A Study on Recommender Systems Based on Dempster-Shafer Theory

Van-Doan Nguyen

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

A Study on Recommender Systems Based on Demspter-Shafer Theory

Van-Doan Nguyen

Supervisor: Associate Professor Van-Nam Huynh

School of Knowledge Science
Japan Advanced Institute of Science and Technology

(March 2017)

Abstract

Recommender systems (RSs) have been developing rapidly since they were introduced in 1990s; and in practice, these systems have been applied in a variety of e-commerce applications. Usually, RSs provide rating domains representing as finite sets and allow users (customers) to evaluate items (products or services) with hard ratings which are known as single values in the sets. However, user preferences are subjective and qualitative; therefore, in some scenarios, representing user preferences as hard ratings is not suitable. Moreover, most previous studies on recommendation techniques have unfortunately neglected the important issue of imperfect information which may be present due to ambiguities and uncertainties in user ratings.

More recently, using soft ratings represented as subsets of a rating domain is considered to be a strategy to model not only subjective and qualitative information but also imperfect information about user preferences in RSs. According to the literature, RSs offering soft ratings are developed based on Dempster-Shafer theory (DST) which is known as one of the most general theories for modeling imperfect information. Furthermore, these days, communication and collaboration in social networks have become more and more convenient and frequent, and social relations in social networks can naturally influence individual behaviors as well as decisions including the ones on buying items. In this research, we have developed two novel collaborative filtering RSs based on DST, which exploit community context information and community preferences extracted from social networks for improving accuracy of recommendations. One of the developed systems is able to deal with the sparsity problem, and the other can overcome both the sparsity and cold-start problems.

In RSs based on DST, context information, community context information or community preference is employed for predicting unprovided ratings, and then both predicted and provided ratings are used for computing user-user similarities. As predicted ratings are not one hundred percent accurate, while the provided ratings are actually evaluated by users, in this research, we have proposed a new method for computing user-user similarities, in which provided ratings are considered more significant than predicted ones.

As observed, Dempster's rule is currently applied for combining information about user preferences in RSs based on DST. However, when using this method, the combined results usually contain many focal elements with very low probabilities and a few focal elements with high probabilities. The focal elements with very low probabilities can lead to unsatisfactory results in case of combining highly conflicting mass functions. Therefore, in this research, we have developed two new combination methods, called *2-probabilities focused* combination and *noise-averse* combination, which are capable of reducing the focal elements with very low probabilities. Moreover, Dempster's rule does not allow to combine totally conflicting mass functions which are common in RSs based on DST due to the diversity of users; thus, we have also developed two new mixed combination methods that support combining totally conflicting mass functions. In fact, the new combination methods developed in this research can be employed as useful tools for fusing information about user preferences from different sources in RSs based on DST.

Keywords: Recommender System, Dempster-Shafer Theory, Social Network, Information Fusion, Uncertain Reasoning.

# Acknowledgments

First of all, I would like to thank the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for awarding me a scholarship. Without such financial support from MEXT, I could not have completed this research. Additionally, I would like to thank Japan Advanced Institute of Science and Technology (JAIST) for offering me a professional research environment.

Then, I would like to express my respect and deep gratitude to Associate Professor Van-Nam Huynh from JAIST. He introduced me to JAIST and supervised my research. Importantly, he has taught me not only how to be a real researcher but also how to become better and stronger in life. The knowledge and experience that he conveyed to me will go along with me all of my life.

Besides my supervisor, I would like to thank the rest of members in the evaluation committee, Professor Mitsuru Ikeda, Professor Takashi Hashimoto, Professor Sadaaki Miyamoto, and Associate Professor Hieu Chi Dam, for their hard questions, insightful comments, and thoughtful suggestions. These questions, comments and suggestions were valuable for improving my research.

My sincere thanks again go to Professor Takashi Hashimoto for supervising me conducting the minor research. His vision and suggestions are precious for me. In addition, I would like to thank Mr. Guanhong Li, a PhD candidate in Hashimoto-Lab, for preparing the EEG data and supporting me in the minor research.

I would like to extend thanks to Dr. Tomoya Iwakura and Dr. Hiyori Yoshikawa for supervising me in the internship at Fujitsu Laboratories. The experience in working at Fujitsu Laboratories is really helpful for my research career.

Also, I would like to give my thanks to all members in Huynh-Lab for helping me in doing research as well as supporting me in my daily life.

Last but not least, I would like to express my gratitude to my parents for their endless love and encouragement; additionally, I especially thank my wife and little daughter for their support, patience and understanding.

Table of Contents

Abstract **i**

Acknowledgments **iii**

Table of Contents **iv**

List of Figures **vi**

List of Tables **viii**

List of Symbols **xi**

List of Abbreviations **xv**

1 Introduction **1**

2 Background and Literature Review **7**

2.1 Dempster-Shafer Theory 7

2.2 Literature Review 10

2.2.1 Overview of Recommendation Techniques 10

2.2.2 RSs Based on DST 14

2.2.3 Evaluation Criteria 15

3 Using Community Context Information **17**

3.1 Introduction 17

3.2 Data Modeling 18

3.3 Detecting Overlapping Communities 20

3.4 Performing on Communities 21

3.4.1 Generating Unprovided Ratings 21
3.4.2 Computing User-User Similarities 25
3.4.3 Selecting Neighborhoods 28
3.4.4 Estimating Ratings 28
3.5 Generating Recommendations 29
3.6 Experiment 29
3.6.1 Data Sets 29
3.6.2 Results of Experiment on Movielens Data Set 32
3.6.3 Results of Experiment on Flixster Data Set 37
3.7 Conclusion 41

4 Using Community Preferences 43

4.1 Introduction 43
4.2 Data Modeling 44
4.3 Performing in a Community 46
4.3.1 Extracting Community Preference 46
4.3.2 Overcoming the Sparsity Problem 49
4.3.3 Overcoming the Cold-Start Problem: New Items 51
4.3.4 Overcoming the Cold-Start Problem: New Users 52
4.4 Experiment 53
4.5 Conclusion 55

5 Two-Probabilities Focused Combination 57

5.1 Introduction 57
5.2 The Proposed Combination Method 60
5.3 Experiment 64
5.3.1 Experiment on Movielens Data Set 65
5.3.2 Experiment on Flixster Data Set 67
5.4 Conclusion 69

6 Noise-Averse Combination 73

6.1 Introduction 73
6.2 The Proposed Combination Method 74

6.3 Experiment 76

6.4 Conclusion 80

7 Mixed Rules of Combination 81

7.1 Introduction 81

7.2 Characteristics of Combining Information 82

7.3 Existing Combination Methods 84

7.3.1 Dempster's Rule of Combination 84

7.3.2 Smets's Rule of Combination 85

7.3.3 Yager's Rule of Combination 86

7.3.4 Dubois and Prade's Rule of Combination 87

7.3.5 Dubois and Prade's "hybrid" Rule of Combination 88

7.3.6 Averaging Rule of Combination 88

7.3.7 Summary 88

7.4 Two Mixed Rules of Combination 89

7.5 Experiment 91

7.6 Conclusion 97

8 Conclusion and Future Work 98

8.1 Conclusion 98

8.2 Suggestions for Future Research 99

List of Figures

1.1 The content of the dissertation . . . 5

2.1 Classification of recommendation techniques . . . 12
2.2 The process of recommendations in CoFiDS . . . 14

3.1 The context information influencing on users and items . . . 19
3.2 The process of recommendations in the proposed system . . . 19
3.3 The domains of w_1 and w_2 . . . 28
3.4 Visualizing overall MAE (Movielens data set) . . . 34
3.5 Visualizing overall *DS-MAE* (Movielens data set) . . . 34
3.6 Overall MAE versus K (Movielens data set) . . . 35
3.7 Overall *DS-MAE* versus K (Movielens data set) . . . 35
3.8 Visualizing overall MAE (Flixster data set) . . . 39
3.9 Visualizing overall *DS-MAE* (Flixster data set) . . . 40
3.10 Overall MAE versus K (Flixster data set) . . . 40
3.11 Overall *DS-MAE* versus K (Flixster data set) . . . 41

4.1 Underlying the social network in the proposed system . . . 45
4.2 The process of recommendations of the proposed system . . . 45
4.3 Community preference on item I_k regarding group $G_{p,q}$ . . . 46
4.4 Community preference on item I_k regarding concept C_p . . . 47
4.5 Community preference on item I_k . . . 48
4.6 Overall *DS-MAE* versus K . . . 55
4.7 Overall *DS-Precision* versus K . . . 56
4.8 Overall *DS-Recall* versus K . . . 56

5.1 Overall DS-MAE versus K (Movielens data set) 66
5.2 Overall time of computation versus K (Movielens data set) 66
5.3 Overall DS-MAE versus K (Flixster data set) 68
5.4 Overall time of computation versus K (Flixster data set) 69

6.1 Results evaluated by criterion *DS-MAE* versus K 77
6.2 Results evaluated by criterion DS-$Precision$ versus K 77
6.3 Results evaluated by criterion *DS-Recall* versus K 78
6.4 Results evaluated by criterion DS-$F1$ versus K 78
6.5 Overall time of computation versus K 79

7.1 Performances according to *DS-MAE* versus K 94
7.2 Performances according to *DS-Recall* versus K 94
7.3 Performances according to DS-$Precision$ versus K 95
7.4 Performances according to DS-$F1$ versus K 95

List of Tables

2.1 Modeling with imperfect information 8

3.1 Rating matrix 23
3.2 Ratings are represented as mass functions 23
3.3 The values of the reliability function $\mu(x_{i,k}, x_{j,k})$ 27
3.4 Overlapping communities in Flixster data set 32
3.5 Overall MAE versus w_1 and w_2 (Movielens data set) 33
3.6 Overall DS-MAE versus w_1 and w_2 (Movielens data set) 33
3.7 The comparison in hard decisions (Movielens data set) 36
3.8 The comparison in soft decisions (Movielens data set) 37
3.9 Overall MAE versus w_1 and w_2 (Flixster data set) 38
3.10 Overall DS-MAE versus w_1 and w_2 (Flixster data set) 38
3.11 The comparison in hard decisions (Flixster data set) 41
3.12 The comparison in soft decisions (Flixster data set) 42

4.1 Comparing two RSs based on DST 44
4.2 Experiment results 54

5.1 A combined result representing as mass function m 58
5.2 *Triplet* mass function $\bar{m}$ 59
5.3 Mass function m_1 59
5.4 Mass function m_2 59
5.5 *Triplet* mass function $\bar{m}_1^{(1)}$ 60
5.6 *Triplet* mass function $\bar{m}_1^{(2)}$ 60
5.7 *Triplet* mass function $\bar{m}_1^{(3)}$ 60
5.8 *Triplet* mass function $\bar{m}_2$ 60

5.9 *Triplet* mass function $\bar{m}^{(1)}$ 61
5.10 *Triplet* mass function $\bar{m}^{(2)}$ 61
5.11 *Triplet* mass function $\bar{m}^{(3)}$ 61
5.12 Mass function m'_1 62
5.13 Mass function m'_2 62
5.14 *2-probabilities focused* mass function $\ddot{m}'_1$ 63
5.15 *2-probabilities focused* mass function $\ddot{m}'_2$ 63
5.16 *2-probabilities focused* mass function $\ddot{m}'$ 64
5.17 *2-probabilities focused* mass function $\ddot{m}_1$ 64
5.18 *2-probabilities focused* mass function $\ddot{m}_2$ 65
5.19 *2-probabilities focused* mass function $\ddot{m}$ 65
5.20 Users belonging to four overlapping communities 70
5.21 *DS-MAE* varies with twenty four combinations (part 1) 71
5.22 *DS-MAE* varies with twenty four combinations (part 2) 72

7.1 Ratings on item I_k 83
7.2 Ratings on item I_t 86
7.3 Summary of analyzing popular combination methods 89
7.4 Results of combining two totally conflicting mass functions 90
7.5 Performance comparison according to *DS-MAE* 92
7.6 Changes of performances according to *DS-MAE* 96

List of Symbols

$\mathbf{U}$	Set of users
U_i	A user
M	The number of users
$\mathbf{I}$	Set of items
I_k	An item
N	The number of items
Θ	The rating domain
θ_i	A single rating value
L	The number of rating elements or preference labels in the rating domain Θ
2^{Θ}	The power set of the rating domain Θ
$\mathbf{C}$	Context information
P	The number of concepts
C_p	A concept
2^{C_p}	The power set of concept C_p
$G_{p,q}$	A group that belongs to concept C_p
Q_p	The number of groups in concept C_p
f_p	A function that identifies the groups (in concept C_p) in which a user is interested
g_p	A function that identifies the groups (in concept C_p) to which an item belongs
$\mathbf{R}$	The rating matrix
$\mathfrak{r}_{i,k}$	The hard rating of user U_i on item I_k
$\hat{\mathfrak{r}}_{i,k}$	The estimated hard rating of user U_i on item I_k
$r_{i,k}$	The soft rating of user U_i on item I_k
$\hat{r}_{i,k}$	The estimated soft rating of user U_i on item I_k
$\mathbf{r}_{i,k}$	The overall soft ratings of all members in neighborhood set $\mathcal{N}_{i,k}$ on item I_k
$\alpha_{i,k}$	A trust factor corresponding to the rating of user U_i on item I_k
$\sigma_{i,k}$	A dispersion factor corresponding to the rating of user U_i on item I_k

${}^{R}\mathbf{I}_i$	Set of items rated by user U_i
${}^{R}\mathbf{U}_k$	Set of users who have rated item I_k
$\mathbf{G}$	A graph representing a social network
$\mathbf{F}$	Set of friend relationships
$\boldsymbol{\mathcal{C}}$	Set of overlapping communities
$\mathcal{C}_v$	An overlapping community
V	The number of overlapping communities in graph $\mathbf{G}$
Max	The maximum number of users in a community
Min	The minimum number of users in a community
T	The maximum number of iteration in SLPA algorithm
${}^{G}m_{p,q}$	The community preference regarding group $G_{p,q}$
${}^{G}m_{k,p,q}$	The preference of users on item I_k regarding group $G_{p,q}$ (chapter 3) or the community preference on item I_k regarding group $G_{p,q}$ (chapter 4)
${}^{G}m_{i,k,p,q}$	The preference of user U_i on item I_k regarding group $G_{p,q}$
${}^{C}m_{k,p}$	The community preference on item I_k regarding concept C_p
${}^{C}m_{i,k,p}$	The preference of user U_i on item I_k regarding concept C_p
${}^{\mathbf{C}}m_k$	The community preference on item I_k regarding context $\mathbf{C}$
${}^{\mathbf{C}}m_{i,k}$	The preference of user U_i on item I_k regarding context $\mathbf{C}$
$\mathbf{\Theta}$	The cross product space $\mathbf{\Theta}$
$\boldsymbol{\theta}_i$	An element in the cross product space
$cyl_{\mathbf{\Theta}}$	The cylindrical extensions of focal elements of a soft rating to the cross product $\mathbf{\Theta}$
$M_{i,k}$	A mass function defined on the cross product $\mathbf{\Theta}$
M_i	The user-PBA of user U_i
m	A mass function defined on Θ
m'	A mass function defined on Θ
m_i	A mass function defined on Θ
$\bar{m}$	A *triplet* mass function defined on Θ
$\bar{m}_i$	A *triplet* mass function defined on Θ
$\ddot{m}$	A *2-probabilities focused* mass function defined on Θ
$\ddot{m}_i$	A *2-probabilities focused* mass function defined on Θ
$\ddot{m}'$	A *2-probabilities focused* mass function defined on Θ

$\ddot{m}'_i$	A *2-probabilities focused* mass function defined on Θ
$\dot{m}$	A *noise-averse* mass function defined on Θ
$\dot{m}_i$	A *noise-averse* mass function defined on Θ
δ	The probability of reliability of the source providing mass function m
Bp	The pignistic probability distributions according to mass function m
Bp_i	The pignistic probability distributions according to mass function M_i
$Bp_{i,k}$	The pignistic probability distributions according to mass function $r_{i,k}$
D	The distance between two user-BPAs M_i and M_j in CoFiDS
$\hat{D}$	The proposed distance between two user-BPAs M_i and M_j
CD	The Chan and Darwiche distance measure
μ	The reliability function (chapter 3)
	or mean (chapters 5 and 7)
$x_{i,k}$	The trust of evaluation of user U_i on item I_k
w_1	The reliability coefficient representing the state when a user has rated an item
w_2	The reliability coefficient representing the state when two users have rated an item
ψ	A monotonically decreasing function
$\mathbf{S}$	The user-user similarity matrix
$s_{i,j}$	The user-user similarity between two users U_i and U_j
τ	A user-user similarity threshold
$\mathcal{N}_{i,k}$	The neighborhood set of user U_i regarding item I_k
$\oplus$	Dempster's rule of combination
$\otimes$	Smets' rule of combination
$\boxplus$	Yager's rule of combination
$\circledcirc$	DP's rule of combination
$\ominus$	DP's "hybrid" rule of combination
$\circledast$	Averaging rule of combination
$\uplus$	*Two-probabilities focused* combination
$\mathbin{\rotatebox[origin=c]{180}{$\uplus$}}$	*Noise-averse* combination
$\boxminus$	Mixed 1 rule of combination
$\boxtimes$	Mixed 2 rule of combination
$\odot$	A general combination method

K	The maximum number of members in a neighborhood set
$\mathcal{K}$	The degree of conflicting of two mass functions
F	A focal set of a mass function
$F\prime$	The set of focal elements of a mass function excluding the whole set element
n	The number of elements in $F\prime$ (chapter 5)
	or the number of mass functions (chapter 5)
	or the number of *2-probabilities focused* mass functions (chapter 5)
	or the number of *noise-averse* mass functions (chapter 6)
ϵ	A infinitesimal threshold
η_1	A threshold of degree of conflicting between two mass functions
η_2	A threshold of degree of conflicting between two mass functions
A	A subset of the rating domain Θ
A_i	A subset of the rating domain Θ
B	A subset of the rating domain Θ
C	A subset of the rating domain Θ
r	A mass function defined on the rating domain Θ
r_i	A mass function defined on the rating domain Θ
β	A coefficient in evaluation method F_β
γ	A coefficient in the monotonically decreasing function ψ
p_i	The probability of a focal element

List of Abbreviations

BPA	Basic Probability Assignment
CoFiDS	Collaborative Filtering Based on Dempster-Shafer Belief-Theoretic Framework
DS	Dempster-Shafer
DST	Dempster-Shafer Theory
DP	Dubois and Prade
FoD	Frame of Discernment
IR	Tackling Incomparable Ratings problem
LPA	Label Propagation Algorithm
MAE	Mean Absolute Error
NSC	Tackling Non-Sense Combination Problem
RS	Recommender System
SD	Standard Deviation
SLPA	Speaker-Listener Label Propagation
TCC	Tackling Totally-Conflicting Combination Problem

Chapter 1

Introduction

As we have seen, on the one hand, in doing online business, online providers make efforts to suggest suitable items to online users in order to increase sales growth; on the other hand, while doing shopping on the Internet, online users want to not only share their opinions with one another but also be recommended items related to what they are looking for. Consequently, RSs [1, 2, 3, 4] have been developed to satisfy both online suppliers and online users. Commonly, RSs collect information about user preferences from multiple sources, estimate user preferences on unseen items, and then generate suitable recommendations to each user mainly based on estimated data. Logically, quality of recommendations depends on the way to model user preferences and accuracy of estimations.

According to the literature, most RSs provide rating domains represented as finite sets and allow users to express their preferences on items by hard ratings. Regarding online providers' point of view, a challenge of RSs is how to generate a short list of suitable items to each specific user among a huge number of potential items, whereas information about user preferences is commonly imperfect (uncertain, imprecise or incomplete). Even though a user has evaluated an item by using a hard rating, this rating might contain imperfect information. For example, let us consider a RS with a rating domain containing 5 elements and assume that a user has rated an item with a hard rating as 3; in this case, we cannot know exactly what this user thought about the item because the user probably wanted to evaluate the item as $\{2, 3\}$, $\{3, 4\}$ or $\{2, 3, 4\}$, but the system only allows to select one value, consequently, this user chose 3.

Additionally, different users can have different opinions about the same item, and naturally user preferences are qualitative [5]; therefore, each hard rating might encode subjective and qualitative information inside. According to previous studies, using soft ratings [6] is known not only as a strategy for modeling subjective, qualitative, and imperfect information but also as a more realistic and flexible way for representing user preferences. Soft ratings are represented as subsets of a rating domain; for instance, with a rating domain $\Theta = \{1, 2, 3, 4, 5\}$, a user can rate an item as $\{2, 3\}$ with a probability of 1.0, or $\{2\}$ with a probability of 0.3 and $\{3\}$ with a probability of 0.7.

More recently, RSs [6] based on DST [7, 8] have been studied and developed. Comparing to traditional RSs [1, 2, 3, 4] which represent user preferences as hard ratings, RSs [6, 9, 10] based on DST have some advantages such as (1) offering soft ratings, (2) modeling subjective, qualitative, and imperfect information about user preferences, and (3) supporting combining information about user preferences from different sources.

In this research, we aim at making an intensive study on RSs based on DST. The objectives of the research are as follows

- Exploiting information about user preferences from social networks to improve quality of recommendations in RSs based on DST
- Addressing the problems of fusing information about user preferences in RSs based on DST

The main contributions of this research are as below

- *Integrating RSs based on DST with social networks.* As observed, these days social networks are growing very fast as well as increasingly playing a significant role on the Internet; and these networks contain huge amount of information that could be useful for RSs. Practically, in a social network, users are naturally formed into communities whose members interact frequently with one another [11]; and, for example, when consulting for advice to buy a new item, people tend to believe in the recommendations from their relatives and friends in the same community rather than recommendations from anonymous users. Therefore, we have proposed to integrate RSs based on DST with social networks and use community context information about user preferences and community preferences extracted from the networks for reducing the imperfection of information about user preferences and dealing with the sparsity and cold-start problems.

- *Developing a new method for computing user-user similarities.* Commonly, RSs based on DST predict unprovided ratings, and then employ both predicted and provided ratings for computing user-user similarities. However, these systems consider the role of predicted ratings to be the same as that of provided ratings. Obviously, predicted ratings are not one hundred percent accurate because of being generated by computer programs, while provided ratings are actually evaluated by users; so it is unreasonable to treat these two kinds of ratings equally. Thus, we have developed a new method for computing user-user similarities, which considers the significant role of provided ratings to be higher than that of predicted ones.

- *Developing two new reducing combination methods.* In RSs based on DST, user preferences are represented as mass functions and tasks of combining mass functions are executed frequently [12]. In addition, Dempster's rule [7] is currently employed to combine mass functions in these systems. However, when using this combination method, the combined results usually contain a large number of focal elements with very low probabilities and a few focal elements with high probabilities. Moreover, the focal elements with very low probabilities can lead to unsatisfactory results [13, 14] in case of combining highly conflicting mass functions. Thus, we have developed two new combination methods that are capable of eliminating focal elements with very low probabilities. The first new method, called *2-probabilities focused* combination, concentrates on significant focal elements defined as the ones with probabilities in top two highest probabilities and ignores the other focal elements; and this method helps to (1) handle combining highly conflicting mass functions, (2) improve time of computation, and (3) overcome the weakness of an alternative method known as *2-points focused* combination [15, 16, 17]. Regarding the second new method, called *noise-averse* combination, focal elements whose probabilities are less than or equal to an infinitesimal threshold are considered as noise that may be caused by the process of fusing information, and then eliminated. *Noise averse* combination method also has the advantages which *2-probabilities focused* combination method possesses and especially can prevent loss of valuable information about user preferences.

- *Developing two new mixed rules of combination.* As mentioned above, Dempster's rule of combination is currently used for combing mass functions in RSs based on DST. However, this combination method does not allow to combine totally conflicting mass functions; thus, in the existing RSs based on DST, totally conflicting mass functions need to be eliminated in the data sets. In general, some people can express their preferences on an item with rating values that are completely different from the others. In other words, totally conflicting mass functions are common in the systems because of the diversity of users. In this research, we have also developed two new mixed rules of combination, which are capable of handling totally conflicting mass functions when combining information about user preferences in RSs based on DST.

- *Enriching knowledge science.* It can be seen that user preference is one kind of tacit knowledge. In this research, we have developed a new methodology for modeling this kind of knowledge, discovering and extracting the knowledge which is hidden in the process of human communication, gathering and synthesizing the knowledge from different sources, handling conflicting knowledge, justifying the knowledge, and creating new knowledge.

In the experiments, the prototype applications that implement the proposed RSs based on DST as well the proposed combination methods were built by using SQL Server 2012 Standard Edition and Visual Basic 6.0; and all experiments were conducted in the environment as follows

- Processor: Intel (R) Core (TM) i5-43000U CPU @1.90 GHz 2.50 GHz

- System type: 64-bit operating system, x64-based processor

- Installed memory (RAM): 4.00 GB

- Operation system: Windows 8.1 Enterprise

This dissertation contains 8 chapters as illustrated Figure 1.1. The content of each chapter is briefly described as follows

- Chapter 1 introduces the context of research, research objectives, and contributions.

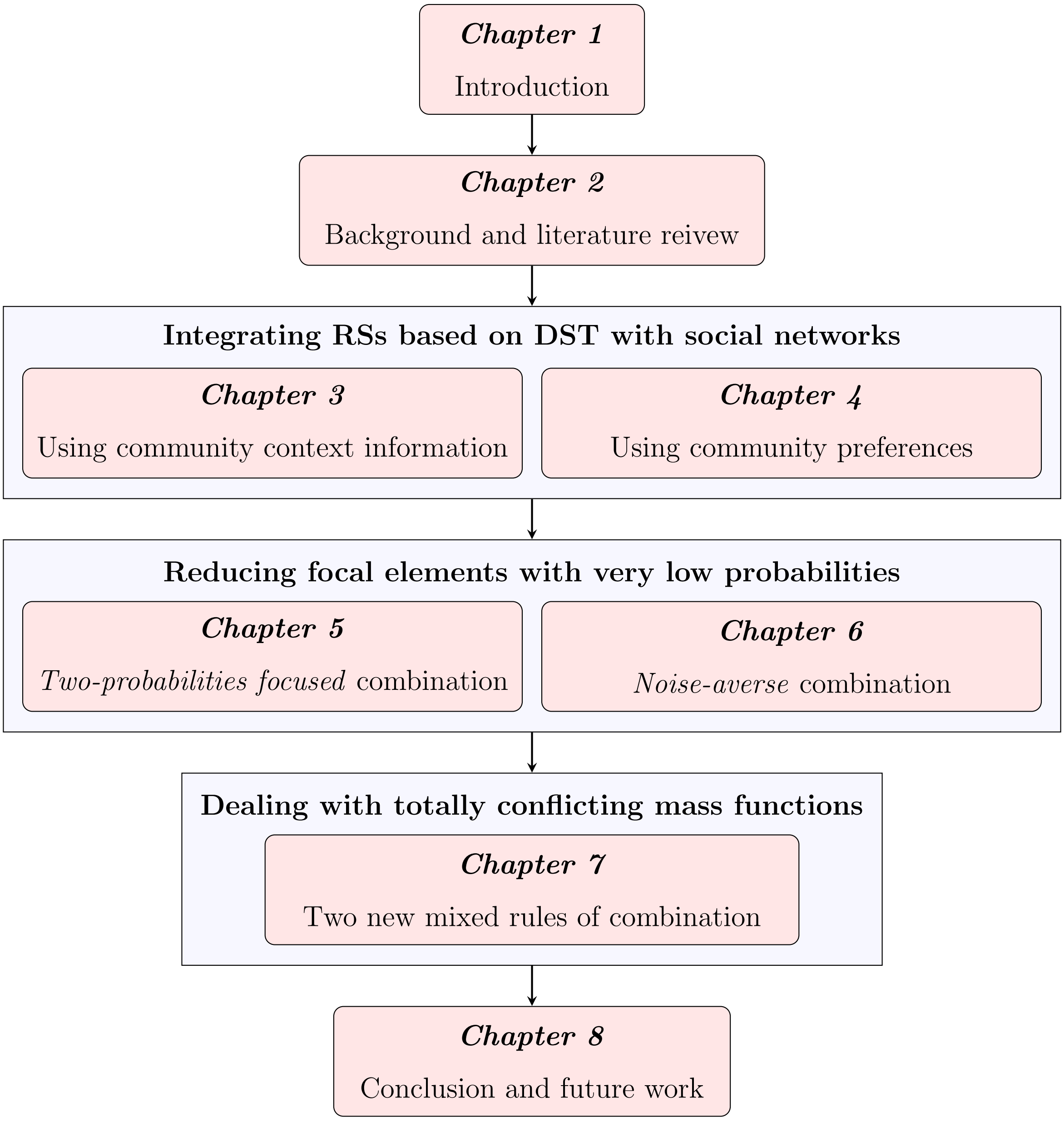

Figure 1.1: The content of the dissertation

- Chapter 2 provides background information about DST, and presents the literature review of RSs.

- Chapter 3 describes the novel RS based on DST, which exploits community context information extracted from the social network consisting of all users for reducing the imperfection of information about user preferences as well as overcoming the sparsity problem. Also, this chapter presents the new method to compute user-user similarities (*PRICAI 2014*; *ECSQARU 2015*; *IEEE Transactions on Systems, Man, and Cybernetics: Systems*).

- Chapter 4 presents the novel RS based on DST, which is capable of exploiting community preferences extracted from the social network for dealing with both the sparsity and cold-start problems (*CSoNet 2016*; *Knowledge-Based Systems*).

- Chapter 5 describes the new combination method called *2-probabilities focused* combination (*IUKM 2015*; *International Journal of Approximate Reasoning*).

- Chapter 6 presents the new combination method called *noise-averse* combination (*ICTAI 2016(1)*).

- Chapter 7 shows two new mixed rules of combination (*ICTAI 2016(2)*).

- Chapter 8 provides a summary of the dissertation as well as suggestions for the future research.

Chapter 2

Background and Literature Review

In this chapter, we first provide basic information about DST and its applications including RSs. Then, we present the literature review of RSs.

2.1 Dempster-Shafer Theory

Over the years, management of imperfect information has become increasingly important; and a number of mathematical theories have been developed for representing inperfect information, such as probability theory [18], possibility theory [19], rough set theory [20], DST [7, 8]. Most these approaches are capable of representing a specific aspect of imperfect information [21]. Importantly, among these, DST is considered to be the most general one [6, 22], as depicted in Table 2.1 (in this table, $r_{i,k}$ is a rating of a user on an item with a rating domain $\Theta = \{\theta_1, \theta_2, ..., \theta_L\}$).

DST, so-called evidence theory or theory of belief functions, is capable of modeling uncertain, imprecise, and incomplete information. Also, DST provides a powerful tool to combine information from multiple sources and arrive at a degree of belief, which is represented by a mathematical object called belief function that takes into account all the available information. So far, this theory has grown into an active research topic [23]. Particularly, DST has been applied in a large number of applications such as multiple attribute decision making [24, 25, 26], RSs [6, 9, 10], image processing [27, 28], computer vision and robotics [29, 30], filtering and tracking [31, 32, 33], classification [34, 35, 36, 37], clustering [38, 39, 40, 41], decision analysis [42, 43, 44], and so on.

Table 2.1: Modeling with imperfect information

Type of Imperfection	Proposition	$r_{i,k}$	Remarks
Hard	θ_i	1.0	
Probabilistic	θ_1	0.1	Singleton focal elements only
	θ_2	0.7	
	θ_3	0.2	
Possibilistic	θ_1	0.7	Consonant focal elements only
	(θ_1, θ_3)	0.2	
	Θ	0.1	
Ambiguity	(θ_1, θ_2)	1.0	Inability to discern among ratings
Vacuous	Θ	1.0	Missing/unknown entry
DST	$\sum\limits_{A \subseteq \Theta} r_{i,k}(A) =$	1.0	The most general

In the context of this theory, a problem domain is represented by a finite set $\Theta = \{\theta_1, \theta_2, ..., \theta_L\}$ of mutually exclusive and exhaustive hypotheses, called the frame of discernment (FoD) [8]. A function $m : 2^\Theta \rightarrow [0, 1]$ is called a mass function or basic probability assignment (BPA) if it satisfies $m(\emptyset) = 0$ and $\sum\limits_{A \subseteq \Theta} m(A) = 1$, where 2^Θ is the power set of Θ. Mass function m is vacuous if $m(\Theta) = 1$ and $\forall A \neq \Theta, m(A) = 0$. A subset $A \subseteq \Theta$ with $m(A) > 0$ is called a focal element of mass function m. Let us consider the example, adapted from [45]. Mr. Jones has been murdered; and we know that there are three suspects, Peter, Paul and Mary. The evidence that we have initially is that Mrs. Jones saw the murder going away, but she is short-signed and she only saw that it was a man. We also know that Mrs. Jones is drunk 20% of the time. When modeling the information in this situation by using DST, we have a set of hypotheses $\Theta = \{Peter, Paul, Mary\}$ and a mass function $m : 2^\Theta \rightarrow [0, 1]$, which represents the evidence, with $2^\Theta = \{\emptyset, \{Peter\}, \{Paul\}, \{Mary\}, \{Peter, Paul\}, \{Peter, Mary\}, \{Paul, Mary\},$ $\{Peter, Paul, Mary\}\}$. Since we know that 80% the murder is a man and we know nothing about the remaining probability, we have

$$\begin{aligned} m(\{Peter, Paul\}) &= 0.8; \\ m(\{Peter, Paul, Mary\}) &= 0.2. \end{aligned} \tag{2.1}$$

As we can see, the focal set of mass function m contains two focal elements $\{Peter, Paul\}$ and $\{Peter, Paul, Mary\}$.

For RSs, DST helps to model user preferences with uncertain, imprecise, and incomplete information. For example, in a RS with a rating domain $\Theta = \{1, 2, 3, 4, 5\}$, when asking a user to evaluate an item, this user can have some possible answers as below

1. "I will rate it as 4 and I am sure about it"; (*The rating is precise and certain*).
2. "I will rate it as 4 and I am 90% sure about it"; (*The rating is precise and uncertain*).
3. "I will rate it at least 4 and I am sure about it"; (*The rating is imprecise and certain*).
4. "I will rate it at least 4 and I am 90% sure about it"; (*The rating is imprecise and uncertain*).
5. "I will not rate it now"; (*The rating is incomplete*).

In this scenario, these answers can be modeled by mass functions, respectively, as follows

1. $r_1(\{4\}) = 1.0$.
2. $r_2(\{4\}) = 0.9; r_2(\Theta) = 0.1$.
3. $r_3(\{4, 5\}) = 1.0$.
4. $r_4(\{4, 5\}) = 0.9; r_4(\Theta) = 0.1$.
5. $r_5(\Theta) = 1.0$.

Furthermore, two useful operations that play a vital role in this theory are Dempster's rule of combination and discounting [8]. Let us consider two mass functions m_1 and m_2

defined on the same frame Θ. Dempster's rule of combination, denoted by $m = m_1 \oplus m_2$, can be used for combining these two mass functions as below

$$\begin{aligned} m(\emptyset) &= 0; \\ m(A) &= \frac{1}{1-\mathcal{K}} \sum_{B,C\subseteq\Theta, B\cap C=A} m_1(B)m_2(C); \\ \text{where } \mathcal{K} &= \sum_{B,C\subseteq\Theta, B\cap C=\emptyset} m_1(B)m_2(C) \neq 0. \end{aligned} \tag{2.2}$$

Here, $\mathcal{K}$ represents basic probability mass associated with conflict. As remarked in the literature, Dempster's rule serves as a powerful tool for fusing information from different sources [46]. The discounting operation is used when the information source providing mass function m has probability δ of reliability. In this case, one may adopt $1-\delta \in [0,1]$ as one's discount rate, resulting in a new mass function m^δ defined by

$$m^\delta(A) = \begin{cases} \delta m(A), & \text{for } A \subset \Theta; \\ \delta m(\Theta) + (1-\delta), & \text{for } A = \Theta. \end{cases} \tag{2.3}$$

Besides, regarding Smets' two-level view in the so-called transferable belief model [47, 48], when a decision needs to be made, the mass function m encoded the available evidence should be transformed into a probability distribution called pignistic probability function $Bp : \Theta \to [0,1]$ defined by

$$Bp(\theta_i) = \sum_{A\subseteq\Theta|\theta_i\in A} \frac{m(A)}{\mid A \mid} \tag{2.4}$$

In RSs, Dempster's rule of combination and discounting operations are useful for combining information about user preferences from different sources, and pignistic probability function can be used when the systems need to make a decision to provide a suitable recommendation to a user.

2.2 Literature Review

2.2.1 Overview of Recommendation Techniques

The roots of RSs can be traced back to the extensive work in cognitive science, approximation theories, information retrieval, forecasting theories; in addition, RS also have links to management science and to consumer choice modeling in marketing [1]. Since the mid-1990s, RSs has emerged as an independent research area [49, 50].

Specially, during the last two decades, developing RSs has become an active research topic [1]. RSs are known as software tools and techniques usually offering personalize recommendations to users [51, 52]. The recommendations are commonly represented as ranked lists of items; and in performing this ranking, RSs try to colect information about user preferences and estimate user preferences on unseen items [3]. For increasing accuracy of estimating user preferences on unseen items, RSs commonly try to gather information about user preferences from different sources. Naturally, the more sources of information about user preferences are available, the more effective estimations will be. As we can observe, there are two main methods to collect information about user preferences in RSs. The first is to obtain the information explicitly from user profiles [53, 54] or ratings [55, 56]. The second is to gather the information implicitly by monitoring user behaviors [57, 58, 59], or by extracting information from context [60, 61, 62] or social networks [63, 64].

For online users, RSs help to deal with information overload by providing a list of suitable items to each specific user [1]. For online providers, these systems are employed as an effective tool for increasing sale growths, selling more diverse items, improving the user satisfaction and fidelity, and better understanding what a user wants [3].

According to the literature, recommendation techniques can be classified into three categories [1, 65] as bellow

- Collaborative filtering recommendations: The recommendations are provided based on the assumption that if users shared the same interests in the past, they will also have similar tastes in the future.
- Content-based recommendations: Each user will be recommended items which are similar to the items this user preferred in the past.
- Hybrid approaches: A hybrid method is a combination of content-based and collaborative filtering methods for the purpose of overcoming certain limitations.

The classification of recommendation techniques is illustrated in illustrated in Figure 2.1.

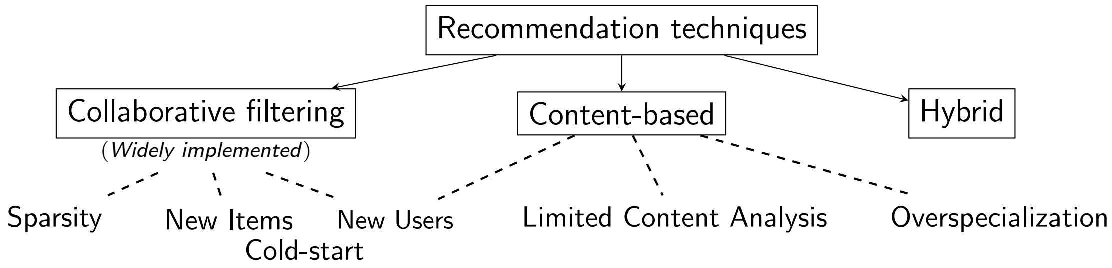

Figure 2.1: Classification of recommendation techniques

As remarked in the previous studies, among these three categories, collaborative filtering is considered to be the most popular and widely implemented approach [3, 66]. To generate suitable recommendations to an active user, collaborative filtering RSs try to find other users who share the same rating patterns with this user, and use their existing ratings for predicting preferences of this user on unseen items, after that generate suitable recommendations based on the predicted preferences; in other words, these systems are capable of understanding and exploiting relationships between items that are linked implicitly through the users who have rated them [67]. However, collaborative filtering RSs suffer from two fundamental problems [1] which are briefly described as below

- Number of items in the RSs can be very large; but, each user only rates a small subset of them. As a consequence, rating matrixes are usually very sparse. This issue is known as the sparisty problem which is supposed to majorly affect performances of recommendations [68].

- When a new item has just added, the RSs do not have information about people's preferences on it; in such a case, it is difficult to recommend this item to users. Furthermore, when a new user has just joined in, the RSs have no knowledge about preference of this user; so, it is also difficult to generate suitable recommendations for him/her. Both new users and new items cause the cold-start problem.

So far, many researchers focus on solving the sparsity and cold-start problems in collaborative filtering RSs, and various methods have been developed for overcoming these problems. A popular method is known as matrix factorization [69, 70, 71] that identifys latent factors and employs these factors for predicting all unprovided ratings. Additionally, in [72], the authors proposed to combine collaborative filtering with content-based approaches for tackling these problems. Besides, some authors suggested using additional information from other sources, such as demographic information [73, 74, 75] or implicit preferences inferred from users' actions that relate to specific item [67]. In addition, as observed, social networks have become an effective communication medium which can connect a larger number of people; and integrating with social networks has emerged as an active research topic [76, 77]. Actually, social networks contain a large amount of information that can be useful for dealing with the sparsity and cold-start problems in collaborative filtering RSs. Up to now, a variety of collaborative filtering RSs have been developed based on social networks [76, 78, 79]; and most of these systems employ social trust [80, 81, 82] for overcoming the sparsity and cold-start problems.

The content-based recommendations have their roots in information retrieval and information filtering search; and content-based RSs suggest items whose features are similar to the features of items which the user liked in the past. The basic process performed by a content-based RS consists in matching up the features of a user profiles with the features of a content item [3]. In fact, content-based RSs have several limitations [1], as follows

- The new users who have rated very few or no items would not be able to get accurate recommendations. This issue is also known as new user problem.

- In content-based RSs, the features need to be explicitly associated with items. Thus, to have a sufficient set of features, the content must either be in a form that can be parsed automatically by computer programs or features should be assigned to items manually. It can be seen that content-based RSs work well in extracting features form text documents, and these systems have an inherent problem with automatics feature extraction from other domains such as images, audio and video streams. This drawback in content-based RSs is called limited content analysis.

- In content-based RSs, a user is usually limited to being recommend items that are similar to those already rated. This issue is known as the overspecialization problem.

As mentioned earlier, most RSs provide rating domains represented as finite sets and allow users to express their preferences on items by hard ratings and each hard rating can encode qualitative, subjective and imperfect information about user preferences. In addition, in some cases, hard ratings are not suitable, as illustrated in Example 1.

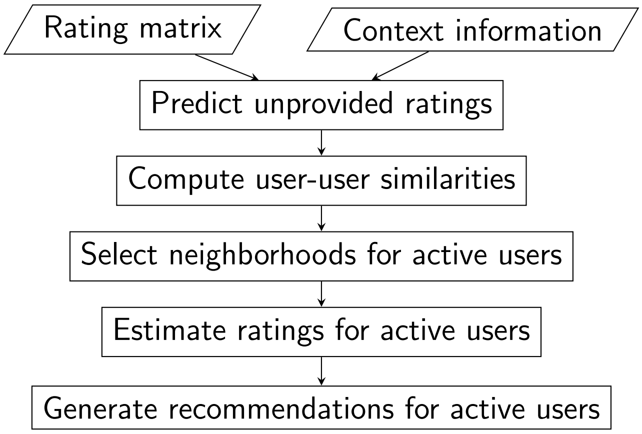

Figure 2.2: The process of recommendations in CoFiDS

RSs [6] based on DST [7, 8] have been studied and developed. Comparing to traditional RSs [1, 2, 3, 4] which represent user preferences as hard ratings, RSs [6, 9, 10] based on DST have some advantages such as (1) offering soft ratings, (2) modeling subjective, qualitative, and imperfect information about user preferences, and (3) supporting combining information about user preferences from different sources.

Example 1 Let us consider a RS offering hard ratings with a rating domain containing 5 elements $\Theta = \{1, 2, 3, 4, 5\}$ and assume that a user has rated two items I_1 and I_2 as 3 and 4 respectively. Now, this user would like to rate an item I_2 with a rating value representing that item I_2 is better than item I_1 but worse than item I_3. However, the user can only rate item I_2 with a rating value in the rating domain, such as 3 or 4; that means hard ratings are not suitable in this scenario.

2.2.2 RSs Based on DST

More resently, RSs [6] based on DST have been studied and developed. Comparing to traditional collaborative filtering RSs [1, 3, 83], the systems based on DST have serveral advantages [84] such as offering soft ratings, modeling user preferences with not only uncertain, imprecise, and incomplete but also subjective and qualitative information, and combining information about user preferences from different sources easily.

Particularly, it is seen that CoFiDS [6] could be considered as a pioneer RS that offers soft ratings. In addition, this system overcomes the sparsity problem by exploiting context information for generating unprovided ratings and deals with imperfect information by using DST for modeling ratings. The general process of recommendations in CoFiDS consists of 5 steps as illustrated in Figure 2.2. First, unprovided ratings are predicted by using the context information. Then, user-user similarities are calculated by employing both provided and predicted ratings. Next, for an active user, a neighborhood set according to each unrated item is selected, and the rating of this user on the item is estimated based on the ratings of his/her neighbors. Finally, the estimated ratings on all unrated items are ranked, and suitable items are selected to recommend to the active user.

However, CoFiDS considers predicted ratings to normally be the same as provided ones, and this system is not able to predict all unprovided ratings as it is expected. Moreover, in CoFiDS, the cold-start problem has not been solved and the other sources of information about user preferences, such as social networks, has not been exploited.

2.2.3 Evaluation Criteria

For evaluating RSs offering hard ratings and hard decisions, MAE (Mean Absolute Error) is widely used for performance evaluation. MAE measures the different between the true ratings and the corresponding estimated ratings as follows

$$MAE(\theta_j) = \frac{1}{\mid D_j \mid} \sum_{(i,k) \in D_j} \mid \mathfrak{r}_{i,k} - \hat{\mathfrak{r}}_{i,k} \mid \tag{2.5}$$

where D_j is the testing set identifying the user-item pairs whose true rating is $\mathfrak{r}_{i,k} = \theta_j \in \Theta$, $\hat{\mathfrak{r}}_{i,k} \in \Theta$ are the true rating and predicted rating of user U_i on item I_k respectively. In addition, when considering tasks of recommendation as calssfication tasks, triditional evaluations such as *Precision*, *Recall* and F_β [85] can be used.

Furthermore, most recently, researchers have developed some new assessment methods which are capable of measuring performances of RSs that offers soft ratings, such as *DS-Precision*, *DS-Recall* [22] and *DS-MAE*, *DS-*F_β [6]. Let us denote that the finally estimated rating (which is used for generating recommendation) of user U_i on item I_k is $\hat{r}_{i,k}$; and the pignistic probability distributions according to $\hat{r}_{i,k}$ is represented as $\widehat{Bp}_{i,k}$.

The new assessment methods can be described as below

$$\begin{aligned} DS\text{-}MAE(\theta_j) &= \frac{1}{\mid D_j \mid} \times \sum_{(i,k)\in D_j;\theta_l\in\Theta} \widehat{Bp}_{i,k}(\theta_l) \times \mid \theta_j - \theta_l \mid; \\ DS\text{-}Precision(\theta_j) &= \frac{TP(\theta_j)}{TP(\theta_j)+FP(\theta_j)}; \\ DS\text{-}Recall(\theta_j) &= \frac{TP(\theta_j)}{TP(\theta_j)+FN(\theta_j)}; \\ DS\text{-}F_\beta(\theta_j) &= \frac{(\beta^2+1)\times DS\text{-}Precision(\theta_j)\times DS\text{-}Recall(\theta_j)}{\beta^2\times DS\text{-}Precision(\theta_j)+DS\text{-}Recall(\theta_j)}; \end{aligned} \tag{2.6}$$

with D_j is the testing set identifying the user-item pairs whose true rating is $\theta_j \in \Theta$, $\beta \geq 1$, and

$$\begin{aligned} TP(\theta_j) &= \sum_{(i,k)\in D_j} \widehat{Bp}_{i,k}(\theta_j); \\ FP(\theta_j) &= \sum_{(i,k)\in D_l;j\neq l} \widehat{Bp}_{i,k}(\theta_j); \\ FN(\theta_j) &= \sum_{(i,k)\in D_j} \widehat{Bp}_{i,k}(\theta_l). \end{aligned} \tag{2.7}$$

Chapter 3

Using Community Context Information

3.1 Introduction

As mentioned previously, these days social networks are growing very fast and play a vital role on the Internet. Social networks are also known as an effective communication and collaboration medium that can connect many people. For RSs, these networks provide additional information about users and their friends, and this information can be used for not only better understanding user behaviors and ratings but also interpreting user preferences more precisely [77]. Moreover, in a social network, people are formed into some communities; and in each of which, members usually interact with one another very often [11] and share information about a variety of topics as well as products or services.

Under such an observation, in this chapter, we develop a novel collaborative filtering RS based on DST, which exploits community context information extracted from the social network containing all users for improving quality of recommendations. In this system, the extracted information is employed for predicting unprovided ratings, and then both predicted and provided ratings are used for computing user-user similarities. As predicted ratings are not one hundred percent accurate, while the provided ratings are actually evaluated by users, we also develop a new method for computing user-user similarities, in which provided ratings are considered more significant than predicted ones.

3.2 Data Modeling

Let $\mathbf{U} = \{U_1, U_2, ..., U_M\}$ be the set of all users and let $\mathbf{I} = \{I_1, I_2, ..., I_N\}$ be the set of all items. Note that in real-world applications, the number of elements in $\mathbf{U}$ and $\mathbf{I}$ can be very large. Each rating of user U_i on item I_k is defined as a mass function $r_{i,k} : 2^\Theta \to [0, 1]$ spanning over a rating domain $\Theta = \{\theta_1, \theta_2, ..., \theta_L\}$, a rank-order set of L preference labels, where $\theta_i < \theta_j$ whenever $i < j$. Additionally, Dempster's rule of combination is applied for combining mass functions in the system.

Each unprovided rating can be modeled by vacuous and considered as manifest lack of evidence. However, it can be seen that vacuous representation is high uncertain. Thus, community context information is employed for predicting unprovided ratings to reduce the uncertainty introduced by vacuous. Context information that might influence user preferences on items can be considered as a set of concepts for grouping users or items [6]. For example, in a movie RS, characteristics such as user gender, user occupation, movie genre are to be regarded as concepts. Each concept can consist of a number of groups, e.g. the movie genre might contain some groups such as drama, comedy, action, mystery, horror, animation. Supposing that, in the system, there are P characteristics which can be considered as concepts, and each concept C_p with $1 \leq p \leq P$ consists of a maximum of Q_p groups. Formally, context information is denoted by $\mathbf{C}$ and represented as follows

$$\begin{aligned} \mathbf{C} &= \{C_1, C_2, ..., C_P\}; \\ C_p &= \{G_{p,1}, G_{p,2}, ..., G_{p,Q_p}\}, \text{ where } p = \overline{1, P}. \end{aligned} \tag{3.1}$$

As illustrated in Figure 3.1, each user U_i can be interested in several groups and each item I_k may belong to some groups from the same concept. For a given concept C_p, the groups in which user U_i is interested are identified by mapping functions f_p as below

$$\begin{aligned} f_p &: \mathbf{U} \to 2^{C_p} \\ U_i &\mapsto f_p(U_i) \subseteq C_p; \end{aligned} \tag{3.2}$$

and the groups to which item I_k belongs are determined by mapping function g_p as follows

$$\begin{aligned} g_p &: \mathbf{I} \to 2^{C_p} \\ I_k &\mapsto g_p(I_k) \subseteq C_p, \end{aligned} \tag{3.3}$$

where 2^{C_p} is the power set of C_p [6].

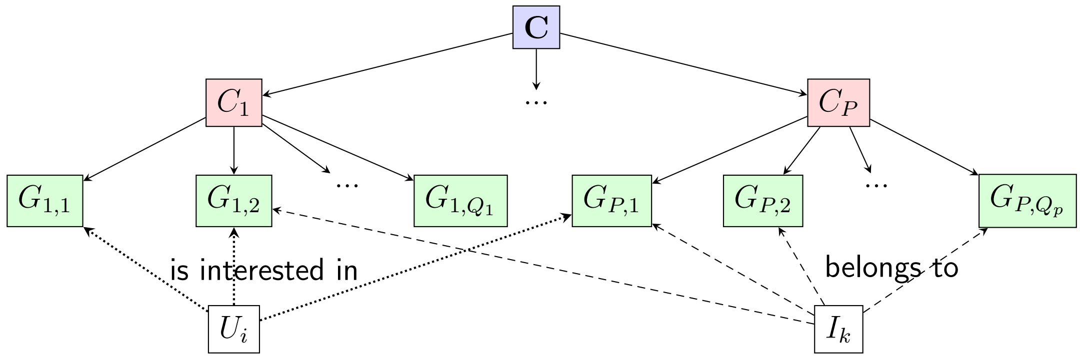

Figure 3.1: The context information influencing on users and items

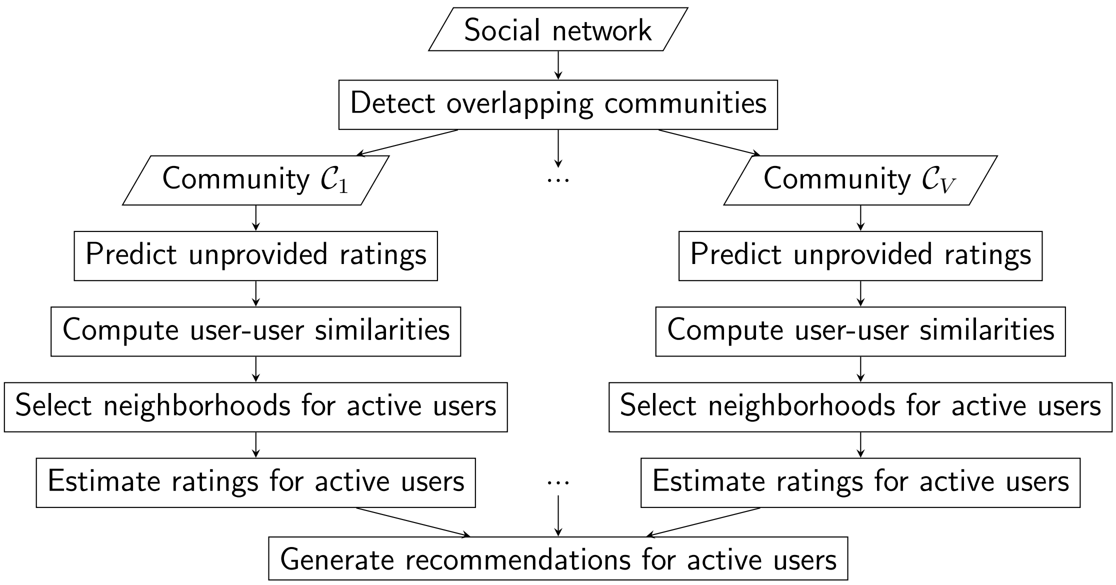

Figure 3.2: The process of recommendations in the proposed system

Basically, a social network can be considered to be a social structure made of nodes regarding individuals or organizations, and edges that connect nodes in various relations such as friendship or kinship [11]; and the social structure is usually represented by a graphical graph or an adjacency matrix. In addition, in the social network, an individual can simultaneously belong to a variety of communities. Here, we assume that all users join in a social network represented as an undirected graph $\mathbf{G} = (\mathbf{U}, \mathbf{F})$, with $\mathbf{U}$ is the set of all users (nodes) and $\mathbf{F}$ is the set of all friend relationships (edges).

Generally, the process of recommendations in the proposed system is illustrated in Figure 3.2. As can be seen in this figure, first, overlapping communities in the social network are detected; and assuming that after detecting, we achieve V overlapping communities. Second, tasks such as predicting unprovided ratings, computing user-user similarities, selecting neighborhoods and estimating ratings for active users on unseen items are performed in each community independently. Finally, the estimated ratings of each active user in the communities to which he/she belongs are combined, and then suitable recommendations to this user are generated mainly based on the combined results. In the rest of this section, details of tasks in this process will be presented.

3.3 Detecting Overlapping Communities

Over the years, numerous techniques have been developed for detecting communities in social networks, such as removal of high-betweenness [86, 87, 88], mimicking human pair wise communication [89], modularity optimization [90, 91], detection of dense sub-graphs [92]. In this research, we adopt Speaker-Listener Label Propagation (SLPA) algorithm [89] for naturally uncovering communities in the social network. The reason is that this algorithm is able to not only effectively detect overlapping communities in a large-scale network with the time complexity scaling linearly with the number of edges, but also avoid producing a number of small size communities.

SLPA algorithm is an extension of Label Propagation Algorithm (LPA) [93]. Regarding LPA algorithm, each node holds only a single label, but in SLPA algorithm, each node has a memory of the labels received in the past and takes its content to account to make the current decision. Additionally, SLPA algorithm simulates human pairwise communication, and in each communication step, one node is a speaker, and the other is a listener. Briefly, SLPA algorithm consists of three stages represented as follows

1. First, the memory of each node is initialized with a unique label.
2. Next, the following steps are repeated until the maximum iteration T is reached:
    - One node is selected as a listener.
    - Each neighbor of the listener randomly selects a label with probability proportional to the occurrence frequency of this label in its memory and sends the selected label to the listener.
    - The listener adds the most popular label received to its memory.
3. Finally, overlapping communities are identified based on the labels in the memories.

Note that some communities detected by using SLPA algorithm might consist of a large or a small number of users. Therefore, we continue applying this algorithm to separate the large communities into several smaller communities (if possible); and, for each member in the small communities, we assign it to the community containing most of its neighbors. Formally, let Min and Max be the minimum and maximum number users in a community which we expect to achieve. The communities consisting of more than Max or less than Min users are considered as large or small communities, respectively. The values of Min and Max can be selected according to each specific application.

We assume that, after executing SLPA algorithm, we get V overlapping communities, denoted by $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_V\}$. Next, as mentioned earlier, predicting unprovided ratings, computing user-user similarities, selecting neighborhoods and estimate ratings will be performed in each overlapping community independently.

3.4 Performing on Communities

Note that the tasks described in this subsection are performed in a community $\mathcal{C}_v \in \boldsymbol{\mathcal{C}}$.

3.4.1 Generating Unprovided Ratings

The ratings of all users in the community is denoted by a rating matrix $\mathbf{R} = \{r_{i,k}\}$ with $U_i \in \mathcal{C}_v$ and $k = \overline{1, N}$. The unprovided ratings in the matrix are generated mainly based on the method suggested in [6]. Let us consider that the items being rated by user U_i and the users who have rated item I_k are denoted by ${}^R\mathbf{I}_i = \{I_l \in \mathbf{I} | r_{i,l} \neq \text{vacuous}\}$ and ${}^R\mathbf{U}_k = \{U_l \in \mathbf{U} | r_{l,k} \neq \text{vacuous}\}$, respectively.

For a given concept C_p with $p = \overline{1, P}$, the preference of all users on item I_k regarding each group $G_{p,q} \in g_p(I_k)$, with $1 \leq q \leq Q_p$, defined by ${}^G m_{k,p,q} : 2^\Theta \to [0, 1]$, is calculated by combining provided ratings of users who are interested in group $G_{p,q}$ and have rated item I_k, as below

$$ {}^G m_{k,p,q} = \bigoplus_{\{i | I_k \in {}^R\mathbf{I}_i, G_{p,q} \in f_p(U_i), G_{p,q} \in g_p(I_k)\}} r_{i,k}. \tag{3.4} $$

Supposing that user U_i has not rated item I_k, the process for predicting unprovided rating $r_{i,k}$ regarding the preference of user U_i on item I_k is performed as follows

- First, since user U_i belongs to community $\mathcal{C}_v$, U_i's preference on item I_k is influenced by the preferences of members in this community. Additionally, it would appear that users who are interested in the same group of a given concept can be expected to possess similar preferences regarding this group. Under such an observation, for each concept C_p, if user U_i is interested in a group $G_{p,q} \in C_p$ then U_i's preference on item I_k regarding group $G_{p,p}$, denoted by ${}^G m_{i,k,p,q} : 2^\Theta \to [0,1]$, can be assigned the preference of users regarding group $G_{p,q}$ in community $\mathcal{C}_v$ on item I_k regarding group $G_{p,q}$, as follows
$$ {}^G m_{i,k,p,q} = {}^G m_{k,p,q}. \tag{3.5}$$

- Second, U_i's preference on item I_k regarding concept C_p, denoted by ${}^C m_{i,k,p} : 2^\Theta \to [0,1]$, is computed by combining preferences of user U_i on item I_k regarding group $G_{p,q}$ with $1 \leq q \leq Q_p$, as below
$$ {}^C m_{i,k,p} = \bigoplus_{\{q|G_{p,q} \in f_p(U_i), G_{p,q} \in g_p(I_k)\}} {}^G m_{i,k,p,q}. \tag{3.6}$$

- Next, U_i's preference on item I_k regarding context $\mathbf{C}$, denoted by ${}^{\mathbf{C}} m_{i,k} : 2^\Theta \to [0,1]$, is achieved by combining all preferences of user U_i on item I_k regarding group C_p with $1 \leq p \leq P$, as follows
$$ {}^{\mathbf{C}} m_{i,k} = \bigoplus_{p=\overline{1,P}} {}^C m_{i,k,p}. \tag{3.7}$$

- Finally, if context information $\mathbf{C}$ does not affect user U_i on item I_k, $f_p(U_i) \cap g_p(I_k) = \emptyset$ with $1 \leq p \leq P$, unprovided rating $r_{i,k}$ is assigned the result obtained by combining all existing ratings on item I_k as shown below
$$ r_{i,k} = \bigoplus_{\{j|U_j \in {}^R\mathbf{U}_k\}} m_{j,k}. \tag{3.8}$$

  Otherwise, unprovided rating $r_{i,k}$ is assigned the U_i's preference regarding context $\mathbf{C}$, as follows
$$ r_{i,k} = {}^{\mathbf{C}} m_{i,k}. \tag{3.9}$$

A demonstration of generating unprovided ratings is depicted in Example 2.

Table 3.1: Rating matrix

	I_k
U_1	4 for 90%
U_2	At least 3
U_3	
U_4	5 for 80%
U_5	

Table 3.2: Ratings are represented as mass functions

	I_k
U_1	$r_{1,k}(\{4\}) = 0.9; r_{1,k}(\Theta) = 0.1$
U_2	$r_{2,k}(\{3, 4, 5\}) = 1$
U_3	
U_4	$r_{3,k}(\{5\}) = 0.8; r_{3,k}(\Theta) = 0.2$
U_5	

Example 2 Let us consider a simple movie RS with a rating domain $\Theta = \{1, 2, 3, 4, 5\}$. Assuming that the context information in this system is represented as bellow

$$\begin{aligned} \mathbf{C} = \{C_1\} = \{Genre\}; \\ C_1 = \{G_{1,1}, G_{1,2}, G_{1,3}, G_{1,4}\} = \{Drama, Classics, Western, Horror\}. \end{aligned} \tag{3.10}$$

In this RS, let us consider an item I_k and a community containing four users U_1, U_2, U_3, U_4 and U_5. Supposing that the genres to which I_k belongs are as follows

$$g_1(I_k) = \{G_{1,3}, G_{1,4}\} = \{Western, Horror\}; \tag{3.11}$$

and the genres in which each user is interested are as below

$$\begin{aligned} f_1(U_1) = \{G_{1,1}, G_{1,3}\} = \{Drama, Western\}; \\ f_1(U_2) = \{G_{1,2}, G_{1,4}\} = \{Classics, Horror\}; \\ f_1(U_3) = \{G_{1,1}, G_{1,2}, G_{1,4}\} = \{Drama, Classics, Horror\}; \\ f_1(U_4) = \{G_{1,2}, G_{1,3}, G_{1,4}\} = \{Classics, Western, Horror\}; \\ f_1(U_5) = \{G_{1,1}, G_{1,2}\} = \{Drama, Classics\}. \end{aligned} \tag{3.12}$$

Additionally, assuming that the flexible ratings on item I_k are shown in Table 3.1. In the system, these ratings are represented as mass functions as illustrated in Table 3.2. It is seen that item I_k belongs to genre $Western$, and two users U_1 and U_4 who are also interested in genre $Western$ have rated this item; therefore, the ratings of users U_1 and U_4 are considered as pieces of the preference of users in the community on item I_k regarding genre $Western$. As a result, the preference of users on item I_k regarding genre $Western$ is obtained by combining all related pieces (using Equation (3.4)) as below

$$^{G}m_{k,1,3} = r_{1,k} \oplus r_{4,k}. \tag{3.13}$$

After computing, we have

$$\begin{aligned} ^{G}m_{k,1,3}(\{4\}) &\approx 0.643; \\ ^{G}m_{k,1,3}(\{5\}) &\approx 0.286; \\ ^{G}m_{k,1,3}(\Theta) &\approx 0.071. \end{aligned} \tag{3.14}$$

Similarly, the preference of users on item I_k regarding genre $Horror$ is computed as follows

$$^{G}m_{k,1,4} = r_{2,k} \oplus r_{4,k}; \tag{3.15}$$

and after computing we have

$$\begin{aligned} ^{G}m_{k,1,4}(\{5\}) &= 0.8; \\ ^{G}m_{k,1,4}(\{3,4,5\}) &= 0.2. \end{aligned} \tag{3.16}$$

As observed, two users U_3 and U_5 has not rated item I_k; thus, the unprovided ratings of these users, $r_{3,k}$ and $r_{5,k}$, need to be generated. The process to generate $r_{3,k}$ is as follows

- As we can see, $f_1(U_3) \cap g_1(I_k) = \{Horror\}$. In addition, since user U_3 is a member in the community, the preference of user U_3 on item I_k regarding genre $Horror$ is probably influenced by the preference of users in the community on item I_k regarding genre $Horror$. Then, the preference of user U_3 on item I_k regarding genre $Horror$ is calculated by using Equation (3.5) as below

$$^{G}m_{3,k,1,4} = \ ^{G}m_{k,1,4}. \tag{3.17}$$

- Then, the preference of user U_3 on item I_k regarding concept C_1, $Genre$, is achieved by applying Equation (3.6) as follows

$$^{C}m_{3,k,1} = \ ^{G}m_{3,k,1,4}. \tag{3.18}$$

- Next, the preference of user U_3 on item I_k regarding context $\mathbf{C}$ is obtained by applying Equation (3.7) as below

$$^{\mathbf{C}}m_{3,k} = {}^{G}m_{3,k,1}. \tag{3.19}$$

- Finally, using Equation (3.9), the unprovided rating $r_{3,k}$ is achieved as below

$$r_{3,k} = {}^{\mathbf{C}}m_{3,k}. \tag{3.20}$$

  In other words, we have

$$\begin{aligned} r_{3,k}(\{5\}) &= 0.8; \\ r_{3,k}(\{3,4,5\}) &= 0.2. \end{aligned} \tag{3.21}$$

As observed, the context information does on affect user U_5 on item I_k because $f_1(U_5) \cap g_1(I_k) = \emptyset$. Consequently, the unprovided rating $r_{5,k}$ is generated by using Equation (3.8) as follows

$$r_{5,k} = r_{1,k} \oplus r_{2,k} \oplus r_{4,k}. \tag{3.22}$$

After computing we get

$$\begin{aligned} r_{5,k}(\{4\}) &\approx 0.643; \\ r_{5,k}(\{5\}) &\approx 0.286; \\ r_{5,k}(\{3,4,5\}) &\approx 0.071. \end{aligned} \tag{3.23}$$

3.4.2 Computing User-User Similarities

In the rating matrix $\mathbf{R} = \{r_{i,k}\}$, each entry $r_{i,k}$ represents the user U_i's preference toward a single item I_k. The user U_i's preference toward all items as a whole can be represented by the cross-product FoD $\boldsymbol{\Theta} = \Theta_1 \times \Theta_2 \times ... \times \Theta_N$, where $\Theta_i = \Theta, \forall i = \overline{1,N}$ [6, 22]. Let us consider a focal element $A \subseteq \Theta$ of mass function $r_{i,k}$; the cylindrical extension of this focal element to the cross-product $\boldsymbol{\Theta}$ is $cyl_{\boldsymbol{\Theta}}(A) = [\Theta_1 ... \Theta_{i-1} A \Theta_{i+1} ... \Theta_N]$. The mapping $M_{i,k} : 2^{\boldsymbol{\Theta}} \to [0,1]$, where $M_{i,k}(B) = r_{i,k}(A)$ for $B = cyl_{\boldsymbol{\Theta}}(A)$ and 0 otherwise, generates a valid mass function defined on the FoD $\boldsymbol{\Theta}$ [22]. The mass function $M_i : 2^{\boldsymbol{\Theta}} \to [0,1]$, where $M_i = \bigoplus_{k=1}^{N} M_{i,k}$, is referred to as the user-BPA of user U_i in community $\mathcal{C}_v$ [6].

Let us consider user U_i's user-BPA M_i and mass functions $r_{i,k}$ with $k = \overline{1, N}$. The pignistic probability of a singleton $\theta_{i_1} \times ... \times \theta_{i_N} \in \boldsymbol{\Theta}$, is

$$Bp_i\left(\theta_{i_1} \times ... \times \theta_{i_N}\right) = \prod_{k=1}^{N} Bp_{i,k}(\theta_{i_k}), \tag{3.24}$$

where $\theta_{i_k} \in \Theta$, and Bp_i and $Bp_{i,k}$ are user U_i's pignistic probability distributions corresponding to its user-BPA and user preference mass function, respectively [6].

So as to compute distances among users, we use the distance measure method introduced in [94]. According to this method, the distance between two user-BPAs M_i and M_j defined over the same cross-product $\boldsymbol{\Theta}$ is $D(M_i, M_j) = CD(Bp_i, Bp_j)$, where Bp_i and Bp_j are the pignistic probability distributions corresponding to M_i and M_j, respectively, and CD refers to the Chan and Darwiche distance measure [94] represented as below

$$CD(Bp_i, Bp_j) = \ln \max_{\boldsymbol{\theta}_i \in \boldsymbol{\Theta}} \frac{Bp_j(\boldsymbol{\theta}_i)}{Bp_i(\boldsymbol{\theta}_i)} - \ln \min_{\boldsymbol{\theta}_i \in \boldsymbol{\Theta}} \frac{Bp_j(\boldsymbol{\theta}_i)}{Bp_i(\boldsymbol{\theta}_i)}. \tag{3.25}$$

The distance between the two user-BPAs M_i and M_j is

$$D(M_i, M_j) = \sum_{k=1}^{N} CD(Bp_{i,k}, Bp_{j,k}), \tag{3.26}$$

where $Bp_{i,k}$ and $Bp_{j,k}$ refer to the pignistic probability distributions corresponding to preference ratings of user U_i and U_j on item I_k, respectively [6].

According to equation (3.26), the role of each expression $CD(Bp_{i,k}, Bp_{j,k})$ is considered to be the same for all items regardless of whether ratings of users U_i and U_j on them are predicted or provided. Obviously, the expression based on provided ratings must be more reliable than the one based on predicted ratings. Moreover, for each item I_k, it is easy to recognize as follows

- In case neither user U_i nor user U_j has rated item I_k that means both ratings $r_{i,k}$ and $r_{j,k}$ are predicted ones. Since $Bp_{i,k}$ and $Bp_{j,k}$ are derived from ratings $r_{i,k}$ and $r_{j,k}$ respectively, the value of the expression $CD(Bp_{i,k}, Bp_{j,k})$ is not fully reliable in this case.

- The value of the expression $CD(Bp_{i,k}, Bp_{j,k})$ is also not fully reliable if either user U_i or user U_j has not rated item I_k.

Table 3.3: The values of the reliability function $\mu(x_{i,k}, x_{j,k})$

$x_{i,k}$	$x_{j,k}$	$\mu(x_{i,k}, x_{j,k})$
0	0	1
0	1	$1 - w_1$
1	0	$1 - w_1$
1	1	$1 - 2 \times w_1 - w_2$

- The value of the expression $CD(Bp_{i,k}, Bp_{j,k})$ is only fully reliable if both user U_i and U_j have rated item I_k.

In order to improve the accuracy of measuring the distance between two users, we need to consider the importance of provided ratings to be higher than that of predicted ones. To achieve this goal, we propose a new method to compute the distance between two user-BPAs M_i and M_j, as below

$$\hat{D}(M_i, M_j) = \sum_{k=1}^{N} \mu(x_{i,k}, x_{j,k}) \times CD(Bp_{i,k}, Bp_{j,k}), \tag{3.27}$$

where $\mu(x_{i,k}, x_{j,k}) \in [0, 1]$ is a reliability function referring to the trust of the evaluation of both user U_i and user U_j on item I_k. $\forall(i,k), x_{i,k} \in \{0,1\}$; $x_{i,k} = 1$ when U_i has rated I_k, otherwise U_i has not.

Since $\mu(x_{i,k}, x_{j,k}) \in [0, 1]$, the distinguishing of provided and predicted ratings does not destroy the elegance of the selected distance measure method [94]. When $\mu(x_{i,k}, x_{i,k}) < 1$ indicates that the distance between user U_i and user U_j is shorter than it actually is; that means user U_i has a higher opportunity for being a member in user U_j's neighborhood set, and vice versa.

In practice, the reliability function $\mu(x_{i,k}, x_{j,k})$ can be selected according to each specific application. In the general case, we suggest a reliability function as below

$$\mu(x_{i,k}, x_{j,k}) = 1 - w_1 \times (x_{i,k} + x_{j,k}) - w_2 \times x_{i,k} \times x_{j,k}, \tag{3.28}$$

where $w_1 \geq 0$ and $w_2 \geq 0$ are the reliability coefficients representing the state when a user has rated an item and two users together have rated an item, respectively. Because of $\forall(i,k), x_{i,k} \in \{0,1\}$, the function $\mu(x_{i,k}, x_{j,k})$ has to belong to one of four cases as

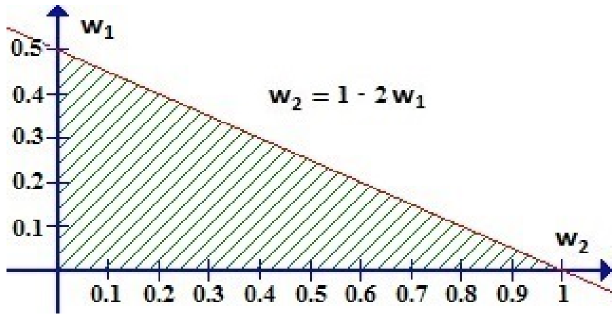

Figure 3.3: The domains of w_1 and w_2

shown in Table 3.3. In addition, under the condition $0 \leq \mu(x_{i,k}, x_{j,k}) \leq 1$, the domains of coefficients w_1 and w_2 must be in the parallel diagonal line shading area in Figure 3.3.

Consider a monotonically decreasing function ψ: $[0, \infty] \to [0, 1]$ satisfying $\psi(0) = 1$ and $\psi(\infty) = 0$. Then, with respect to ψ, $s_{i,j} = \psi(\hat{D}(M_{i,k}, M_{j,k}))$ is referred to as the user-user similarity between users U_i and U_j. To compute user-user similarities, we select the monotonically decreasing function as follows

$$\psi(x) = e^{-\gamma \times x}, \text{ where } \gamma \in (0, \infty). \tag{3.29}$$

Consequently, the user-user similarity matrix is generated as $\mathbf{S} = \{s_{i,j}\}$ with $U_i, U_j \in \mathcal{C}_v$.

3.4.3 Selecting Neighborhoods

We adopt the method proposed in [95] for selecting neighborhoods for active users. This method is effective because of the inclusion of two popular strategies known as K-nearest neighbor and minimum similarity shareholding. Formally, in order to select neighborhood set $\mathcal{N}_{i,k}$ for user U_i regarding item I_k, users who have rated item I_k and whose similarity with user U_i is equal or greater than a threshold τ is extracted. Next, K users with highest similarity with user U_i is selected from the extracted list.

3.4.4 Estimating Ratings

After obtaining neighborhood set $\mathcal{N}_{i,k}$, the rating $r_{j,k}$ of each neighbor $U_j \in \mathcal{N}_{i,k}$ is discounted by the user-user similarity $s_{i,j} \in \mathbf{S}$ between user U_i and U_j as follows

$$r_{j,k}^{s_{i,j}}(A) = \begin{cases} s_{i,j} \times r_{j,k}(A), & \text{for } A \subset \Theta; \\ s_{i,j} \times r_{j,k}(\Theta) + (1 - s_{i,j}), & \text{for } A = \Theta. \end{cases} \tag{3.30}$$

The estimated rating for user U_i on unrated item I_k is represented as below

$$\begin{aligned} \hat{r}_{i,k} &= r_{i,k} \oplus \mathbf{r}_{i,k}, \\ \text{where } \mathbf{r}_{i,k} &= \bigoplus_{\{j|U_j \in \mathcal{N}_{i,k}\}} r_{j,k}^{s_{i,j}}. \end{aligned} \tag{3.31}$$

3.5 Generating Recommendations

The suitable recommendations for an active user U_i are generated according to the number of communities to which this user belongs. If user U_i is a member of only one community $\mathcal{C}_v$, the finally estimated rating of this user on item I_k is user U_i's estimated rating of user U_i on item I_k in community $\mathcal{C}_v$. In case user U_i belongs to a variety of communities simultaneously, the finally estimated rating of this user on item I_k is achieved by combining the estimated ratings on item I_k in the communities to which user U_i belongs.

For a hard decision on a singleton $\theta_i \in \Theta$, the pignistic probability is applied, and then the singleton having the highest probability is selected as the preference label. In case a preference label (a singleton or a composite) is needed, the maximum belief with nonoverlapping interval strategy [96] is applied; if such as preference label can not be found, the decision is made according to the favor of composite preference label that has the maximum belief and those singletons have a higher plausibility [6].

3.6 Experiment

3.6.1 Data Sets

In the research area of RSs, Movielens 100K data set[1], collected by the GroupLens Research Project at the University of Minnesota, is widely used for measuring performances of recommendations. Thus, this data set was selected in the experiments.

Movielens 100K data set consists of 943 users, 1682 movies, and 100,000 hard ratings with a rating domain containing 5 elements $\Theta = \{1, 2, 3, 4, 5\}$ ($L = 5$). Each user has rated at least 20 movies. In this data set, context information, considered for grouping

[1]http://grouplens.org/datasets/movielens/

user, is represented as below

$$\begin{aligned}\mathbf{C} = \{C_1\} = \{Genre\};\\ C_1 = \{G_{1,1}, G_{1,2}, ..., G_{1,19}\} = \{Unknown, Action, Adventure, Animation,\\ Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film\text{-}Noir,\\ Horror, Musical, Mystery, Romance, Sci\text{-}Fi, Thriller, War, Western\}.\end{aligned} \tag{3.32}$$

However, information about genres in which a user interested is not available; thus, we assume that the genres in which a user is interested are assigned by genres of all items which have been rated by this user. In addition, each hard rating $\mathfrak{r}_{i,k} = \theta_l \in \Theta$ of user U_i on item I_k was transformed into the corresponding soft rating $r_{i,k}$ by the DS modeling function suggested in [6] as follows

$$r_{i,k}(A) = \begin{cases} \alpha_{i,k} \times (1 - \sigma_{i,k}), & \text{for } A = \{\theta_l\};\\ \alpha_{i,k} \times \sigma_{i,k}, & \text{for } A = B;\\ 1 - \alpha_{i,k}, & \text{for } A = \Theta;\\ 0, & \text{otherwise,} \end{cases} \tag{3.33}$$
$$\text{with } B = \begin{cases} \{\theta_1, \theta_2\}, & \text{if } l = 1;\\ \{\theta_{L-1}, \theta_L\}, & \text{if } l = L;\\ \{\theta_{l-1}, \theta_l, \theta_{l+1}\}, & \text{otherwise.} \end{cases}$$

Here, $\alpha_{i,k} \in [0, 1]$ and $\sigma_{i,k}$ are a trust factor and a dispersion factor according to the rating $r_{i,k}$, respectively [6]. The trust factor quantifies how likely the user assigned rating reflects the user's true perception; and the dispersion factor quantifies how likely the user assigned rating would span a larger set [6].

Note that the social network information is not available in Movielens 100K data set; thus, with this data set, we can only evaluate the impact of assigning weights to rating data in the proposed system. To measure the full ability of the proposed system, we selected Flixster data set[2].

Flixster data set contains friend relationships and hard ratings on movies with a rating domain containing 10 elements $\Theta = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$ $(L = 10)$.

[2]https://www.cs.ubc.ca/ jamalim/datasets/

However, in this data set the genres to which each movie belongs are not available. Thus we have enriched the data set by crawling the genres of movies.

After crawling and cleaning, we achieved a new Flixster data set that contains 49,410 friend relationships, 535,013 hard ratings from 3,827 users on 1210 movies. In the new data set, each user has rated at least 15 movies; and all the genres considered as context information are represented as below

$$\begin{aligned} \mathbf{C} &= \{Genre\}; \\ Genre &= \{Drama, Comedy, Action\ \&\ Adventure, Television, \\ &\quad Mystery\ \&\ Suspense, Horror, Science\ Fiction\ \&\ Fantasy, \\ &\quad KiDS\&\ Family, Art\ House\ \&\ International, Romance, Classics, \\ &\quad Musical\ \&\ Performing\ Arts, Anime\ \&\ Manga, Animation, Western, \\ &\quad Documentary, Special\ Interest, Sports\ \&\ Fitness, Cult\ Movies\}. \end{aligned} \tag{3.34}$$

In this data set, information about genres in which a user interested is not available; therefore, we also assume that the genres in which user $U_i \in \mathbf{U}$ is interested are assigned by genres of all items rated by user U_i. To transform each hard rating $\mathfrak{r}_{i,k} = \theta_l \in \Theta$ of user U_i on item I_k into the corresponding soft rating $r_{i,k}$, we applied the DS modeling function as shown below

$$r_{i,k}(A) = \begin{cases} \alpha_{i,k}(1-\sigma_{i,k}), & \text{for } A = \{\theta_l\}; \\ \frac{3}{5}\alpha_{i,k}\sigma_{i,k}, & \text{for } A = B; \\ \frac{2}{5}\alpha_{i,k}\sigma_{i,k}, & \text{for } A = C; \\ 1-\alpha_{i,k}, & \text{for } A = \Theta; \\ 0, & \text{otherwise}, \end{cases} \quad \text{where } B = \begin{cases} \{\theta_1, \theta_2\}, & \text{if } l = 1; \\ \{\theta_{L-1}, \theta_L\}, & \text{if } l = L; \\ \{\theta_{l-1}, \theta_l, \theta_{l+1}\}, & \text{otherwise}, \end{cases}$$

$$\text{and } C = \begin{cases} \{\theta_1, \theta_2, \theta_3\}, & \text{if } l = 1; \\ \{\theta_1, \theta_2, \theta_3, \theta_4\}, & \text{if } l = 2; \\ \{\theta_{L-3}, \theta_{L-2}, \theta_{L-1}, \theta_L\}, & \text{if } l = L-1; \\ \{\theta_{L-2}, \theta_{L-1}, \theta_L\}, & \text{if } l = L; \\ \{\theta_{l-2}, \theta_{l-1}, \theta_l, \theta_{l+1}, \theta_{l+2}\}, & \text{otherwise}, \end{cases} \tag{3.35}$$

and $\alpha_{i,k} \in [0, 1]$ is a trust factor, $\sigma_{i,k} \in [0, 1]$ is a dispersion factor [6].

Table 3.4: Overlapping communities in Flixster data set

Community IDs.	Total number of users
16	226
49	377
50	2749
86	712
90	1011
113	460
147	105

All users join in a social network whose nodes are connected by undirected friendships. After detecting communities from this social network by using SLPA algorithm, we obtained 7 overlapping communities, as illustrated in Table 3.4

To evaluate the proposed system on Movielens and Flixster data sets, we adopted all assessment methods mentioned in chapter 2. Moreover, we also selected CoFiDS for performance comparison.

3.6.2 Results of Experiment on Movielens Data Set

First, 10% of users in Movielens data set were randomly selected. Then, for each selected user, we randomly withheld 5 ratings, the withheld ratings were used as testing data and the remaining ratings were considered as training data. Finally, recommendations were computed for the testing data. We repeated this process for 10 times, and the average results of 10 splits were represented in this section. Additionally, in the experiments, some parameters were selected as follows: $\gamma = 10^{-4}$, $\beta = 1$, $\forall(i,k)\{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 2/9\}$.

Table 3.5 and 3.6 show the overall MAE and $DS\text{-}MAE$ criterion results computed by mean of these evaluation criteria with $K = 15, \tau = 0$ according to two coefficients w_1 and w_2, respectively. The statistics in these tables indicate that the performance of the proposed system is almost linearly dependent on values of coefficient w_1; this finding is the same for the other evaluation criteria. Coefficient w_2 just slightly influences the performance in hard decisions, but seems not to affect the performance in soft decisions;

Table 3.5: Overall MAE versus w_1 and w_2 (Movielens data set)

		w_1					
		0.0	0.1	0.2	0.3	0.4	0.5
	0.0	0.8361	0.8366	0.8363	0.8350	0.8342	0.8334
	0.1	0.8363	0.8363	0.8363	0.8347	0.8342	
	0.2	0.8366	0.8363	0.8361	0.8342	0.8342	
	0.3	0.8366	0.8363	0.8361	0.8339		
	0.4	0.8363	0.8363	0.8358	0.8339		
w_2	0.5	0.8363	0.8363	0.8355			
	0.6	0.8363	0.8361	0.8355			
	0.7	0.8363	0.8361				
	0.8	0.8361	0.8361				
	0.9	0.8358					
	1.0	0.8358					

Table 3.6: Overall $DS\text{-}MAE$ versus w_1 and w_2 (Movielens data set)

		w_1					
		0.0	0.1	0.2	0.3	0.4	0.5
	0.0	0.8406	0.8406	0.8405	0.8402	0.8399	0.8397
	0.1	0.8406	0.8406	0.8405	0.8401	0.8399	
	0.2	0.8406	0.8406	0.8404	0.8401	0.8400	
	0.3	0.8406	0.8405	0.8404	0.8400		
	0.4	0.8406	0.8405	0.8405	0.8400		
w_2	0.5	0.8406	0.8406	0.8404			
	0.6	0.8406	0.8405	0.8404			
	0.7	0.8406	0.8405				
	0.8	0.8406	0.8405				
	0.9	0.8405					
	1.0	0.8406					

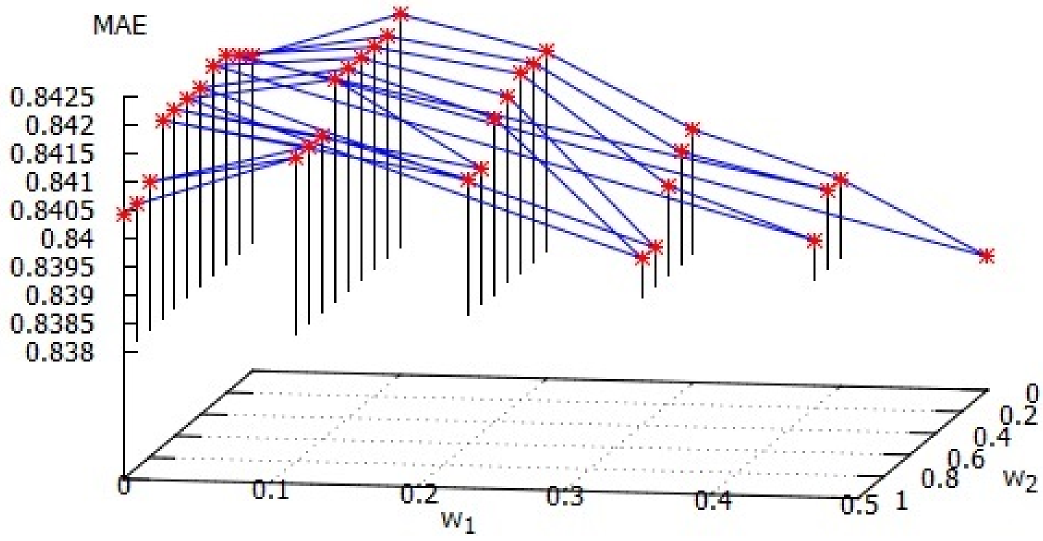

Figure 3.4: Visualizing overall MAE (Movielens data set)

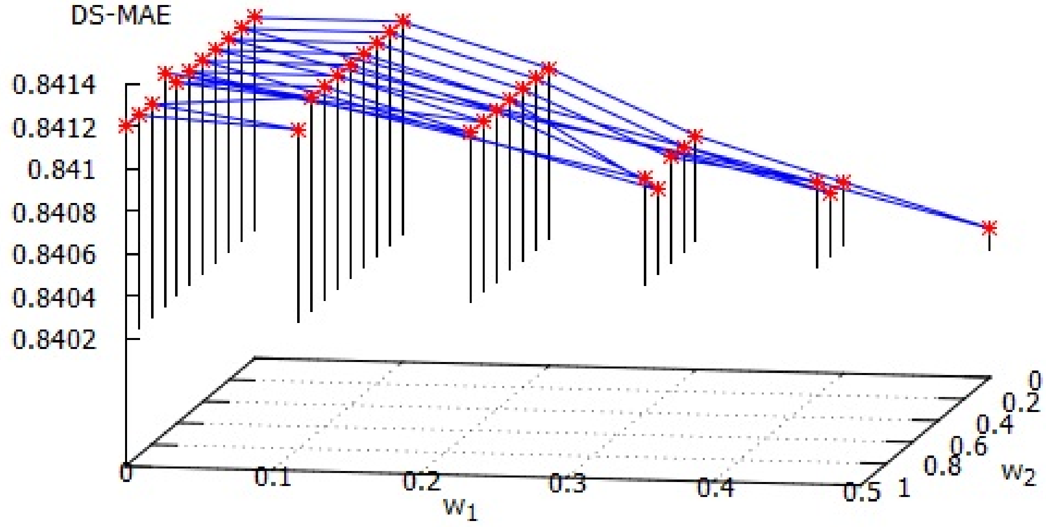

Figure 3.5: Visualizing overall $DS\text{-}MAE$ (Movielens data set)

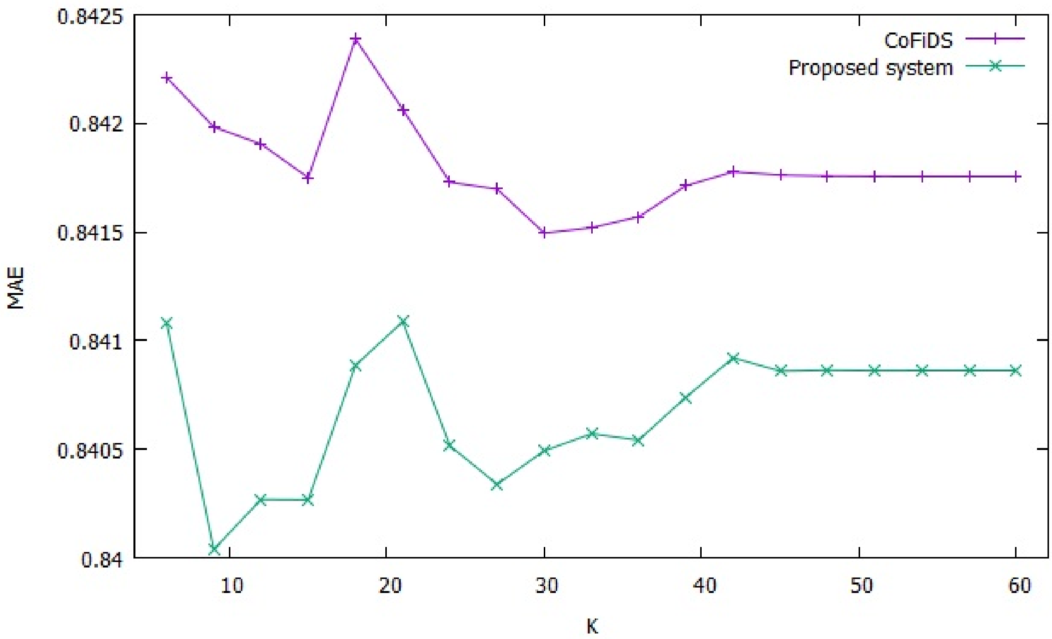

Figure 3.6: Overall MAE versus K (Movielens data set)

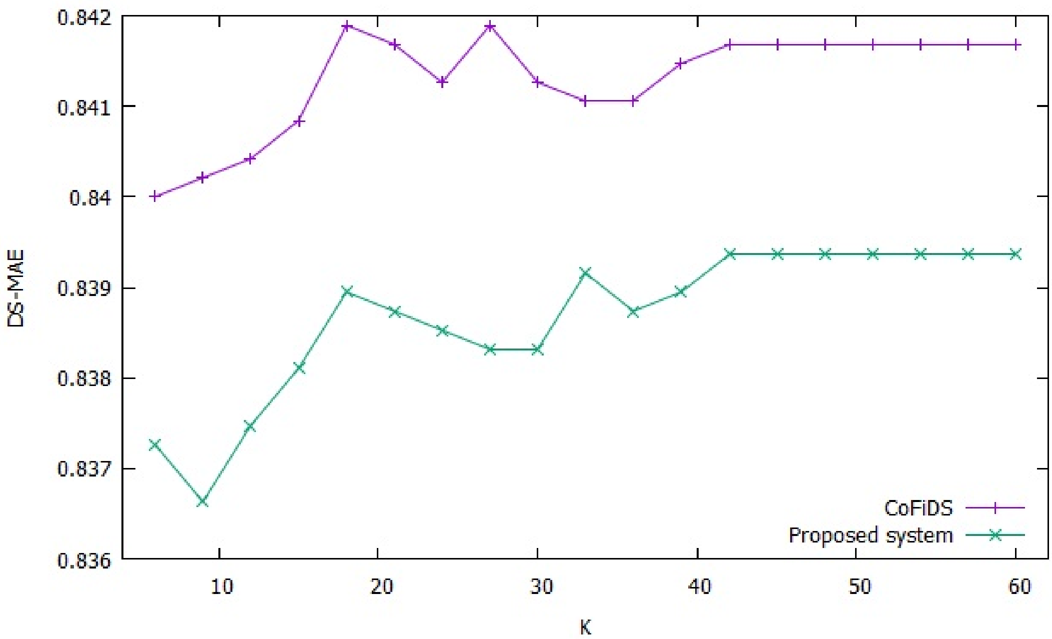

Figure 3.7: Overall DS-MAE versus K (Movielens data set)

Table 3.7: The comparison in hard decisions (Movielens data set)

Metric	True Rating 1	2	3	4	5	Overall
Proposed system:						
Precision	**0.3201**	0.2210	**0.3188**	**0.4002**	**0.4179**	**0.3630**
Recall	**0.0906**	**0.0892**	**0.3179**	0.6413	**0.1885**	**0.3709**
MAE	**2.1368**	**1.4242**	**0.7790**	**0.4212**	**1.0175**	**0.8383**
F_1	**0.1205**	**0.1245**	**0.3170**	**0.4924**	**0.2571**	**0.3384**
CoFiDS:						
Precision	0.3118	**0.2151**	0.3177	0.3996	0.4171	0.3609
Recall	0.0873	0.0872	0.3157	**0.6418**	0.1866	0.3697
MAE	2.1435	1.4325	0.7813	0.4224	1.0202	0.8413
F_1	0.1148	0.1216	0.3152	0.4921	0.2551	0.3366

the reason is that, in the data set, when considering two users, the number of movies rated by these two users is very small while the total number of movies is large. Figure 3.4 and Figure 3.5 depict the same information visually.

To compare with CoFiDS, we conducted the experiments with $w_1 = 0.5, w_2 = 0, \tau = 0$, and several values of K. Figure 3.6 and Figure 3.7 show the overall MAE and DS-MAE criterion results of both the proposed system and CoFiDS change with neighborhood size K. According to these figures, the performances of two systems are fluctuated when $K < 42$, and then appear to be stable with $K \geq 42$; in particular, both figures show that the proposed system is more effective than CoFiDS in all cases. Note that, with $K \geq 42$, members in neighborhood sets become stable; this leads to the stable performances as we can observe in Figure 3.6 and Figure 3.7.

Tables 3.7 and 3.8 show summarized results of the performance comparisons between the proposed system and CoFiDS in hard and soft decisions, respectively, with $K = 30, w_1 = 0.5, w_2 = 0$ and $\tau = 0$. In each category in these tables, every rating has its own column, the bold values indicate the better performance, and underlined values illustrate equal performance. Importantly, the statistics in both tables show that, except for soft decisions with true rating value $\theta_4 = 4$, the proposed system achieves better performance

Table 3.8: The comparison in soft decisions (Movielens data set)

DS-Metric	True Rating 1	2	3	4	5	Overall
Proposed system:						
Precision	**0.3001**	**0.2035**	**0.3150**	0.3990	0.4016	**0.3551**
Recall	**0.0663**	0.0926	**0.3164**	0.6391	**0.1847**	**0.3680**
MAE	**2.1963**	**1.4313**	**0.7721**	0.4122	**1.0317**	**0.8405**
F_1	**0.1036**	**0.1248**	**0.3147**	0.4909	**0.2507**	**0.3349**
CoFiDS:						
Precision	0.2926	0.2032	0.3148	0.3990	**0.4020**	0.3547
Recall	0.0658	**0.0934**	0.3155	**0.6398**	0.1837	0.3679
MAE	2.1973	1.4323	0.7724	**0.4118**	1.0359	0.8415
F_1	0.1028	0.1255	0.3141	**0.4911**	0.2500	0.3347

in all selected measurement criteria. However, absolute values of the performance of the proposed system are just slightly higher than those of CoFiDS. The reason is that MovieLens data set contains a small number of provided ratings. In case more provided ratings are available, the proposed system can be much better than CoFiDS.

3.6.3 Results of Experiment on Flixster Data Set

For each user in Flixster data set, we withheld randomly 5 ratings. The withheld ratings were used as the testing data; and the remaining ratings were considered as the training data. In all experiments, some parameters are selected as follows: $\gamma = 10^{-5}$, $\beta = 1$, $\forall(i,k)\{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 2/9\}$, $Min = 100$, $Max = 500$, and $T = 100$.

In order to measure the impact of two coefficients w_1 and w_2 on Flixster data set, we selected $K = 25$ and $\tau = 0.75$ for the experiments. Tables 3.9 and 3.10 represent results of overall MAE and DS-MAE criteria, respectively. It can be seen in these tables that when $w_1 \leq 0.2$ and $w_2 \geq 0.5$, the performance of the proposed system is mostly linearly dependent on values of both coefficients w_1 and w_2; in the other cases, only some values of coefficients w_1 and w_2 effect the proposed system. Table 3.9 and Table 3.10 are visualized

Table 3.9: Overall MAE versus w_1 and w_2 (Flixster data set)

		w_1					
		0.0	0.1	0.2	0.3	0.4	0.5
w_2	0.0	0.8337	0.8337	0.8338	0.8333	0.8338	0.8338
	0.1	0.8337	0.8335	0.8338	0.8334	0.8336	
	0.2	0.8339	0.8335	0.8337	0.8336	0.8337	
	0.3	0.8339	0.8337	0.8338	0.8334		
	0.4	0.8339	0.8339	0.8337	0.8338		
	0.5	0.8339	0.8339	0.8336			
	0.6	0.8339	0.8338	0.8337			
	0.7	0.8337	0.8337				
	0.8	0.8337	0.8337				
	0.9	0.8337					
	1.0	0.8337					

Table 3.10: Overall $DS\text{-}MAE$ versus w_1 and w_2 (Flixster data set)

		w_1					
		0.0	0.1	0.2	0.3	0.4	0.5
w_2	0.0	0.8340	0.8337	0.8338	0.8338	0.8340	0.8337
	0.1	0.8340	0.8338	0.8337	0.8339	0.8339	
	0.2	0.8341	0.8337	0.8337	0.8340	0.8339	
	0.3	0.8339	0.8338	0.8337	0.8339		
	0.4	0.8339	0.8338	0.8338	0.8340		
	0.5	0.8340	0.8339	0.8338			
	0.6	0.8341	0.8339	0.8339			
	0.7	0.8339	0.8339				
	0.8	0.8339	0.8339				
	0.9	0.8339					
	1.0	0.8339					

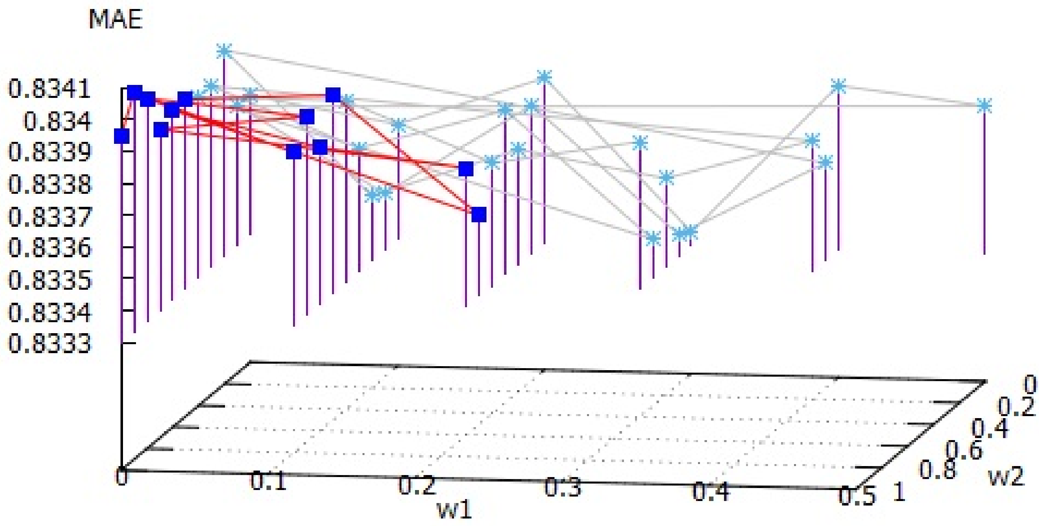

Figure 3.8: Visualizing overall MAE (Flixster data set)

in Figure 3.8 and Figure 3.9, respectively.

To compare with CoFiDS, we selected $w_1 = 0.2, w_2 = 0.5, \tau = 0$, and several values of neighborhood size K. The results are shown in Figure 3.10 and Figure 3.11. According to these figures, the proposed system achieves better performance in both hard and soft decisions in all selected values of K, specially when $K \leq 45$. These results indicate that using community context information for predicting unprovided ratings and assigning weights to rating data when computing user-user similarity is capable of improving the performance of recommendations.

Tables 3.11 and 3.12 show summarized results of the performance comparisons between the proposed system and CoFiDS, with $K = 40, \tau = 0$, in hard and soft decisions, respectively. In these tables, every rating value has its own column; underlined values indicate the better performance, bold values illustrate equal performances, and italic values mention that they are incomparable for comparison. Note that, in the data set, the number users rated as 1.0, 1.5, 2.0 or 2.5 is very small compared to the number of people rated as higher values. So, in these two tables, it can be seen that the columns regarding rating values ranging from 1.0 to 2.5 contain some values as 0 or N/A (Not applicable).

As we have seen from statistics in both Tables 3.11 and 3.12, the proposed system achieves better performance in all selected measurement criteria in most of true rating

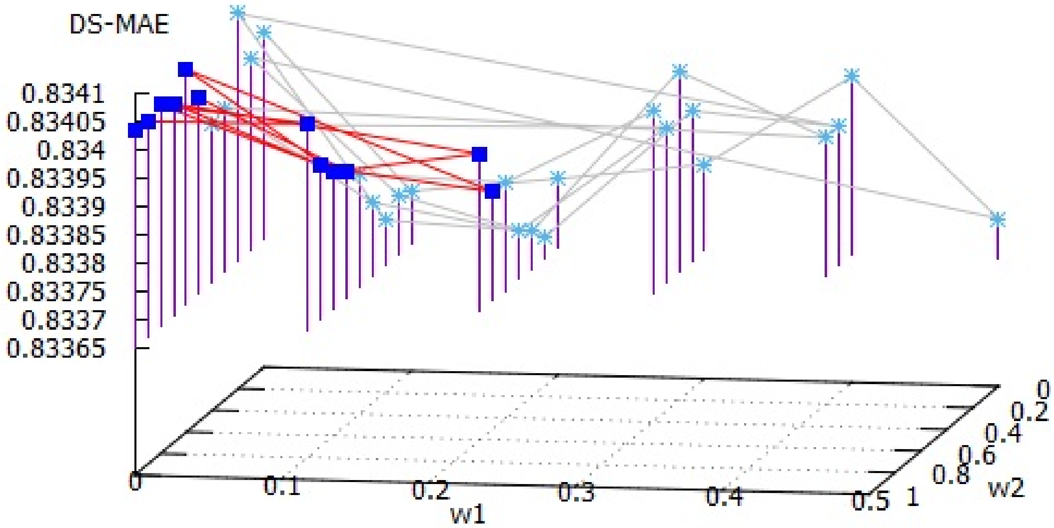

Figure 3.9: Visualizing overall $DS\text{-}MAE$ (Flixster data set)

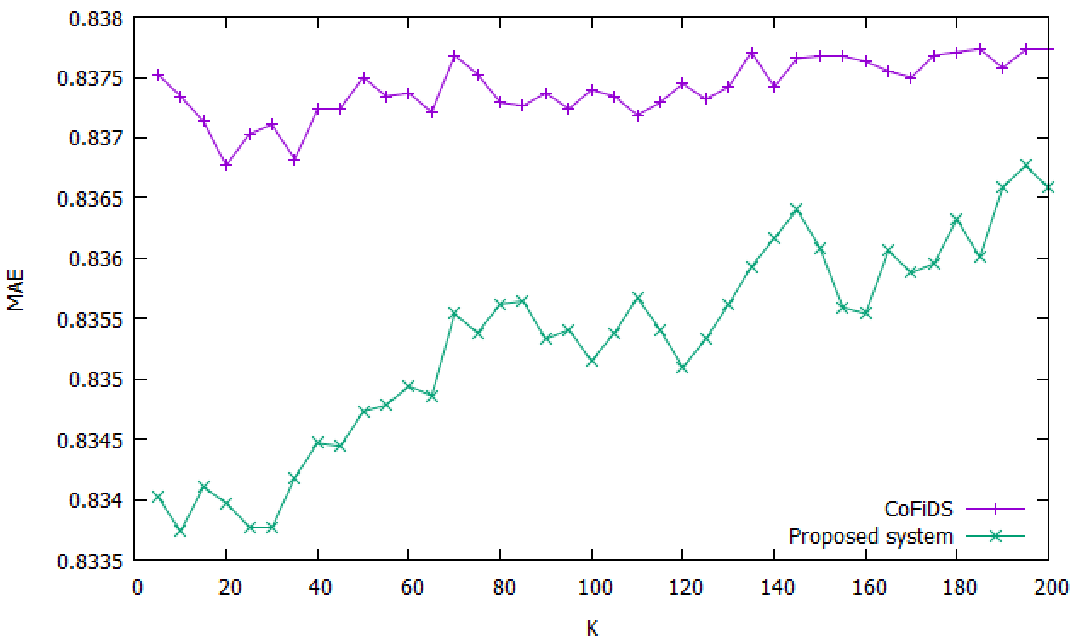

Figure 3.10: Overall MAE versus K (Flixster data set)

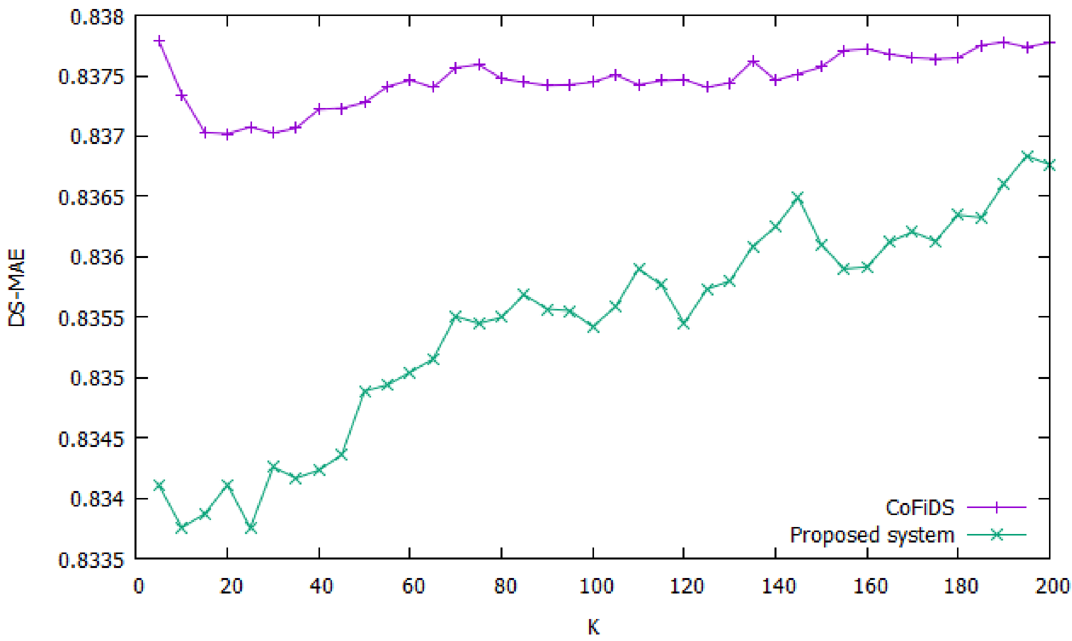

Figure 3.11: Overall *DS-MAE* versus K (Flixster data set)

Table 3.11: The comparison in hard decisions (Flixster data set)

Metric	True rating value										Overall
	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	
Proposed system:											
MAE	**3.3170**	**2.8783**	**2.3949**	**1.8790**	**1.3992**	**0.8983**	**0.4795**	0.1515	0.5722	**0.9740**	**0.8346**
Precision	0.8750	*0*	*0*	*0*	*0.2000*	**0.2100**	**0.1789**	**0.2019**	**0.1616**	0.3811	*0.2357*
Recall	0.0221	0	0	0	**0.0010**	**0.0674**	**0.1477**	0.7809	**0.0138**	**0.0911**	**0.2099**
F_1	0.0431	*N/A*	*N/A*	*N/A*	*0.0020*	0.**1021**	**0.1618**	0.3208	**0.0254**	**0.1470**	*N/A*
CoFiDS:											
MAE	3.3281	2.9006	2.4205	1.898	1.4052	0.9017	0.4796	**0.1226**	**0.5701**	0.998	0.8372
Precision	0.8750	*N/A*	*N/A*	*N/A*	*N/A*	0.1998	0.1778	0.2005	0.1000	**0.3960**	*N/A*
Recall	0.0221	0	0	0	0	0.0607	0.122	**0.8247**	0.0013	0.07	0.2069
F_1	0.0431	*N/A*	*N/A*	*N/A*	*N/A*	0.0931	0.1447	**0.3226**	0.0026	0.1189	*N/A*

values. However, the same as in Movielens data set, the absolute values of the performance of the proposed system are just slightly higher than those of CoFiDS. When more provided ratings are availabe and/or we identify communities in the social network by using another information such as the number of messages, emails, comments, tags, and so on, the different absolute values may be much greater.

3.7 Conclusion

In summary, in this chapter, we have developed a novel collaborative filtering system that uses DST for representing ratings and employs community context information for

Table 3.12: The comparison in soft decisions (Flixster data set)

DS-Metric	True rating value										Overall
	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	
Proposed system:											
MAE	**3.3172**	**2.8811**	**2.3965**	**1.879**	**1.3969**	**0.8994**	**0.4794**	0.1519	0.5710	**0.9725**	**0.8342**
Precision	**0.8637**	0	0	**0.1406**	**0.1152**	**0.2088**	**0.1787**	**0.2018**	**0.1682**	0.3813	**0.2368**
Recall	0.0221	0	0	**0.0007**	**0.0005**	**0.0667**	**0.1478**	0.7796	**0.0148**	**0.0920**	**0.2100**
F_1	0.0431	0	0	**0.0015**	**0.0011**	**0.1011**	**0.1618**	0.3206	**0.0273**	**0.1482**	**0.1426**
CoFiDS:											
MAE	3.3283	2.9014	2.4208	1.8971	1.4060	0.9009	0.4802	**0.1226**	**0.5706**	0.9976	0.8372
Precision	0.8750	0	0	0.0001	0.0002	0.2031	0.1770	0.2006	0.0966	**0.3965**	0.2198
Recall	0.0221	0	0	0	0	0.0617	0.1217	**0.8245**	0.0014	0.0702	0.207
F_1	0.0431	0	0	0	0	0.0947	0.1442	**0.3227**	0.0028	0.1193	0.1292

predicting unprovided ratings. After predicting, suitable recommendations are generated by using both predicted and provided ratings with the important aspect being the stipulation that provided ratings are more important than predicted ones. Remarkably, the developed system is capable of dealing with both the sparsity problem and imperfect information. Moreover, the experiment results show that performance of the proposed system has improved in both hard and soft decisions compared with CoFiDS.

Chapter 4

Using Community Preferences

4.1 Introduction

Naturally, in a community different members can have different preferences on an item; and the overall preference of all members is called the community preference. Actually, for each user, community preferences can influence his/her preferences on unseen items. In this chapter, we develop a new collaborative filtering RS, which is capable of extracting community preferences from the network and employing the extracted preferences for improving quality of recommendations. Especially, in the developed system, we have introduced a new perspective to deal with the sparsity and cold-start problems by using community preferences. To evaluate the new system, we have selected the community-based RS offering soft ratings, which is presented in the previous chapter, for performance comparison.

The difference between the RSs based on DST (System 2) developed in this chapter and the RSs based on DST presented in chapter 3 (System 1) is summarized in Table 4.1. As can be seen in this table, both systems are capable of integrating with the social network containing all users as well as using the new method developed in chapter 3 for computing user-user similarities. As we can see, in System 1, community context information is employed for overcoming the sparsity problem; and in case the community context information does not affect a user on an item, the existing ratings on the item is used. However, System 1 is not effective when dealing with the cold-start problem because when the community context information does not effect users (including new users) on

Table 4.1: Comparing two RSs based on DST

	System 1	System 2
Integrating with the social network	✓	✓
Using the new method for computing user-user similarities	✓	✓
Overcoming the sparsity problem	Existing ratings ($\mathbf{C}$ does not affect)	Community preferences ($\mathbf{C}$ does not affect)
Overcoming the cold-start problem	Vacuous	Community preferences

a new item, the existing ratings on this item are not available. As mentioned previously, missing ratings can be modeled by vacuous; but vacuous representation is highly uncertain. In System 2, we propose to exploit community preferences for overcoming the sparsity and cold-start problems.

4.2 Data Modeling

Let $\mathbf{U} = \{U_1, U_2, ..., U_M\}$ be the set of M users and $\mathbf{I} = \{I_1, I_2, ..., I_N\}$ be the set of N items. Each rating of user U_i on item I_k is represented as a mass function $r_{i,k}$ spanning over a rating domain $\Theta = \{\theta_1, \theta_2, ..., \theta_L\}$. In addition, Dempster's rule of combination [7] is selected for fusing information.

As remarked in [6, 12, 97], context information that may significantly influence user preferences on items can be considered as concepts. Assuming that, in the system, context information consists of P concepts denoted by $\mathbf{C} = \{C_1, C_2, ..., C_P\}$ and each concept $C_p \in \mathbf{C}$ contains at most Q_p groups, denoted by $C_p = \{G_{p,1}, G_{p,2}, ..., G_{p,Q_p}\}$. In addition, for concept C_p with $1 \leq p \leq P$, a user $u_i \in \mathbf{U}$ can be interested in several groups, and an item $I_k \in \mathbf{I}$ can belong to one or some groups of this concept. The groups in which user U_i is interested and the groups to which item I_k belongs are identified by mapping functions f_p and g_p as presented in Equations (3.2) and (3.3) respectively.

In addition, items and users are separated into two different spaces, called item space and user space. In the user space, we assume that all users together join in a social network representing as an undirected graph $\mathbf{G} = (\mathbf{U}, \mathbf{F})$, where $\mathbf{U}$ is the set of all users

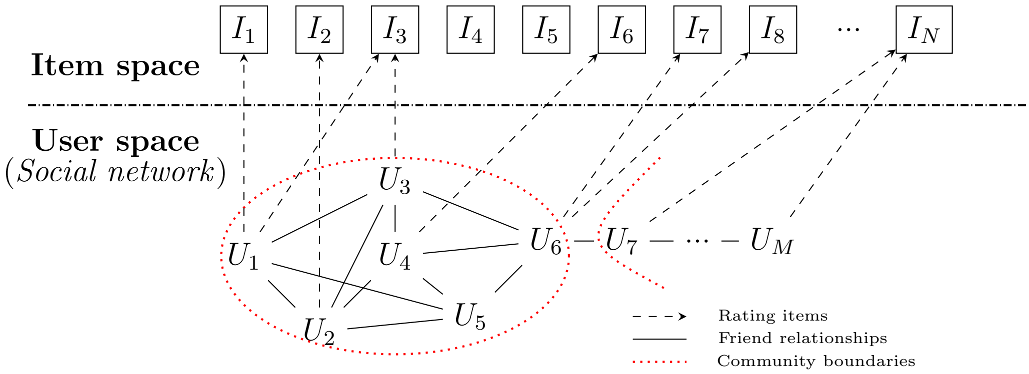

Figure 4.1: Underlying the social network in the proposed system

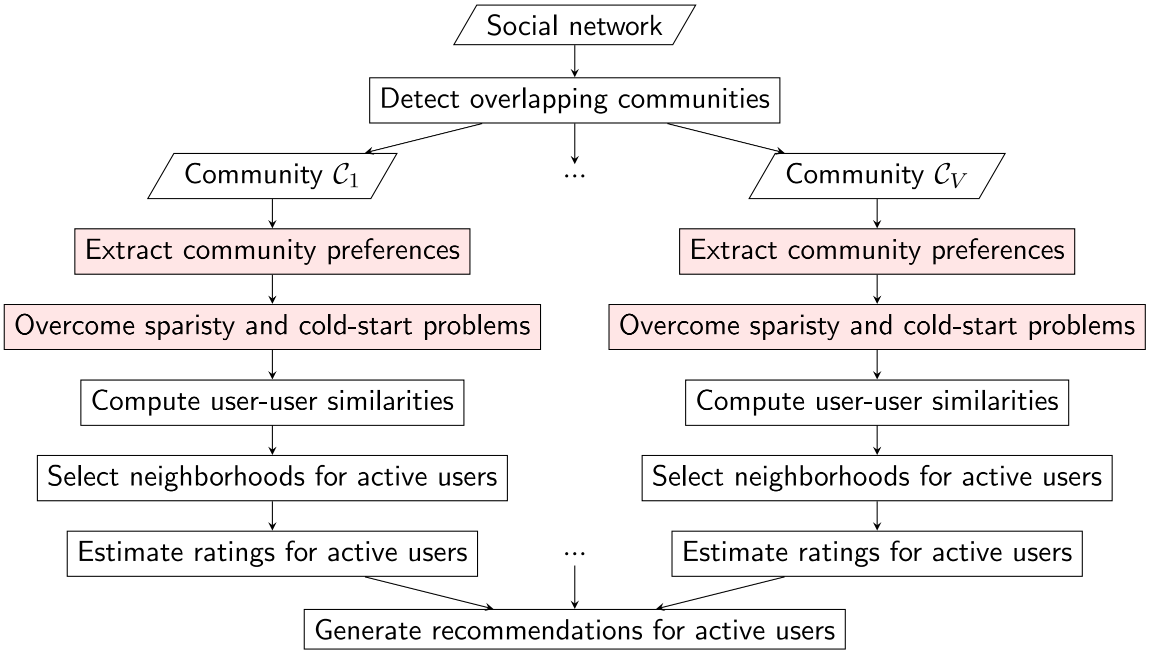

Figure 4.2: The process of recommendations of the proposed system

(nodes) and **F** is the set of all friend relationships (edges). In this social network, users can form into several communities, and a user can belong to one or several communities at the same time. The visual information about the underlying social network is depicted in Figure. 4.1 (adapted from [98]). In this figure, supposing that six users $U_1, U_2, ..., U_6$ belong to a community and they have rated several items.

The general process of recommendations of the system is illustrated in Figure. 4.2. As can be seen in this figure, first, the communities are uncovered in the social network; and then main tasks, such as extracting community preferences, dealing with the sparsity and cold-start problems, computing user-user similarities, selecting neighborhoods,

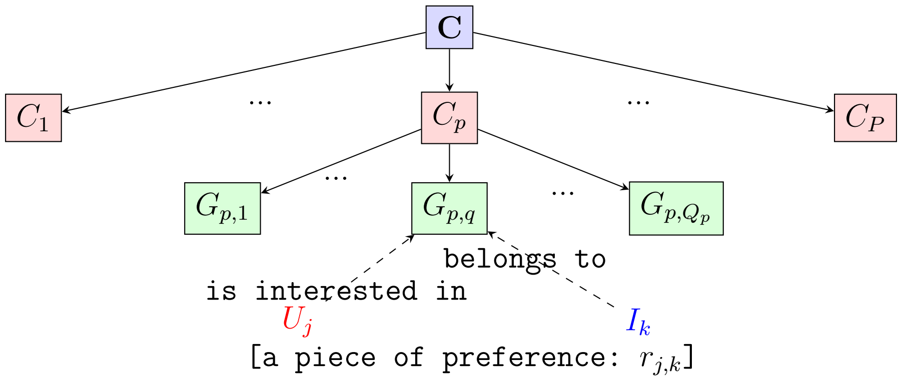

Figure 4.3: Community preference on item I_k regarding group $G_{p,q}$

and estimating ratings on unseen items are performed in each community independently; after that, the suitable recommendations for each active user will be generated based on estimated ratings on the communities to which this user belongs.

In the process of recommendations the task of detecting overlapping communities in the social network is the same as the task presented in Section 3.3 in chapter 3; after completing this task, we also assume that we achieve V overlapping communities denoted by $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_V\}$. Additionally, the tasks such as computing user-user similarities, selecting neighborhoods, estimating ratings, and generating recommendations are the same as the corresponding tasks presented in chapter 3.

4.3 Performing in a Community

In this section, we will present about details of the tasks of extracting community preferences and overcoming the sparsity and cold-start problems in a community $\mathcal{C}_v \in \boldsymbol{\mathcal{C}}$ with $1 \leq v \leq V$.

4.3.1 Extracting Community Preference

The rating matrix of all members in the community is denoted by $\mathbf{R} = \{r_{i,k}\}$ with $r_{i,k}$ is the rating of user U_i on item I_k. In addition, all items rated by user U_i and all users who have rated item I_k are denoted by ${}^R\mathbf{I}_i$ and ${}^R\mathbf{U}_k$, respectively. In addition, in the community, each member has his/her own preference on an item, and the overall preference of all members is called the community preference on the item.

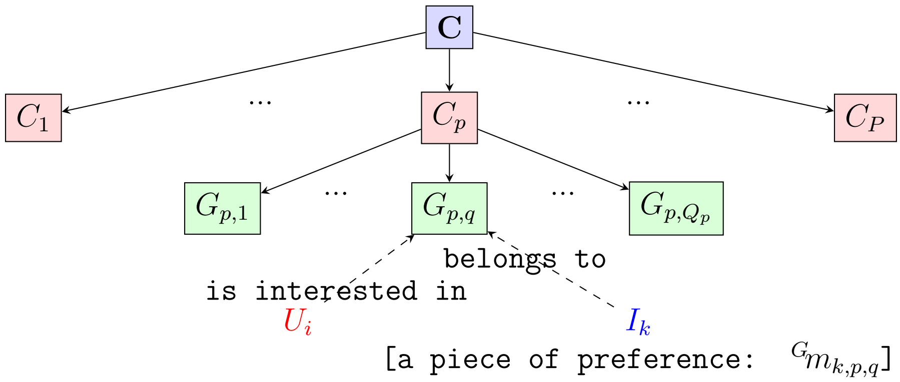

Figure 4.4: Community preference on item I_k regarding concept C_p

Let consider an item I_k. If this item belongs to group $G_{p,q}$ of a concept C_p then each rating of user U_j, who is interested in group $G_{p,q}$, on this item can be considered as a piece of community preference on item I_k regarding group $G_{p,q}$, as illustrated in Figure. 4.3. Consequently, community preference on item I_k regarding group $G_{p,q}$, denoted by ${}^{G}m_{k,p,q}$, can be obtained by combining all related pieces as below

$$ {}^{G}m_{k,p,q} = \bigoplus_{\{j | I_k \in {}^{R}\mathbf{I}_j, G_{p,q} \in f_p(U_j) \cap g_p(I_k)\}} r_{j,k}. \tag{4.1} $$

Since $G_{p,q} \in C_p$, community preference on item I_k regarding group $G_{p,q}$ reflects a piece of community preference on item I_k regarding concept C_p, as illustrated in Figure. 4.4. Thus, community preference on item I_k regarding concept C_p, denoted by ${}^{C}m_{k,p}$, is computed as follows

$$ {}^{C}m_{k,p} = \bigoplus_{\{q | G_{p,q} \in C_p, G_{p,q} \in g_p(I_k)\}} {}^{G}m_{k,p,q}. \tag{4.2} $$

Obviously, community preference on item I_k regarding concept C_p is regarded as a piece of community preference on item I_k, as illustrated in Figure. 4.5. Therefore, community preference on item I_k as a whole, denoted by ${}^{\mathbf{C}}m_k$, is calculated as below

$$ {}^{\mathbf{C}}m_k = \bigoplus_{\{p | \exists G_{p,q} \in C_p, G_{p,q} \in g_p(I_k)\}} {}^{C}m_{k,p}. \tag{4.3} $$

Besides, community preference on item I_k regarding group $G_{p,q}$, ${}^{G}m_{k,p,q}$, is viewed as a piece of community preference regarding group $G_{p,q}$ overall all items. As a result, community preference regarding group $G_{p,q}$, denoted by ${}^{\mathbf{G}}m_{p,q}$, is obtained as follows

$$ {}^{\mathbf{G}}m_{p,q} = \bigoplus_{\{k | 1 \le k \le N, \exists G_{p,q} \in C_p, G_{p,q} \in g_p(I_k)\}} {}^{G}m_{k,p,q}. \tag{4.4} $$

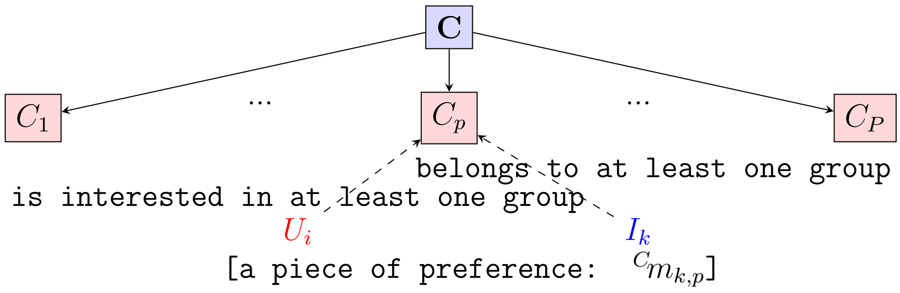

Figure 4.5: Community preference on item I_k

A demonstration of extracting community preferences is illustrated in Example 3.

Example 3 Let us consider the move RS described in Example 2. The community presences on the community containing five users U_1, U_2, U_3, U_4, and U_5 are computed as follows

- It can be seen that, item I_k belong to genres $Western$ and $Horror$. Thus, each rating of a user who is interested in genre $Western$ (or $Horror$) is considered as a piece of community preference on item I_k regarding genre $Western$ (or $Horror$). In addition, community preference on item I_k regarding genre $Western$ (or $Horror$) are generated by combining all related pieces. Applying Equation (4.1), community preferences on item I_k regarding genres $Western$ and $Horror$ are achieved as below

$$\begin{aligned} {}^{G}m_{k,1,3} &= r_{1,k} \oplus r_{4,k}; \\ {}^{G}m_{k,1,4} &= r_{2,k} \oplus r_{4,k}. \end{aligned} \tag{4.5}$$

  After computing, we obtain the values of ${}^{G}m_{k,1,3}$ and ${}^{G}m_{k,1,4}$ as presented in Equations (3.14) and (3.16), respectively.

- The community preference on item I_k regarding concept $Genre$ is obtained by combing all community preferences on item I_k regarding all genres to which item I_k belongs. Applying Equation (4.2), the community preference on item I_k regarding concept $Genre$ is achieved as below

$${}^{C}m_{k,1} =^{G} m_{k,1,3} \oplus {}^{G}m_{k,1,4}. \tag{4.6}$$

After computing, we have

$$\begin{aligned} {}^{C}m_{k,1}(\{4\}) &\approx 0.643; \\ {}^{C}m_{k,1}(\{5\}) &\approx 0.286; \\ {}^{C}m_{k,1}(\Theta) &\approx 0.071. \end{aligned} \tag{4.7}$$

- Community preference on item I_k regarding context **C** is obtained by combing all community preferences on item I_k regarding all concepts. Thus, applying Equation (4.3), we have community preference on item I_k as follows

$$^{\mathbf{C}}m_k = {}^{C}m_{k,1}. \tag{4.8}$$

- Assuming that, there is only one item I_k in the system, the community preferences regarding genres $Western$ and $Horror$ are computed by using Equation (4.4) as below

$$\begin{aligned} ^{\mathbf{G}}m_{1,3} &= {}^{G}m_{k,1,4} \\ ^{\mathbf{G}}m_{1,4} &= {}^{G}m_{k,1,4} \end{aligned} \tag{4.9}$$

4.3.2 Overcoming the Sparsity Problem

In the system, unprovided ratings need to be generated in order to deal with the sparsity problem. Assuming that users who are interested in the same group of a concept can be expected to possess similar preferences regarding to that group. Additionally, supposing that, user U_i has not rated item I_k; the process for generating unprovided rating of user U_i on item I_k, denoted by $r_{i,k}$, contains five steps as follows

- First, as mentioned earlier, user U_i is a member in the community; so, the preference of user U_i is influenced by the preference of the community. Because of this, if user U_i is interested in a group $G_{p,q}$ to which item I_k belongs, user U_i's preference on item I_k regarding group $G_{p,q}$, denoted by ${}^{G}m_{i,k,p,q}$, is signed by community preference on item I_k regarding the same group $G_{p,q}$, as below

$$^{G}m_{i,k,p,q} = {}^{G}m_{k,p,q}. \tag{4.10}$$

- Second, it can be seen that user U_i's preference on item I_k regarding group $G_{p,q}$ reflects a piece of user U_i's preference on the item regarding concept C_p. Consequently, user U_i's preference on item I_k regarding concept C_p, denoted by ${}^{C}m_{i,k,p}$, is computed as follows

$$^{C}m_{i,k,p} = \bigoplus_{\{q|G_{p,q}\in C_p, G_{p,q}\in f_p(U_i)\cap g_p(I_k)\}} {}^{G}m_{i,k,p,q}. \tag{4.11}$$

- Then, user U_i's preference on item I_k regarding concept C_p is a piece of user U_i's preference on item I_k as a whole. As a result, user U_i's preference on item I_k, denoted by ${}^{\mathbf{C}}m_{i,k}$, is achieved by combining user U_i's preferences on item I_k regarding all concepts, as shown below

$$^{\mathbf{C}}m_{i,k} = \bigoplus_{\{p|\exists G_{p,q}\in C_p, G_{p,q}\in f_p(U_i)\cap g_p(I_k)\}} {}^{C}m_{i,k,p}. \tag{4.12}$$

- Next, unprovided rating $r_{i,k}$ is assigned user U_i's preference on item I_k, as shown below

$$r_{i,k} \;=\; {}^{\mathbf{C}}m_{i,k}. \tag{4.13}$$

- Finally, in case, unprovided rating $r_{i,k}$ cannot be generated by using equations (4.10) (4.11), (4.12) and (4.13) because the context information does not affect user U_i and item I_k, in other words $\forall p, f_p(U_i) \cap g_p(I_k) = \emptyset$. We propose that $r_{i,k}$ is assigned community preference on item I_k, as follow

$$r_{i,k} = ^{\mathbf{C}} m_k. \tag{4.14}$$

It is seen that when community context information affects a user on an item, the result of generating unprovided rating of this user on the item by using community preferences is the same as the result obtained by using community context information. However, when community context information does not affect, the result obtained by using community preferences will be different, as depicted in Example 4.

Example 4 Let us consider the movie RS described in Examples 2 and 3. In this scenario, $f_1(U_5) \cap g_1(I_k) = \emptyset$, thus the unprovided rating $r_{5,k}$ of user U_5 on item I_k is assigned by the community preference on item I_k generated by using Equation (3.8) as follows

$$r_{5,k} \;=\; {}^{C}m_{k,1}. \tag{4.15}$$

The value of ${}^{C}m_{k,1}$ is presented in Equation (4.7). As we can observed, this value is different from the value presented in Equation (3.23).

4.3.3 Overcoming the Cold-Start Problem: New Items

Let consider a new item $I_{k'}$, and supposing that, for each concept C_p, the information about the groups, to which this item belong are available. We assume that if $G_{p,q} \in g_p(I_{k'})$ then the community preference on group $G_{p,q}$ is considered to be the community preference on this item regarding group $G_{p,q}$. Based on the assumption, all unprovided ratings on item $I_{k'}$ can be generated. After that, this item is treated the same as any other one. As a result, the new item problem is completely eliminated from the system.

For each group $G_{p,q} \in g_p(I_{k'})$, community preference on item $I_{k'}$ regarding group $G_{p,q}$ is assigned by community preference on group $G_{p,q}$, as below

$$^{G}m_{k',p,q} =^{\mathbf{G}} m_{p,q}. \tag{4.16}$$

Then, for each user U_i, applying Equations (4.10), (4.11), (4.12), (4.13) and (4.14) to generate the unprovided rating of user U_i on item $I_{k'}$.

In the special situation, the groups to which item $I_{k'}$ belongs are very new for the community. In other words, $^{\mathbf{G}}m_{p,q}$ corresponding to $\forall G_{p,q} \in g_p(I_{k'})$ does not exist. If there are some users, who are not interested in any $G_{p,q} \in g_p(I_{k'})$ but have looked at and rated the item, the rating on $I_{k'}$ is assigned by combining all existing ratings on the item as follows

$$r_{i,k'} = \bigoplus_{\{j | I_k \in {}^{R}\mathbf{I}_j, G_{p,q} \in g_p(I_k), G_{p,q} \notin f_p(U_j)\}} r_{j,k}, \text{ for } U_i \in \mathbf{U}. \tag{4.17}$$

Otherwise, if nobody in the community has rated this item, $U_{k'} = \emptyset$, then for each user $U_i \in \mathbf{U}$, $r_{i,k'}$ is assigned by vacuous.

A demonstration of generating unprovided ratings for new items is illustrated in Example 5.

Example 5 Let us consider the move RS described in Examples 2 and 3. Assuming that a new item $I_{k'}$ belonging to genres *Drama* and *Horror* is just added into this system. In this case, community preference regarding genre *Horror* is considered as community preference on this item regarding genre *Horror*. Applying Equation (4.16) we have

$$^{G}m_{k',1,4} = {}^{\mathbf{G}}m_{1,4}. \tag{4.18}$$

As observed, users U_2, U_3 and U_4 are interested in genre *Horror*, thus, the preferences of these users on item $I_{k'}$ regarding genre *Horror* is obtained by using Equations (4.10) as follows

$$^{G}m_{2,k',1,4} \;=\; ^{G}m_{3,k',1,4} \;=\; ^{G}m_{4,k',1,4} =^{G} m_{k',1,4}; \tag{4.19}$$

Then, applying Equations (4.11), (4.12), and (4.13), the unprovided ratings $r_{2,k'}, r_{3,k'}$ and $r_{4,k'}$ will be generated as below

$$r_{2,k'} = r_{3,k'} = r_{4,k'} = \;^{\mathbf{G}}m_{1,4}. \tag{4.20}$$

However, in this example, there is no users who are interested in genre *Drama* have ratings. So, community preference regarding genre *Drama* is vacuous. As the result, the unproved ratings of users U_1 and U_5 are vacuous.

4.3.4 Overcoming the Cold-Start Problem: New Users

Let consider a new user $U_{i'}$. In the system, all unprovided ratings regarding this user are generated; and then, there is no difference between user $U_{i'}$ and the other ones in terms of being recommended. As a result, the new user problem is solved.

In case the profile of user $U_{i'}$ contains information about the groups of each concept C_p, in which user $U_{i'}$ is interested, the unprovided ratings of this user on each item I_k are generated as follows

- Community preference on item I_k regarding a group $G_{p,q} \in f_p(U_{i'})$ is considered as $U_{i'}$'s preference on item I_k regarding a group $G_{p,q}$, as below

$$^{G}m_{i',k,p,q} =^{G} m_{k,p,q}. \tag{4.21}$$

- Applying the equations (4.11), (4.12), (4.13) and (4.14) for user $U_{i'}$, the unprovided rating $r_{i',k}$ will be created.

Otherwise, if the information about the groups in which user $U_{i'}$ is interested is not available, the unprovided rating of this user on an item I_k is assigned community preference on item I_k as follows

$$r_{i',k} =^{\mathbf{C}} m_k \tag{4.22}$$

In Example 2, users U_3 and U_5 can be considered as new users because they have not rated any item. The process of generating unproved ratings of these users can be seen in Example 4.

At this point, all unprovided ratings related new items as well as new users have been created. As a result, the cold-start problem is completely eliminated in the system.

4.4 Experiment

To evaluate the proposed system, the system introduced in [12] was selected as a baseline for performance comparison. The same as the proposed system in this chapter, the baseline is also developed based on DST. To deal with the sparsity problem, the baseline exploits community context information extracted from the social network. The baseline is not capable of tackling the cold-start problem; however, unprovided ratings are signed by vacuous for new items as well as users who are not affected by the context information. In addition, to measure recommendation performances, evaluation methods *DS-MAE* [6], and *DS-Precision*, *DS-Recall* [22] were chosen.

Flixster data set [12], described in Section 3.6.1, was selected in the experiments. In this data set, the genres in which a user is interested are not available; thus, we assume that a user is interested in a genre if this user has rated at least 11 movies belonging to that genre. Since the proposed system works with soft ratings, the Dempster-Shafer modeling function presented in Equation 3.35 was adopted for transforming hard ratings in this data set into soft ratings. All users in this data set belong to a social network whose nodes are linked by undirected friendships. To discover overlapping communities in this social network, SLPA algorithm [89] was also selected.

The values of parameters in the experiments were selected mainly based on the analyzed results published in [97, 6], as follows: $\gamma = 10^{-4}$, $w_1 = 0.3, w_2 = 0.1$ and $\forall(i,k)\{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 2/9\}$. In addition, to choose the value for parameter τ, all values of user-user similarity in matrix $\mathbf{S}$ were sorted in ascending order, and the lowest value of top 50% of highest values in $\mathbf{S}$ was selected. In addition, this data set was separated into two parts, testing data and training data; the first one contains random selections of 5 ratings of each user, and the other consists of the remaining ratings.

The results of comparison according to overall *DS-MAE*, *DS-Precision* and *DS-Recall*

Table 4.2: Experiment results

K	*DS-MAE*		*DS-Precision*		*DS-Recall*	
	Baseline	New system	Baseline	New system	Baseline	New system
5	0.95978	**0.85227**	0.20442	**0.22506**	0.17956	**0.18872**
10	0.95263	**0.85241**	0.20533	**0.22495**	0.18101	**0.18878**
15	0.94846	**0.85241**	0.20561	**0.22498**	0.18174	**0.18884**
20	0.94643	**0.85236**	0.20580	**0.22511**	0.18202	**0.18894**
25	0.94492	**0.85235**	0.20596	**0.22516**	0.18231	**0.18895**
30	0.94378	**0.85231**	0.20596	**0.22514**	0.18247	**0.18895**
35	0.94259	**0.85231**	0.20589	**0.22517**	0.18263	**0.18898**
40	0.94151	**0.85229**	0.20600	**0.22522**	0.18281	**0.18902**
45	0.94057	**0.85229**	0.20594	**0.22522**	0.18295	**0.18905**
50	0.94007	**0.85229**	0.20595	**0.22525**	0.18300	**0.18908**
55	0.93964	**0.85227**	0.20597	**0.22531**	0.18308	**0.18912**
60	0.93919	**0.85226**	0.20600	**0.22536**	0.18315	**0.18916**
65	0.93900	**0.85226**	0.20599	**0.22540**	0.18317	**0.18918**
70	0.93891	**0.85226**	0.20600	**0.22539**	0.18320	**0.18918**
75	0.93869	**0.85226**	0.20602	**0.22541**	0.18322	**0.18919**
80	0.93853	**0.85227**	0.20605	**0.22538**	0.18325	**0.18919**
85	0.93841	**0.85226**	0.20607	**0.22539**	0.18327	**0.18920**
90	0.93836	**0.85226**	0.20606	**0.22540**	0.18329	**0.18921**
95	0.93830	**0.85226**	0.20607	**0.22540**	0.18330	**0.18921**
100	0.93825	**0.85226**	0.20607	**0.22539**	0.18331	**0.18922**
105	0.93822	**0.85226**	0.20607	**0.22539**	0.18331	**0.18922**
110	0.93820	**0.85226**	0.20606	**0.22538**	0.18331	**0.18922**
115	0.93818	**0.85227**	0.20604	**0.22538**	0.18331	**0.18922**
120	0.93817	**0.85227**	0.20603	**0.22536**	0.18332	**0.18921**
125	0.93816	**0.85227**	0.20603	**0.22536**	0.18332	**0.18921**
130	0.93814	**0.85227**	0.20603	**0.22536**	0.18332	**0.18922**
135	0.93814	**0.85227**	0.20601	**0.22535**	0.18333	**0.18922**
140	0.93812	**0.85227**	0.20601	**0.22535**	0.18333	**0.18922**
145	0.93812	**0.85227**	0.20600	**0.22534**	0.18333	**0.18923**
150	0.93810	**0.85227**	0.20600	**0.22534**	0.18333	**0.18923**

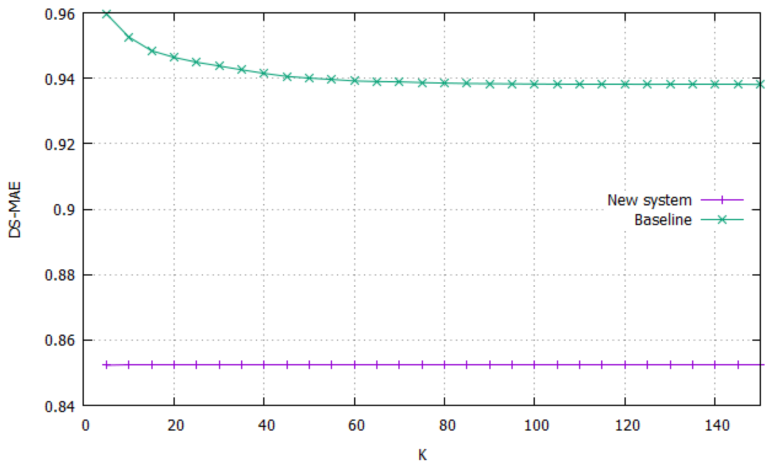

Figure 4.6: Overall $DS\text{-}MAE$ versus K

varying with changing neighborhood size K are summarized and illustrated Table. 4.2. Note that, in this table, the bold values indicate the better ones. As observed, the new system is more effective than the baseline in all cases. That means integrating with community preferences is capable of improving the quality of recommendations.

The results as illustrated in Table 4.2 are also visualized in Figures 4.6, 4.7, and 4.8. According to these figures, the performances of the baseline increase with $K <= 20$ for $DS\text{-}Precision$ and $K <= 40$ for $DS\text{-}MAE$ and $DS\text{-}Recall$, and become stable when $K > 20$ for $DS\text{-}Precision$ and $K > 40$ for $DS\text{-}MAE$ and $DS\text{-}Recall$. Specially, the performances of the new system are better and more stable than those of the baseline. These results indicate that using community preferences is more effective than using community context information for generating unprovided ratings in RSs based on DST.

4.5 Conclusion

In this chapter, we have developed a new collaborative filtering RS that is capable of integrating with a social network consisting of all users. In this system, community preferences are extracted from the social network and represented as mass functions first, and then the extracted preferences are employed for predicting all unprovided ratings of users on items (including new users and new items), after that, suitable recommendations are generated mainly based on both provided and predicted ratings without concerning about whether users or items are new or not.

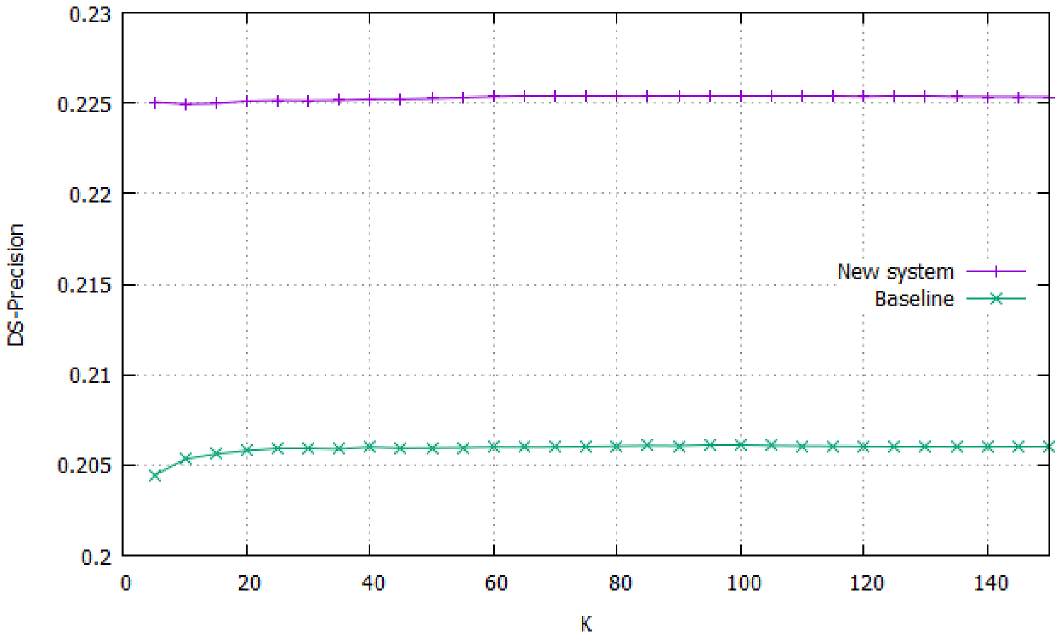

Figure 4.7: Overall *DS-Precision* versus K

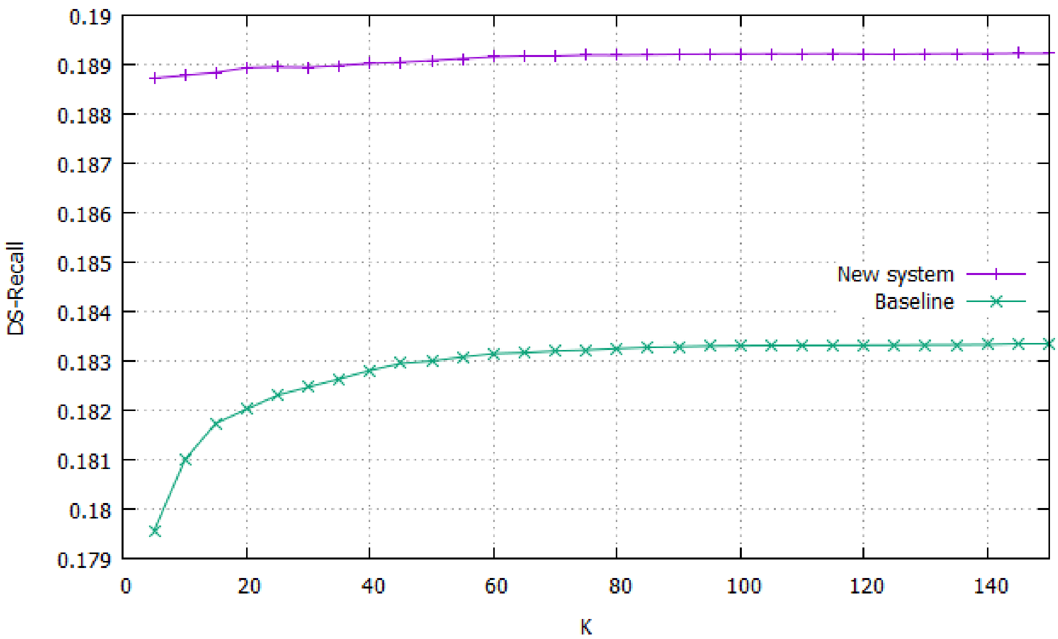

Figure 4.8: Overall *DS-Recall* versus K

Chapter 5

Two-Probabilities Focused Combination

5.1 Introduction

As can be seen that in the research area of DST [7, 8], Dempster's rule of combination plays a significant role [99] as well as being employed as a powerful tool to combine information in a variety of applications [37, 100, 101, 102]. Particularly, this combination method is currently applied in RSs [6, 10, 12, 97] based on DST.

In RSs based on DST, ratings or user preferences on items (products or services) are represented as mass functions and tasks of combining mass functions are executed frequently [6, 12, 97]. However, when combining information by using Dempster's rule of combination, the combined results usually contain many focal elements with very low probabilities and a few focal elements with high probabilities [103], a typical example of this kind of combined results is illustrated in Table 5.1.

In addition, when combining two highly conflicting mass functions by using Dempster's rule, the focal elements with very low probabilities can lead to unsatisfactory results [13, 14]. Moreover, in RSs based on DST, highly and even totally conflicting ratings are very common because of the diversity of users, and ratings from different users are not verified to be totally reliable and independent. Under those circumstances, combining information about user preferences in these systems by using Dempster's rule may be not effective and other rules of combination can be more appropriate.

Table 5.1: A combined result representing as mass function m

$m(\{3.5\})$	=	0.999999980892241
$m(\{3.0\})$	=	1.91077577843934E-08
$m(\{4.0\})$	=	7.86327480839232E-16
$m(\{4.5\})$	=	4.85693565681365E-43
$m(\{2.5\})$	=	4.64318514160755E-48
$m(\{3.0, 3.5\})$	=	2.09075154133808E-64
$m(\{3.5, 4.0\})$	=	8.60391580797565E-67
$m(\{5.0\})$	=	5.00831844339378E-68
$m(\{2.5, 3.0\})$	=	2.22082105935693E-75
$m(\{4.0, 4.5\})$	=	3.76097996470203E-80
$m(\{2.0\})$	=	2.39546318569105E-87
$m(\{2.5, 3.0, 3.5\})$	=	1.33165249421964E-91
$m(\{4.5, 5.0\})$	=	4.43884166473141E-92
$m(\{3.5, 4.0, 4.5\})$	=	5.48005141571796E-94
$m(\{1.5, 2.0, 2.5\})$	=	2.66666663099885E-100
$m(\{\Theta\})$	=	1E-100

In [15, 16, 17], the authors have developed a combination method, known as *2-points focused* combination, which is capable of distinguishing focal elements with very low probabilities from the ones with high probabilities. Regarding this method, mass functions are reduced into *triplet* or *2-points focused* mass functions [15, 16, 17] whose focal sets consist of two focal elements with the highest probabilities and the whole set element; for instance, the mass function illustrated in Table 5.1 will be reduced into a *triplet* mass function $\bar{m}$ as shown in Table 5.2. Moreover, this method can help RSs based on DST to handle combining highly conflicting mass functions and improve time of computation [104]. However, in case the focal set of a mass function contains more than two focal elements with the same highest probabilities (excluding the whole set one), this mass function can be reduced into several *triplet* mass functions, as illustrated in Example 6; and when combining this mass function with another one by using *2-points focused* combination method, we can achieve differently combined results depending on which

Table 5.2: *Triplet* mass function $\bar{m}$

$\bar{m}(\{3.5\})$	=	0.999999980892241
$\bar{m}(\{3.0\})$	=	1.91077577843934E-08
$\bar{m}(\Theta)$	=	8.13198474256578E-16

Table 5.3: Mass function m_1

$m_1(\{1\})$	=	0.30
$m_1(\{3\})$	=	0.30
$m_1(\{4\})$	=	0.04
$m_1(\{5\})$	=	0.30
$m_1(\{1,2,3,4,5\})$	=	0.06

Table 5.4: Mass function m_2

$m_2(\{1\})$	=	0.40
$m_2(\{2\})$	=	0.10
$m_2(\{3\})$	=	0.07
$m_2(\{4\})$	=	0.40
$m_2(\{1,2,3,4,5\})$	=	0.03

corresponding *triplet* mass function is selected.

Example 6 Assuming that, in a RS based on DST with a rating domain $\Theta = \{1,2,3,4,5\}$, we need to combine two ratings by using *2-points focused* combination method. These ratings are represented by two mass functions denoted by m_1 and m_2 as well as being depicted in Tables 5.3 and 5.4, respectively. When converting into *triplet* mass functions, mass function m_1 can be one of three different *triplet* mass functions, called $\bar{m}_1^{(1)}$, $\bar{m}_1^{(2)}$, and $\bar{m}_1^{(3)}$, as shown in Tables 5.5, 5.6 and 5.7, respectively; and mass function m_2 has an only one *triplet* mass function, denoted by $\bar{m}_2$, described in Table 5.8. Regarding three options for selecting *triplet* mass function corresponding to mass function m_1, when combing two mass functions m_1 and m_2 using *2-points focused* combination method, we can achieve three possible results denoted as $\bar{m}^{(1)}, \bar{m}^{(2)}$ and $\bar{m}^{(3)}$, as shown in Tables 5.9, 5.10, and 5.11 respectively. We can observe that *triplet* mass function $\bar{m}^{(3)}$ is significantly different from *triplet* mass functions $\bar{m}^{(1)}$ and $\bar{m}^{(2)}$. As a result, *2-points focused* combination method is not effective in this scenario.

In this chapter, we develop a new combination method that concentrates on significant focal elements defined as the ones with probabilities in top two highest probabilities and

Table 5.5: *Triplet* mass function $\bar{m}_1^{(1)}$

$\bar{m}_1^{(1)}(\{1\})$	$=$	0.30
$\bar{m}_1^{(1)}(\{3\})$	$=$	0.30
$\bar{m}_1^{(1)}(\{1,2,3,4,5\})$	$=$	0.40

Table 5.6: *Triplet* mass function $\bar{m}_1^{(2)}$

$\bar{m}_1^{(2)}(\{1\})$	$=$	0.30
$\bar{m}_1^{(2)}(\{5\})$	$=$	0.30
$\bar{m}_1^{(2)}(\{1,2,3,4,5\})$	$=$	0.40

Table 5.7: *Triplet* mass function $\bar{m}_1^{(3)}$

$\bar{m}_1^{(3)}(\{3\})$	$=$	0.30
$\bar{m}_1^{(3)}(\{5\})$	$=$	0.30
$\bar{m}_1^{(3)}(\{1,2,3,4,5\})$	$=$	0.40

Table 5.8: *Triplet* mass function $\bar{m}_2$

$\bar{m}_2(\{1\})$	$=$	0.40
$\bar{m}_2(\{4\})$	$=$	0.40
$\bar{m}_2(\{1,2,3,4,5\})$	$=$	0.20

ignores the other focal elements excluding the whole set element. With this characteristic, the new method also has the advantages which *2-points focused* combination method possesses. Particularly, when combing two mass functions by using the proposed method, we can get only one combined result; that means this method is capable of tackling the weakness of *2-points focused* combination method.

In the experiments, to measure the effectiveness and efficiency of the new method, it was integrated in the RSs based on DST, introduced in [12, 97, 6], using Movielens and Flixster data sets and compared with *2-points focused* combination method. The experiment results indicate that, regarding to accuracy of recommendations, the new method is better than *2-points focused* combination method; and the time of computation of the new method can be comparable to that of *2-points focused* combination method whose time complexity is linear [104].

5.2 The Proposed Combination Method

In RSs based on DST, let us consider a mass function $m : 2^\Theta \rightarrow [0,1]$ defined on a rating domain $\Theta = \{\theta_1, \theta_2, ..., \theta_L\}$. The focal set of mass function m is denoted by F. As mentioned previously, in the focal set F, especially when mass function m is a combined result, usually most of focal elements have infinitesimal probabilities whereas a few focal elements have high probabilities.

We propose that, in focal set F, only focal elements with probabilities in top two

Table 5.9: *Triplet* mass function $\bar{m}^{(1)}$

$\bar{m}^{(1)}(\{1\})$	=	0.53
$\bar{m}^{(1)}(\{4\})$	=	0.25
$\bar{m}^{(1)}(\{1,2,3,4,5\})$	=	0.22

Table 5.10: *Triplet* mass function $\bar{m}^{(2)}$

$\bar{m}^{(2)}(\{1\})$	=	0.53
$\bar{m}^{(2)}(\{4\})$	=	0.25
$\bar{m}^{(2)}(\{1,2,3,4,5\})$	=	0.22

Table 5.11: *Triplet* mass function $\bar{m}^{(3)}$

$\bar{m}^{(3)}(\{1\})$	=	0.31
$\bar{m}^{(3)}(\{4\})$	=	0.31
$\bar{m}^{(3)}(\{1,2,3,4,5\})$	=	0.38

highest ones are retained, and other focal elements excluding the whole set element are treated as noise that may be caused due to superficial rating or resulting from the process of information fusion and then eliminated. Note that, the probabilities of the eliminated focal elements are transferred to the whole set element in order to make sure that the achieved mass function is still well-defined.

Formally, assuming that $F' = F \backslash \{\Theta\}$ and F' contains n elements. After sorting all elements in F' by descending probabilities, we obtain $F' = \{A_1, A_2, ..., A_n\}$, where $m(A_i) = p_i$ with $A_i \subset \Theta$, and $p_1 \geq p_2 \geq p_3 \geq ... \geq p_n$. Based on mass function m, *2-probabilities focused* mass function $\ddot{m} : 2^\Theta \to [0,1]$ is defined as follows

$$\ddot{m}(A) = \begin{cases} m(A), \text{ for } A \subset \Theta \text{ and } (m(A) = p_1 \text{ or } m(A) = p_2); \\ 1 - \sum\limits_{\{B \subset \Theta | m(B) = p_1\}} m(B) - \sum\limits_{\{C \subset \Theta | m(C) = p_2\}} m(C), \text{ if } A = \Theta; \\ 0, \text{ otherwise.} \end{cases} \tag{5.1}$$

Let us consider two *2-probabilities focused* mass functions $\ddot{m}_1$ and $\ddot{m}_2$ defined on the same frame of discernment Θ. The method to combine these two *2-probabilities focused* mass functions, denoted by $\ddot{m} = \ddot{m}_1 \uplus \ddot{m}_2$ and called *2-probabilities focused* combination, contains two steps as shown below

- First, *2-probabilities focused* mass functions $\ddot{m}_1$ and $\ddot{m}_2$ are combined by using Dempster's rule regarding Equation (2.2). Let m' denote the combined result, we have $m' = \ddot{m}_1 \oplus \ddot{m}_2$.

Table 5.12: Mass function m'_1

$m'_1(\{1\})$	$=$	0.85
$m'_1(\{1,2\})$	$=$	0.12
$m'_1(\{2\})$	$=$	0.02
$m'_1(\{3\})$	$=$	0.01

Table 5.13: Mass function m'_2

$m'_2(\{3\})$	$=$	0.01
$m'_2(\{4\})$	$=$	0.05
$m'_2(\{5\})$	$=$	0.94

- Second, mass function m' is converted into corresponding *2-probabilities focused* mass function $\bar{m}$ according to Equation (5.1).

Supposing that we need to combine n *2-probabilities focused* mass functions, defined on the same frame Θ, by using *2-probabilities focused* combination method. In the best case, when n *2-probabilities focused* mass functions as well as the temporary combined results are *2-points focused* mass functions, the proposed method will be the same as *2-points focused* combination method. Moreover, as remarked in [104], the time complexity of *2-points focused* combination method is linear $O(n)$. Thus, the time complexity of the proposed method is linear in the best case.

In the worst case known as when there no focal elements are eliminiated from both n *2-probabilities focused* mass functions and the temporary combined results, the time complexity of *2-probabilities focused* combination is the same as that of Dempster's rule whose time complexity is exponential $O(|\Theta|^{n-1})$ [17]. Therefore, the time complexity of *2-probabilities focused* combination exponential in the worst case.

It can be seen that, with *2-probabilities focused* combination method, RSs based on DST can have three advantages as follows

- First, by transferring probabilities of eliminated focal elements to the whole set element, this method can help RSs based on DST handle combining highly conflicting mass function. For example, in a RS based on DST with a rating domain $\Theta = \{1, 2, 3, 4, 5\}$, let us consider two ratings represented as two mass functions shown in Tables 5.12 and 5.13 respectively. When combining these two mass functions by using Dempster's rule of combination, $m' = m'_1 \oplus m'_2$, we will get a unsatisfactory result, $m'(\{3\}) = 1$. With *2-probabilities focused* combination method, two mass functions m'_1 and m'_2 are transformed into two *2-probabilities focused* mass functions shown in Tables 5.14 and 5.15 respectively, and the combined result of

Table 5.14: *2-probabilities focused* mass function $\ddot{m}'_1$

$\ddot{m}'_1(\{1\})$	$=$	0.85
$\ddot{m}'_1(\{1,2\})$	$=$	0.12
$\ddot{m}'_1(\{1,2,3,4,5\})$	$=$	0.03

Table 5.15: *2-probabilities focused* mass function $\ddot{m}'_2$

$\ddot{m}'_2(\{4\})$	$=$	0.05
$\ddot{m}'_2(\{5\})$	$=$	0.94
$\ddot{m}'_2(\{1,2,3,4,5\})$	$=$	0.01

these two *2-probabilities focused* mass functions is more reasonable, as illustrated in Table 5.16.

- Second, when reducing the number of focal elements in focal sets of mass functions, logically, it takes less time to combine them together. Thus, time of computation in the system is improved.

- Finally, from a given mass function, we can induce only one *2-probabilities focused* mass function; thus, we get only one combined result when combining mass functions together by using *2-probabilities focused* combination method. Let us consider Example 6 again. Regarding mass function m_1, there is only one *2-probabilities focused* mass function $\ddot{m}_1$ as depicted in Table 5.17; and the *2-probabilities focused* mass function $\ddot{m}_2$ corresponding to mass function m_2 is shown in Table 5.18. Consequently, after combining two *2-probabilities focused* mass functions $\ddot{m}_1$ and $\ddot{m}_2$ using *2-probabilities focused* combination method, we achieve one combined result as illustrated in Table 5.19.

However, *2-probabilities focused* combination method is not associative; in other words, when combining several mass functions by using this combination method, the combined result is influenced by the order of inputs. So as to evaluate the effect of this weakness on RSs based on DST, we have conducted the experiment on Flixster data set. Details and results of this experiment are presented in Section 5.3.2.

Generally, in RSs based on DST, we can model user preferences by *t-probabilities*

Table 5.16: *2-probabilities focused* mass function $\dddot{m}'$

$\dddot{m}'(\{1\})$	$=$	0.214105793
$\dddot{m}'(\{5\})$	$=$	0.710327456
$\dddot{m}'(\{1,2,3,4,5\})$	$=$	0.075566751

Table 5.17: *2-probabilities focused* mass function $\dddot{m}_1$

$\dddot{m}_1(\{1\})$	$=$	0.30
$\dddot{m}_1(\{3\})$	$=$	0.30
$\dddot{m}_1(\{4\})$	$=$	0.04
$\dddot{m}_1(\{5\})$	$=$	0.30
$\dddot{m}_1(\{1,2,3,4,5\})$	$=$	0.06

focused mass functions as well as using *t-probabilities focused* combination method with t is an integer number ranging from 1 to $2^{|\Theta|} - 2$ for combining information.

5.3 Experiment

To evaluate *2-probabilities focused* combination method, *2-points focused* combination method [15, 16, 17] were selected for the purpose of comparisons in both accuracy of recommendations and time of computation. In addition, to measure accuracy of recommendations, evaluation methods *DS-MAE* was chosen.

We conducted experiments on two RSs based on DST, which consist of characteristics as described in the previous section. The first system, similar to the system proposed in [97, 6], does not integrate with social networks. In contrast, the second one, the same as the system introduced in [12], is capable of integrating with community information extracted from the social network containing all users. Note that in these two systems, Equation (3.27) is employed to compute distance between two users.

Since the two systems work with domains with soft ratings, the method suggested in [6] was adopted for generating data sets in the experiments. Regarding this method, data sets with hard ratings are selected first, and then a DS modeling function is applied to transform the hard ratings into corresponding soft ratings. Here, Movielens and Flixster

Table 5.18: *2-probabilities focused* mass function $\ddot{m}_2$

$\ddot{m}_2(\{1\})$	$=$	0.40
$\ddot{m}_2(\{2\})$	$=$	0.10
$\ddot{m}_2(\{4\})$	$=$	0.40
$\ddot{m}_2(\{1,2,3,4,5\})$	$=$	0.10

Table 5.19: *2-probabilities focused* mass function $\ddot{m}$

$\ddot{m}(\{1\})$	$=$	0.60
$\ddot{m}(\{4\})$	$=$	0.15
$\ddot{m}(\{1,2,3,4,5\})$	$=$	0.25

data sets described in Section 3.6.1 were used then first and the second systems, respectively. In these data sets, each hard rating of a user U_i on an item I_k was transformed into the corresponding soft rating which is presented by a *2-probabilities focused* mass function $\ddot{m}_{i,k}$ by using the DS modeling function presented Equation (3.33). Besides, in both Movielens and Flixster data sets, the information about the genres in which a user is interested is not available. Thus, we assume that if a user has rated an item then this user is interested in all genres to which the item belongs.

In the rest of this section, experiments on the first system with Movielens data set as well as those on the second system with Flixster data set are provided. Note that, the values of parameters in these systems are selected mainly based on the analyzed results published in [6, 97].

5.3.1 Experiment on Movielens Data Set

The values of parameters were selected as follows: $\gamma = 10^{-4}$, $w_1 = 0.3, w_2 = 0.1$, and $\forall(i,k)\{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 2/9\}$. Particularly, it is unreasonable to select a fixed value for parameter τ to use in the experiments. The reason is that, with different combination methods, the values of user-user similarities of two specific users are different. Thus, to select value for parameter τ, all values in user-user similarity matrix were sorted in ascending, and then, a value of $s_{i,j}$ that can retain top 30% of the highest values in the rating matrix was chosen for τ.

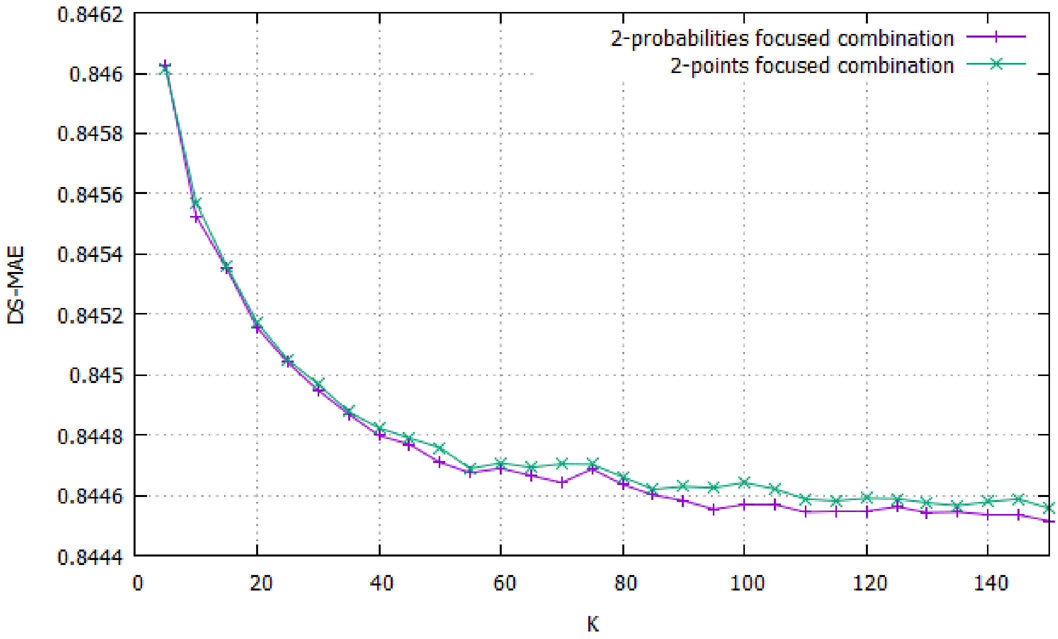

Figure 5.1: Overall $DS\text{-}MAE$ versus K (Movielens data set)

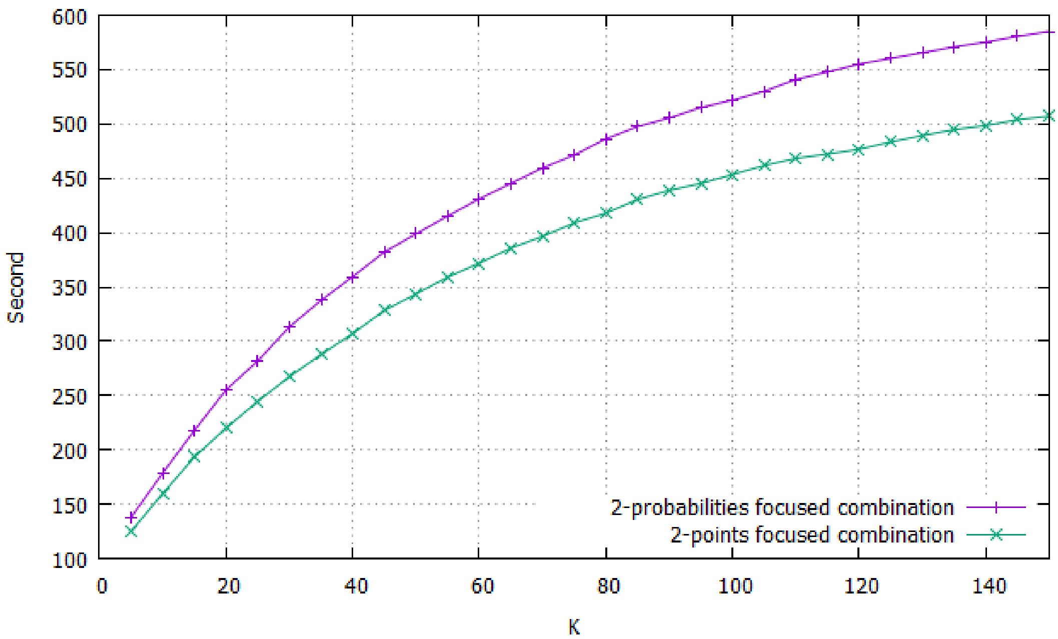

Figure 5.2: Overall time of computation versus K (Movielens data set)

Additionally, 10-fold cross validation was used in the experiments. Firstly, ratings in this data set were divided into 10 folds; each fold contains random 10% ratings of each user. Then, the experiments were conducted 10 times; in each time, one of 10 folds was selected as testing data and the remaining ratings were employed as training data. The average results of 10 times will be represented in the remainder of this section.

Figure 5.1 demonstrates overall $DS\text{-}MAE$ criterion results changes with neighborhood size K. Note that, in this figure, the smaller values are the better ones. As can be seen in the figure, with $K \leq 40$ performances of the two methods increase sharply as well as being the same as each other. With $K > 40$, performances of both methods become stable; and especially, *2-probabilities focused* combination is slightly better than *2-points focused* combination method.

Execution time for the task of estimating ratings varies with neighborhood size K is depicted in Figure 5.2. As can be seen in this figure, the time taken by *2-probabilities focused* is quite effective as well as being comparable to *2-points focused* combination.

5.3.2 Experiment on Flixster Data Set

All users in Flixster data set belong to a social network whose nodes are linked by undirected friendships. So as to discover overlapping communities in this social network, SLPA algorithm [89] was selected. After executing this algorithm, 7 overlapping communities were detected and they are depicted in Table 3.4.

The rating matrix containing all rating data in the Flixster data set was divided into 7 sub-rating matrices according to 7 communities. Each sub-rating matrix consists of the ratings of members in the corresponding community. After that, tasks of predicting unrated data, computing user-user similarities, selecting neighborhood, and estimating rating data were performed in each community independently. The finally estimated rating data for an active user were generated by combining all estimated rating data of this user in the communities to which he/she belongs. The suitable recommendations will be generated based on the finally estimated rating data.

The values of parameters were selected as follows: $\gamma = 10^{-4}$, $w_1 = 0.3, w_2 = 0.1$ and $\forall(i,k)\{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 2/9\}$. To choose the value for parameter τ, all values in user-user similarity matrix were sorted in ascending, and then, a value of $s_{i,j}$ that can retain top

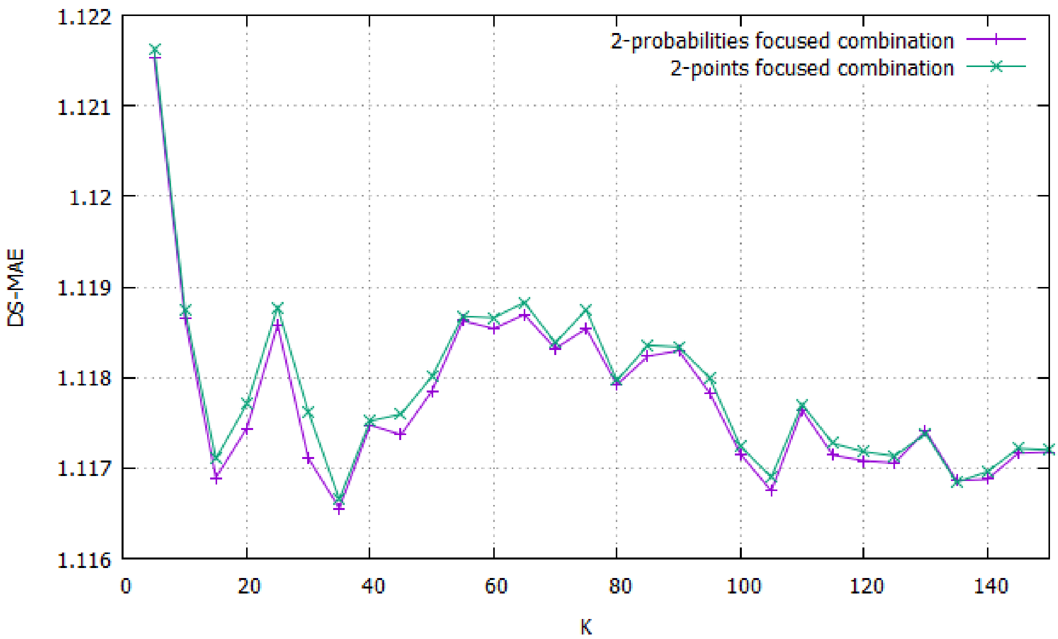

Figure 5.3: Overall $DS\text{-}MAE$ versus K (Flixster data set)

50% of the highest values in the rating matrix. In addition, this data set was separated into two parts, testing data and training data; the first one contains random 5 ratings of each user, and the other consists of the remaining ratings.

Overall $DS\text{-}MAE$ criterion results varies with neighborhood size K is depicted in Figure 5.3 . This figure shows that the performances of both combination methods are similar to each other and rise sharply when $K \leq 15$; with K in between 15 and 110, the performances are fluctuated; and then become quite stable when $K > 110$. As observed in this feature, regarding recommendation accuracy, *2-probabilities focused* combination is slightly better than *2-points focused* combination.

The computation time for the task of estimating ratings changes with neighborhood size K is depicted in Figure 5.4. As observed, execution time of *2-probabilities focused* combination is somewhat worse than but comparable to that of *2-points focused* combination. Additionally, this result is consistent with the result illustrated in Figure 5.2.

To evaluate the weakness of *2-probabilities focused* combination method, an experiment was conducted as follows. Seventeen users, each of them belongs to 4 communities concurrently, were selected; and these users as well as their corresponding communities are shown in Table 5.20. For each user, the estimated ratings on an item in his/her communities are considered as pieces of evidence of the finally estimated rating on the item.

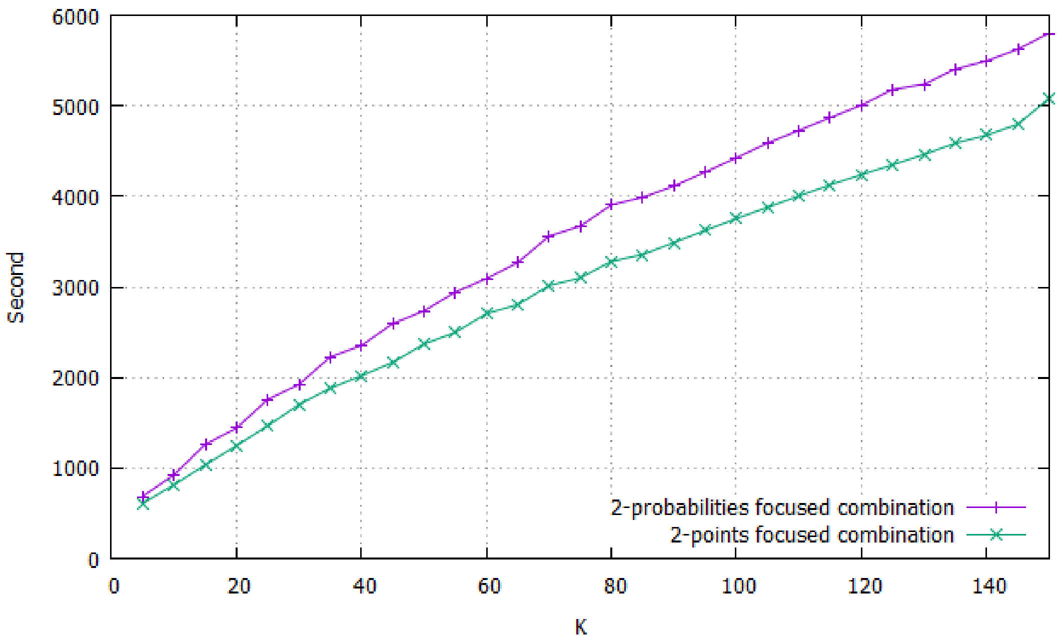

Figure 5.4: Overall time of computation versus K (Flixster data set)

Thus, the finally estimated rating is generated by combining corresponding 4 pieces of evidence by using this method. There are 24 combinations of the inputs when combing 4 pieces of evidence. The performances of recommendations regarding 24 combinations were evaluated by using $DS\text{-}MAE$ evaluation criterion; and the results with $K = 45$ are illustrated in Tables 5.21 and 5.22.

In Tables 5.21 and 5.22, each column presents the overall $DS\text{-}MAE$ for one user; and μ and SD are means and standard deviations of overall $DS\text{-}MAE$ over 24 combinations respectively. As observed, the standard deviations are very small (SD is smaller than 0.01 for 4 users, in between 0.01 and 0.1 for 12 users, and about 0.1059 for one user). That means, in the RSs, when combining information by using *2-probabilities focused* combination, input order is just minor affected the combined results.

5.4 Conclusion

In RSs based on DST, tasks of combining information are performed very open and play a vital role. However, highly conflicting mass functions representing user preferences are very common in the systems. Thus, in this chapter, we have developed a new combination method, called *2-probabilities focused* combination, which can help the systems handle

Table 5.20: Users belonging to four overlapping communities

User IDs	Community IDs					
	16	49	50	86	90	113	147
90			✓	✓	✓	✓	
206			✓	✓	✓	✓	
601		✓	✓		✓	✓	
1106		✓	✓		✓	✓	
1523			✓	✓	✓	✓	
1611			✓	✓	✓	✓	
1820		✓	✓		✓	✓	
2302			✓	✓	✓	✓	
2441			✓	✓	✓	✓	
2523		✓	✓		✓	✓	
2825			✓	✓	✓	✓	
3012		✓	✓		✓	✓	
3021		✓	✓		✓	✓	
3024		✓	✓		✓	✓	
3061			✓	✓	✓	✓	
3282			✓	✓	✓	✓	
3481			✓	✓	✓	✓	

combining highly conflicting mass functions. Also, the new method is capable of improving time of computation and tackling the weakness of an alternative combination method known as *2-points focused* combination.

Table 5.21: $DS\text{-}MAE$ varies with twenty four combinations (part 1)

No.	User IDs								
	90	206	601	1106	1523	1611	1820	2302	2441
1	1.20663	0.78436	1.19467	0.67151	1.09809	0.76991	1.19449	0.51485	1.13954
2	1.15631	0.77841	1.19384	0.88947	1.09985	0.77171	1.11603	0.52434	1.13692
3	1.18254	0.83257	1.19132	0.67529	1.10304	0.77412	1.17816	0.65597	1.13778
4	1.06767	0.74446	1.18026	0.89375	1.10540	0.76072	1.10144	0.71870	1.18033
5	1.04333	0.71440	1.18976	0.89158	1.10707	0.76905	1.12229	0.55391	1.16420
6	1.04686	0.72602	1.17988	0.89321	1.10642	0.75059	1.09904	0.58141	1.12196
7	1.12474	0.73366	1.18902	0.89312	1.10520	0.85901	1.04451	0.81300	1.18558
8	1.13057	0.70265	1.18832	0.89324	1.11029	0.76606	1.04080	0.69762	1.17637
9	1.12119	0.76086	1.18543	0.89281	1.10424	0.84659	1.12487	0.82616	1.17124
10	1.06767	0.74446	1.18026	0.89375	1.10540	0.76072	1.10144	0.71870	1.18033
11	1.05450	0.70331	1.18430	0.89323	1.10712	0.75395	1.10842	0.58004	1.15608
12	1.04686	0.72602	1.17988	0.89321	1.10642	0.75059	1.09904	0.58141	1.12196
13	1.14624	0.68937	1.19437	0.89390	1.10560	0.79425	1.04196	0.60060	1.14811
14	1.13057	0.70265	1.18832	0.89324	1.11029	0.76606	1.04080	0.69762	1.17637
15	1.13490	0.79629	1.19928	0.84851	1.09768	0.76911	1.06677	0.54010	1.13746
16	1.15631	0.77841	1.19384	0.88947	1.09985	0.77171	1.11603	0.52434	1.13692
17	1.05450	0.70331	1.18430	0.89323	1.10712	0.75395	1.10842	0.58004	1.15608
18	1.04333	0.71440	1.18976	0.89158	1.10707	0.76905	1.12229	0.55391	1.16420
19	1.14624	0.68937	1.19437	0.89390	1.10560	0.79425	1.04196	0.60060	1.14811
20	1.12474	0.73366	1.18902	0.89312	1.10520	0.85901	1.04451	0.81300	1.18558
21	1.13490	0.79629	1.19928	0.84851	1.09768	0.76911	1.06677	0.54010	1.13746
22	1.20663	0.78436	1.19467	0.67151	1.09809	0.76991	1.19449	0.51485	1.13954
23	1.12119	0.76086	1.18543	0.89281	1.10424	0.84659	1.12487	0.82616	1.17124
24	1.18254	0.83257	1.19132	0.67529	1.10304	0.77412	1.17816	0.65597	1.13778
μ	1.11796	0.74720	1.18920	0.85247	1.10417	0.78209	1.10323	0.63389	1.15463
SD	0.05274	0.04295	0.00583	0.08275	0.00377	0.03411	0.04882	0.10588	0.02020

Table 5.22: DS-MAE varies with twenty four combinations (part 2)

No.	User IDs							
	2523	2825	3012	3021	3024	3061	3282	3481
1	1.54115	1.13978	1.50942	1.29970	0.97805	0.58283	1.93319	1.09029
2	1.53108	1.08298	1.49625	1.29963	0.94727	0.64155	1.92527	1.09210
3	1.53921	1.10816	1.43177	1.29970	0.93814	0.57665	1.88074	1.06980
4	1.52362	1.06260	1.45589	1.29946	0.93895	0.70653	1.81822	1.07599
5	1.53234	0.95396	1.49751	1.29937	0.95869	0.75097	1.89421	1.02075
6	1.51809	0.99396	1.46803	1.29937	0.94765	0.74716	1.80553	1.00148
7	1.53751	1.01736	1.53441	1.29946	1.00114	0.51073	1.83120	1.29152
8	1.53106	1.03701	1.52828	1.29947	1.00664	0.70271	1.83379	1.29183
9	1.53389	1.10749	1.43813	1.29951	0.92185	0.62258	1.80976	1.13714
10	1.52362	1.06260	1.45589	1.29946	0.93895	0.70653	1.81822	1.07599
11	1.53366	1.00921	1.49699	1.29940	0.97276	0.74295	1.84090	0.99981
12	1.51809	0.99396	1.46803	1.29937	0.94765	0.74716	1.80553	1.00148
13	1.54034	1.03765	1.59069	1.29957	1.00905	0.56031	1.87270	1.15649
14	1.53106	1.03701	1.52828	1.29947	1.00664	0.70271	1.83379	1.29183
15	1.54321	1.05644	1.58039	1.29975	1.00663	0.56641	1.92080	1.09462
16	1.53108	1.08298	1.49625	1.29963	0.94727	0.64155	1.92527	1.09210
17	1.53366	1.00921	1.49699	1.29940	0.97276	0.74295	1.84090	0.99981
18	1.53234	0.95396	1.49751	1.29937	0.95869	0.75097	1.89421	1.02075
19	1.54034	1.03765	1.59069	1.29957	1.00905	0.56031	1.87270	1.15649
20	1.53751	1.01736	1.53441	1.29946	1.00114	0.51073	1.83120	1.29152
21	1.54321	1.05644	1.58039	1.29975	1.00663	0.56641	1.92080	1.09462
22	1.54115	1.13978	1.50942	1.29970	0.97805	0.58283	1.93319	1.09029
23	1.53389	1.10749	1.43813	1.29951	0.92185	0.62258	1.80976	1.13714
24	1.53921	1.10816	1.43177	1.29970	0.93814	0.57665	1.88074	1.06980
μ	1.53376	1.05055	1.50231	1.29953	0.96890	0.64261	1.86386	1.11015
SD	0.00720	0.05229	0.04950	0.00013	0.03047	0.08275	0.04570	0.09533

Chapter 6

Noise-Averse Combination

6.1 Introduction

As mentioned previously, in RSs based on DST, when several or a large number of mass functions are combined together by using Dempster's rule [7], in the focal set of the combined result, most focal elements usually have very low probabilities whereas a few focal elements have high probabilities. To deal with this issue, in [15, 17, 104], the authors introduced a combination method called *2-points focused* combination; and in the previous chapter, we also developed a new combination method called *2-probabilities* focused combination.

However, when using *2-points focused* or *2-probabilities focused* combination methods, focal elements with probabilities that are not very small might also be eliminated in some cases. With *2-points focused* combination method, focal elements (excluding the whole set element) which are not selected to the corresponding *triplet* mass function will always be eliminated regardless of whether their probabilities are infinitesimal or not. Additionally, when *2-probabilities focused* combination method is used for combining information, the focal elements whose probabilities are not in top two highest probabilities are constantly eliminated except for the whole set element.

Intuitively, focal elements with probabilities which are not very small could contain information that is valuable for the RSs. When these focal elements are transferred to the whole set element, the information is treated as ignorance in the reasoning process of generating suitable recommendations to a specific user. In other words, the elimination of this kind of focal elements could affect quality of recommendations because of loss of valuable information.

Under such an observation, in this chapter, we develop a new combination method that is capable of eliminating only the focal elements having very low probabilities. Regarding this method, first, a threshold to identify very low probabilities is clearly defined, and then focal elements with probabilities which are less than or equal to this threshold are eliminated. It is seen that reducing mass functions by using the threshold can help the RSs avoid unsatisfactory results, prevent loss of valuable information, improve computation time, and overcome the weakness of *2-points focused* combination method. Moreover, the new method is tested in a wide range of experiments on Movielens data set, and the experiment results indicate that the proposed method is much better than the baselines in accuracy of recommendations.

6.2 The Proposed Combination Method

Let us consider a RS based on DST with a rating domain which consists of L levels of preferences, $\Theta = \{\theta_1, \theta_2, ..., \theta_L\}$; and a mass function $m : 2^\Theta \to [0, 1]$. As mentioned previously, when mass function m is a combined result by using Dempster's rule of combination, only some focal elements in the focal set of this mass function have high probabilities whereas many others have very low probabilities [103].

Assuming that ϵ denotes a threshold to distinguish very low probabilities with the others; actually, ϵ should be a very small positive number. We propose that, in mass function m, focal elements with probabilities are less than or equal to threshold ϵ should be treated as noise and then eliminated. Note that, the probabilities of the eliminated focal elements must be added to that of the whole set element. After eliminating noise or the focal elements with very low probabilities, the achieved mass function is then called *noise-averse* mass function.

Formally, based on mass function m and threshold ϵ, a *noise-averse* mass function $\dot{m} : 2^\Theta \to [0, 1]$ is defined as follows

$$\begin{aligned} &\dot{m}(A) = m(A), \text{ for } A \subset \Theta \text{ and } m(A) > \epsilon; \\ &\dot{m}(\Theta) = m(\Theta) + \sum_{\{B \subset \Theta | m(B) \leq \epsilon\}} m(B). \end{aligned} \tag{6.1}$$

Subsequently, in RSs based on DST, ratings or user preferences are represented by *noise-averse* mass functions instead of general ones.

Additionally, let us consider two *noise-averse* mass functions $\dot{m}_1$ and $\dot{m}_2$ which are defined on the same rating domain Θ. The new combination method called *noise-averse* combination for combining these two mass functions, denoted by $\dot{m} = \dot{m}_1 \boxplus \dot{m}_2$, is defined as follows

$$\begin{aligned} \dot{m}(A) &= m'(A), \text{ for } A \subset \Theta \text{ and } m'(A) > \epsilon; \\ \dot{m}(\Theta) &= m'(\Theta) + \sum_{\{B \subset \Theta | m'(B) \leq \epsilon\}} m'(B), \\ &\text{where, } m' = \dot{m}_1 \oplus \dot{m}_2. \end{aligned} \tag{6.2}$$

As can be seen in Equation (6.2), first two *noise-averse* mass functions $\dot{m}_1$ and $\dot{m}_2$ are combined by using Dempster's rule of combination as shown in Equation (2.2), and then the combined result is transformed into corresponding *noise-averse* mass function $\dot{m}$ according to Equation (6.1).

By transferring the focal elements with probabilities less than or equal to threshold ϵ to the whole set element, the *noise-averse* combination method obtains the advantages as follows

- First, this method is capable of avoiding unsatisfactory results that are caused by combining highly conflicting mass functions by using Dempster's rule of combination.

- Second, this method helps to not only prevent loss of valuable information but also improve quality of recommendations because of its ability to retain all focal elements whose probabilities are greater than the threshold.

- Third, this method helps to improve time of computation because eliminating focal elements with very low probabilities will reduce time of computation.

- Finally, this method overcomes the problem of unstable results caused by the non-uniqueness of *triplet* mass functions when using *2-points focused* combination method.

However, *noise-averse* combination method is not associative. In the future, we will measure the effect of input orders on performances of recommendations when this method is used for combining information.

Supposing that we need to combine n *noise-averse* mass functions, which are defined on the same rating domain Θ, by using the new combination method. In the worst case, known as when the probabilities of focal elements (excluding the whole set element) in focal sets of both n *noise-averse* mass functions and the temporary results are all greater than threshold ϵ, it means that no focal elements are eliminated. In this situation, the time complexity of the new combination is the same as that of Dempster's rule of combination whose time complexity is exponential $O(|\Theta|^{n-1})$ [17]. Therefore, it can be said that the time complexity of the new combination method is still exponential in the worst case.

In the best case, when both n *noise-averse* focused mass functions and the temporary results are *2-points focused* mass functions, *noise-averse* combination method will be the same as *2-points focused* combination method whose time complexity is linear $O(n)$ [104].

6.3 Experiment

To evaluate *noise-averse* combination method, we integrated it into the RS based on DST [97], and selected *2-points focused* and *2-probabilities focused* combination methods for performance comparison on both accuracy of recommendations as well as time of computation. In addition, we also chose evaluation criteria *DS-MAE* [6], *DS-Precision* [22], *DS-Recall* [22], and *DS-F*1 [6] for measuring performances.

MovieLens data set described in Section 3.6.2 was used in experiments. In this data set, *genres* of movies are considered as context information; and we assumed that if a user has rated an item then this user is said to be interested in all genres to which the item belongs. Moreover, the RS [97] works with soft ratings; thus, each hard rating in the data set was transformed into the corresponding soft rating by using Equation 3.33.

In the experiments, we assumed that focal elements with probabilities which are less than or equal to 10^{-10} are considered as noise; thus, $\epsilon = 10^{-10}$ was selected for *noise-averse* combination method. Additionally, the values of the other parameters were selected based on the analyzed results published in [97, 6], as follows: $w_1 = 0.3, w_2 = 0.1$, $\gamma = 10^{-5}$, $\beta = 1$, and $\forall i, k\{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 2/9\}$. Particularly, it is unreasonable to select a fixed value for parameter τ to use in the experiments. The reason is that, with different

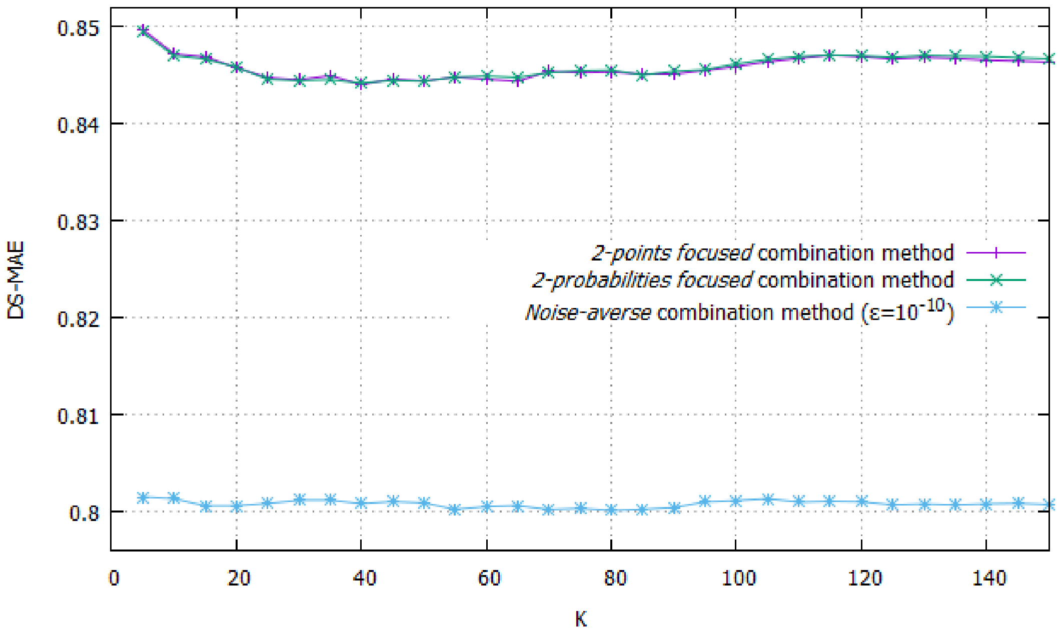

Figure 6.1: Results evaluated by criterion *DS-MAE* versus K

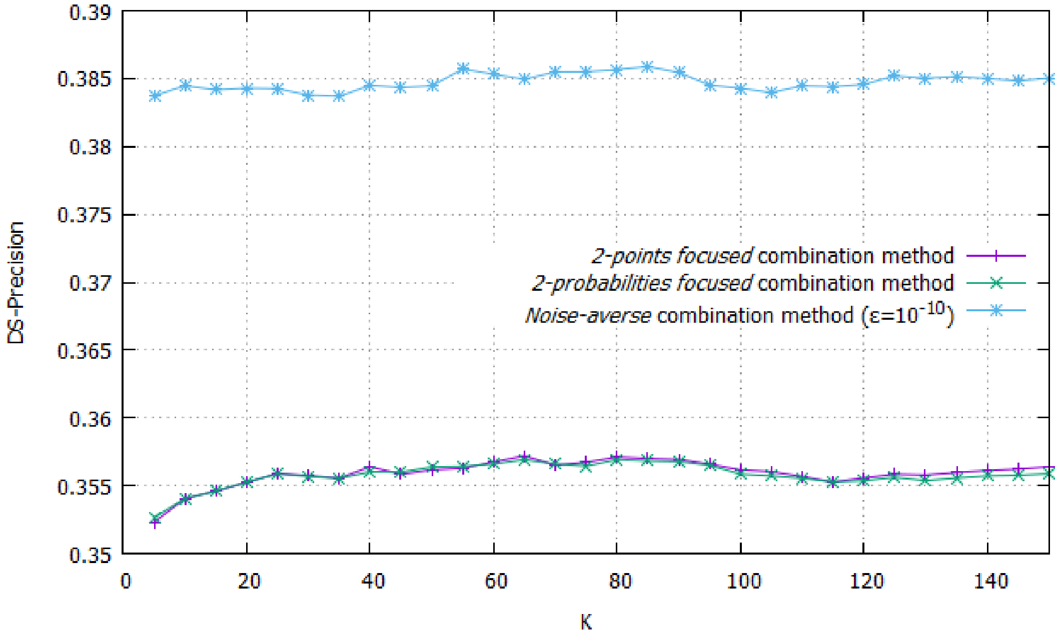

Figure 6.2: Results evaluated by criterion *DS-Precision* versus K

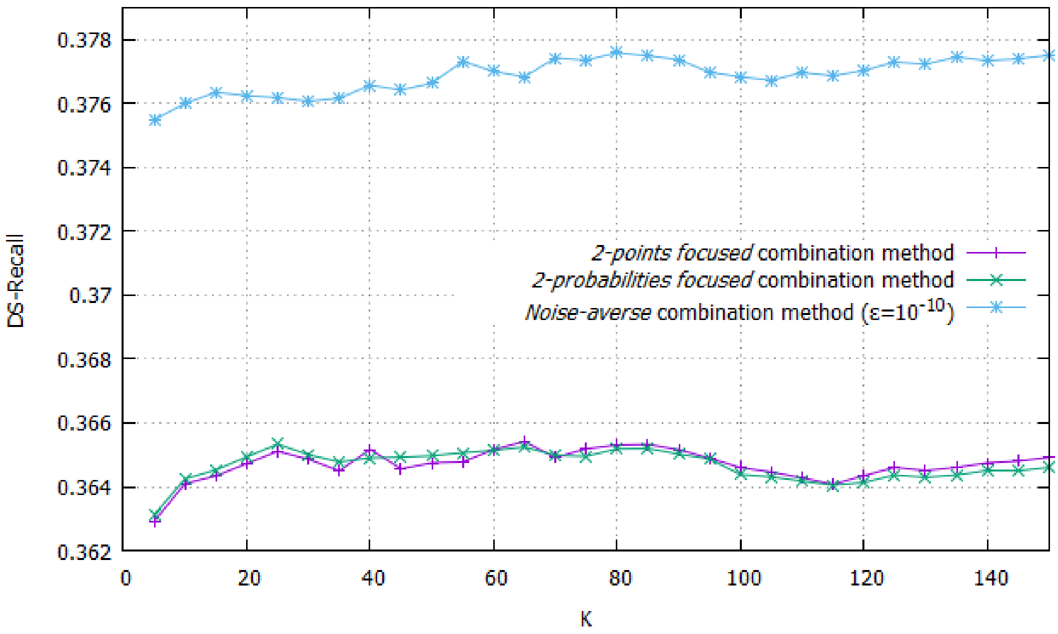

Figure 6.3: Results evaluated by criterion *DS-Recall* versus K

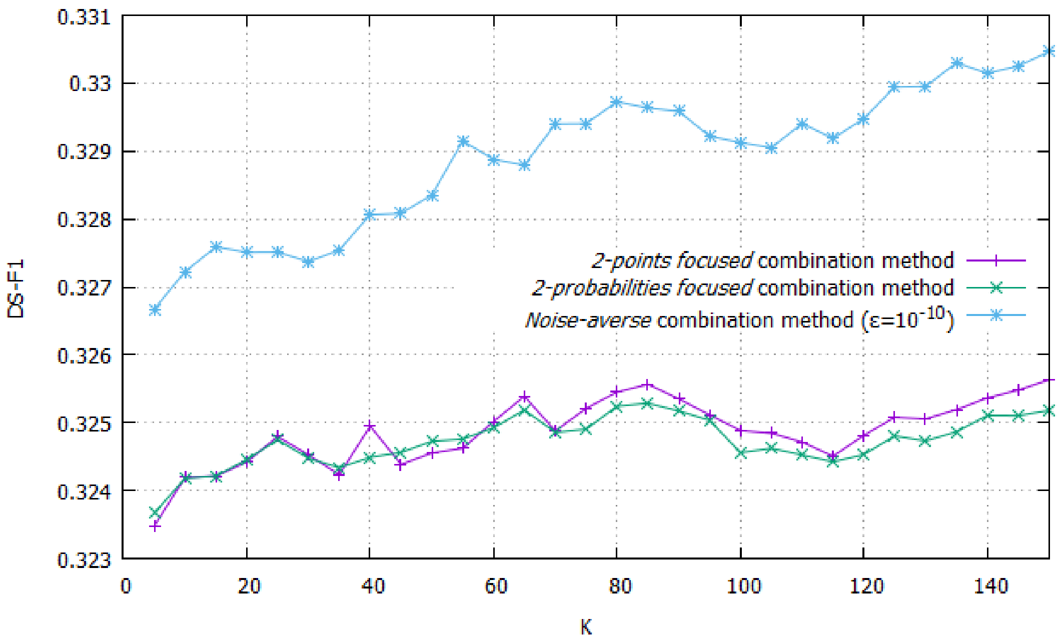

Figure 6.4: Results evaluated by criterion $DS\text{-}F1$ versus K

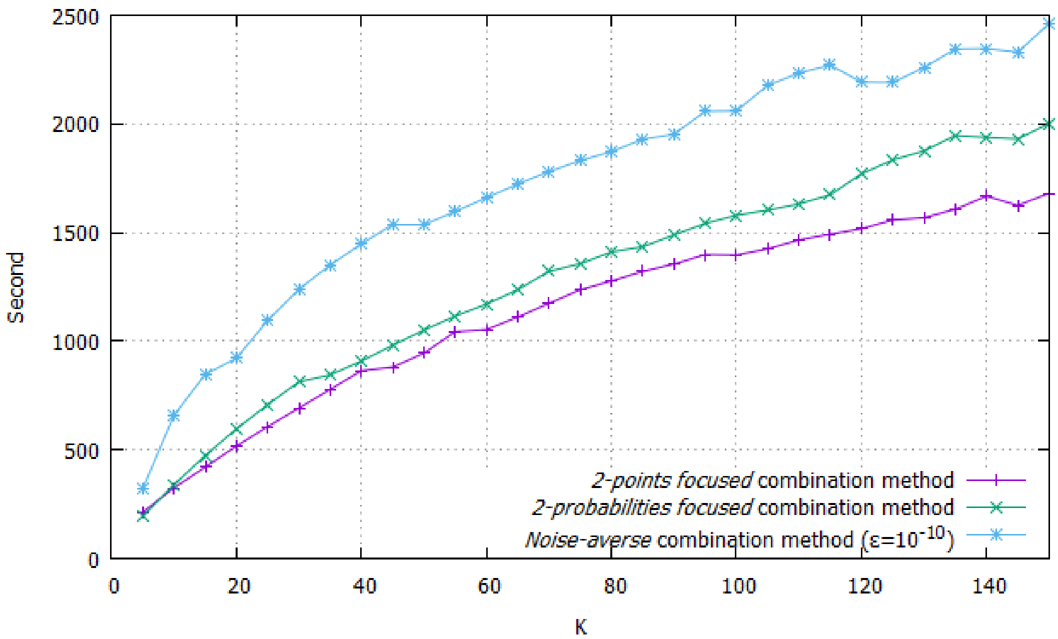

Figure 6.5: Overall time of computation versus K

combination methods, the values of user-user similarities of two specific users are different. To select the value for parameter τ, we first sorted all values in the user-user similarity matrix [97] in ascending order, and then we selected the value which can retain top 65% of the highest values in rating matrix as the value for parameter τ. Besides, 10% ratings of each user were randomly selected to be the testing set and the remaining ratings were considered as the training set.

Figure 6.1 illustrates the overall results of recommendation performances evaluated by criterion *DS-MAE* changes with neighborhood size K. As observed in this figure, performances of *2-points focused* and *2-probabilities focused* combination methods are similar to each other. In particular, the performance of *noise-averse* combination method is much better than those of *2-points focused* and *2-probabilities focused* combination methods. Furthermore, Figures 6.2, 6.3, and 6.4 show the same comparison results obtained by evaluating with criteria *DS-Precision*, *DS-Recall*, and *DS-F*1 respectively.

As for comparing performances in time of computation, we measured the execution time for the task of estimating ratings [97] varies with neighborhood size K. The comparison results are depicted in Figure 6.5. Regarding this figure, *2-points focused* combination method is the most effective one; the second place is *2-probabilities focused* combination method; and *noise-averse* combination method is in the last place. However, as observed, the performance in time of computation of the new combination method is not much worse than those of the baselines.

6.4 Conclusion

In this chapter, we developed a new reducing combination method, called *noise-averse* combination, for combining information in RSs based on DST. By reducing focal elements whose probabilities are less than or equal to a very small threshold ϵ, this method is capable of preventing loss of valuable information about user preferences in the systems. Regarding the experimental results, the new method outperforms the baselines in accuracy of recommendations; in addition, time of computation of this method is not much worse than that of *2-points focused* combination method whose time complexity is linear.

Chapter 7

Mixed Rules of Combination

7.1 Introduction

In RSs based on DST, tasks of combining information play a significant role as well as being used very often; and, in almost all cases, Dempster's rule of combination [7] is applied. However, this combination method does not allow to combine totally conflicting mass functions. Therefore, in the existing RSs based on DST, totally conflicting mass functions need to be eliminated in the data sets. To do this, values of dispersion factors in the Dempster-Shafer modeling functions [6] have been selected to be less than 1 to make sure that the whole set element is added to corresponding mass functions. In general, some people can express their preferences on an item with rating values that are completely different from the ones evaluated by some others. In other words, several mass functions corresponding to the ratings on an item can be totally conflicting. Thus, in RSs based DST, these mass functions need to be combined together.

In this chapter, we first discuss the characteristics of combining information in RSs based on DST, and then analyze six popular combination methods in the context of RSs. Based on the analysis results, we propose two new mixed combination methods that help to handle combining totally conflicting mass functions in RSs based on DST. To evaluate the new methods, we integrate them into a typical RS based on DST, and then measure recommendation performances on Movielens data set.

7.2 Characteristics of Combining Information

In RSs based on DST, user preferences on items are represented as mass functions, and recommendations are mainly generated based on combining these mass functions. To work with these systems, a binary combination operator, denoted by $\odot$, is required to satisfy three basic requirements as below

- Commutativity. $m_1 \odot m_2 = m_2 \odot m_1$ for any two mass functions m_1 and m_2;
- Associativity. $(m_1 \odot m_2) \odot m_3 = m_1 \odot (m_2 \odot m_3)$ for any three mass functions m_1, m_2 and m_3;
- Idempotence. $m = m \odot m$ for any mass function m.

Among these requirements, only the first one is mandatory. In addition, if operator $\odot$ is not associative but quasi-associative [105], it is still accepted. Operator $\odot$ is quasi-associative if it supports updating an existing combined result when a new mass function needs to be combined with that result [106, 105]. In case, this operator is neither associative nor quasi-associative, but orders of inputs just slightly affect recommendation performances; it can be used in the systems.

As observed in the RSs, conflicting ratings usually occur because of the diversity of users. When handling conflict by using Yager's rule of combination [107], the conflict is transferred into the whole set element; therefore, in most cases, combined results will be vacuous. On the contrary, if Smets' rule of combination [108] is applied, the conflict is transferred into the empty set element; and in this situation, combined results have a high possibility to be a mass function whose focal set contains only the emptyset, called *fully-subnormal mass function.* Therefore, with both Yager's and Smets' rules of combination, the achieved information is very high uncertainty. Furthermore, in the systems, the number of ratings on each item increases over time, and new ratings are required to combine with the existing ones on this item for the purpose of updating knowledge about users' preferences on the item. When combing a combined result, which is vacuous, with any other new mass function by using Yager's or DP's [109] rule of combination, the obtained result will always be vacuous. Whereas, with Smets's rule of combination, when combining *fully-subnormal mass function* with a new mass function, the combined result will still be *fully-subnormal mass function* regardless of the new

Table 7.1: Ratings on item I_k

Users	Items	Ratings
U_i	I_k	$r_{i,k}(\{5\}) = 0.15$ $r_{i,k}(\{4\}) = 0.85$
U_j	I_k	$r_{j,k}(\{3\}) = 0.90$ $r_{j,k}(\{2\}) = 0.10$

one. The issue of unchanged results when combining with new mass functions is called *non-sense combination*; and in the RSs, it is better if operator $\odot$ can void this issue.

Additionally, in the RSs, generally different users can have different preferences on an item; thus they can evaluate opposite ratings on the same item. For this reason, totally conflicting mass functions are common in the systems. In fact, these mass functions also need to be combined together, as illustrated in Example 7. Therefore, operator $\odot$ must be capable of dealing with this issue, called *totally-conflicting combination*; and this requirement cannot be optional.

Example 7 Let us consider a RS based on DST with a rating domain $\Theta = \{1, 2, 3, 4, 5\}$. Supposing that, in this system, (1) each rating of a user on an item is considered as a piece of users' preference on the item, and (2) users' preference on an item is computed by combing all pieces of users' preference on this item. In addition, assuming that there are only two users U_i and U_j who have rated item k, and the ratings are represented as two mass functions $r_{i,k}$ and $r_{j,k}$ shown in Table 7.1. As we can see in this table, these mass functions are totally conflicting. Obviously, the users' preference on item k does exist; thus, in this case, two mass functions $r_{i,k}$ and $r_{j,k}$ must be combined together.

As for generating recommendations for a particular user, the RSs commonly predict the ratings of this user on all unseen items, rank these ratings, and then select the highest ranked items. In the predicting process, combining operations are used frequently for combining pieces of evidence of this user's preference on each unseen item [12, 97, 6]. However, with Smets's rule of combination, a predicted rating can be fully subnormal; thus it cannot compare with other ones, this issue is called *incomparable ratings.* Therefore, in order to be used in the systems, operator $\odot$ need to make sure that predicted ratings on unseen items are able to be ranked; and this requirement is mandatory.

In short, beside three basic requirements, operator $\odot$ needs to be capable of overcoming three problems as listed below

- Non-sense combination;
- Totally-conflicting combination;
- Incomparable ratings.

7.3 Existing Combination Methods

In a RS based on DST, let us consider n mass functions $m_1, m_2, ..., m_n$ defined on the same rating domain Θ. Assuming that the sources providing these mass functions are independent. In this section, six popular combination methods for combining two or all of these mass functions together will be analyzed according to the requirements listed in the previous section.

7.3.1 Dempster's Rule of Combination

Dempster's rule of combination [7] for combining two mass functions m_1 and m_2 has been presented in Section 2.1. Nevertheless, for better understanding in this chapter, this combination method is re-denoted by $m_\oplus = m_1 \oplus m_2$ as below

$$\begin{aligned} m_\oplus(\emptyset) &= 0; \\ m_\oplus(A) = \frac{1}{1-\mathcal{K}} \sum_{B,C\subseteq\Theta, B\cap C=A} m_1(B)m_2(C),\ &\text{for } \emptyset \neq A \subseteq \Theta; \\ \text{where } \mathcal{K} = \sum_{B,C\subseteq\Theta, B\cap C=\emptyset} m_1(B)m_2(C) &\neq 0. \end{aligned} \tag{7.1}$$

Here, $\mathcal{K}$ represents the basic probability mass associated with conflict before normalization. When $\mathcal{K} = 1$, $m_\oplus$ does not exist; that means, this combination method is not able to overcome *totally-conflicting combination* problem.

It can be seen that Dempster's rule of combination is commutative, associative, but not idempotent [7, 106]. Additionally, it is capable of dealing with *non-sense combination* problem. Besides, predicted ratings that are generated by using this combination method are always comparable.

7.3.2 Smets's Rule of Combination

Smets's rule of combination [108], also known as conjunctive rule of combination, for combining two mass functions m_1 and m_2, denoted by $m_{\otimes} = m_1 \otimes m_2$, is defined as follows

$$m_{\otimes}(A) = \sum_{B,C \subseteq \Theta, B \cap C = A} m_1(B) m_2(C), \text{ for } A \subseteq \Theta. \tag{7.2}$$

As can be seen in this equation, masses are not re normalized and the conflict is stored in the mass given to the empty set. The same as Dempster's rule of combination, this combination method is commutative, associative, but not idempotent [108]. Furthermore, it is capable of tackling *totally-conflicting combination* problem. However, when combining two totally conflicting ones together, the focal set of the combined result will always be *fully-subnormal mass function*, as illustrated in Example 8.

Example 8 Let us consider the RS in Example 7 again. Assuming that only two users U_i and U_j have rated item I_t, and the ratings are represented as two mass functions $r_{i,t}$ and $r_{j,t}$ shown in Table 7.2. When computing users' preferences by using Smets' rule of combination, users' preference on item k is the same as that on item t, as shown below

$$\begin{aligned} (r_{i,k} \otimes r_{j,k})(\emptyset) &= 1; \\ (r_{i,k} \otimes r_{j,k})(A) &= 0, \text{ for } \emptyset \neq A \subseteq \Theta; \\ (r_{i,t} \otimes r_{j,t})(\emptyset) &= 1; \\ (r_{i,t} \otimes r_{j,t})(A) &= 0, \text{ for } \emptyset \neq A \subseteq \Theta. \end{aligned} \tag{7.3}$$

As can be seen, these combined results are unreasonable because ratings on item i are significantly different from those on item i'.

When combining n mass functions $m_1, m_2, ..., m_n$ together by using Smets's rule of combination; if at least two mass functions m_i and m_j with $i \neq j$ are totally conflicting, regardless of the other mass functions, the combined result will always be fully subnormal. In addition, in some cases, the predicted ratings can also be fully subnormal. Thus,

Table 7.2: Ratings on item I_t

Users	Items	Ratings
U_i	I_t	$r_{i,t}(\{2,3\}) = 0.05$ $r_{i,t}(\{2\}) = 0.95$
U_j	I_t	$r_{j,t}(\{1\}) = 1.00$

as mentioned earlier, this combination method can not tackle both *no-sense combination* and *incomparable ratings* problems.

7.3.3 Yager's Rule of Combination

Yager's rule of combination [107] for combining two mass functions m_1 and m_2, denoted by $m_{\boxplus} = m_1 \boxplus m_2$, is defined as below

$$m_{\boxplus}(A) = \begin{cases} 0, \text{ if } A = \emptyset; \\ \displaystyle\sum_{B,C \subseteq \Theta, B \cap C = A} m_1(B)m_2(C), \text{for } \emptyset \neq A \subset \Theta; \\ m_1(\Theta)m_2(\Theta) + \displaystyle\sum_{B,C \subseteq \Theta, B \cap C = \emptyset} m_1(B)m_2(C), \text{if } A = \Theta. \end{cases} \tag{7.4}$$

This combination method is commutative, but neither associative nor idempotent [107]. Actually, it is quasi-associative [105]; and the corresponding n-ary operator [106] to combine n mass functions $m_1, m_2, ..., m_n$ is given by

$$\begin{aligned} m_{\boxplus}(\emptyset) &= 0; \\ m_{\boxplus}(A) &= \sum_{\cap_{i=1}^{n} A_i = A} m_1(A_1)m_2(A_2)...m_n(A_n), \text{ for } \emptyset \neq A \subset \Theta; \\ m_{\boxplus}(\Theta) &= m_1(\Theta)m_2(\Theta)...m_n(\Theta) + \sum_{\cap_{i=1}^{n} A_i = \emptyset} m_1(A_1)m_2(A_2)...m_n(A_n); \\ & \text{where } A_i \subseteq \Theta \text{ with } i = 1, ..., n. \end{aligned} \tag{7.5}$$

With this n-ary operator, a combined result is capable of updating when a new mass function is available [106].

In addition, with Yager's rule of combination, the conflict is transferred to the whole set element Θ. For this reason, this combination method is able to deal with *totally-conflicting combination* problem. But, when two totally conflicting mass functions are combined together, the combined result is always vacuous, as demonstrated in Example 9.

Example 9 Let us consider the RS in Examples 7 and 8. When computing users' preferences on item U_i and U_j by using Yager' rule of combination, combined results are similar, as below

$$\begin{aligned} (r_{i,k} \boxplus r_{j,k})(\Theta) &= 1; \\ (r_{i,k} \boxplus r_{j,k})(A) &= 0, \text{ for } A \subset \Theta. \\ (r_{i,t} \boxplus r_{j,t})(\Theta) &= 1; \\ (r_{i,t} \boxplus r_{j,t})(A) &= 0, \text{ for } A \subset \Theta. \end{aligned} \tag{7.6}$$

As observed, the combined results are unreasonable.

Moreover, when combining n mass functions, $m_1, m_2, ..., m_n$ by using Yager's rule of combination, the combined result will always be vacuous if there are at least two mass functions m_i and m_j with $i \neq j$ are totally conflicting. Thus, this combination method is not able to overcome *non-sense combination* problem. Besides, predicted ratings generated based on this combination method can be ranked.

7.3.4 Dubois and Prade's Rule of Combination

Dubois and Prade (DP)'s rule of combination [109], called disjunctive rule of combination, for combining two mass functions m_1 and m_2, denoted by $m_{\circledcirc} = m_1 \circledcirc m_2$, is given by

$$m_{\circledcirc}(A) = \sum_{B,C \subseteq \Theta, B \cup C = A} m_1(B) m_2(C), \text{ for } A \subseteq \Theta. \tag{7.7}$$

This combination method is commutative, associative, but not idempotent [109]. In addition, it supports combining totally conflicting mass functions. However, when combining a large number of mass functions together by using this method, the combined result is vacuous in most of cases. The reason is that, with union operator, $\cup$, the larger number of mass functions are combined, the higher possibility of the union result containing all elements in rating domain.

Moreover, when the combined result is vacuous, it will still be vacuous regardless of other new mass functions that are added. Besides, this combination method is capable of tackling *incomparable ratings* problem.

7.3.5 Dubois and Prade's "hybrid" Rule of Combination

DP's "hybrid" rule of combination [110] for combining two mass functions m_1 and m_2, denoted by $m_{\ominus} = m_1 \ominus m_2$, is defined as follows

$$\begin{aligned} m_{\ominus}(\emptyset) &= 0; \\ m_{\ominus}(A) &= \sum_{B,C\subseteq\Theta, B\cap C=A} m_1(B)m_2(C) \\ &\quad + \sum_{B,C\subseteq\Theta, B\cap C=\emptyset, B\cup C=A} m_1(B)m_2(C), \text{ for } \emptyset \neq A \subseteq \Theta. \end{aligned} \tag{7.8}$$

This combination method is commutative, but neither associative nor idempotent [110]. In addition, it is able to overcome *totally-conflicting combining*, *non-sense combination*, and *incomparable ratings* problems.

7.3.6 Averaging Rule of Combination

The averaging rule of combination [111], also known as mixing combination rule, for combining two mass functions m_1 and m_2, denoted by $m_{\circledast} = m_1 \circledast m_2$, is defined as below

$$m_{\circledast}(A) = \frac{1}{2}\big(m_1(A) + m_2(A)\big), \text{ for } A \subseteq \Theta. \tag{7.9}$$

This combination method is commutative and idempotent. In addition, it is not associative, but quasi-associative [106]. Let $m_{\circledast}$ denotes the combined result after combining n mass functions $m_1, m_2, ..., m_n$ with $n > 1$, $m_{\circledast} = m_1 \circledast m_2 \circledast ... \circledast m_n$. When a new mass function m' needs to be combined with this combined result, it can be updated as follows

$$m_{\circledast}(A) = \frac{m_{\circledast}(A) \times n + m'(A)}{n+1}, \text{ for } \forall A \subseteq \Theta. \tag{7.10}$$

Besides, the same as DP's "hybrid" rule of combination, this combination method is capable of dealing with the three problems.

7.3.7 Summary

The summary of analyzing six popular combination methods is shown in Table 7.3. As observed in this table, all six selected combination methods satisfy the first mandatory requirement (Commutativity). However, Dempster's rule of combination can not meet the next mandatory requirement (TCC), and Smets' rule of combination does not satisfy

Table 7.3: Summary of analyzing popular combination methods

Methods	Requirements					
	Commutativity⋆	Associativity	Idempotence	NSC	TCC⋆	IR⋆
Dempster's	Yes	Yes	No	Yes	No	Yes
Smets's	Yes	Yes	No	No	Yes	No
Yager's	Yes	No‡	No	No	Yes	Yes
DP's	Yes	Yes	No	No	Yes	Yes
DP's "hybrid"	Yes	No	No	Yes	Yes	Yes
Averaging	Yes	No‡	Yes	Yes	Yes	Yes

NSC: *Tackling non-sense combination problem.*

TCC: *Tackling totally-conflicting combination problem.*

IR: *Tackling incomparable ratings problem.*

⋆ *Requirements are mandatory.*

‡ *Combination methods are not associative, but quasi-associative.*

the last mandatory requirement (IR). Therefore, both of these combination methods are not suitable for RSs based on DST.

The last four combination methods listed in this table can be used for combining information in the systems. However, when combining totally conflicting mass functions with Yager's rule of combination, the combined result will always be vacuous; and when combining a large number of mass functions together, both Yager's and DP's combination rules suffer from the *non-sense combination* problem. Moreover, only DP's "hybrid" rule of combination is neither associative nor quasi-associative; thus, in the experiments we will measure the effect of input orders on recommendation performances.

7.4 Two Mixed Rules of Combination

According to the literature, among the six popular combination methods, Dempster's rule of combination is the most well-known as well as playing an important role in the research area of DST [99]. So far, this combination method has been applied in various applications [100, 37, 112]. However, as mentioned earlier, it cannot work with RSs based on DST because of inability to overcome *fully-conflicting combination* problem. Moreover, when combining highly conflicting mass functions, this combination method can generate

Table 7.4: Results of combining two totally conflicting mass functions

Dempter's	Smets's	Yager's	DP's or DP's "hybrid"	Averaging
N/A	$r_{\otimes}(\emptyset) = 1$	$r_{\boxplus}(\{1,2,3,4,5\}) = 1$	$r_{\circledcirc}(\{3,5\}) = r_{\ominus}(\{3,5\}) = 0.135$ $r_{\circledcirc}(\{2,5\}) = r_{\ominus}(\{2,5\}) = 0.015$ $r_{\circledcirc}(\{3,4\}) = r_{\ominus}(\{3,4\}) = 0.765$ $r_{\circledcirc}(\{2,4\}) = r_{\ominus}(\{2,4\}) = 0.085$	$r_{\circledast}(\{5\}) = 0.075$ $r_{\circledast}(\{4\}) = 0.425$ $r_{\circledast}(\{3\}) = 0.45$ $r_{\circledast}(\{2\}) = 0.05$

Mass function r is the combined result after combining two mass functions $r_{u,i}$ and $r_{u',i}$.

counterintuitive results [14].

As for better understanding abilities to deal with totally conflicting mass functions of six methods, let us consider two examples illustrated in Tables 7.4. Table 7.4 shows results after combining two totally conflicting ratings presented in Table 7.1. As can be seen in this table, when combining two totally conflicting mass functions, DP's, DP's "hybrid" and averaging rule of combinations can achieve the most reasonable results.

Under such an observation, we propose two new mixed combination methods for combining information in RSs based on DST. Formally, let us consider two mass functions m_1 and m_2 defined on rating domain Θ. In order to combine these two mass functions, a combination of DP's and Dempster's rules of combination, called *mixed 1 rule of combination* and denoted by $m_{\boxminus} = m_1 \boxminus m_2$, is defined as follows

$$m_{\boxminus}(A) = \begin{cases} 0, & \text{for } A = \emptyset; \\ \dfrac{1}{1-\mathcal{K}} \displaystyle\sum_{B,C \subseteq \Theta, B \cap C = A} m_1(B) m_2(C), & \text{for } \mathcal{K} < \eta_1, \emptyset \neq A \subseteq \Theta; \\ \displaystyle\sum_{B,C \subseteq \Theta, B \cup C = A} m_1(B) m_2(C), & \text{for } \mathcal{K} \geq \eta_1, \emptyset \neq A \subseteq \Theta; \end{cases} \tag{7.11}$$

$$\text{where } \mathcal{K} = \sum_{B,C \subseteq \Theta, B \cap C = \emptyset} m_1(B) m_2(C) \text{ and } \eta_1 \in [0,1].$$

The second new method is a combination of Dempster's and averaging rule of combination, called *mixed 2 rule of combination* and denoted by $m_{\boxtimes} = m_1 \boxtimes m_2$, is given by

$$m_{\boxtimes}(A) = \begin{cases} 0, & \text{for } A = \emptyset; \\ \dfrac{1}{1-\mathcal{K}} \displaystyle\sum_{B,C \subseteq \Theta, B \cap C = A} m_1(B)m_2(C), & \text{for } \mathcal{K} < \eta_2, \emptyset \neq A \subseteq \Theta; \\ \dfrac{1}{2}(m_1(A) + m_2(A)), & \text{for } \mathcal{K} \geq \eta_2, \emptyset \neq A \subseteq \Theta; \end{cases} \tag{7.12}$$

$$\text{where } \mathcal{K} = \sum_{B,C \subseteq \Theta, B \cap C = \emptyset} m_1(B)m_2(C) \text{ and } \eta_2 \in [0,1].$$

The parameters η_1 and η_2 represent a threshold for separating low conflicting from high conflicting mass functions; and the values for them are selected according to a specific RS based on DST. When two mass functions are low conflicting, they are combined by using Dempster's rule of combination; otherwise, DP's or averaging rules of combination is employed.

It can be seen that two new mixed combination methods can satisfy three mandatory requirements and an optional one (N.S.C). However, these combination methods are neither idempotent nor associative. In the experiments, we will measure the influence of input orders on recommendation performances when these methods are applied.

7.5 Experiment

In order to evaluate two new mixed combination methods, they were integrated into a RS [97], and the recommendation performances according to these methods were measured. Note that, for these methods, we selected $\eta_1 = \eta_2 = 1$; that means DP's or averaging rules of combination are only employed to combine totally conflicting mass functions.

As mentioned earlier, among the six popular combination methods, four of them can be applied into RSs based on DST. In the experiments, all of these four methods were used as baselines. In addition, to measure recommendation performances, evaluation criteria $DS\text{-}MAE$ [6], *DS-Precision* [22], *DS-Recall* [22], and $DS\text{-}F1$ [6] were selected.

Movielens data set described in Section 3.6.1 was selected for the experiments. In this data set, information about the genres, in which a user is interested, is not available; thus, we assumed that a user is interested in a genre if this user has rated at least 5 items belonging to that genre. In addition, each hard rating was transformed into a corresponding soft rating by using Equation (3.33). Note that, when computing user-user

Table 7.5: Performance comparison according to *DS-MAE*

K	Combination Methods					
	Yager's	DP's	DP's "hybrid"	Averaging	Mixed 1	Mixed 2
5	1.51340	1.49701	0.97056	1.08703	0.86083	0.85828
10	1.51341	1.49969	0.97846	1.08775	0.85347	0.85159
15	1.51341	1.50121	0.98228	1.08832	0.85203	0.84956
20	1.51340	1.50212	0.98622	1.08876	0.84884	0.84715
25	1.51339	1.50269	0.99078	1.08909	0.84846	0.84713
30	1.51338	1.50304	0.99389	1.08945	0.84740	0.84649
35	1.51337	1.50324	0.99993	1.08970	0.84723	0.84568
40	1.51336	1.50337	1.00344	1.08994	0.84638	0.84542
45	1.51336	1.50346	1.00628	1.09024	0.84603	0.84581
50	1.51335	1.50352	1.00811	1.09043	0.84640	0.84625
55	1.51335	1.50356	1.00789	1.09055	0.84626	0.84601
60	1.51334	1.50358	1.00831	1.09065	0.84636	0.84591
65	1.51334	1.50360	1.00941	1.09073	0.84612	0.84516
70	1.51334	1.50361	1.01020	1.09083	0.84577	0.84493
75	1.51334	1.50362	1.00990	1.09093	0.84574	0.84475
80	1.51334	1.50363	1.01048	1.09108	0.84595	0.84517
85	1.51334	1.50363	1.01044	1.09117	0.84621	0.84541
90	1.51334	1.50363	1.01049	1.09126	0.84631	0.84564
95	1.51334	1.50363	1.01131	1.09133	0.84613	0.84592
100	1.51334	1.50363	1.01062	1.09138	0.84628	0.84567
105	1.51333	1.50364	1.01132	1.09142	0.84618	0.84557
110	1.51333	1.50364	1.01162	1.09147	0.84621	0.84548
115	1.51333	1.50364	1.01028	1.09151	0.84581	0.84558
120	1.51333	1.50364	1.00987	1.09156	0.84573	0.84524
125	1.51333	1.50364	1.00930	1.09162	0.84558	0.84527
130	1.51333	1.50364	1.00952	1.09166	0.84551	0.84494
135	1.51333	1.50364	1.00964	1.09170	0.84547	0.84482
140	1.51333	1.50364	1.01081	1.09176	0.84575	0.84494
145	1.51333	1.50364	1.01083	1.09182	0.84545	0.84488
150	1.51333	1.50364	1.01105	1.09187	0.84582	0.84506

similarity, if the focal set of a rating does not consist of the whole set element, the whole set element with very low probability ϵ, an infinitesimal positive number, needs to be added into the focal set. The user-user similarity between two users was computed by using the Equations 3.27, 3.28, and 3.29. To select the value for threshold τ, all values of user-user similarity were sorted in ascending, and then the value that can retain top 80% of highest values was chosen. The values of the other parameters were selected as follows: $\forall(i,k), \{\alpha_{i,k}, \sigma_{i,k}\} = \{0.9, 0.1\}, w_1 = 0.3, w_2 = 0.1, \gamma = 10^{-6}$ and $\epsilon = 10^{-300}$. Besides, 10% of ratings in the data set were randomly selected as testing data, the remaining ratings were used as training data.

Table 7.5 summarizes recommendation performances changing with neighborhood size K according to $DS\text{-}MAE$. In this table, overall performance of each combination method is presented in a column, the underlined value indicates the best one in each row. As can be seen, two new mixed combination methods outperform the other ones for all the cases. In fact, the second mixed combination method is in the first place; following is the first mixed combination method; the third place is DP's "hybrid" rule of combination; averaging rule of combination is better than DP's rule of combination; and Yager's rule of combination is the worst one. Besides, recommendation performance of DP's rule of combination is just slightly better than that of Yager's rule of combination; and performances of these two methods are significantly worse than the others. The reason is that when combining a large number of mass functions with these two methods, the combined results usually are vacuous.

In Fig. 7.1, the similar information, as presented in Table 7.5, is depicted in a visualization way. Additionally, the result of performance comparison according *DS-Recall* is the same as the comparison result according to $DS\text{-}MAE$, as illustrated in Fig. 7.2. That means two proposed combination methods also outperform the baselines with criterion *DS-Recall*.

Fig. 7.3 and Fig. 7.4 show the results of performance comparison according to criteria *DS-Precision* and $DS\text{-}F1$, respectively. Regarding these figures, DP's "hybrid" rule of combination is the best one, and Yager's and DP's rules of combination are still significantly worse than the others. In addition, the performances of two proposed combination methods can be comparable to DP's "hybrid" and averaging rules of combination.

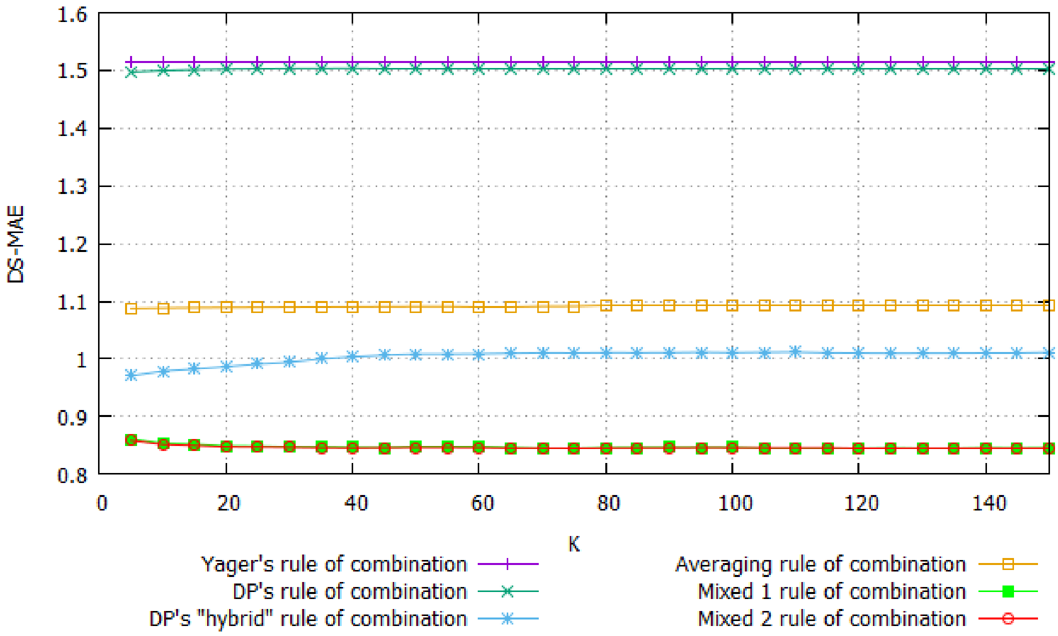

Figure 7.1: Performances according to *DS-MAE* versus K

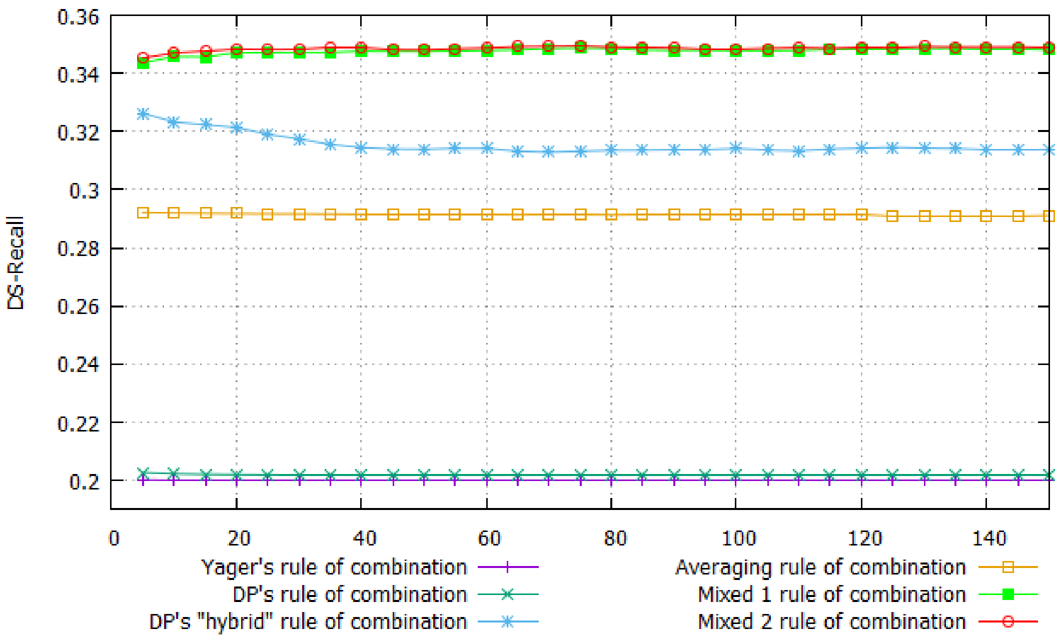

Figure 7.2: Performances according to *DS-Recall* versus K

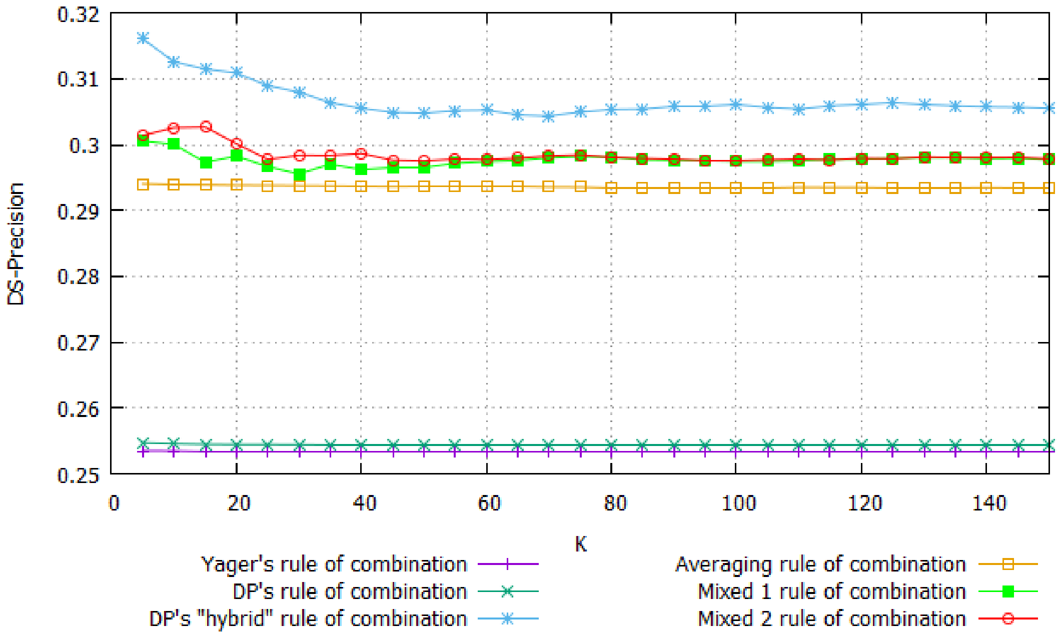

Figure 7.3: Performances according to *DS-Precision* versus K

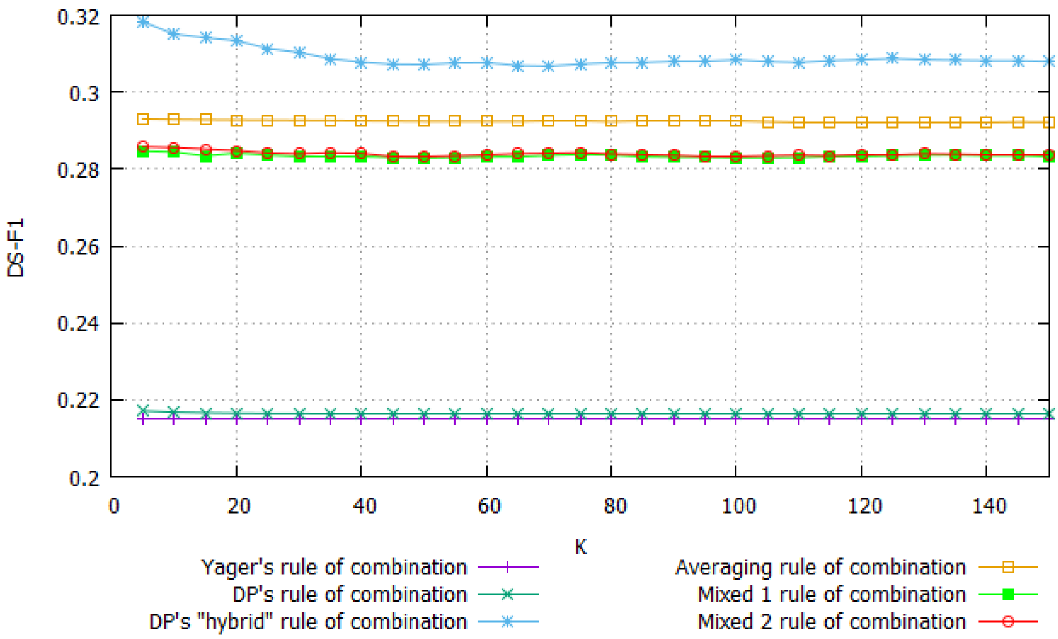

Figure 7.4: Performances according to $DS\text{-}F1$ versus K

Table 7.6: Changes of performances according to *DS-MAE*

No.	Combination Methods		
	DP's "hybrid"	Mixed 1	Mixed 2
1	0.951786	0.851369	0.850781
2	0.953341	0.851369	0.850781
3	0.952666	0.851369	0.850781
4	0.954449	0.851369	0.850781
5	0.958122	0.851369	0.850781
6	0.956171	0.851369	0.850781
7	0.952064	0.851369	0.850781
8	0.954039	0.851369	0.850781
9	0.951597	0.851369	0.850781
10	0.954449	0.851369	0.850781
11	0.957031	0.851369	0.850781
12	0.956171	0.851369	0.850781
13	0.952015	0.851369	0.850781
14	0.954039	0.851369	0.850781
15	0.952465	0.851369	0.850781
16	0.953341	0.851369	0.850781
17	0.957031	0.851369	0.850781
18	0.958122	0.851369	0.850781
19	0.952015	0.851369	0.850781
20	0.952064	0.851369	0.850781
21	0.952465	0.851369	0.850781
22	0.951786	0.851369	0.850781
23	0.951597	0.851369	0.850781
24	0.952666	0.851369	0.850781
μ	0.953812	0.851369	0.850781
SD	4.66123e-06	1.28619e-32	2.05790e-31

To evaluate the effect of input orders on two new mixed combination methods as well as DP's "hybrid" rule of combination, we conducted an experiment as follows. First, we selected neighborhood size $K = 4$ and threshold $\tau = 0$. Then, we measured the recommendation performances for each user on an item according to 24 possible combinations of 4 neighborhoods. The performances according to $DS\text{-}MAE$ are presented in Table 7.6. As observed, the standard deviations are very small in all the cases. That means, in RSs based on DST, when combining information by using two new mixed combination methods or DP's "hybrid" of combination, input orders are just minorly affected the recommendation performances.

7.6 Conclusion

RSs based on DST are effective in not only modeling user preferences with uncertain, imprecise and incomplete information but also combining information from different sources. For users, these systems offer a more realistic and flexible way to represent ratings (soft ratings). In this chapter, have investigated tasks of combining information in these systems, and listed three basic requirements as well as three particular problems (*non-sense combination*, *fully-conflicting combination* and *incomparable ratings*) for these tasks. In addition, we have analyzed six popular combination methods and pointed out which of them can be used in the systems. Especially, we have developed two new mixed combination methods for information fusion in RSs based on DST. To test and evaluate these methods, we have integrated them into the RS introduced in [97] and measured recommendation performances on Movielens data set.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

More recently, some researchers have focused on developing RSs based on DST. Comparing to traditional RSs which offer hard ratings, the RSs based on DST have some advantages such as representing user preferences in a more realistic and flexible way, modeling user preferences with subjective, qualitative, and imperfect information, and combining information about user preferences from different sources easily. However, so far, research on RSs based on DST is still limited. In this research, we have presented an intensive study of integrating RSs based on DST with social networks, computing user-user similarities, and combining information in RSs based on DST.

In the research, we pointed out that using community context information or community preferences helps to improve accuracy of recommendations and tackle the sparsity and cold-start problems in RSs based on DST. Additionally, when computing user-user similarities, provided ratings should be weighted more important than predicted ones.

As observed, in RSs based on DST, tasks of combining information are performed very open and play a vital role. In addition, highly conflicting mass functions which represent user preferences on items are very common. Thus, in this research, we have developed two new combination methods, called *2-probabilities focused* combination and noise-averse combination, which help the systems handle combining highly conflicting mass functions, improve time of computation, and overcoming the weakness of an alternative combination method known as *2-points focused* combination.

Also, we have investigated tasks of combining information in these systems, and listed three basic requirements and three particular problems (*non-sense combination*, *fully-conflicting combination* and *incomparable ratings*) for these tasks. In addition, we have analyzed six popular combination methods and pointed out which of them can be used in the systems. Especially, we have developed two new mixed combination methods for information fusion in the systems.

8.2 Suggestions for Future Research

As we can see, more work needs to be done before we can draw more solid conclusions about RSs based on DST. For the future research, we suggest some possible directions as follows

- The quality of community preferences can be improved not only by detecting overlapping communities in social networks by using other attributes such as tagging, messaging, frequency of discussing, but also by dealing with the grey-sheep users problem [113].

- Further evaluating as well as demonstrating the advantages of *noise-averse* combination method need to be done, such as measuring the dependence of accuracy of recommendations and time of computation on the threshold ϵ, and measuring the effect of input orders on performances of recommendations when combining information by using this combination method.

- Two new mixed rules of combination also need to be further evaluated with different data sets and the best values for parameters η_1 and η_2 need to be investigated.

- Combination of *noise-averse* combination method with averaging rule of combination could be an effective way for fusing information about user preferences in RSs based on DST.

Bibliography

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, 2013.

[3] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook.* Springer, 2011.

[4] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, 2015.

[5] G. R. Santhanam, S. Basu, and V. Honavar, "Representing and reasoning with qualitative preferences for compositional systems," *J. Artif. Intell. Res. (JAIR)*, vol. 42, pp. 211–274, 2011.

[6] T. L. Wickramarathne, K. Premaratne, M. Kubat, and D. T. Jayaweera, "CoFiDS: A belief-theoretic approach for automated collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 2, pp. 175–189, 2011.

[7] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Ann. Math. Stat.*, vol. 38, pp. 325–339, 1967.

[8] G. Shafer, *A mathematical theory of evidence.* Princeton University Press, 1976.

[9] J. Iglesias, A. M. Bernardos, and J. R. Casar, "An evidential and context-aware recommendation strategy to enhance interactions with smart spaces," in *HAIS*, 2013, pp. 242–251.

[10] L. Troiano, L. J. Rodríguez-Muñiz, and I. Díaz, "Discovering user preferences using Dempster-Shafer theory," *Fuzzy Set. Syst.*, vol. 278, pp. 98–117, 2015.

[11] L. Tang and H. Liu, *Community detection and mining in social Media*, ser. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2010.

[12] V.-D. Nguyen and V.-N. Huynh, "A community-based collaborative filtering system dealing with sparsity problem and data imperfections," in *PRICAI*, 2014, pp. 884–890.

[13] P. Smets, "Analyzing the combination of conflicting belief functions," *Inf. Fusion*, vol. 8, no. 4, pp. 387–412, 2007.

[14] L. A. Zadeh, "Book review: A mathematical theory of evidence," *AI Mag.*, vol. 5, no. 3, pp. 81–83, 1984.

[15] D. A. Bell, J. W. Guan, and Y. Bi, "On combining classifier mass functions for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 10, pp. 1307–1319, 2005.

[16] Y. Bi, "The impact of diversity on the accuracy of evidential classifier ensembles," *Int. J. Approx. Reasoning*, vol. 53, no. 4, pp. 584–607, 2012.

[17] Y. Bi, J. Guan, and D. A. Bell, "The combination of multiple classifiers using an evidential reasoning approach," *Artif. Intell.*, vol. 172, no. 15, pp. 1731–1751, 2008.

[18] W. Feller, *An introduction to probability theory and its applications.* J. Wiley & Sons, 1968, vol. 1.

[19] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Set. Syst.*, vol. 1, pp. 3–28, 1978.

[20] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, 1992.

[21] B. Khaleghi, A. Khamis, F. Karray, and S. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28 – 44, 2013.

[22] K. Hewawasam, K. Premaratne, and M.-L. Shyu, "Rule mining and classification in a situation assessment application: A belief-theoretic approach for handling data imperfections," *IEEE Trans. Syst. Man Cybern., Part B*, vol. 37, no. 6, pp. 1446–1459, 2007.

[23] T. Denoeux, "40 years of Dempster-Shafer theory," *Int. J. Approx. Reasoning*, vol. 79, pp. 1–6, 2016.

[24] M. A. Boujelben, Y. D. Smet, A. Frikha, and H. Chabchoub, "Building a binary outranking relation in uncertain, imprecise and multi-experts contexts: The application of evidence theory," *Int. J. Approx. Reasoning*, vol. 50, no. 8, pp. 1259–1278, 2009.

[25] Y. Wang, J. Yang, D. Xu, and K. Chin, "The evidential reasoning approach for multiple attribute decision analysis using interval belief degrees," *Eur. J. Oper. Res.*, vol. 175, no. 1, pp. 35–66, 2006.

[26] V.-N. Huynh, J. Lawry, and Y. Nakamori, "Integrated uncertainty management for decision making," *Ann. Oper. Res.*, vol. 195, no. 1, pp. 1–4, 2012.

[27] V. M. Mondéjar-Guerra, R. Muñoz-Salinas, M. J. Marín-Jiménez, Á. C. Poyato, and R. M. Carnicer, "Keypoint descriptor fusion with Dempster-Shafer theory," *Int. J. Approx. Reasoning*, vol. 60, pp. 57–70, 2015.

[28] B. Lelandais, I. Gardin, L. Mouchard, P. Vera, and S. Ruan, "Dealing with uncertainty and imprecision in image segmentation using belief function theory," *Int. J. Approx. Reasoning*, vol. 55, no. 1, pp. 376–387, 2014.

[29] M. Kurdej, J. Moras, V. Cherfaoui, and P. Bonnifait, "Controlling remanence in evidential grids using geodata for dynamic scene perception," *Int. J. Approx. Reasoning*, vol. 55, no. 1, pp. 355–375, 2014.

[30] J. Clemens, T. Reineking, and T. Kluth, "An evidential approach to SLAM, path planning, and active exploration," *Int. J. Approx. Reasoning*, vol. 73, pp. 1–26, 2016.

[31] J. Klein, C. Lecomte, and P. Miché, "Hierarchical and conditional combination of belief functions induced by visual tracking," *Int. J. Approx. Reasoning*, vol. 51, no. 4, pp. 410–428, 2010.

[32] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "Tracking an unknown number of targets using multiple sensors: A belief propagation method," in *FUSION*, 2016, pp. 719–726.

[33] F. Faux and F. Luthon, "Theory of evidence for face detection and tracking," *Int. J. Approx. Reasoning*, vol. 53, no. 5, pp. 728–746, 2012.

[34] T. Reineking, "Active classification using belief functions and information gain maximization," *Int. J. Approx. Reasoning*, vol. 72, pp. 43–54, 2016.

[35] S. Trabelsi, Z. Elouedi, and P. Lingras, "Classification systems based on rough sets under the belief function framework," *Int. J. Approx. Reasoning*, vol. 52, no. 9, pp. 1409–1432, 2011.

[36] T. Denoeux and M. Masson, "Evidential reasoning in large partially ordered sets - application to multi-label classification, ensemble clustering and preference aggregation," *Ann. Oper. Res.*, vol. 195, no. 1, pp. 135–161, 2012.

[37] V.-N. Huynh, N. T. Tri, and C. A. Le, "Adaptively entropy-based weighting classifiers in combination using Dempster-Shafer theory for word sense disambiguation," *Comput. Speech Lang.*, vol. 24, no. 3, pp. 461–473, 2010.

[38] M. Masson and T. Denoeux, "Ensemble clustering in the belief functions framework," *Int. J. Approx. Reasoning*, vol. 52, no. 1, pp. 92–109, 2011.

[39] Z. Liu, Q. Pan, J. Dezert, and G. Mercier, "Credal c-means clustering method based on belief functions," *Knowl.-Based Syst.*, vol. 74, pp. 119–132, 2015.

[40] T. Denoeux, O. Kanjanatarakul, and S. Sriboonchitta, "EK-NNclus: A clustering procedure based on the evidential K-nearest neighbor rule," *Knowl.-Based Syst.*, vol. 88, pp. 57–69, 2015.

[41] Q. Luo, Y. Peng, X. Peng, and A. El-Saddik, "Uncertain data clustering-based distance estimation in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6584–6605, 2014.

[42] P. H. Giang, "Decision making under uncertainty comprising complete ignorance and probability," *Int. J. Approx. Reasoning*, vol. 62, pp. 27–45, 2015.

[43] T. Flaminio, L. Godo, and H. Hosni, "Coherence in the aggregate: A betting method for belief functions on many-valued events," *Int. J. Approx. Reasoning*, vol. 58, pp. 71–86, 2015.

[44] H. Tang, "A novel fuzzy soft set approach in decision making based on grey relational analysis and dempster-shafer theory of evidence," *Appl. Soft Comput.*, vol. 31, pp. 317–325, 2015.

[45] P. Smets, "Belief functions," in *Non-Standard Logics for Automated Reasoning*, P. Smets, E. H. Mamdani, D. Dubois, and H. Prade, Eds. Academic Press, 1988, pp. 1–25.

[46] V.-N. Huynh, "Discounting and combination scheme in evidence theory for dealing with conflict in information fusion," in *MDAI*, 2009, pp. 217–230.

[47] P. Smets, "Belief induced by the partial knowledge of the probabilities," in *UAI*, 1994, pp. 523–530.

[48] ——, "Practical uses of belief functions," in *UAI*. Morgan Kaufmann Publishers Inc., 1999, pp. 612–621.

[49] D. Goldberg, D. A. Nichols, B. M. Oki, and D. B. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[50] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *SIGKDD*, 2009, pp. 627–636.

[51] T. Mahmood and F. Ricci, "Improving recommender systems with adaptive conversational strategies," in *HYPERTEXT*, 2009, pp. 73–82.

[52] P. Resnick and H. R. Varian, “Recommender systems - introduction to the special section,” *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[53] C. Martínez-Cruz, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma, “A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling,” *Inf. Sci.*, vol. 311, pp. 102–118, 2015.

[54] A. M. Rashid, G. Karypis, and J. Riedl, “Learning preferences of new users in recommender systems: An information theoretic approach,” *SIGKDD Explor.*, vol. 10, no. 2, pp. 90–100, 2008.

[55] E. Damiani, P. Ceravolo, F. Frati, V. Bellandi, R. Maier, I. Seeber, and G. Waldhart, “Applying recommender systems in collaboration environments,” *Comput. Hum. Behav.*, vol. 51, pp. 1124–1133, 2015.

[56] H. Langseth and T. D. Nielsen, “A latent model for collaborative filtering,” *Int. J. Approx. Reasoning*, vol. 53, no. 4, pp. 447–466, 2012.

[57] N. Cao, C. Shi, W. S. Lin, J. Lu, Y. Lin, and C. Lin, “Targetvue: Visual analysis of anomalous user behaviors in online communication systems,” *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 280–289, 2016.

[58] E. Costa-Montenegro, A. B. Barragáns-Martínez, and M. Rey-López, “Which app? A recommender system of applications in markets: Implementation of the service for monitoring users’ interaction,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9367–9375, 2012.

[59] E. R. Núñez-Valdéz, J. M. C. Lovelle, O. S. Martínez, V. García-Díaz, P. O. de Pablos, and C. E. M. Marín, “Implicit feedback techniques on recommender systems applied to electronic books,” *Comput. Hum. Behav.*, vol. 28, no. 4, pp. 1186–1193, 2012.

[60] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, “Context-aware recommender systems,” *AI Mag.*, vol. 32, no. 3, pp. 67–80, 2011.

[61] Z. D. Champiri, S. R. Shahamiri, and S. S. B. Salim, "A systematic review of scholar context-aware recommender systems," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1743–1758, 2015.

[62] L. O. Colombo-Mendoza, R. Valencia-García, A. R. González, G. Alor-Hernández, and J. J. Samper-Zapater, "Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1202–1222, 2015.

[63] J. Bernabé-Moreno, Á. Tejeda-Lorente, C. Porcel, H. Fujita, and E. Herrera-Viedma, "CARESOME: A system to enrich marketing customers acquisition and retention campaigns using social media information," *Knowl.-Based Syst.*, vol. 80, pp. 163–179, 2015.

[64] Y. Li, C. Chou, and L. Lin, "A social recommender mechanism for location-based group commerce," *Inf. Sci.*, vol. 274, pp. 125–142, 2014.

[65] M. Balabanovic and Y. Shoham, "Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[66] D. Jannach, M. Zanker, M. Ge, and M. Gröning, "Recommender systems in computer science and information systems - A landscape of research," in *EC-Web*, 2012, pp. 76–87.

[67] M. Grcar, D. Mladenic, B. Fortuna, and M. Grobelnik, "Data sparsity issues in the collaborative filtering framework," in *WebKDD*, 2005, pp. 58–76.

[68] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, 2004.

[69] X. Luo, Y. Xia, and Q. Zhu, "Incremental collaborative filtering recommender based on regularized matrix factorization," *Knowl.-Based Syst.*, vol. 27, pp. 271–280, 2012.

[70] L. Pu and B. Faltings, "Understanding and improving relational matrix factorization in recommender systems," in *RecSys*, 2013, pp. 41–48.

[71] J. Sun, D. Parthasarathy, and K. Varshney, "Collaborative kalman filtering for dynamic matrix factorization," *IEEE Trans. Sign. Proc.*, vol. 62, no. 14, pp. 3499–3509, 2014.

[72] C. Jiang, R. Duan, H. K. Jain, S. Liu, and K. Liang, "Hybrid collaborative filtering for high-involvement products: A solution to opinion sparsity and dynamics," *Decis. Support Syst.*, vol. 79, pp. 195–208, 2015.

[73] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5-6, pp. 393–408, 1999.

[74] H. Kim, A. Ji, I. Ha, and G. Jo, "Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation," *Electron. Commer. Res. Appl.*, vol. 9, no. 1, pp. 73–83, 2010.

[75] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2065–2073, 2014.

[76] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *SIGIR*, 2009, pp. 195–202.

[77] Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang, and H. Yan, "Recommender systems based on social networks," *J. Syst. Software*, vol. 99, pp. 109–119, 2015.

[78] S. M. Kywe, E. Lim, and F. Zhu, "A survey of recommender systems in Twitter," in *SocInfo*, 2012, pp. 420–433.

[79] C. D. Maio, G. Fenza, V. Loia, and M. Parente, "Time aware knowledge extraction for microblog summarization on Twitter," *Inf. Fusion*, vol. 28, pp. 60–74, 2016.

[80] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowl.-Based Syst.*, vol. 57, pp. 57–68, 2014.

[81] W. Yuan, D. Guan, Y. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowl.-Based Syst.*, vol. 23, no. 3, pp. 232–238, 2010.

[82] H. Wu, K. Yue, Y. Pei, B. Li, Y. Zhao, and F. Dong, "Collaborative topic regression with social trust ensemble for recommendation in social media systems," *Knowl.-Based Syst.*, vol. 97, pp. 111–122, 2016.

[83] J. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," *Knowl.-Based Syst.*, vol. 22, no. 4, pp. 261–265, 2009.

[84] V. Nguyen and V. Huynh, "Two-probabilities focused combination in recommender systems," *Int. J. Approx. Reasoning*, vol. 80, pp. 225–238, 2017.

[85] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[86] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.

[87] S. Gregory, "An algorithm to find overlapping community structure in networks," in *PKDD*, 2007, pp. 91–102.

[88] ——, "A fast algorithm to find overlapping communities in networks," in *ECML/PKDD (1)*, 2008, pp. 408–423.

[89] J. Xie and B. Szymanski, "Towards linear time overlapping community detection in social networks," in *PAKDD (2)*, 2012, pp. 25–36.

[90] M. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, 2003.

[91] ——, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.

[92] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[93] U. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E*, vol. 76, no. 3, pp. 036 106+, 2007.

[94] H. Chan and A. Darwiche, "A distance measure for bounding probabilistic belief change," *Int. J. Approx. Reasoning*, vol. 38, no. 2, pp. 149–174, 2005.

[95] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *SIGIR*, 1999, pp. 230–237.

[96] I. Bloch, "Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account," *Pattern Recogn. Lett.*, vol. 17, no. 8, pp. 905–919, 1996.

[97] V.-D. Nguyen and V.-N. Huynh, "A reliably weighted collaborative filtering system," in *ECSQARU*, 2015, pp. 429–439.

[98] M. Papagelis, D. Plexousakis, and T. Kutsuras, "Alleviating the sparsity problem of collaborative filtering using trust inferences," in *iTrust*, 2005, pp. 224–239.

[99] T. Denoeux, "Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence," *Artif. Intell.*, vol. 172, no. 2-3, pp. 234–264, 2008.

[100] B. Chang, H. Tsai, and P. Yu, "Character recognition based on neural network and Dempster-Shafer theory," in *ICTAI*, 2007, pp. 418–423.

[101] T. Denoeux, "Maximum likelihood estimation from uncertain data in the belief function framework," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 119–130, 2013.

[102] B. Quost, M. Masson, and T. Denoeux, "Classifier fusion in the Dempster-Shafer framework using optimized t-norm based combination rules," *Int. J. Approx. Reasoning*, vol. 52, no. 3, pp. 353–374, 2011.

[103] V.-D. Nguyen and V.-N. Huynh, "Evidence combination focusing on significant focal elements for recommender systems," in *IUKM*, 2015, pp. 290–302.

[104] Y. Bi, "An efficient triplet-based algorithm for evidential reasoning," *Int. J. Intell. Syst.*, vol. 23, no. 4, pp. 483–516, 2008.

[105] R. R. Yager, "Quasi-associative operations in combination of evidence," *Kybernetes*, vol. 16, pp. 37–41, 1987.

[106] K. Sentz and S. Ferson, "Combination of evidence in Dempster-Shafer theory," *Technical report, Sandia National Laboratories*, 2002.

[107] R. R. Yager, "On the Dempster-Shafer framework and new combination rules," *Inf. Sci.*, vol. 41, no. 2, pp. 93–137, 1987.

[108] P. Smets, "Decision making in the TBM: The necessity of the pignistic transformation," *Int. J. Approx. Reasoning*, vol. 38, no. 2, pp. 133–147, 2005.

[109] D. Dubois and H. Prade, "A set-theoretic view of belief functions: Logical operations and approximations by fuzzy sets," *Int. J. Gen. Syst.*, vol. 12, no. 3, pp. 193–226, 1986.

[110] ——, "Representation and combination of uncertainty with belief functions and possibility measures," *Comput. Intell.*, vol. 4, pp. 244–264, 1988.

[111] C. K. Murphy, "Combining belief functions when evidence conflicts," *Decis. Support Syst.*, vol. 29, no. 1, pp. 1–9, 2000.

[112] C. R. Parikh, M. J. Pont, and N. B. Jones, "Application of Dempster-Shafer theory in condition monitoring applications: A case study," *Pattern Recogn. Lett.*, vol. 22, no. 6/7, pp. 777–785, 2001.

[113] M. A. Ghazanfar and A. Prügel-Bennett, "Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3261–3275, 2014.

Publications

Journal papers

1. Nguyen, V.-D. and Huynh, V.-N., "Two-Probabilities Focused Combination in Recommender Systems," *International Journal of Approximate Reasoning*, vol. 80, pp. 225-238, January 2017.

2. Nguyen, V.-D., Huynh, V.-N. and Sriboonchitta, S., "Integrating Community Context Information into a Reliably Weighted Collaborative Filtering System Using Soft Ratings," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (minor revision, resubmitted for the second round of review).

3. Nguyen, V.-D., Sriboonchitta, S. and Huynh, V.-N., "Using Community Preference for Solving Sparsity and Cold-Start Problems in Collaborative Filtering Approach," *Electronic Commerce Research and Applications* (under review).

Peer-reviewed conference/symposium papers

4. Nguyen, V.-D. and Huynh, V.-N., "Noise-Averse Combination Method," in *Proceedings of the 28th International Conference on Tools with Artificial Intelligence (ICTAI 2016)*, pp. 86-90, 6-8 November 2016, San Jose, California, United States.

5. Nguyen, V.-D. and Huynh, V.-N., "On Information Fusion in Recommender Systems Based-on Dempster-Shafer Theory," in *Proceedings of the 28th International Conference on Tools with Artificial Intelligence (ICTAI 2016)*, pp. 78-85, 6-8 November 2016, San Jose, California, United States.

6. Nguyen, V.-D. and Huynh, V.-N., "Integrating with Social Network to Enhance Recommender System Based-on Dempster-Shafer Theory," in *Proceedings of the*

5th International Conference on Computational Social Networks (CSoNet 2016), pp. 170-181, LNCS 9795, 2-4 August 2016, Ho Chi Minh City, Vietnam.

7. Nguyen, V.-D. and Huynh, V.-N., "Evidence Combination Focusing on Significant Focal Elements for Recommender Systems," in *Proceedings of the 4th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM 2015)*, pp. 290-302, LNCS 9376, 15-17 October 2015, Nha Trang, Vietnam.

8. Nguyen, V.-D. and Huynh, V.-N., "A Reliably Weighted Collaborative Filtering System," in *Proceedings of the 13th European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2015)*, pp. 429-439, LNAI 9161, 15-17 July 2015, Compiegne, France.

9. Nguyen, V.-D. and Huynh, V.-N., "A Community-Based Collaborative Filtering System Dealing with Sparsity Problem and Data Imperfections," in *Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2014)*, pp. 884-890, LNCS 8862, 1-5 December 2014, Gold Coast, Queensland, Australia.