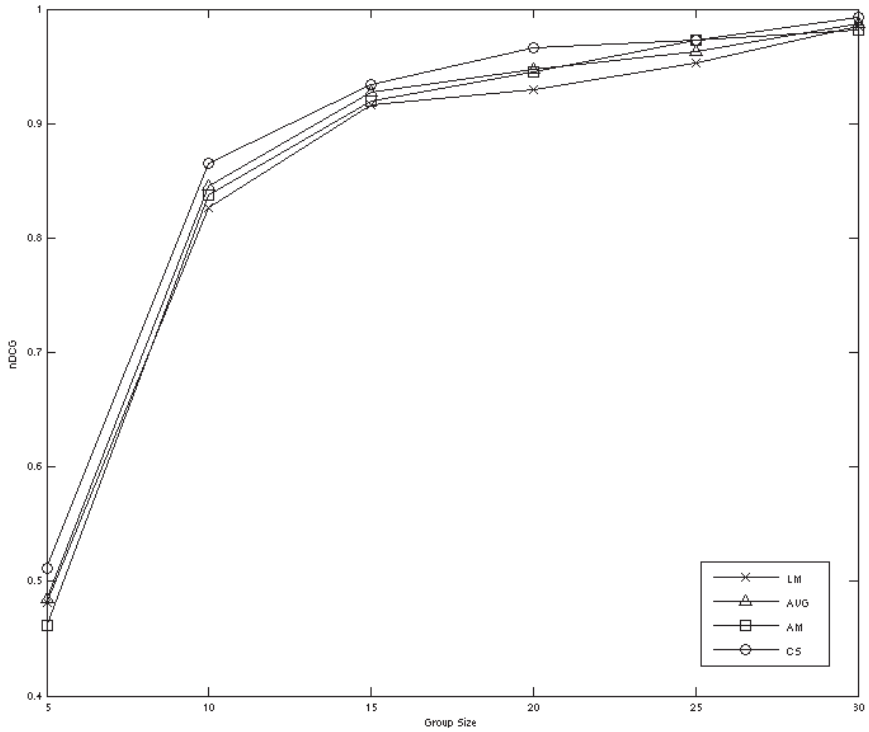


Figure 5-5. nDCG result of Jester.

Even on a sparser dataset, ML1M, the CS and the CS-GR approaches clearly make better recommendations, and the approach using the CS-GR outperforms the approach using CS only. An interesting fact is that for the AVG strategy on ML1M, performance decreases when the group size becomes large. A reasonable explanation for this is that when there is insufficient information, it is difficult to find a fair solution for all the members.

On the Jester dataset, the approaches are better than AVG, AM and LM. When group size is 5, the CS is 5% and CS-GR is 9 % better than AVG, while LM is close to AVG and AM is worse than AVG. When group size increases, which means the group profile covers more items and unknown ratings become less, nDCG results are close to 1 and the performance of CS and MV-GR decreases. CS is 0.9% better than AVG and CS-GR is 1.1% better than AVG when group size is 30.

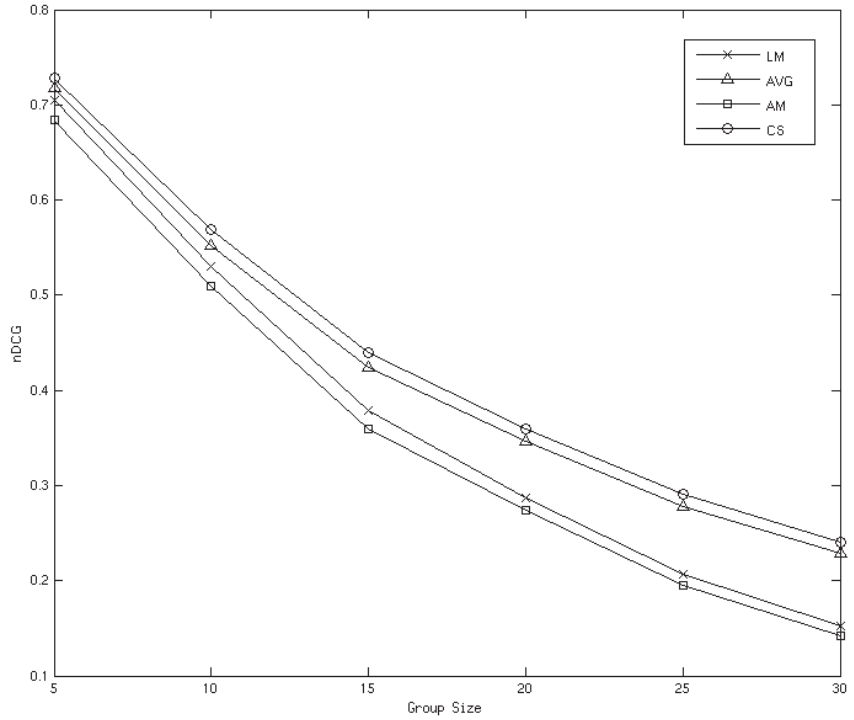


Figure 5-6. F result of MovieLens100K.

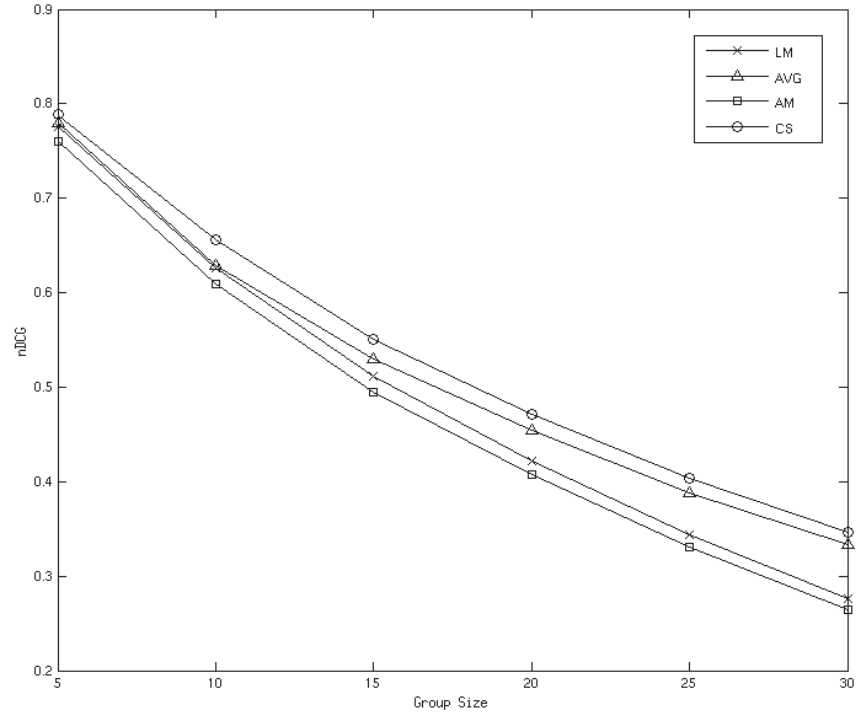


Figure 5-7. F result of MovieLens1M.

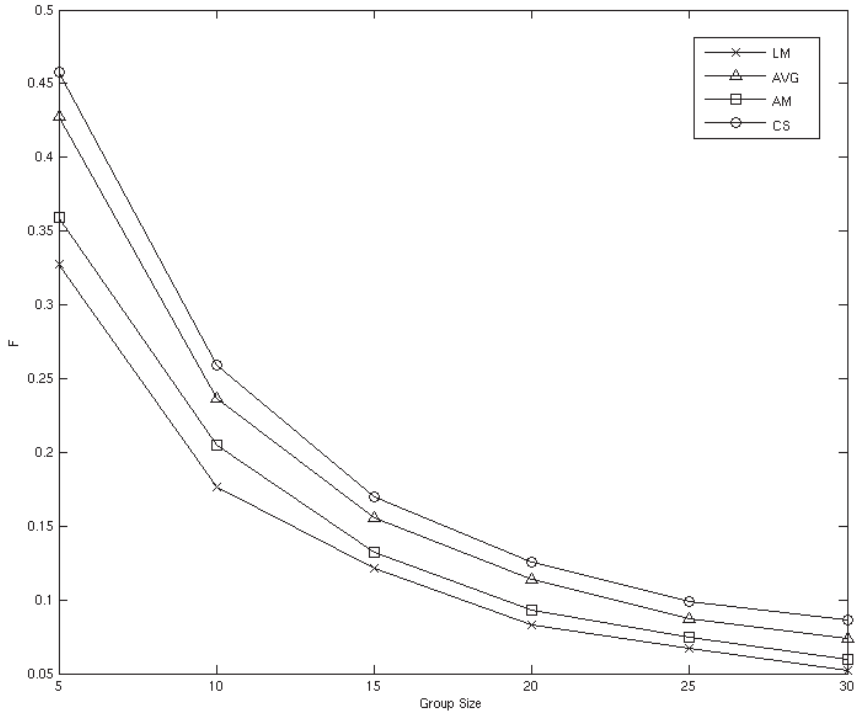


Figure 5-8. F scores F result of Jester.

Figure 5-6 to 5-8 show the F results obtained by LM, AVG, AM and the approach. As shown in figures, AVG is the best approach when the group size is very small, i.e. 5. This is mainly because the performance would be better if the approach could correctly predict the majority opinions. When the group size becomes larger, the approach decreases more slowly than LM, AVG and AM. On the ML100K dataset with a group size of 30, CS is 8% better than AVG, which is the best approach in LM, AVG and AM, and on the ML1M dataset, CS is 6% better than AVG, which is also the best approach in LM, AVG and AM. On the Jester dataset with a group size of 30, CS is 10% better than AVG, which is also the best approach in LM, AVG and AM.

Because a threshold is used in the LCF model to estimate the target item related average rating, several experiments are also performed to examine the sensitivity of performance with this threshold, in which they varied the value of the threshold to generate the predicted rating. The nDCG and *F* results of the proposal were also compared with others.

Figure 5-9 to 5-12 show the results of using CS alone, and 0.2, 0.3 and 0.4 are employed in the CS-LCF model. From Figure 5-9 and Figure 5-10, it can clearly be

observed that the F results are not greatly affected when different parameters are used in the LCF model. The results show that the CS-GR approaches outperform CS and there are no big differences when the parameter is not strictly set. It also can be noticed that the different parameters affect the performance considerably when the group size is relatively small. This demonstrates that for random groups, local average rating tends to be an average rating when the group size is large.

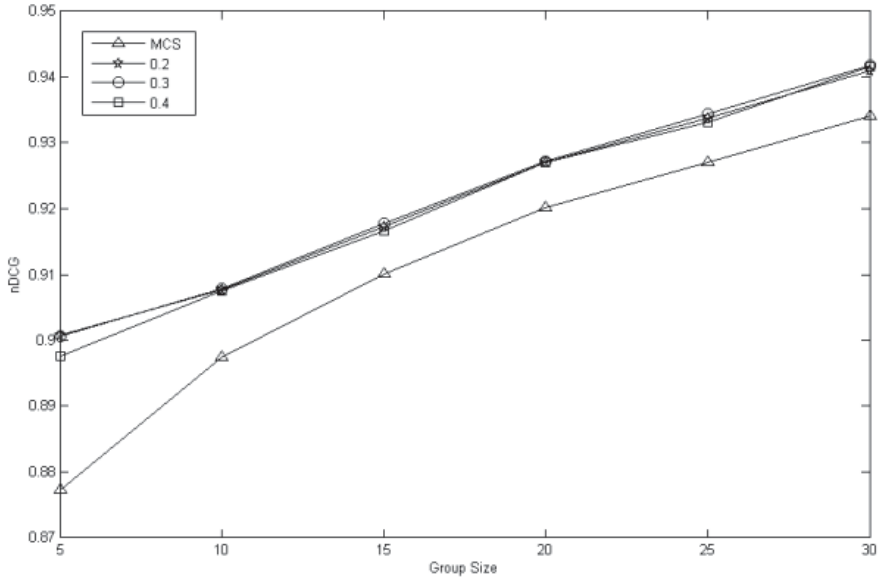


Figure 5-9. nDCG results when using the CS model alone and when combining the LCF model using 0.2, 0.3 and 0.4 thresholds to produce local average ratings on 100K.

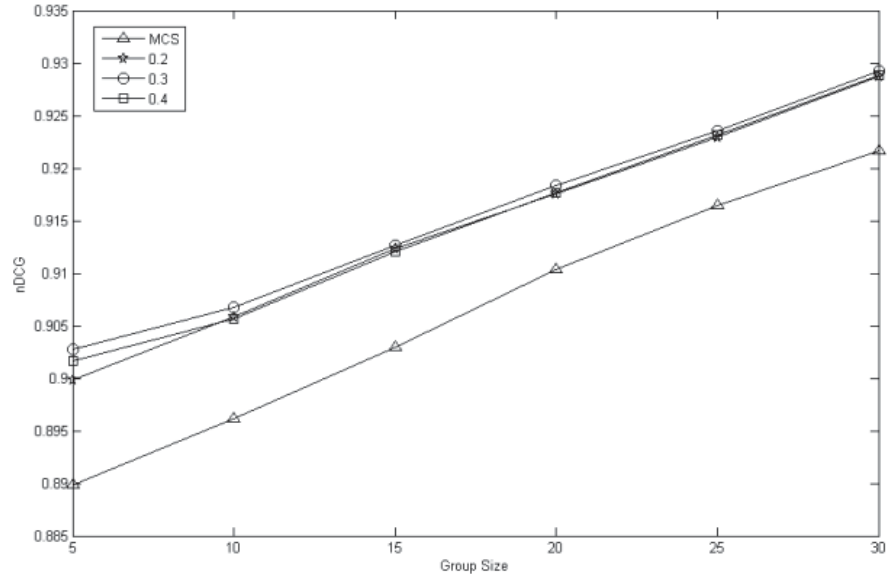


Figure 5-10. nDCG results when using the CS model alone and when combining the LCF model using 0.2, 0.3 and 0.4 thresholds to produce local average ratings on 1M.

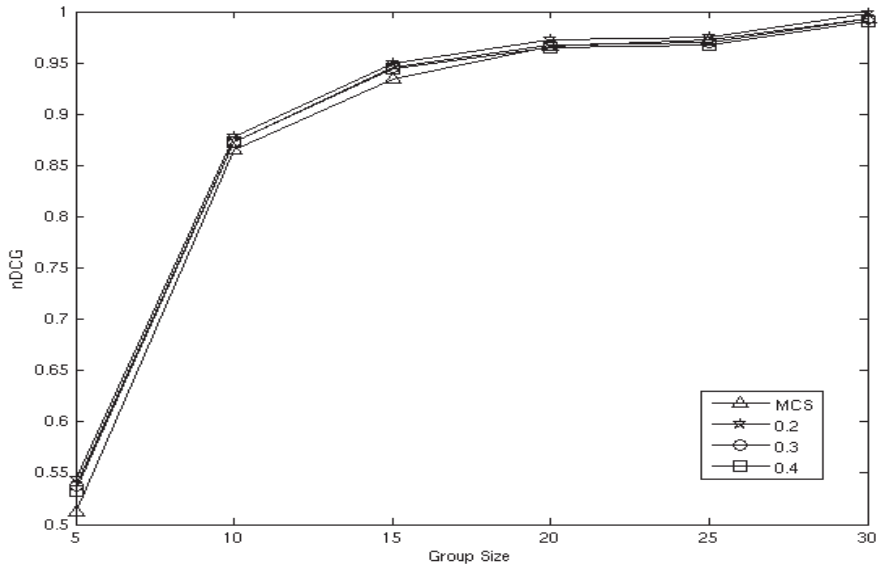


Figure 5-11 nDCG results when using the CS model alone and when combining the LCF model using 0.2, 0.3 and 0.4 thresholds to produce local average ratings on Jester.

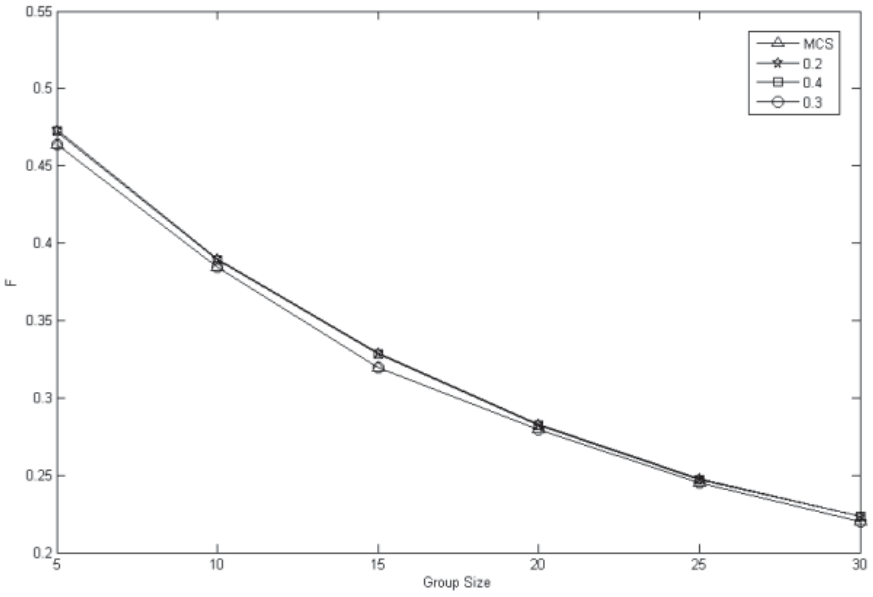


Figure 5-12. F results when using the CS model alone and when combining the LCF model using 0.2, 0.3 and 0.4 thresholds to produce local average ratings on 100K.

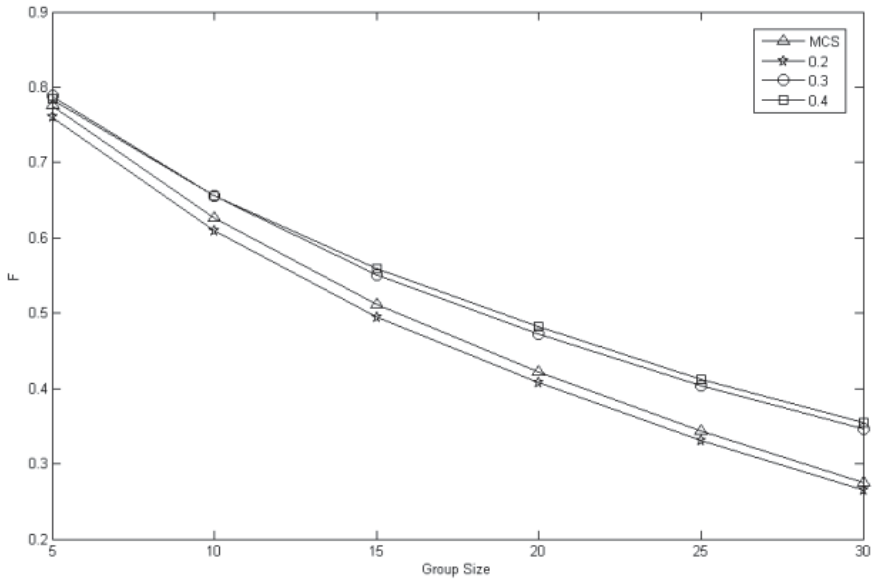


Figure 5-13. F results when using the CS model alone and when combining the LCF model using 0.2, 0.3 and 0.4 thresholds to produce local average ratings on 1M.

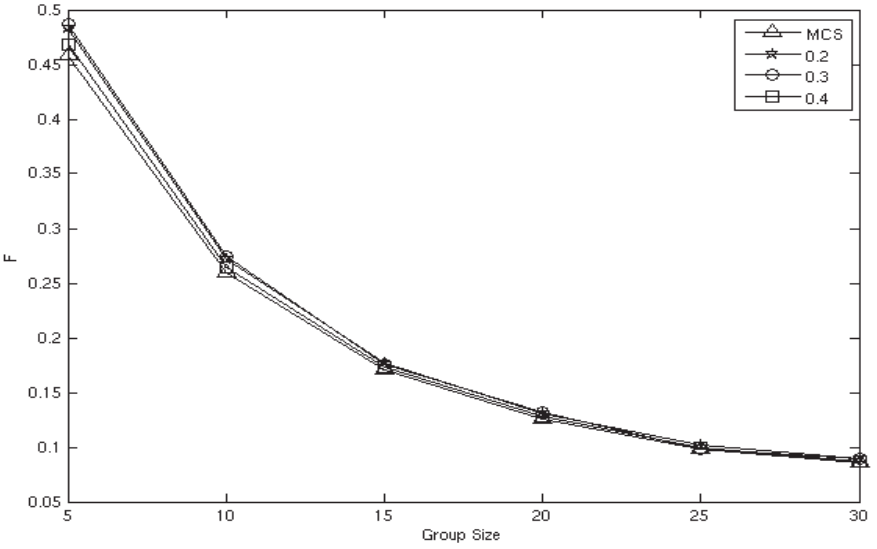


Figure 5-14. F results when using the CS model alone and when combining the LCF model using 0.2, 0.3 and 0.4 thresholds to produce local average ratings on Jester.

5.5 GROTO: A CONTRIBUTION SCORE-BASED GROUP RECOMMENDER SYSTEM FOR TOURISM

Group tourism (GroTo) is a web-based group recommender system that aims to provide personalized recommendation activities for web-based tourist groups in Australia. In this system, the activities are classified and labeled in advance. There are six categories of tourism activity in GroTo: Nature, Sports, Arts, Aboriginal, Attractions and Social. Each category contains detailed activities for users to rate. For example, going to the beach, or visiting state parks and farms, can be rated by users in the Nature category.

The GroTo system has three components: a system interface, a recommender engine and a data server, as shown in Figure 5-15.

- *The system interface* collects information from users who can actively specify their preferences for various tourist activities via web-based interfaces provided by the system. Users' context information can also be passively collected from mobile devices. Note that the preferences and historical visiting information are transformed into structural data, e.g. XML, in the user data collector module. Users' data are passed to the recommender engine for further processing.
- *The recommender engine* parses structural the information of users, and user preferences are transformed into rating vectors in the user data server module. Additionally, every historical location that can be found and labeled in the system is transformed into ratings in the user data server module. Negative UGC is not transformed into ratings but will be used as criteria for pre-selection. The activities filter generates available activities by excluding all the activities that clearly do not appeal to members. The user contribution server models a group profile to describe overall group preferences using the CS model. The group profile and negative list are given to the recommender server to filter appropriate activities to recommend to the group.

- *The data server* is responsible for recording data from the system and individual information, preferences and UGC. It is important to point out that, except for individual information and UGC, the group can be stored as a case for future recommendation. A group can be identified by its members' reason for getting together, such as holiday, conference, business or education. Recommendations can be precisely made to future groups by using a group filter that can find similar cases in the database.

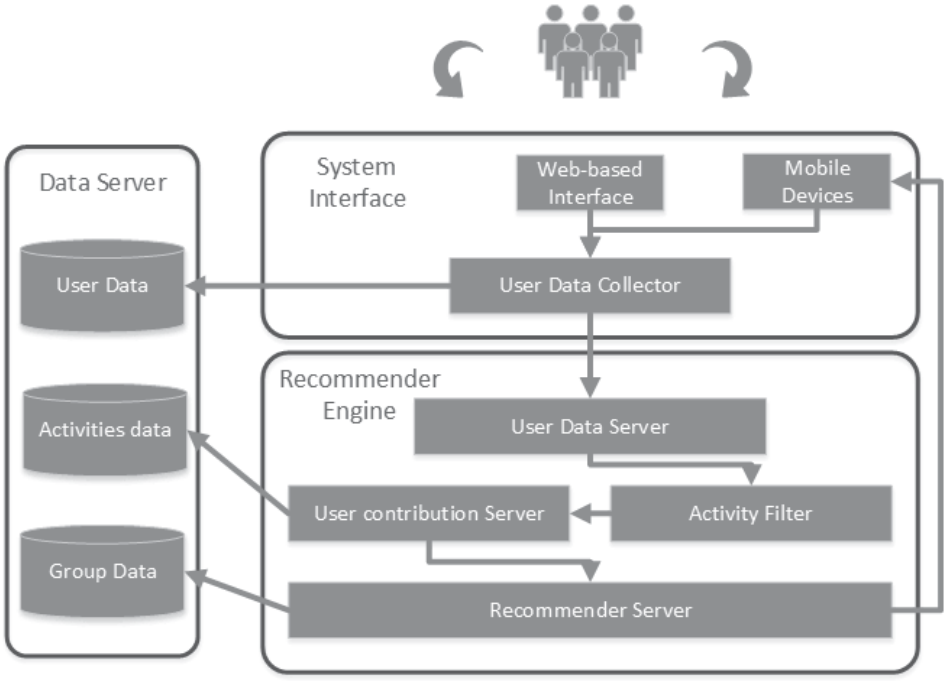


Figure 5-15. Architecture of the tourism recommender system GroTo.

An example is given in which only a selection of activities in GroTo is considered. A group is formed by six members who each nominate their preferences via the system interface. Their inputs are shown in Table 5-2. According to the proposal, their weightings are [0. 1216, 0. 1622, 0. 2162, 0.1622, 0.1622, 0.1757]. According to Equation (5-13), it can be obtained that the group profile, is [$r_{g,beach} = 3$, $r_{g,national\ /\ state\ parks} = 1.62$, $r_{g,farms} = 1.34$, $r_{g,diving} = 2.23$, $r_{g,snow\ sports} = 1.96$, $r_{g,cycling} = 1.71$].

The known ratings are shown in Table 5-3. UBC is used to predict the unknown group ratings for whale/dolphin watching, botanic gardens, fishing, surfing and golf, with the results 2.31, 1.73, 1.89, 2.56 and 1 respectively. If three best activities are recommended to the group, they are surfing, whale/dolphin watching and fishing.

Since the GroTo system's interface is under final development and testing, only the results are shown in these tables. A detailed report on the GroTo system will be presented in another paper.

The results of this example show that the proposal can accurately aggregate individual preferences and produce appropriate recommendations for group.

5.6 SUMMARY

In this chapter, a new group recommendation approach is proposed for modeling group profiles by considering all member contributions to the group's activities. A CS model is also proposed to measure the contribution of each group member in which, by partitioning the item space, members' opinions can be analysed using the SNMF technique. In addition, the local collaborative filtering method has been combined to alleviate the biased rating problem by adaptively calculating the average rating related to the target item when predicting unknown group ratings. Using these two models, the proposal can handle a high level of compromise in the group profile and exclude unnecessary information when generating predictions of user preferences.

The experiments were set up on two popular public datasets, and approach results are compared with three popular approaches are compared in the field of group recommendation. The results show the high effectiveness of the CS-LCF approach.

This study not only has theoretical significance but also potentially has high practical application. Many online services, such as movie or tourism recommendation sites and other websites, could adopt the approach.

The future study will include the extension of the proposed approach to select representative samplings instead of random samplings when sub-space differences are taken into consideration. A possible future improvement is to mathematically define a function to describe the degree of contribution divergence, and to incorporate alternative models when the function has a higher value.

Table 5-2. Ratings of group members on the activities Nature and Sport: each row represents a member.

Nature					Sports					
beach	national / state parks	whale / dolphin watching	botanic gardens	farms	fishing	diving	surfing	snow sports	golf	cycling
5				3		4				2
	4			4						3
4						5		2		
4	1					3				5
	5			2				4		
5						1		5		1

Table 5-3. Observed ratings of non-member users for the activities Nature and Sport

UserID	beach	national / state parks	whale / dolphin watching	botanic gardens	farms	fishing	diving	surfing	snow sports	golf	cycling
1	5		4			2	5	4	2		
2		3		4							
3						5	4			5	
4		4		4				3			4
5	2		1								
6	4	3				4				1	
7		5		4							
8							5				3
9	4		4			1					

CHAPTER 6

HIERARCHY VISUALIZATION FOR GROUP RECOMMENDER SYSTEMS

6.1 INTRODUCTION

Although many group recommendation methods have been developed to provide personalized information for users, many users have trouble interpreting and/or trusting the effectivity and reliability of the results. Unfortunately, most GRSs (Kazienko, Musial & Kajdanowicz 2011; Mao et al. 2015; Nanopoulos 2011; Wang, Zhang & Lu 2015) focus on the generation of recommendation lists, but provide no reasoning for why the lists have been selected. This is a natural limitation of GRSs, and is mainly due to an over concern for theoretical accuracy. As a result, many recommendation procedures have become a black box. However, incorporating explanations into RSs (Herlocker, Konstan & Riedl 2000) is beginning to be studied, and (Tintarev & Masthoff 2007) provides a summarized review of some of these approaches.

Displaying the results of RSs as graphs is very common. Knijnenburg et al. in (Knijnenburg et al. 2012) claims visualization presents data and its relationships in an efficient and inherently human way. It makes information far easier to understand, and provides an effective means of allowing users to interact with systems. Not surprisingly, visualization techniques have been used by many researchers to help users understand RSs, and many combine a number of different graph layouts for more convincing systems, such as references (Baltrunas, Makcinskas & Ricci 2010;

Bederson, Shneiderman & Wattenberg 2002; Bogdanov et al. 2013; Gavalas et al. 2014; Verbert et al. 2013) in the music, tour, academic and group recommendation domains. Yet despite these efforts, providing a visual explanation for GRSs is still challenging. First, most methods are proposed for individual RSs but are not appropriate for group recommendation methods because they include more complex data, such as group and group profile than individual RSs. Secondly, GRSs visualizations typically convey the overall system, and not the influence of individual members within the group. In (Castro et al. 2015), the group consensus and individual member recommendations are mapped to 2D plane using SOM clustering, but understanding the relationships between individual members and the final results is difficult.

The trackable hierarchy method addresses these problems because it supports system explanation and detailed individual influence data (i.e. group ratings, neighbour similarities and predictions). First, a hierarchy graph-based model is developed to build a higher level of abstraction of the system explanation. The model summarizes all involved entities and graphs them as nodes. The edges of the graph show inherited information. Thus the nodes and the edges combined demonstrate the entire recommender process. The method also illustrates the influence of data from different members for each node as a pie chart for individual members to track their influence within the system. The individual influences for each member are detailed in each node using a pie chart, providing some insight with which to gauge the quality of the recommendations.

Furthermore, the fundamental difference among group modeling methods is that the group rating is determined by which member(s). A group rating of an item is calculated by combining all the individual ratings of members specified on it. Many group modeling methods are proposed and generate different combinations. For example, a group rating is defined by the mean of all the ratings when using average modeling method; it is the lowest rating when using least misery modeling method. Therefore, when abstract method-independent components, i.e. member nodes, group rating nodes and edges between them, from these trivial calculation details, the approach can easily adapt to other modeling methods by simply adjusting the edges.

This method can also be applied to visualize individual RSs, by simply using a ‘single member’ group.

The rest of this chapter is organised as follows. Section 6.2 introduces a trackable hierarchy visualization method and corresponding components for GRSs. The method has been implemented as a web-based system based on MovieLens dataset to test the feasibility. The detailed description and discussion of the method are provided in Section 6.3. An extension supporting individual recommender system, SBS system, is developed. The introduction of the extension is presented in Section 6.4. Finally, the summary is given in Section 6.5.

6.2 HIERARCHY VISUALIZATION METHOD FOR GROUP RECOMMENDER SYSTEMS

6.2.1 LAYOUT

There are three main procedures in pseudo user-based methods. The first procedure models a pseudo user with a group profile. The pseudo user is modeled based on group items (i.e. items that have been rated by members). For each group item, group rating for it is calculated by aggregating all the ratings specified on it and the aggregating is defined by group modeling method. The pseudo user then becomes the active user for next two phases. The second procedure identifies the neighbours that have similar preferences to the pseudo user, using a similarity calculation method – typically cosine or Pearson’s correlation. The last procedure predicts unknown item rankings and recommends the items with the highest rankings. The method employs a hierarchy graph to provide a meaningful abstraction of contribution score-based group recommendation method (CS-GR).

The entities (i.e. users, items and ratings) involved in the CS-GR procedures are represented as nodes, and their position and size are determined by numerical features such as rating and similarity. There are four types of entities in the CS-GR method: group members, group profile, neighbours and recommendations. Each type of entity is represented as nodes and rearranged in four different levels of the hierarchy graph.

Nodes:

- Member node (MN): at level 1 and represents a group member.
- Profile node (PN): at level 2 and presents an item has been rated by members.
- Neighbour node (NN): at level 3 and represents a neighbour non-member user.
- Recommendation node (RN): at level 4 and represents a recommendation.

With this design, users are given a clear overview of the whole system. Furthermore the edges that link two nodes in different levels demonstrate the three main procedures in the CS-GR method:

- MN-PN edges: the edges represent pseudo user modeling.
- PN-NN edges: the edges represent neighbour identification.
- NN-RN edges: the edges represent recommendation calculation.

Explanations for each component are given in the following section.

6.2.2 COMPONENTS

The group recommendation problem is defined as:

Definition: Let *U* be all the users of the system $U = \{u_1, u_2 \dots u_{|U|}\}$ and *I* be all the items $I = \{i_1, i_2 \dots i_{|I|}\}$. Group G is a sub-set of U that $G = \{g_1, g_2 \dots g_{|G|}\}$, where $g_m \in U, m = 1 \dots |G|$. Let $I_G = \{i_1^*, i_2^*, \dots, i_{|I_G|}^*\}$ be all the items that have been rated by group members, where $i_n^* \in I, n = 1 \dots |I_G|$. Let g be the pseudo user of the group with profile P_g. P_g is then be defined as a rating vector and every dimension of it is $r_{g,i}$ where $i \in \{i_1^*, i_2^*, \dots, i_{|I_G|}^*\}$. The unknown ratings $r_{g,i}, i \notin I_G$ are predictions from the individual recommendation method *f*, $r_{g,i} = f(g, i)$. The recommendations are generated by choosing the top-*N* items with the highest predictions.

$$rec = \max_{i \notin I_G}(r_{g,i}) = \max_{i \notin I_G}(f(g,i)) \quad (6\text{-}1)$$

In the CS-GR method each member has an important attribute, its contribution, which is a numeric measurement of the representativeness of group members. The CS-GR method aims to develop a recommendation approach that maximizes group satisfaction for users by modeling group preferences through the analysis of member

ratings. The method employs SNMF to calculate contributions in terms of a sub-rating matrix.

All the item-pair subspaces are used as sub-rating matrixes to measure the contribution. Let S be the set of all the subspaces, $S_j \in S$, $j \in [1..|S|]$ and the contribution of member g_m in subspace S_j is

$$c_{g_m}^{S_j} = \begin{cases} 1, & m = \arg\max_k (\arctan 2(r_{g_k,j_1}, r_{g_k,j_2})) \\ 0, & otherwise \end{cases} \quad (6\text{-}2)$$

where j_1 and j_2 are two items which form the S_j. The overall contribution of member g_m is sum of all subspaces according Equation (6-3) as follows:

$$c_{g_m} = \sum_{j=1}^{|S|} c_{g_m}^{S_j} \quad (6\text{-}3)$$

Every member in G is visualized as an MN node and then the MN set is $[MN_{g_1}, MN_{g_2} \ldots MN_{g_{|G|}}]$. How the MN nodes are constructed mainly depends on two factors: the member identifier m and its corresponding contribution c_{g_m}. The member identifier helps members distinguish themselves from the others, and, in the system, the identifiers are shown using different colors. The contributions are shown using different radii of nodes, and the radii are determined by a mapping Equation (6-4). Let $c_{max} = \max(c_{g_m})$, $m = 1 \ldots |G|$ and the maximum radius is R, the radius of MN_{g_m} is

$$R_{MN_{g_m}} = R \frac{c_{g_m}}{c_{max}} . \quad (6\text{-}4)$$

However, the MN level may contain numerous MN nodes and these unorganised nodes make members are difficult to obtain relative relation between themselves and the others, therefore they are sorted by level of contribution for clarity of presentation. The nodes are sorted in each subsequent level, based on similar quantitative information, and these details are given in the corresponding explanation of each component.

Once a member's contribution has been calculated, the member's group rating can be calculated. Let g_m be the member and c_{g_m} be corresponding contribution, for a specific group item i_n^*, r_{g_m,i_n^*} is the rating for g_m of i_n^*. Since the group rating is a weighted sum of all the members that have rated that item, the part of group rating related to g_m is $c_{g_m} r_{g_m,i_n^*}$. Given a member may not rate all the items in the group, an MN-PN edge is drawn when, and only when, a member g_m has rated item i_n^*. From perspective of record, every MN-PN edge represents a historical rating and is only determined by $c_{g_m} r_{g_m,i_n^*}$. In this case, the width of edge is used to visualize $c_{g_m} r_{g_m,i_n^*}$. Let $r_{max} = max(c_{g_m} r_{g_m,i_n^*})$ where $g_m \in G \wedge i_n^* \in I_G$ has a maximum width of W. The width of the MN-PN edge is calculated according to Equation (6-5). The width of the edge indicates degree of preference, giving the user insight into group profile.

$$W_{MN-PN_{g_m,i_n^*}} = W \frac{c_{g_m} r_{g_m,i_n^*}}{r_{max}} \tag{6-5}$$

As previously mentioned, the group rating is calculated with a weighted sum and the equation is shown below:

$$r_{g,i_n^*} = \sum_{g_m \in G} c_{g_m} r_{g_m,i_n^*} \ . \tag{6-6}$$

Next the pseudo user g is modeled by constructing a rating vector group profile P_g which assume can represent the preferences of the whole group. $P_G = [r_{g,i_1^*}, r_{g,i_2^*} \dots r_{g,i_{|I_G|}^*}]$. Every element in P_g is visualized as a PN node, and the PN set becomes $[PN_1, PN_2 \dots PN_{|I_G|}]$. The PN node's construction mainly depends on two factors: the group rating and individual influence within it. A group rating represents the overall preference for an item and is visualized using the radius of the node. Instead draw a node using a solid colour, a pie chart is displayed in the node to illustrate the individual influences.

The radius of a PN node is determined by a mapping Equation (6-7) considering all the group ratings. Let $r_{max} = max(r_{g,i_n^*}), n = 1 \dots |I_G|$ and the maximum radius is R, the radius of PN_n is

$$R_{PN_n} = R\frac{r_{g,i_n^*}}{r_{max}} \tag{6-7}$$

All the PN nodes, once calculated, are sorted and arranged according to their group ratings. This helps members understand relative relationships within the group profile.

The individual influences from different members are illustrated by slices in a pie chart and the ratio of each slice is calculated according to Equation (6-8).

$$ration_{PN_n}^{m} = \frac{c_{g_m} r_{g_m,i_n^*}}{r_{g,i_n^*}} = \frac{c_{g_m} r_{g_m,i_n^*}}{\sum_{m=1}^{|G|} c_{g_m} r_{g_m,i_n^*}} \tag{6-8}$$

They are the normalized products of members, and thus, this chart demonstrates how much degree a member determined the group rating.

Based on the group profile, P_g, the similarities between the pseudo user and non-member group users can be calculated. The PCC similarity, widely used in a number of RSs, has been used for the similarity computation. Let u be a non-group user. Let I_u be the item set that has been rated by u. The PCC similarity between pseudo user g and u is computed based on their common ratings as follows:

$$Sim(g,u) = \frac{\sum_{i\in(I_G\cap I_u)} (r_{g,i} - \bar{r}_g)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i\in(I_G\cap I_u)} (r_{g,i} - \bar{r}_g)^2 \sum_{i\in(I_G\cap I_u)} (r_{u,i} - \bar{r}_u)^2}} \tag{6-9}$$

where $I_G \cap I_u$ is the set of common items, $\bar{r_g}$ is the average rating of g and $\bar{r_u}$ is the average rating of u. It is important to note that for a given i_n^* and P_g, the relationship between g and u on item i_n^* can be measured by:

$$Sim^{i_n^*}(g,u) = \frac{(r_{g,i_n^*} - \bar{r}_g)(r_{u,i_n^*} - \bar{r}_u)}{\sqrt{\sum_{i\in(I_G\cap I_u)} (r_{g,i} - \bar{r}_g)^2 \sum_{i\in(I_G\cap I_u)} (r_{u,i} - \bar{r}_u)^2}} \tag{6-10}$$

It is easy to see that every PN-NN edge represents a historical rating of non-group user on items of group profile. Let $r_{max} = max(Sim^{i_n^*}(g,u))$ where u is a non-member user and the maximum width is *W*, for specific group profile item i_n^* and neighbour *u*, the width of PN-NN edge is:

$$W_{PN-NN_{g_m i_n^*}} = W \frac{Sim^{i_n^*}(g,u)}{r_{max}} \tag{6-11}$$

After all the similarities of non-member users are determined, a collection of neighbours are selected, either according to a predefined number or a similarity threshold, to calculate predictions. The selected neighbours set is noted as *N* where $N = \{n_1, n_2 \dots n_{|N|}\}$. This means that only the PN-NN edges that target the selected neighbours need to be drawn. Every element in *N* is shown as an NN node and where the NN set is $[NN_1, NN_2 \dots NN_{|N|}]$. The radius of an NN node is determined by a mapping function from similarities. Let $sim_{max} = max(sim(g,n_k))$, $k = 1 \dots |N|$ be the similarities between the pseudo user and its neighbours with a maximum radius of R. The radius of NN_k is:

$$R_{NN_k} = \frac{Sim(g,n_k)}{Sim_{max}} \ . \tag{6-12}$$

All the NN nodes are sorted rearranged according to their similarity value, which provides a clear representation of the neighbourhood. For a given neighbour *u*, the influence of a specific group profile item $\boldsymbol{i_n^*}$ is calculated, using Equation (6-13), by normalizing the similarity for every common item.

$$ratio_{NN_k}^{n} = \frac{Sim^{i_n^*}(g,u)}{Sim(g,u)} \tag{6-13}$$

Combining the influence of a specific member to i_n^*, gives the influence of a specific member to a neighbour. A pie chart demonstrates how much a member has influenced a specific neighbour.

$$ratio_{NN_k}^{m} = \sum_{n=1}^{|I_G|} (ratio_{NN_k}^{n} ratio_{PN_n}^{m}) = \sum_{n=1}^{|I_G|} (\frac{Sim^{i_n^*}(g,u)}{Sim(g,u)} \frac{c_{g_m} r_{g_m,i_n^*}}{\sum_{m=1}^{|G|} c_{g_m} r_{g_m,i_n^*}}) \tag{6-14}$$

Unknown group ratings can be predicted after the neighbours are selected. The user-based collaborative filtering is adopted, calculated by the weighted sum of deviations from the average rating of its neighbours. Let $r_{g,i}$ be the unknown group rating for item *i*, and $i \notin I_G$, and $r_{g,i}$ can be computed by Equation (6-15).

$$r_{g,i} = \overline{r}_g + \frac{\sum_{u \in N} (r_{u,i} - \overline{r}_u) \times Sim(g,u)}{\sum_{u \in N} |Sim(g,u)|} \tag{6-15}$$

where $\overline{r_g}$ denotes the average rating of pseudo user *g*. Clearly, $\overline{r_g}$ is a constant for given group, thus the group rating is mainly determined by:

$$\tilde{r}_{g,i} = \overline{r}_g + \frac{\sum_{u \in N} (r_{u,i} - \overline{r}_u) Sim(g,u)}{\sum_{u \in N} |Sim(g,u)|}$$

Therefore, for given neighbour *u* and item *i*, the NN-RN edge is constructed by the mapping Equation (6-16). Let $r_{max} = max(\tilde{\boldsymbol{r}}_{\boldsymbol{g,i}})$ for specific neighbour *u* and the maximum width is *W*, the width of NN-RN edge is:

$$W_{NN-RN_{u,i}} = W \frac{\tilde{r}_{g,i}}{r_{max}} \tag{6-16}$$

Usually, the top-*N* items with highest predictions are selected as the final recommendations using Equation (6-15). From the perspective of recommendations, only NN-RN edges those target to final results need to be visualized. The recommendation set is denoted as *R* where $R = \{r_1, r_2 \ldots r_{|R|}\}$. Every element in *R* is shown as an RN node and the RN set is $[RN_1, RN_2 \ldots RN_{|R|}]$. The radius of an RN node is determined by a mapping function from group ratings. Let $r_{max} = max(r_{g,i})$, $i = 1 \ldots |R|$ and the maximum radius is *R*, the radius of RN_k is

$$R_{RN_k} = R \frac{r_{g,i}}{r_{max}} \tag{6-17}$$

Because the unknown group rating is derived by aggregating predictions from all the neighbours, the influence of specific neighbour *u* for given recommendation *i* can

be measured by $(r_{u,i} - \overline{r_u}) \times sim_{g,u}$. Thus, the individual influence from different neighbours for specific $r_{g,i}$ is calculated by normalizing Equation (6-18):

$$ratio_{RN_i}^{k} = \frac{|(r_{k,i} - \overline{r}_k) \times Sim_{g,k}|}{\sum_{u \in N} (r_{u,i} - \overline{r}_u) \times Sim_{g,u}} \tag{6-18}$$

Combining these ratios with ratio of the member to specific neighbour provides the ratio of the member to specific recommendation.

$$ratio_{NN_k}^{m} = \sum_{k=1}^{|N|} (ratio_{RN_i}^{k} ratio_{NN_k}^{m}) = \sum_{k=1}^{|N|} \left(\frac{|(r_{k,i} - \overline{r}_k) \times Sim_{g,k}|}{\sum_{u \in N} (r_{u,i} - \overline{r}_u) \times Sim_{g,u}} \sum_{n=1}^{|I_G|} \frac{Sim^{i_n^*}(g,u)}{Sim(g,u)} \frac{c_{g_m} r_{g_m, i_n^*}}{\sum_{m=1}^{|G|} c_{g_m} r_{g_m, i_n^*}} \right) \tag{6-19}$$

A pie chart shows the individual influence of every member in the RN node, and becomes a useful tool for members to track the influence they had on each recommendation.

6.3 HIERARCHY VISUALIZATION IMPLEMENTATION

The method is implemented on MovieLens 100K data set to test the feasibility and effictivity. The implementation includes three procedures to calculate the predictions, and they are summarized as follow:

- Procedure 1: CS-based group modeling is used to model the pseudo user.
- Procedure 2: PCC is used to identify the neighbours based on pseudo user's profile.
- Procedure 3: CF is used to calculate the unknown group ratings based on the observed ratings of neighbours.

The detail of implementation and discussion are shown below.

6.3.1 USABILITY

An overview of the result is shown in Figure 6-1, where nodes representing the entities, i.e., the group members, profile items, neighbours and recommendations, are allocated to different levels according their type.

A group consisting of five randomly selecting members was formed. The overall recommendation procedures are represented using different sized nodes and different width edges. First, the CS-GR method calculated each member's contributions, and a pseudo user is modeled from the weighted average of them all. As a result, five nodes of different sizes were rendered in the MN level – their names are displayed to the right side of each node. Focusing on Members 1 and 5, for simplicity, their contributions were 0.31 and 0.06, respectively, and therefore the radius of Member 1's node is much larger than Member 5's. Member 1's node has three edge sources showing that Member 1 rated three items, i.e. P1, P2 and P4, but did not rate P3 and P5.

The width of the edges represents the strength of inheriting information. The MN-PN edge represents the weighted ratings. Since Member 1's contribution is constant, the three edges show that Member 1 gave her highest rating to P1 and her lowest to P4. Given the target group rating is the sum of weighted ratings from all the members, the P1 node which is determined by Member1, becomes the biggest because of Member 1's large contribution. P9's radius is the smallest because the member contributions were small. In this way, all members can easily see what items are in group profile and how the group ratings were derived.

The PN-NN edges represent the ratings specified by neighbours for items in the group profile. Take P1's sources an example. Both Neighbour 1 and 7 rated this item, but the width of the edge to Neighbour 1 is little wider than to Neighbour 7. This demonstrates that Neighbour 1 preferred P1 more than Neighbour7.

Additionally, all the PN-NN edges target a single NN node to represent the overall preference of a neighbour. For example, the final similarity of Neighbour 7 was 0.65 when P1's source was 0.44 and P4's source was 0.21. It is important to point out that a neighbour is selected by the PCC similarity measure, which means it could be no

direct relationships between a group item and a neighbour. For example, Neighbour 14 only has direct connections with P8, which means it was only selected because he preferred P8 and her opinions about the rest of group profile items are not known.

The NN nodes, RN nodes and NN-RN edges work very similarly to the MN nodes, PN nodes and MN-PN edges. The only difference is that the numerical measure for members is their contribution and for neighbour it is their similarity. Because the user-based collaborative filtering is used to generate unknown ratings, the recommendations have direct relationships with at least one neighbour. The width NN-RN edge represents the degree of preference for recommendations for neighbours and is numerically equal to the product of the similarity of a neighbour and the specified ratings in recommendations. For example, R1 is recommended because the top 4 neighbours rated it, and the most similar neighbour preferred it the most, providing members with understanding for why these recommendations were generated and why some ratings are higher.

Individual connections are given as pie charts at every level from the MN nodes to the RN nodes. This makes it easy for members to track their individual influence throughout the recommendation process. For example, tracking Member 1 through P2, then Neighbour 2 and lastly to R4 shows Member 1's individual influence along this path. At P2, she can see her dominance in the group rating, because Member 3's percentage is so much smaller. In the NN node for Neighbour 2, she sees her percentage, or influence, has decreased but Member 3's percentage has increased because Neighbour 2 prefers P3, which is dominated by Member 3. Lastly, at R4, her influence has continued to decrease because many other neighbours have influenced the prediction.

Table 6-1. Visualization Components Summarization

Component	MN	MN-PN	PN	PN-NN	NN	NN-RN	RN
Radius	✓		✓		✓		✓
Width		✓		✓		✓	
Pie			✓		✓		✓

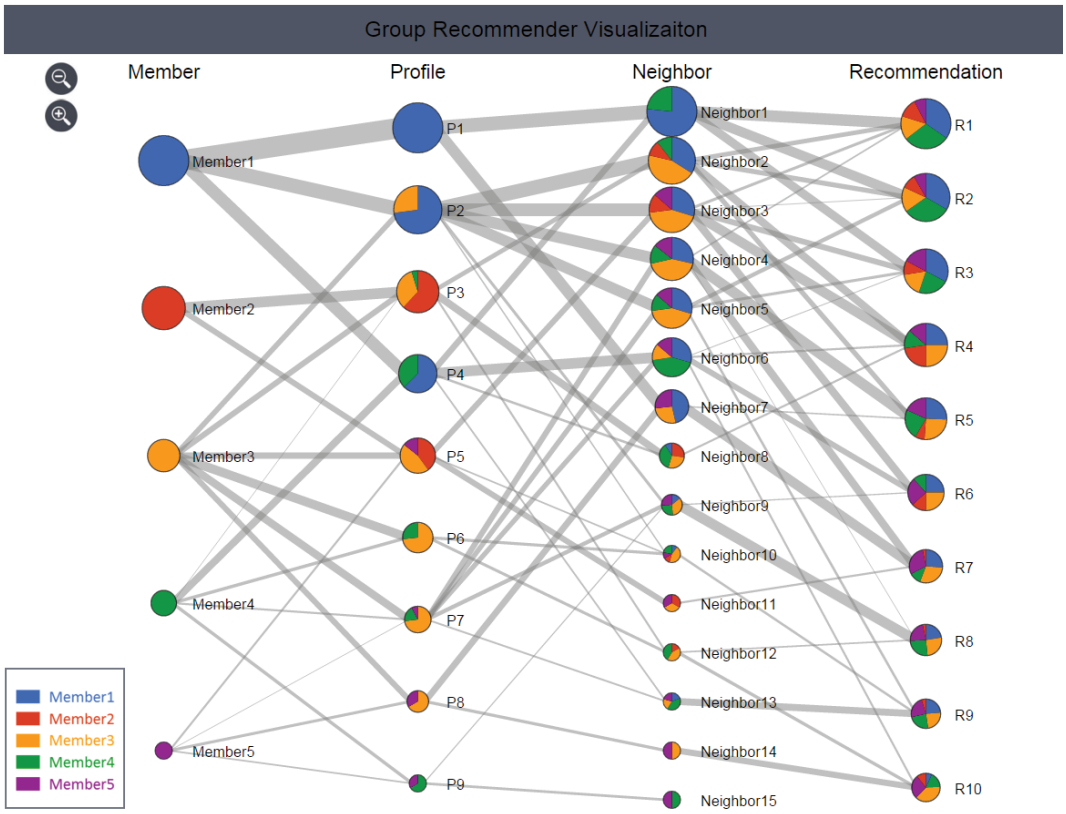


Figure 6-1. A visualization example on real data set MovieLens 100K.

6.3.2 INTERACTIVITY

The zoom and pan techniques allow users to interact with the graph and gain more detailed information at each step. This guarantee increases the efficiency of screen usage and guarantees the scalability of the proposed method. Figure 6-2 gives a demonstration of focusing on the right-top part of the graph using zoom.

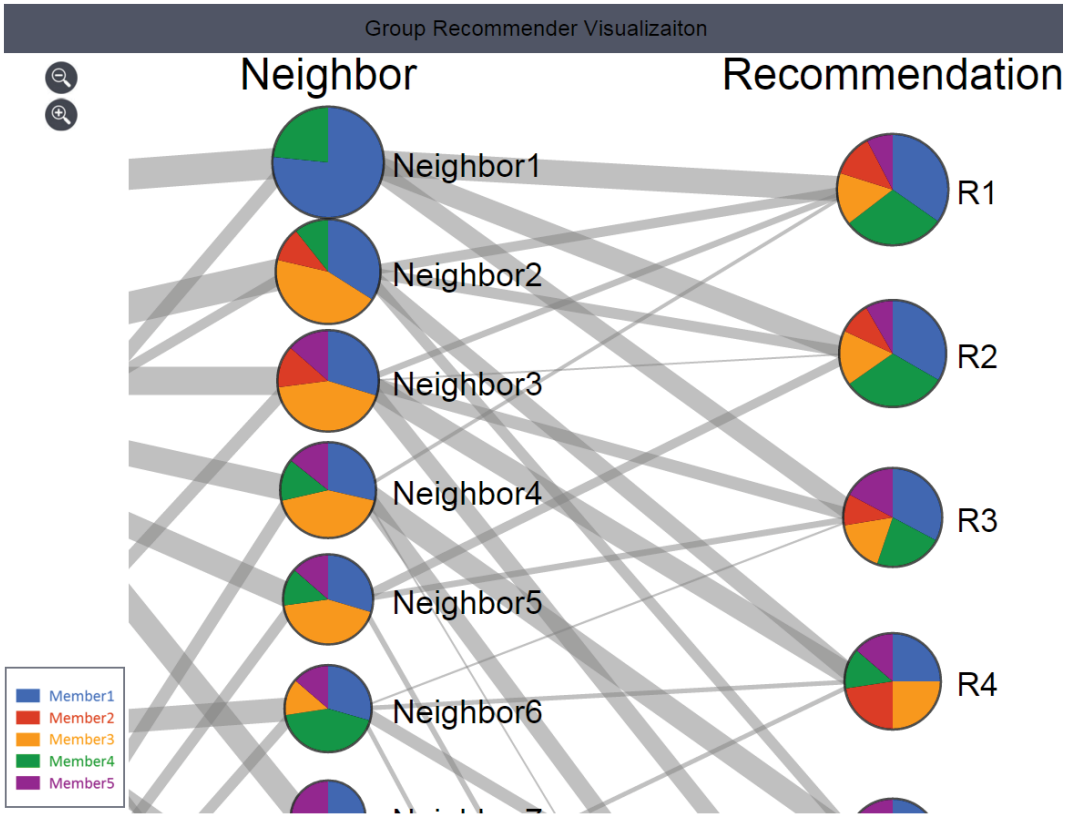


Figure 6-2. Visualization supports zoom and pan to enable interactivity.

6.3.3 ADAPTABILITY

The method is easily adaptable to graph other group modeling methods, like plurality voting, least misery, average etc. The modeling procedure is abstracted using MNs, PNs and MN-PN edges. The XXX method only need to adjust radius of PNs and widths of MN-PN edges to adapt to other modeling methods.

6.3.4 EXPANSIBILITY

The method could also provide solid explanations in collaborative filtering-based recommendation systems, because an individual recommender process can be seen as a ‘special’ group recommender process using group that only contains one member.

Visualization would need to be modified in first two levels to support individual visualization. The first level would contain a single root of only one node, representing the active user. The radius of the circle would represent their average

rating and illustrate their rating pattern. For example, a user's trend for giving higher or lower ratings can be easily identified in this way.

The group profile becomes active user profile, and this profile can still be visualized in PN level. Each node of PN would represent an item has rated by the active user and can be filled with different colour, which is similar of members nodes in group visualization. The pie chart in NN and RN level could represent the influence of every group item within a specific recommendation.

6.4 HIERARCHY VISUALIZATION ON SMARTBIZSEEKER

SBS is a recommender system aims to help businesses to find appropriate partners (suppliers and buyers). It is developed using JAVA and a web-based implementation is deployed in the Glassfish web server. Figure 6-3 shows the login page of SBS.

The application in the web server contains three layers: the presentation layer, the business logic layer and the data access layer.

1) Presentation layer

This layer is responsible for generating the requested web pages and handling the user interface logics and events. When a user requests to view a new page, the presentation layer will invoke corresponding methods in the business logic layer, extract the request data, transform the data into an HTML page and send it back to the client.

2) Business logic layer

The business logic layer defines the business processes and functions of the application, and serves as a mediator between the presentation layer and the recommendation engine and the data access layer. It provides the following main functionalities as follows:

3) Data access layer

The data access layer provides the interfaces to access the data in the database. It deals with the data operations of the database and transfers data with the business logic layer.

From the system point, traditional users and items in RSs are businesses in it. In this case it is a business specifies its preference for other businesses. SBS provides three types of UGCs that enable business users to interact with the system and other users.

- Preference: every user can build her profile including business information, product tree and detail production information. User can specify preferences for system to retrieve and find the perfect matches according the profiles. For example, a grocery user may claim that it accepts only farm eggs. An example for user to specify her preference is shown in Figure 6-4.
- Rating: users can specify ratings for other users scale from 1 to 5. Rating 5 represents the user is strong satisfied and 1 represents she is strongly dissatisfied.
- Comment: users can give short texts to comment other users.

Based on these information, many recommendation methods are implemented on SBS, such as traditional collaborative filtering-based methods and trust-based methods. The use can obtain the personalized recommended supplier/buyer list after she provides some information for system. The results of supplier/buyer recommender results are shown in Figure 6-5 and Figure 6-6 respectively. However, the system lacking necessary explanation makes users hard to believe and trust the recommendation results.

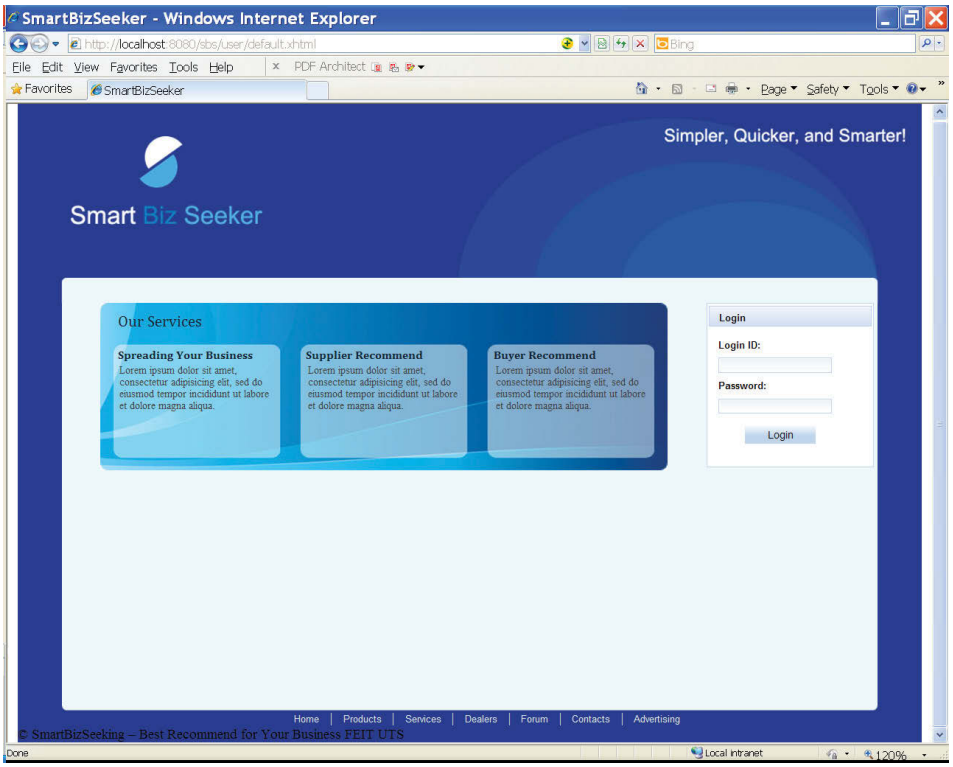


Figure 6-3. The login page of SBS.

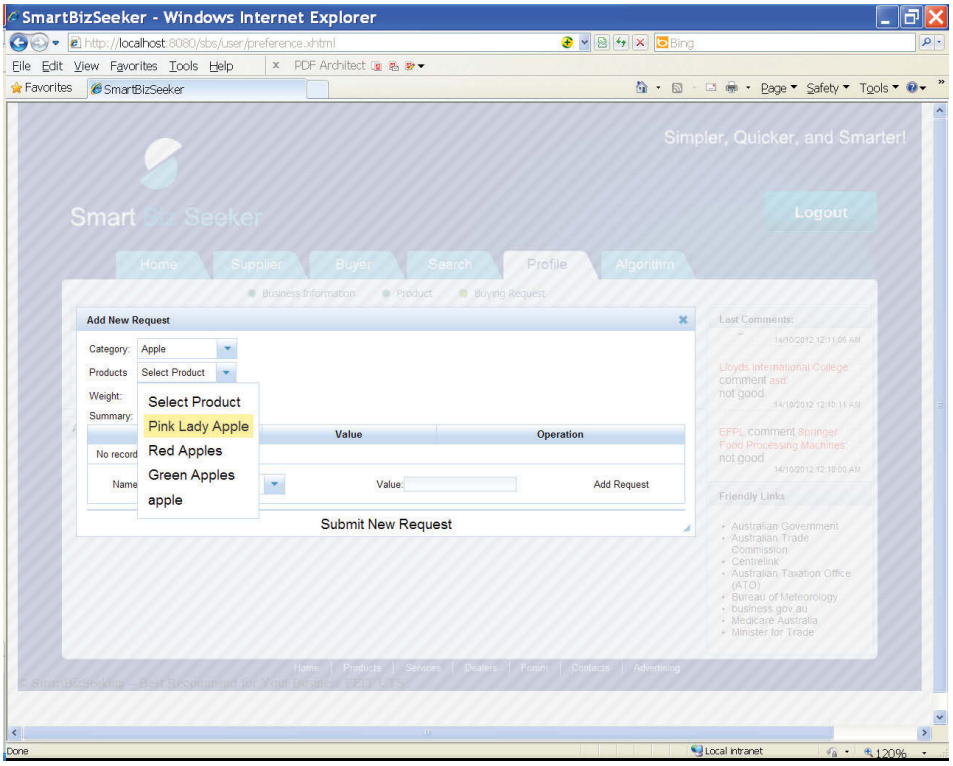


Figure 6-4. Buying request management page.

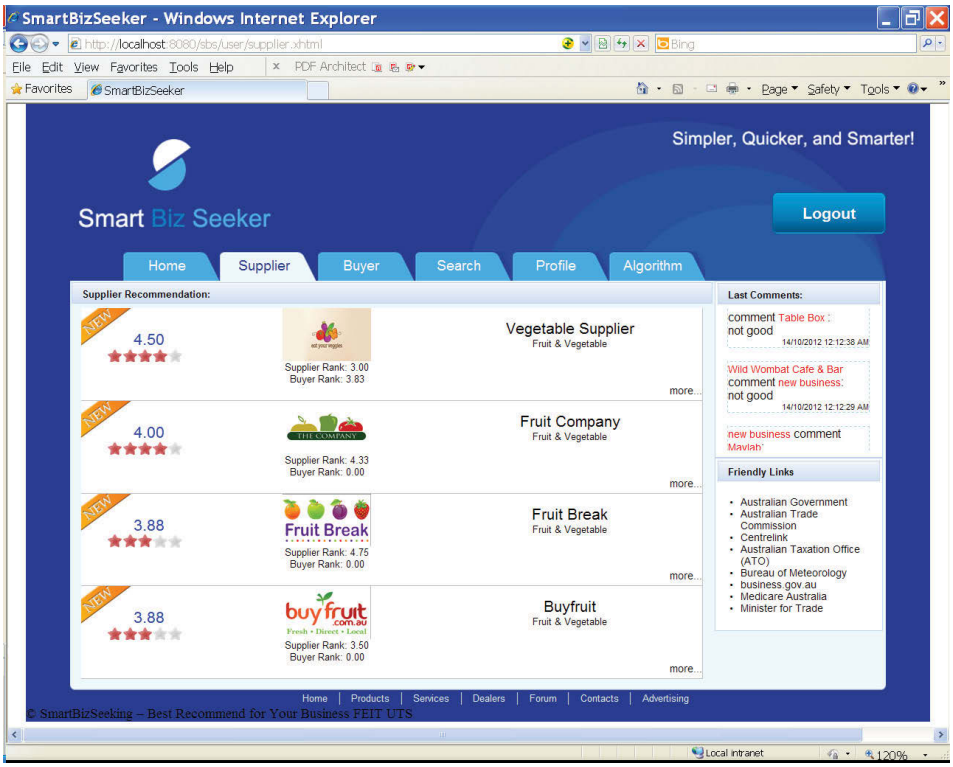


Figure 6-5. The supplier recommendation results.

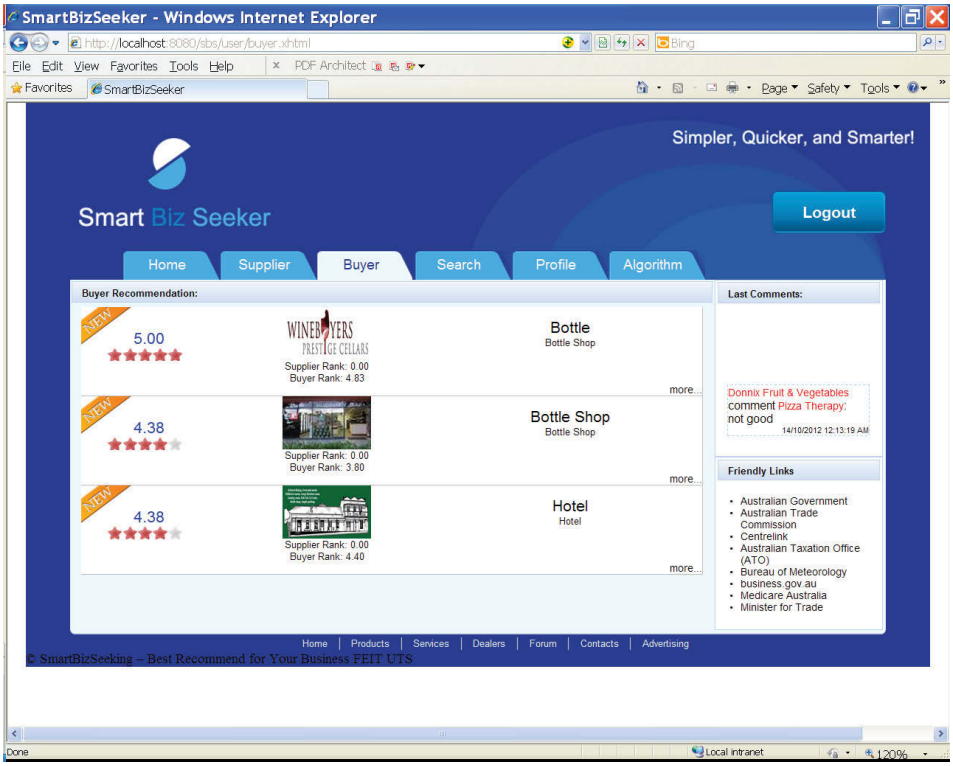


Figure 6-6. The buyer recommendation results.

Based on trackable hierarchy visualization method, an implementation is developed based on SBS. The implementation is described below.

6.4.1 ARCHITECTURE

Comparing to GRSs, SBS is a system designed for individual business users. The separable hierarchy method uses multi-level to represent structural data in recommender process, such as group, group profile and neighbour selection. Some changes are made to adapt to individual RSs. Member node level is removed because it will only contain one node representing the logged in user. Another change is, in SBS, many similarity measures are implemented; hence the corresponding components to illustrate similarity calculation are extracted as an independent module to make the visualization method more adaptable. The details and modifications are shown below.

The architecture of SBS with visualization module is shown in Figure 6-7. It is quite important to point that most recommendation methods implemented in SBS are neighbour-based or their variations (trust-based and fuzzy preference tree-based) and, in neighbour-based recommendation methods, there are two main procedures. The first procedure, for a specific active user (business), is to identify the similar business. The second procedure is to predict the unknown ratings according to similarity results and select top-N recommendations. The module constructs two sub-modules: trackable hierarchy visualization module and similarity visualization module to handle two procedures respectively. The former one illustrates the predicting process and the later one shows the similarity evaluating procedure.

- Trackable hierarchy visualization module: The adapted method consists of two levels: profile level and recommendation level. Every node in profile level represents a business preferred and every node in recommendation level represents a recommended supplier/buyer business.
- Similarity visualization module: An independent layer is constructed to illustrate for similarity visualization. Components in this layer are constructed according to adopted similarity measure.

The details of two modules are described in Section 6.4.2 and 6.4.3.

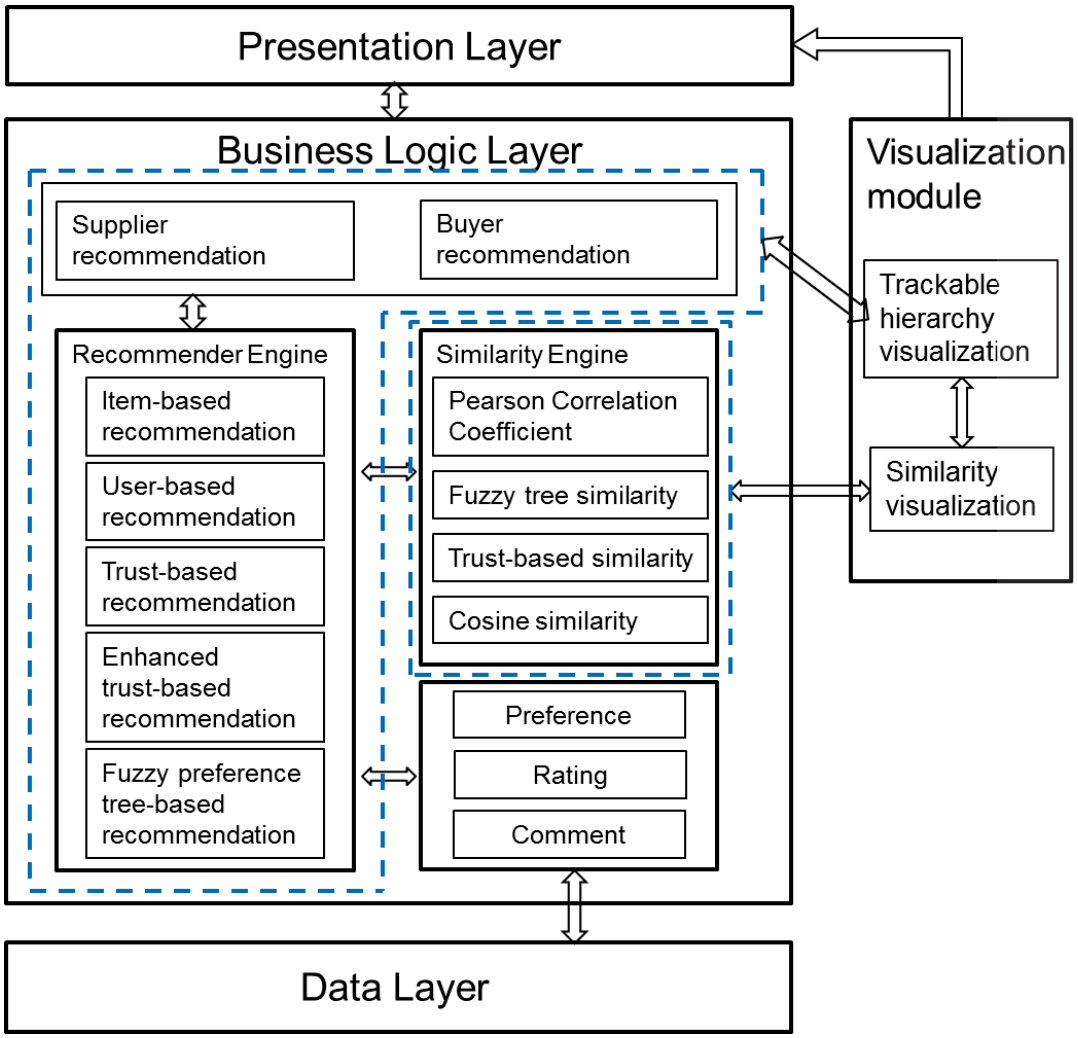


Figure 6-7. SBS system architecture with visualization module.

6.4.2 HIERARCHY VISUALIZATION MODULE

This module is to illustrate the recommender procedure without considering similarity calculation. Therefore, in this module, only two important structure data, user profile and recommendation list, are represented. Similar to the group visualization, PN level and RN level construct a two level hierarchy graph. A profile rating for a business is represented as a PN and radius of the node represents the mean of all the specified ratings on that business. And a recommendation is represented as a RN and radius of the node represents the prediction for it. The two levels are represented as two light grey lines. The recommendation line is at the top and the profile line is at the bottom. Obviously, this user has rated three business and obtain four recommendations. An example of trackable hierarchy visualization is shown in

Figure 6-8. The recommendations are generated for North Strathfield Cellars: the profile is presented in bottom line and is labeled by “Rated Business” and the recommendation is presented in top line and is labeled by “New Business”.

A star diagram is drawn at the right part of page to demonstrate all the ratings that have been specified by the active user. The length of link to the center of the star represents degree of satisfied. This diagram is useful because missing information on parallel diagram can be represented when the neighbours are identified using non-transportation methods such as social network or advanced trust-based methods. For example, a bottle shop has rated 5 wine producers, and another bottle shop will not be recommended with these producers without common ratings or comments using non-transportation methods such as user-based CF method. In this case, SBS provide users with advanced self-digging in the system by clicking businesses on stars.

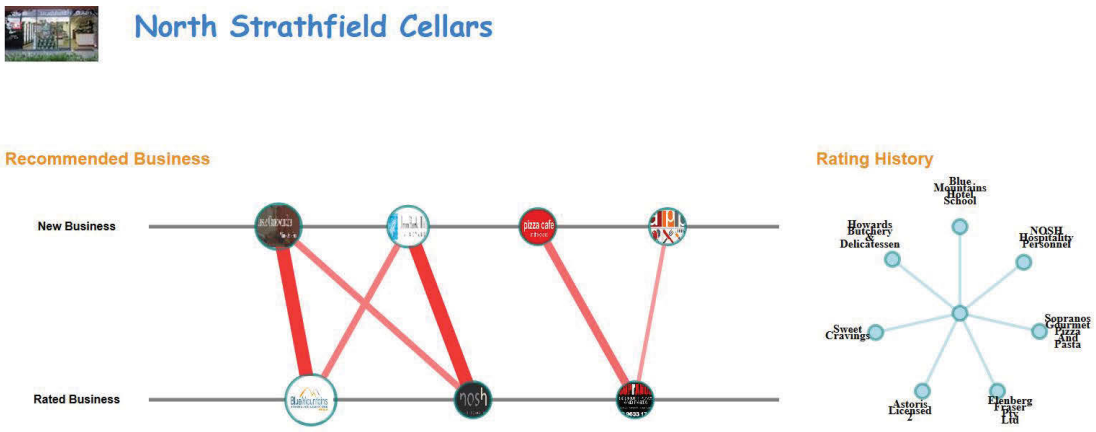


Figure 6-8. Trackable hierarchy visualization result example.

More information including business information and corresponding rating is designed to displayed when the cursor moving over the business nodes. Figure 6-9 and Figure 6-10 are examples of floating business information. The historical rating of rated business or prediction of the recommendation is shown in floating information too. The historical rating is 3 in Figure 6-9 and prediction is 3.33 in Figure 6-10. Because the radius of business node is determined by the rating, node for recommendation “IronBark Hill Vineyard” is litter larger than “Sopranos Gourmet

Pizza and Pasta". It is important to note that distance between two points does not represent similarity between them.

PN-NN edges and NN-RN edges are combined as PN-RN edges to describe the entire recommender process. Red links for PN-RN edges are utilized to illustrate degree of similar between recommendations and profile businesses. A link is wider and redder if the similarity is higher. Using these links, users can have intuitive understand relations of recommendations from that it have specified. Because similarity evaluation procedure is extracted, the PN-RN edges represent calculation results without considering which measure is used.

To calculate recommendations, relations play an important role in recommending process. Since items are arranged into two axes, these relations are links between points from recommendation axis and historical axis. It is important to note that distance between two points does not represent similarity between them. Different link widths, such as curves used in chord diagram, are used to visualize similarity between entities.

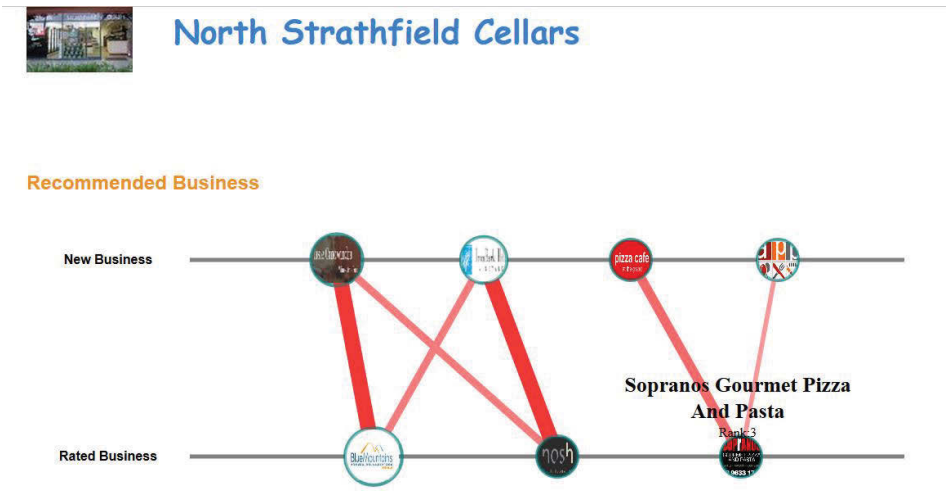


Figure 6-9. Floating business information over node.

North Strathfield Cellars

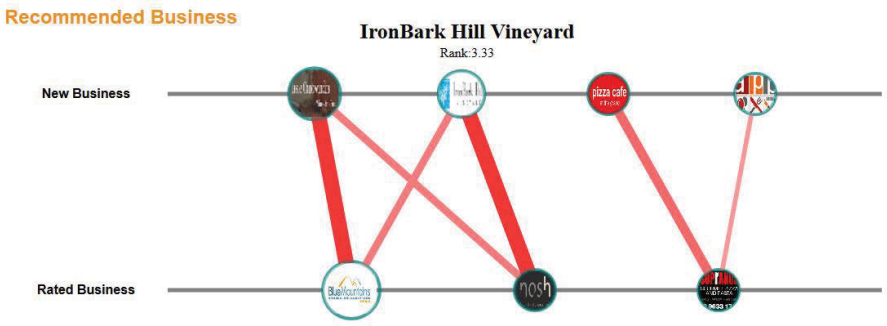


Figure 6-10. Floating business information over node.

North Strathfield Cellars

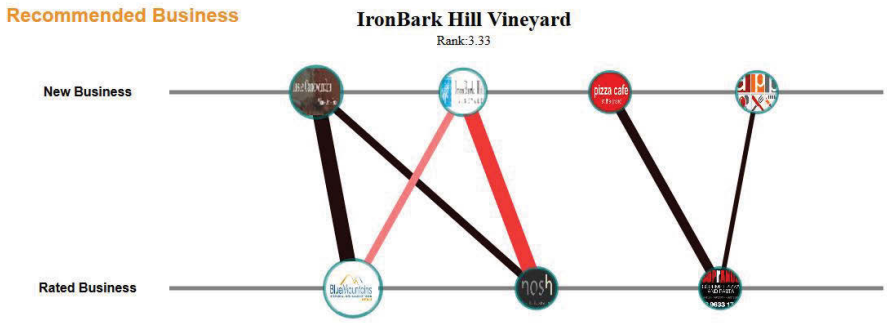


Figure 6-11. Highlighting for a recommendation node.

North Strathfield Cellars

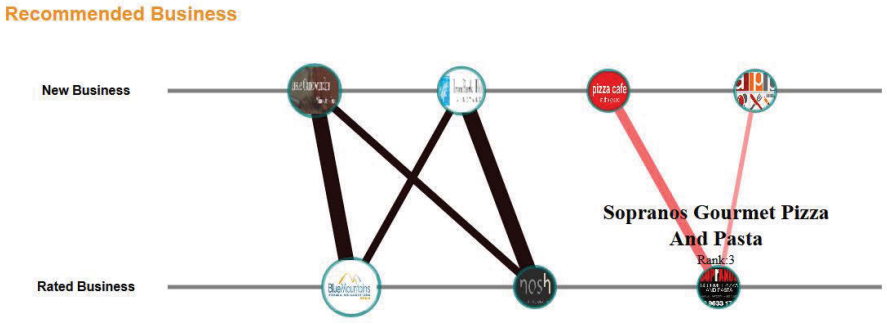


Figure 6-12. Highlighting for a profile node.

6.4.3 SIMILARITY VISUALIZATION MODULE

To calculate recommendations, relations between businesses, i.e. similarity measures, play an important role. The similarity visualization module is designed for demonstrating degree of close between two businesses. In SBS, many similarity measures are implemented including PCC, cosine, trust and their variations. There are great differences in the calculations that cause no uniform design for explanation. Therefore, the explanations of similarity calculating are extracted from the rest of procedures. Hence, similarity visualization designs different layouts for them and an example for fuzzy product tree similarity is shown below.

For every business, product trees are constructed according to its products. Preference tree is constructed according preferences data specified by target business. Fuzzy product tree similarity is to evaluate similarity between two tree-structured data using fuzzy set theory. The neighbours are identified by retrieving the entire possible product trees and select ones match preference tree most.

When mouse moving over an edge, a floating layer is displayed. In this layer, the similarity calculation result is shown at top left corner. Figure 6-14, Figure 6-13 and Figure 6-15 show three similarities between profile business and recommended businesses.

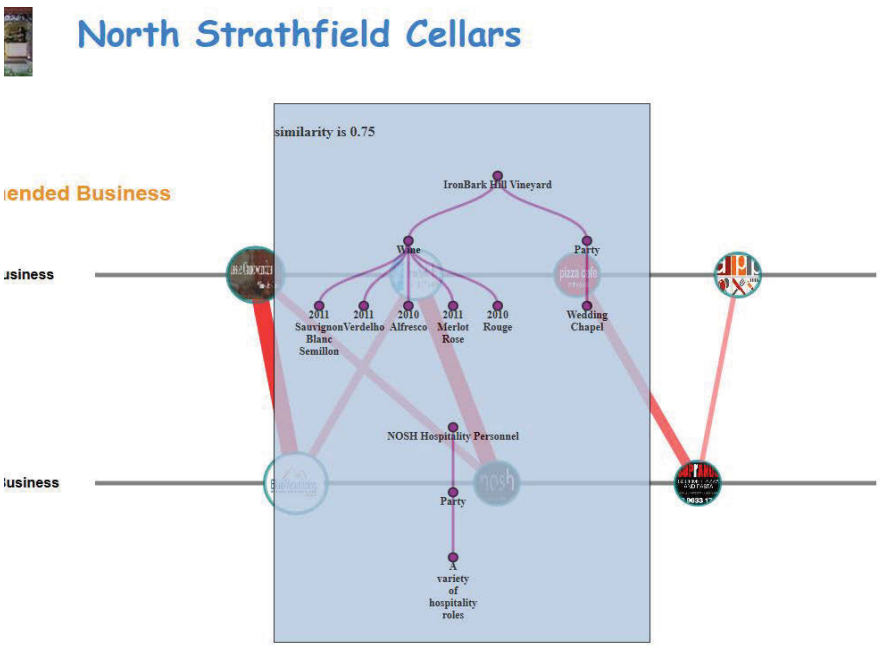


Figure 6-13. Higher tree similarity example for sharing common Party product.

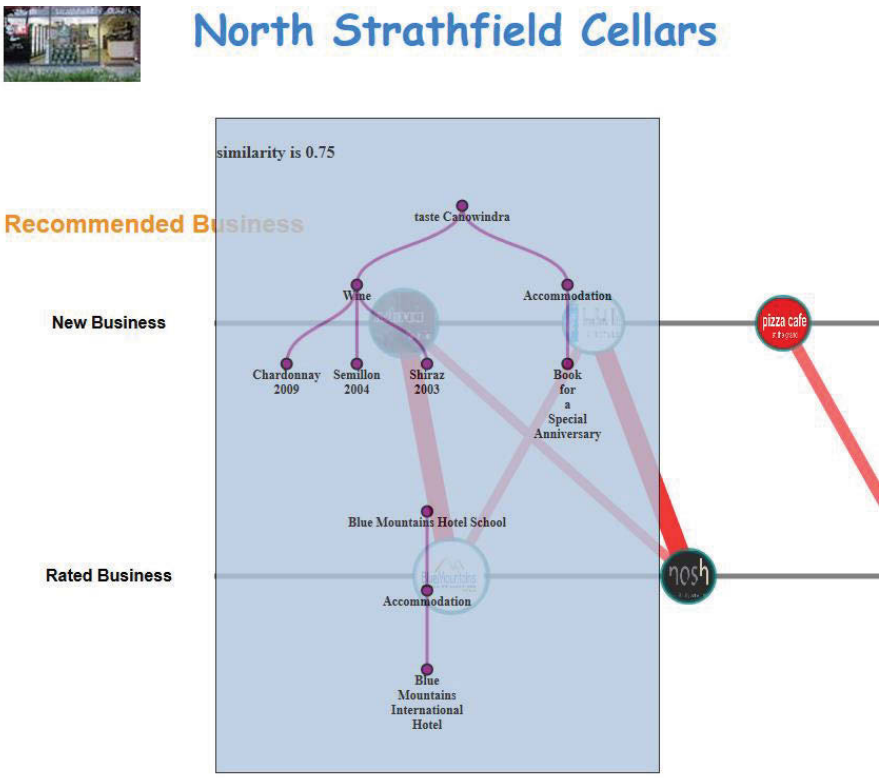


Figure 6-14. Higher tree similarity example for sharing common Accommodation product.

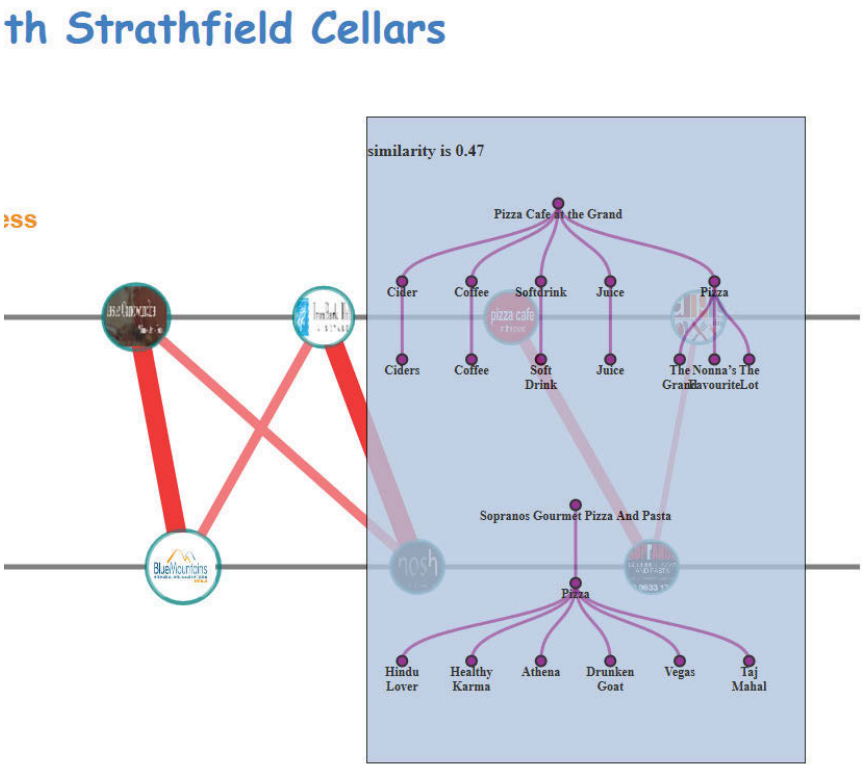


Figure 6-15. Lower tree similarity example for sharing Pizza product.

By combining visualizing methods in layer1 and layer2, the proposal can not only provide explanation for recommending process but also similarity calculation.

6.5 SUMMARY

This chapter presented a novel visualization method providing domain-independent explanation for both group and individual RSs. The method using multi-level layout to categories structure data in recommender system, such as user profile neighbour collection and recommendation list. Edges connecting different levels construct a hierarchy graph to illustrate pseudo user modeling and collaborative filtering procedures. Individual influences can be tracked by using pie chart in each node through all the procedures. Therefore, users are able to glean the overall procedure used to develop recommendations and understand the individual contributions them. The new method is easy to adapt to other pseudo user-based GRSs and is easily extendable to individual RSs. An experiment using a real data set was conducted to test feasibility.

CHAPTER 7

CONCLUSIONS AND FURTHER STUDY

7.1 CONCLUSIONS

This study is motivated by an awareness of practical opportunities and challenges in group recommender systems due to the rapid growth of both technologies and applied domains of intelligent services. Even though group recommender system research has gained considerable attention and undergone rapid developments, there are still some challenges in four aspects:

(1) predicting unknown ratings using CF methods under the biased and complex rating distributions;

(2) identifying high relevance neighbours by deeply analysing the ratings;

(3) constructing the group profile taking into consideration the member representativeness;

(4) providing visual explanation of the systems for users.

All these trends will provide new advanced features of group recommender systems but there exists inadequate studies in related topics in the literature. Hence, this research has conducted comprehensive analysis of each of the mentioned challenges and developed a set of novel recommendation techniques or frameworks.

The main contributions of this study are as follows:

1) It proposes a model to adaptively compute the average ratings considering the target item (to achieve Objective 1) and a local collaborative filtering method (to achieve Objective 2) in Chapter 3.

The adaptively average rating for the active user is computed taking the target item into consideration. For a specific target item, the relevance between it and non-target items are calculated using Manhattan distance measure. The irrelevant items are then filtered out when computing the average rating. Therefore, the local collaborative filtering method calculates an unknown rating for a target item by using this local average rating. In local collaborative filtering method, the predicting deals well with the target ratings on sparse region and far from pseudo user's average rating. The effectiveness of the approach is shown in a case study.

2) It develops a comprehensive similarity measure considering the inter-relationships of the differences between two rating vectors (to achieve Objective 3) and a neighbour identification model to select out highly relevant users/items (to achieve Objective 4) in Chapter 4.

To improve the performance of CF-based approaches, a novel similarity measure is proposed based on information entropy to analyze the relative relationships among all the rating differences when computing similarity between two rating vectors. The new similarity measures the degree of uncertainty of differences and makes co-ratings with higher difference contributing much to similarity only when the number of these items increases. The new similarity is not sensitive to rare impulse differences but is sensitive to continued differences. By hybridizing the entropy-based similarity and PCC-based similarity, the similarities are computed considering both absolute rating differences and relative differences between them. The collaborative filtering method is then based on neighbours using the new similarity measure.

3) It defines a contribution score measure to evaluate the representativeness of the members (to achieve Objective 5) and a group recommendation method using Contribution Scores to model the group profile (to achieve Objective 6) in Chapter 5.

A measure to evaluate the representativeness of the members is defined and is computed by partitioning the item space; members' opinions can be analysed using the SNMF technique. The group profiles are modeled by weighting every member according to their Contribution Scores. A pseudo user-based group recommendation method is developed based on these refined group profiles. Local collaborative filtering method is also used to alleviate the complex biased rating problem. The experiment results show the high effectiveness of the CS-LCF approach. This study not only has theoretical significance but also potentially has high practical application. Many online services, such as movie or tourism recommendation sites and other websites, could adopt the approach.

4) It develops a visualization method to explain the entire recommender process (to achieve Objective 7) and allows the members to track their individual influences in recommending proposals (to achieve Objective 8) Chapter 6.

A novel visualization method is proposed to provide a domain-independent explanation for both group and individual RSs. The method using a multi-level layout to categories structure data in recommender system, such as user profile neighbour collection and recommendation list. Edges connecting different levels construct a hierarchy graph to illustrate pseudo user modeling and collaborative filtering procedures. As for GRSs, individual influences can be tracked by using a pie chart in each node through all the procedures. Therefore, users are able to glean the overall procedure used to develop recommendations and understand the individual contributions to them. The new method is easy to adapt to other pseudo user-based GRSs and is easily extendable to individual RSs.

7.2 FURTHER STUDY

There are still some limitations of the current study:

1) The biased and sparse problem can be alleviated by using the local collaborative filtering method. However, the Manhattan distance measure needs enough common ratings to obtain reliable relevance evaluation between two items.

2) The parameters to hybrid Entropy-based similarity and PCC-based similarity have not been optimized.

3) How to weight the members according to the representativeness is still a complicated problem. Also, modeling the group profile by the weighted average form could be unacceptable when the members do not accept any disappointment.

This research can be fully advanced in the following aspects:

1) In the future, a new method to compute the relevance between two items will be investigated. The method to estimate relevance not only depending on the observed values will be developed.

2) The future study will include the improvement of analysing the rating to optimise the parameter to combine two similarities.

3) The future study will include the extension of the proposed approach to select representative samplings instead of random samplings when sub-space differences are taken into consideration. A possible future improvement is to mathematically define a function to describe the degree of contribution divergence, and to incorporate alternative models when the function has a higher value.

REFERENCES

Aamodt, A. & Plaza, E. 1994, 'Case-based reasoning: foundational issues, methodological variations, and system approaches', *AI Communications*, vol. 7, no. 1, pp. 39-59.

Adomavicius, G. & Tuzhilin, A. 2005, 'Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-49.

Adomavicius, G. & Tuzhilin, A. 2011, 'Context-aware recommender systems', in F. Ricci, L. Rokach, B. Shapira & P.B. Kantor (eds), *Recommender Systems Handbook*, Springer US, pp. 217-53.

Al-hassan, M., Lu, H. & Lu, J. 2011, 'Personalized e-government services: tourism recommender system framework', in J. Filipe & J. Cordeiro (eds), *Web Information Systems and Technologies*, vol. 75, Springer Berlin Heidelberg, pp. 173-87.

Albadvi, A. & Shahbazi, M. 2009, 'A hybrid recommendation technique based on product category attributes', *Expert Systems with Applications*, vol. 36, no. 9, pp. 11480-8.

Ali, K. & Stam, W.v. 2004, 'TiVo: making show recommendations using a distributed collaborative filtering architecture', paper presented to the *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA.

Amer-Yahia, S., Roy, S.B., Chawlat, A., Das, G. & Yu, C. 2009, 'Group recommendation: semantics and efficiency', *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 754-65.

Amoroso, D.L. & Reinig, B.A. 2004, 'Personalization management systems: minitrack introduction', *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, vol. Track 7, Big Island, Hawaii.

Ardissono, L., Goy, A., Petrone, G. & Segnan, M. 2005a, 'A multi-agent infrastructure for developing personalized web-based systems', *ACM Transactions on Internet Technology (TOIT)*, vol. 5, no. 1, pp. 47-69.

Ardissono, L., Goy, A., Petrone, G. & Segnan, M. 2005b, 'A multi-agent infrastructure for developing personalized web-based systems', *ACM Transactions on Internet Technology*, vol. 5, no. 1, pp. 47-69.

Ardissono, L., Goy, A., Petrone, G., Segnan, M. & Torasso, P. 2003, 'Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices', *Applied Artificial Intelligence*, vol. 17, no. 8-9, pp. 687-714.

Armstrong, R., Freitag, D., Joachims, T. & Mitchell, T. 1995, 'Webwatcher: a learning apprentice for the world wide web', *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pp. 6-12.

Asnicar, F.A. & Tasso, C. 1997, 'ifWeb: a prototype of user model-based intelligent agent for document filtering and navigation in the World Wide Web', *Proceedings of Workshop Adaptive Systems and User Modeling on the World Wide Web at 6th International Conference on User Modeling*, Chia Laguna, Sardinia, Italy, pp. 3-11.

Avesani, P., Massa, P. & Tiella, R. 2005, 'Moleskiing.it: a trust-aware recommender system for ski mountaineering', *International Journal for Infonomics*, vol. 20.

Baccigalupo, C. & Plaza, E. 2007, 'A case-based song scheduler for group customised radio', *Proceedings of the 7th International Conference on Case-Based Reasoning*, eds R.O. Weber & M.M. Richter, Springer Berlin Heidelberg, Belfast, Northern Ireland, UK, pp. 433-48.

Baltrunas, L., Makcinskas, T. & Ricci, F. 2010, 'Group recommendations with rank aggregation and collaborative filtering', *Proceedings of the 4th ACM Conference on Recommender Systems*, ACM, Barcelona, Spain, pp. 119-26.

Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A. & Peleteiro, A. 2010, 'A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition', *Information Sciences*, vol. 180, no. 22, pp. 4290-311.

Basu, C., Hirsh, H. & Cohen, W. 1998, 'Recommendation as classification: using social and content-based information in recommendation', *AAAI/IAAI*, pp. 714-20.

Batet, M., Moreno, A., Sánchez, D., Isern, D. & Valls, A. 2012, 'Turist@: Agent-based personalised recommendation of tourist activities', *Expert Systems with Applications*, vol. 39, no. 8, pp. 7319-29.

Battista, G.D., Eades, P., Tamassia, R. & Tollis, I.G. 1999, *Graph drawing: algorithms for the visualization of graphs*, Prentice Hall.

Bederson, B.B., Shneiderman, B. & Wattenberg, M. 2002, 'Ordered and quantum treemaps: making effective use of 2D space to display hierarchies', *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 833-54.

Bellogin, A., Cantador, I., Diez, F., Castells, P. & Chavarriaga, E. 2013, 'An empirical comparison of social, collaborative filtering, and hybrid recommenders', *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 1, pp. 1-29.

Berkovsky, S. & Freyne, J. 2010, 'Group-based recipe recommendations: analysis of data aggregation strategies', *Proceedings of the 4th ACM Conference on Recommender Systems*, ACM, Barcelona, Spain, pp. 111-8.

Billsus, D. & Pazzani, M. 2000, 'User modeling for adaptive news access', *User Modeling and User-Adapted Interaction*, vol. 10, no. 2-3, pp. 147-80.

Bjelica, M. 2010, 'Towards TV recommender system: experiments with user modeling', *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1763-9.

Bobadilla, J., Ortega, F., Hernando, A. & Gutiérrez, A. 2013, 'Recommender systems survey', *Knowledge-Based Systems*, vol. 46, no. 0, pp. 109-32.

Bogdanov, D., Haro, M., Fuhrmann, F., Xambó, A., Gómez, E. & Herrera, P. 2013, 'Semantic audio content-based music recommendation and visualization based on user preference examples', *Information Processing and Management*, vol. 49, no. 1, pp. 13-33.

Bostandjiev, S., O'Donovan, J. & Höllerer, T. 2012, 'TasteWeights: a visual interactive hybrid recommender system', *Proceedings of the 6th ACM Conference on Recommender Systems*, ACM, Dublin, Ireland, pp. 35-42.

Breese, J.S., Heckerman, D. & Kadie, C. 1998a, 'Empirical analysis of predictive algorithms for collaborative filtering', *The14th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., Madison, Wisconsin, USA, pp. 43-52.

Breese, J.S., Heckerman, D. & Kadie, C. 1998b, 'Empirical analysis of predictive algorithms for collaborative filtering', *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Morgan Kaufmann, San Francisco, CA, pp. 43-52.

Burke, R. 2000, 'Knowledge-based recommender systems', *Encyclopedia of Library and Information Systems*, vol. 69, no. 32, pp. 175-86.

Burke, R. 2002, 'Hybrid recommender systems: survey and experiments', *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-70.

Burke, R. 2007, 'Hybrid web recommender systems', in P. Brusilovsky, A. Kobsa & W. Nejdl (eds), *The Adaptive Web*, vol. 4321, Springer-Verlag, Berlin Heidelberg, pp. 377-408.

Burke, R.D., Hammond, K.J. & Young, B.C. 1996, 'Knowledge-based navigation of complex information spaces', paper presented to the *Proceedings of the Thirteenth National Conference on Artificial intelligence - Volume 1*, Portland, Oregon.

Cacheda, F., Carneiro, V., Fernández, D. & Formoso, V. 2011, 'Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems', *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, pp. 1-33.

Cantador, I. 2008, 'Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach', Universidad Autonoma de Madrid.

Capuano, N., Gaeta, M., Ritrovato, P. & Salerno, S. 2014, 'Elicitation of latent learning needs through learning goals recommendation', *Computers in Human Behavior*, vol. 30, no. 0, pp. 663-73.

Card, S.K., Mackinlay, J.D. & Shneiderman, B. 1999, *Readings in information visualization: using vision to think*, Morgan Kaufmann.

Castro, J., Quesada, F.J., Palomares, I. & Martínez, L. 2015, 'A consensus‐driven group recommender system', *International Journal of Intelligent Systems*, vol. 30, no. 8, pp. 887-906.

Celma, Ò. & Serra, X. 2008, 'FOAFing the music: Bridging the semantic gap in music recommendation', *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 4, pp. 250-6.

Chao, D. & Forrest, S. 2003, 'Information immune systems', *Genetic Programming and Evolvable Machines*, vol. 4, no. 4, pp. 311-31.

Chen, C.-M. & Duh, L.-J. 2008, 'Personalized web-based tutoring system based on fuzzy item response theory', *Expert Systems with Applications*, vol. 34, no. 4, pp. 2298-315.

Chen, C.-M., Duh, L.-J. & Liu, C.-Y. 2004, 'A personalized courseware recommendation system based on fuzzy item response theory', *2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '04)*, IEEE, pp. 305-8.

Chen, C.-M., Lee, H.-M. & Chen, Y.-H. 2005, 'Personalized e-learning system using item response theory', *Computers & Education*, vol. 44, no. 3, pp. 237-55.

Chen, G., Wang, F. & Zhang, C. 2009, 'Collaborative filtering using orthogonal nonnegative matrix tri-factorization', *Information Processing & Management*, vol. 45, no. 3, pp. 368-79.

Chen, X., Zheng, Z., Liu, X., Huang, Z. & Sun, H. 2013, 'Personalized QoS-aware web service recommendation and visualization', *IEEE Transactions on Services Computing*, vol. 6, no. 1, pp. 35-47.

Chen, Y.-L., Cheng, L.-C. & Chuang, C.-N. 2008, 'A group recommendation system with consideration of interactions among group members', *Expert Systems with Applications*, vol. 34, no. 3, pp. 2082-90.

Chen, Y., Ma, X., Cerezo, A. & Pu, P. 2014, 'Empatheticons: designing emotion awareness tools for group recommenders', *Proceedings of the 15th International Conference on Human Computer Interaction*, ACM, Puerto de la Cruz, Tenerife, Spain, pp. 1-8.

Chen, Y.L. & Cheng, L.C. 2010, 'An approach to group ranking decisions in a dynamic environment', *Decision Support Systems*, vol. 48, no. 4, pp. 622-34.

Chen, Z., Meng, X., Zhu, B. & Fowler, R.H. 2000, 'WebSail: from on-line learning to web search', *Proceedings of the First International Conference on Web Information Systems Engineering, 2000.*, vol. 1, pp. 206-13 vol.1.

Chesnevar, C.I. & Maguitman, A.G. 2004, 'ArgueNet: an argument-based recommender system for solving Web search queries', *2nd International IEEE Conference on Intelligent Systems*, vol. 1, pp. 282-7 Vol.1.

Cho, Y.H. & Kim, J.K. 2004, 'Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce', *Expert Systems with Applications*, vol. 26, no. 2, pp. 233-46.

Christensen, I.A. & Schiaffino, S. 2011, 'Entertainment recommender systems for group of users', *Expert Systems with Applications*, vol. 38, no. 11, pp. 14127-35.

Colombo-Mendoza, L.O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G. & Samper-Zapater, J.J. 2015, 'RecomMetz: a context-aware knowledge-based mobile recommender system for movie showtimes', *Expert Systems with Applications*, vol. 42, no. 3, pp. 1202-22.

Console, L., Torre, I., Lombardi, I., Gioria, S. & Surano, V. 2003, 'Personalized and adaptive services on board a car: an application for tourist information', *Journal of Intelligent Information Systems*, vol. 21, no. 3, pp. 249-84.

Cornelis, C., Guo, X., Lu, J. & Zhang, G. 2005, 'A fuzzy relational approach to event recommendation', *Proceedings of the Second Indian International Conference on Artificial Intelligence (IICAI-05)*, Pune, INDIA, pp. 2231-42.

Cornelis, C., Lu, J., Guo, X. & Zhang, G. 2007, 'One-and-only item recommendation with fuzzy logic techniques', *Information Sciences*, vol. 177, no. 22, pp. 4906-21.

Crossen, A., Budzik, J. & Hammond, K.J. 2002, 'Flytrap: intelligent group music recommendation', paper presented to the *Proceedings of the 7th International Conference on Intelligent User Interfaces*, San Francisco, California, USA.

Davidson, R. & Harel, D. 1996, 'Drawing graphs nicely using simulated annealing', *ACM Transactions on Graphics*, vol. 15, no. 4, pp. 301-31.

De la Rosa, J.L., Hormazabal, N., Aciar, S., Lopardo, G., Trias, A. & Montaner, M. 2011, 'A negotiation-style recommender based on computational ecology in open negotiation environments', *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2073-85.

De Meo, P., Quattrone, G. & Ursino, D. 2008, 'A decision support system for designing new services tailored to citizen profiles in a complex and distributed e-government scenario', *Data & Knowledge Engineering*, vol. 67, no. 1, pp. 161-84.

Deshpande, M. & Karypis, G. 2004, 'Item-based top-N recommendation algorithms', *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143-77.

Dwivedi, P. & Bharadwaj, K.K. 2015, 'e-Learning recommender system for a group of learners based on the unified learner profile approach', *Expert Systems*, vol. 32, no. 2, pp. 264-76.

Eades, P. 1984, 'A heuristic for graph drawing', *Congressus Numerantium*, vol. 42, pp. 149-60.

Eades, P. 1992, 'Drawing free trees', *Bulletin of the Institute for Combinatorics and its Applications*, vol. 5, pp. 10-36.

Esteban, B., Tejeda-Lorente, Á., Porcel, C., Arroyo, M. & Herrera-Viedma, E. 2014, 'TPLUFIB-WEB: a fuzzy linguistic Web system to help in the treatment of low back pain problems', *Knowledge-Based Systems*, no. 0.

Felfernig, A., Friedrich, G., Jannach, D. & Zanker, M. 2006, 'An integrated environment for the development of knowledge-based recommender applications', *International Journal of Electronic Commerce*, vol. 11, no. 2, pp. 11-34.

Felfernig, A., Gula, B., Leitner, G., Maier, M., Melcher, R. & Teppan, E. 2008, 'Persuasion in knowledge-based recommendation', in H. Oinas-Kukkonen, P. Hasle, M. Harjumaa, K. Segerståhl & P. Øhrstrøm (eds), *Persuasive Technology*, vol. 5033, Springer-Verlag, Berlin Heidelberg, pp. 71-82.

Fesenmaier, D.R., Ricci, F., Schaumlechner, E., Wöber, K. & Zanella, C. 2003, 'DIETORECS: Travel advisory for multiple decision styles', *Information and communication technologies in tourism*, vol. 2003, pp. 232-41.

Freyne, J., Smyth, B., Coyle, M., Balfe, E. & Briggs, P. 2004, 'Further experiments on collaborative ranking in community-based web search', *Artificial Intelligence Review*, vol. 21, no. 3-4, pp. 229-52.

Gansner, E., Hu, Y., Kobourov, S. & Volinsky, C. 2009, 'Putting recommendations on the map: visualizing clusters and relations', *Proceedings of the 3rd ACM Conference on Recommender Systems*, ACM, New York, New York, USA, pp. 345-8.

García-Crespo, A., Chamizo, J., Rivera, I., Mencke, M., Colomo-Palacios, R. & Gómez-Berbís, J.M. 2009, 'SPETA: Social pervasive e-Tourism advisor', *Telematics and Informatics*, vol. 26, no. 3, pp. 306-15.

Garcia, I. & Sebastia, L. 2014, 'A negotiation framework for heterogeneous group recommendation', *Expert Systems with Applications*, vol. 41, no. 4, Part 1, pp. 1245-61.

Garcia, I., Sebastia, L. & Onaindia, E. 2011, 'On the design of individual and group recommender systems for tourism', *Expert Systems with Applications*, vol. 38, no. 6, pp. 7683-92.

Garcia, I., Sebastia, L., Onaindia, E. & Guzman, C. 2009, 'A group recommender system for tourist activities', in T. Noia & F. Buccafurri (eds), *E-Commerce and Web Technologies*, vol. 5692, Springer Berlin Heidelberg, pp. 26-37.

Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S. & Seada, K. 2010a, 'Enhancing group recommendation by incorporating social relationship interactions', paper presented to the *Proceedings of the 16th ACM international conference on Supporting group work*, Sanibel Island, Florida, USA.

Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S. & Seada, K. 2010b, 'Enhancing group recommendation by incorporating social relationship

interactions', *Proceedings of the 16th ACM International Conference on Supporting Group Work*, ACM, Sanibel Island, Florida, USA, pp. 97-106.

Gavalas, D., Konstantopoulos, C., Mastakas, K. & Pantziou, G. 2014, 'Mobile recommender systems in tourism', *Journal of Network and Computer Applications*, vol. 39, pp. 319-33.

Gemmell, J., Schimoler, T., Ramezani, M., Christiansen, L. & Mobasher, B. 2009, 'Improving folkrank with item-based collaborative filtering', *Proceedings of the ACM RecSys'09 Workshop on Recommender Systems & the Social Web*, ACM, New York, NY, USA.

Goldberg, D., Nichols, D., Oki, B.M. & Terry, D. 1992, 'Using collaborative filtering to weave an information tapestry', *Communications of the ACM*, vol. 35, no. 12, pp. 61-70.

Goldberg, K., Roeder, T., Gupta, D. & Perkins, C. 2001, 'Eigentaste: A Constant Time Collaborative Filtering Algorithm', *Information Retrieval*, vol. 4, no. 2, pp. 133-51.

Gonzalez-Carrasco, I., Colomo-Palacios, R., Lopez-Cuadrado, J.L., Garcı´a-Crespo, Á. & Ruiz-Mezcua, B. 2012, 'PB-ADVISOR: a private banking multi-investment portfolio advisor', *Information Sciences*, vol. 206, no. 0, pp. 63-82.

Goren-Bar, D. & Glinansky, O. 2004a, 'FIT-recommend ing TV programs to family members', *Computers & Graphics*, vol. 28, no. 2, pp. 149-56.

Goren-Bar, D. & Glinansky, O. 2004b, 'FIT-recommending TV programs to family members', *Computers & Graphics*, vol. 28, no. 2, pp. 149-56.

Gorla, J., Lathia, N., Robertson, S. & Wang, J. 2013, 'Probabilistic group recommendation via information matching', *Proceedings of the 22nd International Conference on World Wide Web*, ACM, Rio de Janeiro, Brazil, pp. 495-504.

Gotz, D. & Wen, Z. 2009, 'Behavior-driven visualization recommendation', *Proceedings of the 14th International Conference on Intelligent User Interfaces*, ACM, Sanibel Island, Florida, USA, pp. 315-24.

Gretarsson, B., O'Donovan, J., Bostandjiev, S., Hall, C. & Höllerer, T. 2010, 'SmallWorlds: visualizing social recommendations', *Computer Graphics Forum*, vol. 29, no. 3, pp. 833-42.

Guarino, N. & Giaretta, P. 1995, 'Ontologies and knowledge bases: towards a terminological clarification', in N.J.I. Mars (ed.), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pp. 25-32.

Guo, X. & Lu, J. 2007, 'Intelligent e-government services with personalized recommendation techniques', *International Journal of Intelligent Systems*, vol. 22, no. 5, pp. 401-17.

Hamasaki, M., Goto, M. & Nakano, T. 2014, 'Songrium: a music browsing assistance service with interactive visualization and exploration of protect a web of music', *Proceedings of the 23rd International Conference on World Wide Web*, ACM, Seoul, Korea, pp. 523-8.

Hauver, D.B. & French, J.C. 2001, 'Flycasting: using collaborative filtering to generate a playlist for online radio', *First International Conference on Web Delivering of Music.*, pp. 123-30.

Hayes, C. & Cunningham, P. 2001, 'Smart radio—community based music radio', *Knowledge-Based Systems*, vol. 14, no. 3, pp. 197-201.

He, J. & Chu, W. 2010, 'A social network-based recommender system (SNRS)', in N. Memon, J.J. Xu, D.L. Hicks & H. Chen (eds), *Data Mining for Social Network Data*, vol. 12, Springer US, pp. 47-74.

Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R. & Kadie, C. 2001, 'Dependency networks for inference, collaborative filtering, and data visualization', *The Journal of Machine Learning Research*, vol. 1, pp. 49-75.

Herlocker, J., Konstan, J. & Riedl, J. 2002, 'An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms', *Information Retrieval*, vol. 5, no. 4, pp. 287-310.

Herlocker, J.L., Konstan, J.A., Borchers, A. & Riedl, J. 1999, 'An algorithmic framework for performing collaborative filtering', *The 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Berkeley, California, USA, pp. 230-7.

Herlocker, J.L., Konstan, J.A. & Riedl, J. 2000, 'Explaining collaborative filtering recommendations', *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, ACM, Philadelphia, Pennsylvania, USA, pp. 241-50.

Hernando, A., Bobadilla, J., Ortega, F. & Gutiérrez, A. 2013, 'Trees for explaining recommendations made through collaborative filtering', *Information Sciences*, vol. 239, pp. 1-17.

Hernando, A., Moya, R., Ortega, F. & Bobadilla, J. 2014, 'Hierarchical graph maps for visualization of collaborative recommender systems', *Journal of Information Science*, vol. 40, no. 1, pp. 97-106.

Hotho, A., Jäschke, R., Schmitz, C. & Stumme, G. 2006, 'Information Retrieval in Folksonomies: Search and Ranking', in Y. Sure & J. Domingue (eds), *The Semantic Web: Research and Applications*, vol. 4011, Springer Berlin Heidelberg, pp. 411-26.

Hu, X., Meng, X. & Wang, L. 2011, 'SVD-based group recommendation approaches: an experimental study of Moviepilot', *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation*, ACM, Chicago, Illinois, USA, pp. 23-8.

Hung-Wen, T. & Von-Wun, S. 2004, 'A personalized restaurant recommender agent for mobile e-service', *2004 IEEE International Conference on e-Technology, e-Commerce and e-Service. EEE '04*, pp. 259-62.

Hung, L.-p. 2005, 'A personalized recommendation system based on product taxonomy for one-to-one marketing online', *Expert Systems with Applications*, vol. 29, no. 2, pp. 383-92.

Hyeong-Joon, K. & Kwang-Seok, H. 2011, 'Personalized smart TV program recommender based on collaborative filtering and a novel similarity method', *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1416-23.

Jalali, M., Mustapha, N., Sulaiman, M.N. & Mamat, A. 2010, 'WebPUM: A Web-based recommendation system to predict user future movements', *Expert Systems with Applications*, vol. 37, no. 9, pp. 6201-12.

Jamali, M. & Ester, M. 2009, 'TrustWalker: a random walk model for combining trust-based and item-based recommendation', *The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Paris, France, pp. 397-406.

Jameson, A. 2004, 'More than the sum of its members: challenges for group recommender systems', *Proceedings of the Working Conference on Advanced Visual Interfaces*, ACM, Gallipoli, Italy, pp. 48-54.

Jameson, A., Baldes, S. & Kleinbauer, T. 2004, 'Two methods for enhancing mutual awareness in a group recommender system', *The Working Conference on Advanced Visual Interfaces*, ACM, Gallipoli, Italy, pp. 447-9.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L. & Stumme, G. 2007, 'Tag Recommendations in Folksonomies', in J. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič & A. Skowron (eds), *Knowledge Discovery in Databases: PKDD 2007*, vol. 4702, Springer Berlin Heidelberg, pp. 506-14.

Jia, R., Jin, M. & Liu, C. 2010, 'A new clustering method for collaborative filtering', *2010 International Conference on Networking and Information Technology (ICNIT)*, pp. 488-92.

Jøsang, A., Ismail, R. & Boyd, C. 2007, 'A survey of trust and reputation systems for online service provision', *Decision Support Systems*, vol. 43, no. 2, pp. 618-44.

Kaufmann, M. & Wagner, D. 2001, *Drawing graphs: methods and models*, Springer.

Kazienko, P., Musial, K. & Kajdanowicz, T. 2011, 'Multidimensional social network in the social recommender system', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 4, pp. 746-59.

Kim, J.K., Kim, H.K., Oh, H.Y. & Ryu, Y.U. 2010, 'A group recommendation system for online communities', *International Journal of Information Management*, vol. 30, no. 3, pp. 212-9.

Kim, Y.D. & Choi, S. 2009, 'Weighted nonnegative matrix factorization', *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1541-4.

Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H. & Newell, C. 2012, 'Explaining the user experience of recommender systems', *User Modeling and User-Adapted Interaction*, vol. 22, no. 4-5, pp. 441-504.

Kompan, M. & Bielikova, M. 2014, 'Group recommendations: survey and perspectives', *Computing and Informatics*, vol. 33, no. 2, pp. 446-76.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. & Riedl, J. 1997, 'GroupLens: applying collaborative filtering to Usenet news', *Communications of the ACM*, vol. 40, no. 3, pp. 77-87.

Koren, Y., Bell, R. & Volinsky, C. 2009, 'Matrix factorization techniques for recommender systems', *Computer*, vol. 42, no. 8, pp. 30-7.

Koutrika, G., Bercovitz, B. & Garcia-Molina, H. 2009, 'FlexRecs: expressing and combining flexible recommendations', *The 2009 ACM SIGMOD International Conference on Management of Data*, ACM, Providence, Rhode Island, USA, pp. 745-58.

Krulwich, B. 1997, 'Lifestyle finder: Intelligent user profiling using large-scale demographic data', *AI magazine*, vol. 18, no. 2, p. 37.

Lee, D.D. & Seung, H.S. 1999, 'Learning the parts of objects by non-negative matrix factorization', *Nature*, vol. 401, no. 6755, pp. 788-91.

Lee, D.D. & Seung, H.S. 2000, 'Algorithms for non-negativemMatrix factorization', *Advances in Neural Information Processing Systems*, Denver, Colorado, USA, pp. 556-62.

Lee, S.K., Cho, Y.H. & Kim, S.H. 2010, 'Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations', *Information Sciences*, vol. 180, no. 11, pp. 2142-55.

Lee, T., Chun, J., Shim, J. & Lee, S.-g. 2006, 'An ontology-based product recommender system for B2B marketplaces', *International Journal of Electronic Commerce*, vol. 11, no. 2, pp. 125-55.

Lemire, D. & Boley, H. 2003, 'RACOFI: a rule-applying collaborative filtering system', *International Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments*, eds A. Ghorbani & S. Marsh, NRC, Halifax, Nova Scotia, Canada, 2003.

Leung, W.-k.C. 2009, 'Enriching user and item profiles for collaborative filtering: from concept hierarchies to user-generated reviews', The Hong Kong Polytechnic University, Hong Kong.

Linden, G., Smith, B. & York, J. 2003, 'Amazon.com recommendations: item-to-item collaborative filtering', *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80.

Liu, D., Hua, X.S., Yang, L., Wang, M. & Zhang, H.J. 2009, 'Tag ranking', *Proceedings of the 18th International Conference on World Wide Web*, ACM, Madrid, Spain, pp. 351-60.

Lorenzi, F., Santos, F., Ferreira, P., Jr. & Bazzan, A.C. 2008, 'Optimizing Preferences within Groups: A Case Study on Travel Recommendation', in G. Zaverucha & A. Costa (eds), *Advances in Artificial Intelligence - SBIA 2008*, vol. 5249, Springer Berlin Heidelberg, pp. 103-12.

Loveymi, S. & Hamzeh, A. 2015, 'Proposing an evolutionary method based on maximization precision of group recommender systems', *Proceedings of the 7th Conference on Information and Knowledge Technology*, pp. 1-6.

Lu, J. 2004a, 'A personalized e-learning material recommender system', *Proceedings of the 2nd International Conference on Information Technology and Applications*, Harbin, China, CDROM.

Lu, J. 2004b, 'Personalized e-learning material recommender system', *Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)*, pp. 374-9.

Lu, J., Shambour, Q., Xu, Y., Lin, Q. & Zhang, G. 2010, 'BizSeeker: a hybrid semantic recommendation system for personalized government-to-business e-services', *Internet Research*, vol. 20, no. 3, pp. 342-65.

Lu, J., Shambour, Q., Xu, Y., Lin, Q. & Zhang, G. 2013a, 'A web-based personalized business partner recommendatiion system using fuzzy semantic techniques', *Computational Intelligence*, vol. 29, no. 1, pp. 37-69.

Lu, J., Shambour, Q., Xu, Y., Lin, Q. & Zhang, G. 2013b, 'A web-based personalized business partner recommendation system using fuzzy semantic techniques', *Computational Intelligence*, vol. 29, no. 1, pp. 37-69.

Lu, J., Wu, D., Mao, M., Wang, W. & Zhang, G. 2015, 'Recommender system application developments: A survey', *Decision Support Systems*, vol. 74, pp. 12-32.

Lucas, J.P., Luz, N., Moreno, M.N., Anacleto, R., Almeida Figueiredo, A. & Martins, C. 2013, 'A hybrid recommendation approach for a tourism system', *Expert Systems with Applications*, vol. 40, no. 9, pp. 3532-50.

Luo, X., Zhou, M., Xia, Y. & Zhu, Q. 2014, 'An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems', *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273-84.

Mao, M., Lu, J., Zhang, G. & Zhang, J. 2015, 'A fuzzy content matching-based e-commerce recommendation approach', *Proceedings of the 2015 IEEE International Conference on Fuzzy Systems*, IEEE, Istanbul, Turkey, pp. 1-8.

Markellou, P., Mousourouli, I., Sirmakessis, S. & Tsakalidis, A. 2005, 'Personalized e-commerce recommendations', *2005 IEEE International Conference on e-Business Engineering*, Beijing, China, pp. 245-52.

Marriott, K. & Sbarski, P. 2007, 'Compact layout of layered trees', *Proceedings of the 13th Australasian Conference on Computer Science*, Australian Computer Society, Inc., Darlinghurst, Australia, pp. 7-14.

Martinez, L., Rodriguez, R.M. & Espinilla, M. 2009, 'Reja: A georeferenced hybrid recommender system for restaurants', *IEEE/WIC/ACM 2009 International Joint Conferences on Web Intelligence and Intelligent Agent Technologies. WI-IAT '09.*, vol. 3, IET, pp. 187-90.

Masthoff, J. 2011, 'Group recommender systems: combining individual models', in F. Ricci, L. Rokach, B. Shapira & P.B. Kantor (eds), *Recommender Systems Handbook*, Springer US, pp. 677-702.

McCarthy, J.F. 2002a, 'Pocket Restaurant Finder: A situated recommender systems for groups', *Proceeding of Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, ACM, Minneapolis.

McCarthy, J.F. 2002b, 'Pocket RestaurantFinder: A situated recommender systems for groups', *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer System*, ACM, Minneapolis, Minnesota, USA.

McCarthy, J.F. & Anagnost, T.D. 1998, 'MusicFX: an arbiter of group preferences for computer supported collaborative workouts', *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, ACM, Seattle, Washington, USA, pp. 363-72.

McCarthy, K., McGinty, L., Smyth, B. & Salamó, M. 2006, 'The needs of the many: a case-based group recommender system', *Proceedings of the 8th European Conference on Case-Based Reasoning*, eds T. Roth-Berghofer, M. Göker & H.A. Güvenir, vol. 4106, Springer Berlin Heidelberg, Fethiye, Turkey, pp. 196-210.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B. & Nixon, P. 2006a, 'CATS: a synchronous approach to collaborative group recommendation', *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS-06)*, AAAI Press, Melbourne Beach, Florida, USA, pp. 86-91.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B. & Nixon, P. 2006b, 'CATS: a synchronous approach to collaborative group recommendation', *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, AAAI Press, Melbourne Beach, Florida, pp. 86-91.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B. & Nixon, P. 2006c, 'Group recommender systems: a critiquing based approach', paper presented to the *Proceedings of the 11th International Conference on Intelligent User Interfaces*, Sydney, Australia.

McLaughlin, M.R. & Herlocker, J.L. 2004, 'A collaborative filtering algorithm and evaluation metric that accurately model the user experience', *The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Sheffield, United Kingdom, pp. 329-36.

Melville, P., Mooney, R.J. & Nagarajan, R. 2002, 'Content-boosted collaborative filtering for improved recommendations', *Eighteenth National Conference on Artificial intelligence*, American Association for Artificial Intelligence, Edmonton, Alberta, Canada, pp. 187-92.

Middleton, S., Roure, D. & Shadbolt, N. 2009, 'Ontology-based recommender systems', in S. Staab & R. Studer (eds), *Handbook on Ontologies*, Springer Berlin Heidelberg, pp. 779-96.

Middleton, S.E., Shadbolt, N.R. & Roure, D.C.D. 2004a, 'Ontological user profiling in recommender systems', *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 54-88.

Middleton, S.E., Shadbolt, N.R. & Roure, D.C.D. 2004b, 'Ontological user profiling in recommender systems', *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 54-88.

Miller, B.N., Konstan, J.A. & Riedl, J. 2004, 'PocketLens: Toward a personal recommender system', *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 3, pp. 437-76.

Misue, K., Eades, P., Lai, W. & Sugiyama, K. 1995, 'Layout adjustment and the mental map', *Journal of Visual Languages and Computing*, vol. 6, no. 2, pp. 183-210.

Mobasher, B., Cooley, R. & Srivastava, J. 2000, 'Automatic personalization based on Web usage mining', *Communications of the ACM*, vol. 43, no. 8, pp. 142-51.

Mobasher, B., Jin, X. & Zhou, Y. 2004, 'Semantically enhanced collaborative filtering on the web', in B. Berendt, A. Hotho, D. Mladenič, M. Someren, M. Spiliopoulou & G. Stumme (eds), *Web Mining: From Web to Semantic Web*, vol. 3209, Springer Berlin Heidelberg, pp. 57-76.

Moreno, A., Valls, A., Isern, D., Marin, L. & Borràs, J. 2013, 'SigTur/E-Destination: ontology-based personalized recommendation of tourism and leisure activities', *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 633-51.

Moukas, A. 1997, 'Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem', *Applied Artificial Intelligence*, vol. 11, no. 5, pp. 437-57.

Moukas, A. & Maes, P. 1998, 'Amalthaea: An evolving multi-agent information filtering and discovery system for the WWW', *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 59-88.

Nanopoulos, A. 2011, 'Item recommendation in collaborative tagging systems', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 4, pp. 760-71.

Nguyen, Q.V. & Huang, M.L. 2003, 'Space-optimized tree: a connection+enclosure approach for the visualization of large hierarchies', *Information Visualization*, vol. 2, no. 1, pp. 3-15.

Nguyen, T., Lu, H. & Lu, J. 2013, 'Web-page recommendation based on web usage and domain knowledge', *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1041-4347.

Noguera, J.M., Barranco, M.J., Segura, R.J. & Martínez, L. 2012, 'A mobile 3D-GIS hybrid recommender system for tourism', *Information Sciences*, vol. 215, pp. 37-52.

O'Donovan, J., Smyth, B., Gretarsson, B., Bostandjiev, S. & Höllerer, T. 2008, 'PeerChooser: visual interactive recommendation', *Proceedings of the 2008 Conference on Human Factors in Computing Systems*, ACM, Florence, Italy, pp. 1085-8.

O'Connor, M., Cosley, D., Konstan, J. & Riedl, J. 2002, 'PolyLens: a recommender system for groups of users', in W. Prinz, M. Jarke, Y. Rogers, K. Schmidt & V. Wulf (eds), *European Conference on Computer Supported Cooperative Work 2001*, Springer Netherlands, pp. 199-218.

O'Connor, M., Cosley, D., Konstan, J.A. & Riedl, J. 2001, 'PolyLens: a recommender system for groups of users', *Proceedings of the 7th European Conference on*

Computer Supported Cooperative Work, Springer Netherlands, Bonn, Germany, pp. 199-218.

Pashtan, A., Blattler, R., Andi, A.H. & Scheuermann, P. 2003, 'CATIS: a context-aware tourist information system', paper presented to the *The 4th International Workshop of Mobile Computing*, Rostock.

Paterek, A. 2007, 'Improving regularized singular value decomposition for collaborative filtering', *Proceedings of KDD Cup Workshop at the 13th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 39-42.

Pazzani, M. 1999, 'A framework for collaborative, content-based and demographic filtering', *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393-408.

Pazzani, M. & Billsus, D. 2007, 'Content-based recommendation systems', in P. Brusilovsky, A. Kobsa & W. Nejdl (eds), *The Adaptive Web*, vol. 4321, Springer Berlin Heidelberg, pp. 325-41.

Pérez, I.J., Cabrerizo, F.J. & Herrera-Viedma, E. 2010, 'A mobile decision support system for dynamic group decision-making problems', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 6, pp. 1244-56.

Pessemier, T., Dooms, S. & Martens, L. 2013, 'Comparison of group recommendation algorithms', *Multimedia Tools and Applications*, pp. 1-45.

Popescu, G. & Pu, P. 2012, 'What's the best music you have?: designing music recommendation for group enjoyment in groupfun', *Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts*, ACM, Austin, Texas, USA, pp. 1673-8.

Purchase, H. 1997, 'Which aesthetic has the greatest effect on human understanding?', *Proceedings of the 5th International Symposium on Graph Drawing*, ed. G. DiBattista, Springer Berlin Heidelberg, Rome, Italy, pp. 248-61.

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B. & Recio-García, J.A. 2012a, 'Case-based aggregation of preferences for group recommenders', *Proceedings of the 20th International Conference on Case-based Reasoning*, eds B.D. Agudo & I. Watson, Springer Berlin Heidelberg, Lyon, France, pp. 327-41.

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B. & Recio-García, J.A. 2012b, 'A case-based solution to the cold-start problem in group recommenders', *Proceedings of the 20th International Conference on Case-Based Reasoning*, Springer Berlin Heidelberg, Lyon, France, pp. 342-56.

Quijano-Sánchez, L., Díaz-Agudo, B. & Recio-García, J.A. 2014, 'Development of a group recommender application in a Social Network', *Knowledge-Based Systems*, vol. 71, pp. 72-85.

Quijano-Sanchez, L., Recio-Garcia, J.A. & Diaz-Agudo, B. 2011, 'HappyMovie: a facebook application for recommending movies to groups', *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence*, pp. 239-44.

Quijano-Sanchez, L., Recio-Garcia, J.A. & Diaz-Agudo, B. 2014, 'An architecture and functional description to integrate social behaviour knowledge into group recommender systems', *Applied Intelligence*, vol. 40, no. 4, pp. 732-48.

Quijano-Sánchez, L., Recio-García, J.A. & Díaz-Agudo, B. 2010, 'Personality and socialt trust in group recommendations', *Proceedings fo the 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2, pp. 121-6.

Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B. & Jimenez-Diaz, G. 2013, 'Social factors in group recommender systems', *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 1, pp. 1-30.

Recio-Garcia, J.A., Jimenez-Diaz, G., Sanchez-Ruiz, A.A. & Diaz-Agudo, B. 2009, 'Personality aware recommendations to groups', *Proceedings of the 3rd ACM Conference on Recommender Systems*, ACM, New York, New York, USA, pp. 325-8.

Reingold, E.M. & Tilford, J.S. 1981, 'Tidier drawings of trees', *IEEE Transactions on Software Engineering*, vol. 7, no. 2, pp. 223-8.

Rennie, J.D.M. & Srebro, N. 2005, 'Fast maximum margin matrix factorization for collaborative prediction', *Proceedings of the 22nd International Conference on Machine Learning*, ACM, Bonn, Germany, pp. 713-9.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. 1994a, 'GroupLens: an open architecture for collaborative filtering of netnews', *The 1994 ACM Conference on Computer Supported Cooperative Work*, ACM, Chapel Hill, North Carolina, USA, pp. 175-86.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. 1994b, 'GroupLens: an open architecture for collaborative filtering of netnews', *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, ACM, Chapel Hill, North Carolina, United States, pp. 175-86.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. 1994c, 'GroupLens: an open architecture for collaborative filtering of netnews', paper presented to the *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, North Carolina, USA.

Resnik, P. 1995, 'Using information content to evaluate semantic similarity in a taxonomy', *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 1, Morgan Kaufmann Publishers Inc., Montreal, Quebec, Canada, pp. 448-53.

Ricci, F., Rokach, L. & Shapira, B. 2011, *Recommender Systems Handbook*, Springer.

Romero, C., Ventura, S., Zafra, A. & Bra, P.d. 2009, 'Applying web usage mining for personalizing hyperlinks in web-based adaptive educational systems', *Computers & Education*, vol. 53, no. 3, pp. 828-40.

Roy, S.B., Amer-Yahia, S., Chawla, A., Das, G. & Yu, C. 2010, 'Space efficiency in group recommendation', *The VLDB Journal*, vol. 19, no. 6, pp. 877-900.

Ruiz-Montiel, M. & Aldana-Montes, J. 2009, 'Semantically enhanced recommender systems', in R. Meersman, P. Herrero & T. Dillon (eds), *On the Move to*

Meaningful Internet Systems: OTM 2009 Workshops, vol. 5872, Springer-Verlag, Berlin Heidelberg, pp. 604-9.

Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T. & Hyvönen, E. 2013, 'SMARTMUSEUM: A mobile recommender system for the Web of Data', *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 20, no. 0, pp. 50-67.

Salam, M., Mccarthy, K. & Smyth, B. 2012, 'Generating recommendations for consensus negotiation in group personalization services', *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 597-610.

Salter, J. & Antonopoulos, N. 2006, 'CinemaScreen recommender agent: combining collaborative and content-based filtering', *Intelligent Systems, IEEE*, vol. 21, no. 1, pp. 35-41.

Sarwar, B., Karypis, G., Konstan, J. & Reidl, J. 2001, 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th International Conference on World Wide Web*, ACM, pp. 285-95.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. 2000, 'Application of dimensionality reduction in recommender systems–a case study', *In ACM WebKDD 2000 Workshop*.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. 2001, 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th International Conference on World Wide Web*, ACM, Hong Kong, China, pp. 285-95.

Schafer, J., Frankowski, D., Herlocker, J. & Sen, S. 2007, 'Collaborative filtering recommender systems', in P. Brusilovsky, A. Kobsa & W. Nejdl (eds), *The Adaptive Web*, vol. 4321, Springer-Verlag, Berlin Heidelberg, pp. 291-324.

Schiaffino, S. & Amandi, A. 2009, 'Building an expert travel agent as a software agent', *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 1291-9.

Shambour, Q. & Lu, J. 2010, 'A framework of hybrid recommendation system for government-to-business personalized e-services', *Seventh International Conference on Information Technology: New Generations (ITNG)*, pp. 592-7.

Shambour, Q. & Lu, J. 2011a, 'Government-to-business personalized e-services using semantic-enhanced recommender system', in K. Andersen, E. Francesconi, Å. Grönlund & T. Engers (eds), *Electronic Government and the Information Systems Perspective*, vol. 6866, Springer Berlin Heidelberg, pp. 197-211.

Shambour, Q. & Lu, J. 2011b, 'A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations', *2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 1, pp. 71-8.

Shambour, Q. & Lu, J. 2011c, 'A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services', *International Journal of Intelligent Systems*, vol. 26, no. 9, pp. 814-43.

Shambour, Q. & Lu, J. 2012, 'A trust-semantic fusion-based recommendation approach for e-business applications', *Decision Support Systems*, vol. 54, no. 1, pp. 768-80.

Sharon, T., Lieberman, H. & Selker, T. 2003, 'A zero-input interface for leveraging group experience in web browsing', *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ACM, Miami, Florida, USA, pp. 290-2.

Shi, J., Wu, B. & Lin, X. 2015, 'A latent group model for group recommendation', *Proceedings of the 2015 IEEE International Conference on Mobile Services*, pp. 233-8.

Smyth, B. 2007, 'Case-based recommendation', in P. Brusilovsky, A. Kobsa & W. Nejdl (eds), *The Adaptive Web*, vol. 4321, Springer Berlin Heidelberg, pp. 342-76.

Smyth, B. & Balfe, E. 2006, 'Anonymous personalization in collaborative web search', *Information Retrieval*, vol. 9, no. 2, pp. 165-90.

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M. & Boydell, O. 2004, 'Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web Search Engine', *User Modeling and User-Adapted Interaction*, vol. 14, no. 5, pp. 383-423.

Smyth, B. & Cotter, P. 2000a, 'A personalised TV listings service for the digital TV age', *Knowledge-Based Systems*, vol. 13, no. 2–3, pp. 53-9.

Smyth, B. & Cotter, P. 2000b, 'A personalized television listings service', *Communications of the ACM*, vol. 43, no. 8, pp. 107-11.

Sotelo, R., Blanco-Fernandez, Y., Lopez-Nores, M., Gil-Solla, A. & Pazos-arias, J.J. 2009, 'TV program recommendation for groups based on muldimensional TV-anytime classifications', *IEEE Transactions on Consumer Electronics*, vol. 55, no. 1, pp. 248-56.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. & Balakrishnan, H. 2003, 'Chord: a scalable peer-to-peer lookup protocol for internet applications', *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17-32.

Takács, G., Pilászy, I., Németh, B. & Tikk, D. 2009, 'Scalable collaborative filtering approaches for large recommender systems', *The Journal of Machine Learning Research*, vol. 10, pp. 623-56.

Terán, L. & Meier, A. 2010, 'A fuzzy recommender system for eElections', in K. Andersen, E. Francesconi, Å. Grönlund & T. van Engers (eds), *Electronic Government and the Information Systems Perspective*, vol. 6267, Springer Berlin Heidelberg, pp. 62-76.

Tintarev, N. & Masthoff, J. 2007, 'A survey of explanations in recommender systems', *Proceedings of the 23rd International Conference on Data Engineering Workshop*, pp. 801-10.

Verbert, K., Parra, D., Brusilovsky, P. & Duval, E. 2013, 'Visualizing recommendations to support exploration, transparency and controllability',

Proceedings of the 2013 International Conference on Intelligent User Interfaces, ACM, pp. 351-62.

Vildjiounaite, E., Kyllönen, V., Hannula, T. & Alahuhta, P. 2009, 'Unobtrusive dynamic modelling of TV programme preferences in a finnish household', *Multimedia Systems*, vol. 15, no. 3, pp. 143-57.

Wang, J.-C. & Chiu, C.-C. 2008, 'Recommending trusted online auction sellers using social network analysis', *Expert Systems with Applications*, vol. 34, no. 3, pp. 1666-79.

Wang, W., Lu, J. & Zhang, G. 2014, 'A new similarity measure-based collaborative filtering approach for recommender systems', in Z. Wen & T. Li (eds), *Foundations of Intelligent Systems*, vol. 277, Springer Berlin Heidelberg, pp. 443-52.

Wang, W., Makedon, F., Ford, J. & Zhang, S. 2006, 'Learning from incomplete ratings using non-negative matrix factorization', *Proceedings of the 6th SIAM International Conference on Data Mining*, SIAM, Bethesda, Maryland, USA, pp. 549-53.

Wang, W., Zhang, G. & Lu, J. 2015, 'Collaborative filtering with entropy‐driven user similarity in recommender systems', *International Journal of Intelligent Systems*, vol. 30, no. 8, pp. 854-70.

Wang, W., Zhang, G. & Lu, J. 2016, 'Member contribution-based group recommender system', *Decision Support Systems*, vol. 87, pp. 80-93.

Wang, X., Sun, L., Wang, Z. & Meng, D. 2012, 'Group recommendation using external followee for social TV', *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo*, IEEE, Melbourne, Australia, pp. 37-42.

Wongsuphasawat, K. & Gotz, D. 2012, 'Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization', *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2659-68.

Yang, W.-S. & Hwang, S.-Y. 2013, 'iTravel: a recommender system in mobile peer-to-peer environment', *Journal of Systems and Software*, vol. 86, no. 1, pp. 12-20.

Ye, M., Liu, X. & Lee, W.C. 2012, 'Exploring social influence for recommendation: a generative model approach', *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Portland, Oregon, USA, pp. 671-80.

Yee, K.-P., Fisher, D., Dhamija, R. & Hearst, M. 2001, 'Animated exploration of dynamic graphs with radial layout', *Proceedings of the 2001 IEEE Symposium on Information Visualization*, IEEE Computer Society, San Diego, CA, USA, p. 43.

Yu, Z., Zhou, X., Hao, Y. & Gu, J. 2006, 'TV program recommendation for multiple viewers based on user profile merging', *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63-82.

Zaiane, O.R. 2002, 'Building a recommender agent for e-learning systems', *Proceedings of 2002 International Conference on Computers in Education*, pp. 55-9 vol.1.

Zhang, W., Wang, J. & Feng, W. 2013, 'Combining latent factor model with location features for event-based group recommendation', *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Chicago, Illinois, USA, pp. 910-8.

Zhang, Z., Lin, H., Liu, K., Wu, D., Zhang, G. & Lu, J. 2013, 'A hybrid fuzzy-based personalized recommender system for telecom products/services', *Information Sciences*, vol. 235, no. 0, pp. 117-29.

Zheng, N. & Li, Q. 2011, 'A recommender system based on tag and time information for social tagging systems', *Expert Systems with Applications*, vol. 38, no. 4, pp. 4575-87.

Zhou, T., Bian, W. & Tao, D. 2013, 'Divide-and-conquer anchoring for near-separable nonnegative matrix factorization and completion in high dimensions', *Proceedings of the 13th IEEE International Conference on Data Mining*, IEEE Computer Society, Dallas, Texas, USA, pp. 917-26.

Abbreviations

CB	Content-Based
CBR	Case-Based Reasoning
CF	Collaborative Filtering
CS	Contribution Score
EM	Expectation Maximization
KB	Knowledge-Based
GRS	Group Recommender System
LCF	Local Collaborative Filtering
MAE	Mean Absolute Error
MF	Matrix Factorization
NMF	Non-negative Matrix Factorization
PCC	Pearson Correlation Coefficient
RS	Recommender System
SBS	SmartBizSeeker
SNMF	Separable Non-negative Factorization
SPCC	Sigmoid Function-Based PCC
SVD	Singular Value Decomposition
UGC	User-Generated Content
WNMF	Weighted Non-negative Matrix Factorization