

2008-01-01

A Belief Theoretic Approach for Automated Collaborative Filtering

Thanuka Lakmal Wickramarathne
University of Miami, thanuka@gmail.com

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_theses

Recommended Citation

Wickramarathne, Thanuka Lakmal, "A Belief Theoretic Approach for Automated Collaborative Filtering" (2008). *Open Access Theses*. 182.
https://scholarlyrepository.miami.edu/oa_theses/182

This Open access is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Theses by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.

UNIVERSITY OF MIAMI

A BELIEF THEORETIC APPROACH FOR AUTOMATED COLLABORATIVE
FILTERING

By

Thanuka Lakmal Wickramarathne

A THESIS

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science

Coral Gables, Florida

May 2008

UNIVERSITY OF MIAMI

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science

A BELIEF THEORETIC APPROACH FOR AUTOMATED COLLABORATIVE
FILTERING

Thanuka Lakmal Wickramaratne

Approved:

Dr. Kamal Premaratne
Professor of Electrical and
Computer Engineering

Dr. Terri A. Scandura
Dean of the Graduate School

Dr. Miroslav Kubat
Associate Professor of Electrical
and Computer Engineering

Dr. Dushyantha T. Jayaweera
Professor of Clinical Medicine

WICKRAMARATHNE, T. L.

(M.S., Electrical and Computer
Engineering)

A Belief Theoretic Approach for
Automated Collaborative Filtering

(May 2008)

Abstract of a thesis at the University of Miami.

Thesis supervised by Professor Kamal Premaratne.

No. of pages in text. (84)

Automated Collaborative Filtering (ACF) is one of the most successful strategies available for recommender systems. Application of ACF in more sensitive and critical applications however has been hampered by the absence of better mechanisms to accommodate imperfections (ambiguities and uncertainties in ratings, missing ratings, etc.) that are inherent in user preference ratings and propagate such imperfections throughout the decision-making process. Thus one is compelled to make various “assumptions” regarding the user preferences giving rise to predictions that lack sufficient integrity.

With its Dempster-Shafer belief theoretic basis, CoFiDS, the automated Collaborative Filtering algorithm proposed in this thesis, can (a) represent a wide variety of data imperfections; (b) propagate the partial knowledge that such data imperfections generate throughout the decision-making process; and (c) conveniently incorporate contextual information from multiple sources.

The “soft” predictions that *CoFiDS* generates provide substantial flexibility to the domain expert. Depending on the associated DS theoretic belief-plausibility measures, the domain expert can either render a “hard” decision or narrow down the possible set of predictions to as smaller set as necessary. With its capability to accommodate data imperfections, CoFiDS widens the applicability of ACF, from the more popular domains, such as movie and book recommendations, to more sensitive and critical problem domains, such as medical expert support systems, homeland security and surveillance, etc.

We use a benchmark movie dataset and a synthetic dataset to validate CoFiDS and compare it to several existing ACF systems.

*To my parents,
for everything*

Acknowledgements

The research reported in this thesis was carried out at the Department of Electrical and Computer Engineering of the University of Miami, Florida, during the period from May 2006 to Dec 2007.

I extend my sincere gratitude and appreciation to my thesis supervisor and chairman of the thesis committee, Professor Kamal Premaratne, for his guidance, support and suggestions throughout this work. My special thanks also go to Professors Miroslav Kubat of the Department of Electrical and Computer Engineering, and Dushyantha T. Jayaweera of the Division of Infectious Diseases, Department of Internal Medicine of the Miller School of Medicine, for accepting the appointment to the dissertation committee and for their helpful suggestions and support.

The fruitful discussions I had with Professor S. Ramakrishnan of the Department of Mathematics were invaluable in completing this work. I also wish to thank my colleagues K. Wickramarathna and K. K. R. G. K. Hewawasam for their unfailing support.

The support provided through NSF Grant No. IIS-0513702 is also acknowledged. Encouragement and support I received from my friends and colleagues is highly appreciated. My special thanks go to Aunty Benita for supporting me in many ways for my studies and Josephene and Wallace Hanchey for their kindest support.

I would like to thank all my teachers who have guided me throughout the years. Finally, I extend my utmost gratitude to my parents, my brother Kanishka, my sister Gayani, and last but not least, my wife Thanushi for their encouragement, patience and love, which made this work a reality.

THANUKA WICKRAMARATHNE

University of Miami

May 2008

Table of Contents

List of Figures	vi
List of Tables	vii
CHAPTER 1 Introduction	1
1.1 Collaborative Filtering and Related Work	4
1.2 Motivation and Goals	6
1.3 Organization of Presentation	10
CHAPTER 2 Preliminaries	11
2.1 Automated Collaborative Filtering	11
2.1.1 User Preferences and Ratings Matrix	12
2.1.2 Memory-Based ACF Systems	13
2.1.3 Model Based Systems	15
2.1.4 Hybrid Systems	16
2.2 Knowledge Discovery in the Presence of Imperfect Data	16
2.2.1 Bayesian Probability Theory	16
2.2.2 Possibility Theory	18
2.2.3 DS Theory	19
2.3 Contribution	22
2.3.1 Approach for Handling of Data Imperfections	23
2.3.2 Incorporation of Contextual Information	23

CHAPTER 3	CoFiDS Overview	27
3.1	User Preferences	28
3.2	User Similarity and Neighborhood	30
3.3	Predictions and Decision Making	32
CHAPTER 4	User Preference Modeling	34
4.1	Nomenclature	34
4.2	DS Modeling Functions	35
4.2.1	Rated Items	36
4.2.2	Unrated Items	39
4.3	User-BoE	43
4.3.1	An Important Result on User-BoEs	44
CHAPTER 5	Similarity and Neighborhoods	46
5.1	Distance Between User-BoEs	46
5.2	User-User Similarity	49
5.3	User Neighborhood	50
CHAPTER 6	Prediction and Decision Making	52
6.1	Prediction	52
6.2	Decision Making	53
CHAPTER 7	Evaluation Matrices	54
7.1	Nomenclature	54
7.2	Performance Measures for “Crisp” Databases	54
7.3	Performance Measures for “Soft” Databases	57
CHAPTER 8	Experiments and Results	58
8.1	Datasets	58
8.1.1	<i>MovieLens</i>	58

8.1.2	<i>DS–MovieLens</i>	59
8.1.3	<i>PR–MovieLens</i>	61
8.1.4	<i>LKLD–MovieLens</i>	62
8.2	Other ACF algorithms for Comparison	62
8.3	User Preference Modeling	63
8.3.1	Experiments on “Crisp” Dataset	63
8.3.2	Experiment on “Soft” Dataset	64
8.4	Experimental Technique	65
8.5	Comparison	65
8.5.1	Comparison on “Crisp Dataset”	65
8.5.2	Comparison on <i>DS–MovieLens</i>	66
8.6	Results	67
8.6.1	Experiments on <i>MovieLens</i>	67
8.6.2	Evaluations on <i>DS – MovieLens</i>	71
CHAPTER 9 Conclusion and Future Research		74
9.1	Conclusion	74
9.2	Future Research	77
Bibliography		79

List of Figures

8.1	Partial Probability Models used in generating <i>DS-MovieLens</i>	60
8.2	Variation of MAE of CoFiDS with dispersion factor σ for several combinations of $\{K, \tau\}$. Here, CoFiDS $\alpha = 0.9$	68
8.3	Variation of MAE of CoFiDS with neighborhood size K	69
8.4	Variation of MAE of CoFiDS with similarity threshold τ	69
8.5	Variation of DS-MAE of CoFiDS with neighborhood size K	70
8.6	Variation of DS-MAE of CoFiDS with similarity threshold τ	71
8.7	CoFiDS with $\{\alpha, \sigma\} = \{0.9, 2/3\}$: <i>DS-PE1</i> versus p	72
8.8	CoFiDS with $\{\alpha, \sigma\} = \{0.9, 2/3\}$: <i>DS-PE2</i> versus p	72
8.9	CoFiDS with $\{\alpha, \sigma\} = \{0.9, 2/3\}$: <i>DS-PE1</i> versus K when $p = 0.1$	73

List of Tables

2.1	A Simple Ratings Matrix	13
2.2	DS Theoretic Models of Various Types of Data Imperfections	21
3.1	A simple ratings matrix allowing more flexibility in user preferences. Ratings space is a “5-star” system, i.e., users are allowed to pick ratings from $\{1, 2, 3, 4, 5\}$	28
3.2	Examples of CoFiDS Predictions	33
8.1	<i>Observed_Rating = 2</i> of a ± 1 user. Generating <i>DS-MovieLens</i> , <i>PR-MovieLens</i> and <i>LKLD-MovieLens</i> from <i>MovieLens</i>	61
8.2	Performance Comparison with “Hard” Decisions	67
8.3	Performance Comparison with “Soft” Decisions	70
8.4	Performance Comparison on <i>DS-MovieLens</i>	71

CHAPTER 1

Introduction

Information systems that interact with real-world application scenarios must be capable of dealing with imperfect information when processing data and retrieving knowledge, especially when one requires more accurate and realistic results. However, as pointed out in [1], much of the work has been taken up with building elegant and idealized models that are never approached in reality. According to the same author, this has led to the study of imperfections being marginalized instead of accepting the reality of imperfect data in real-world application domains.

There have been many attempts at classifying these imperfections over the years. Although no consistent or coherent classification system is available to date, these classifications are useful in understanding the nature of the imperfections. *Uncertainty*, *imprecision*, *incompleteness*, *inconsistency*, and *vagueness* are some of the terms used by different authors to describe different types of imperfections. Uncertainty arises from the lack of information about a certain *proposition*. Imprecision arises by not being able to measure with a suitable precision, whereas incompleteness arises from the absence of a particular data value. Inconsistency captures the notion of possible existence of two conflicting propositions and vagueness quantifies the vague nature in human expression, e.g., “good boy,” “expensive car.”

There are different sources of imperfections in a typical information system [2]. These imperfections can be a result of unreliable information sources, such as faulty reading instruments, or data input forms filled-out incorrectly (intentionally or inadvertently), e.g., in medical data records; or system errors including input errors, transmission “noise,” delays in processing update transactions, system software/hardware defects, and corrupted data owing to failure or sabotage. Uncertainty is a type of an imperfection inherited in information gathering methods that require estimation or judgement by its own nature. Imperfections could also be a result of restrictions imposed by the information gathering model itself. For example, an integer rating scheme for a movie recommendation system, e.g., **Netflix** movie recommender system. In this type of a setup, uncertainty is trivially embedded in user preferences in cases where he/she is indecisive between two ratings.

Exponential growth in the amount of available information during the past two decades has resulted in technologies for the management of information. These, referred to as *Information Filtering Systems*, are essentially methods of purging information sources via relegation of irrelevant and redundant data using automated computer techniques. But, surprisingly, almost all of these systems are based on idealized models, ignoring the imperfections that are inherent in the information they parse. So, these systems, never attempting to model reality as it is, end up in generating results that may be far from being realistic and acceptable. These techniques proposed in early years of information growth are now facing challenges, due to extremely large amounts of historical data, very high rates of new information being added and contradicting evidence in the stored data, even without taking the inherent imperfections in data into account. This problem of locating only the relevant information in such harsh conditions has been compared to “*locating needles in a haystack that is growing exponentially*” in [3]. These difficulties associated with information

filtering is further exacerbated if one wishes to work with more elaborate and realistic models capable of dealing with the imperfections present in data.

Advanced and sensitive information filtering applications require technologies capable of handling imperfections in data and propagating them throughout the entire process. Thus, the end-user can actually estimate the reliability of the results that he/she gets from such an information filtering system. Indeed, highly sensitive and critical application scenarios (e.g., medical and health care scenarios, battlefield situation and threat assessment, etc.) typically call for decision support systems that assist the end-user make an informed decision, but not systems that replace the decision making task itself.

Recommender Systems are a specific type of information filtering system which recommend information items a user might be interested in, e.g., movies, books, music, news, restaurants, etc. These systems became an important research area in both academia and industry since the first publication on *Collaborative Filtering (CF)* in the early-1990s. Automated collaborative filtering (ACF) has now become perhaps the most successful recommender system [4]. As mentioned in [5], “... *the interest in this research area still remains high because it constitutes a problem-rich research area and because of the abundance of practical applications that help users to deal with information overload and provide personalized recommendations, content, and services to them.*”

Existing work on ACF has not successfully addressed the issue of handling data with imperfections. Indeed, these limitations in existing ACF algorithms have limited their applicability to simpler and non-sensitive domains. To quote [2], “... *uncertainty permeates our understanding of the real world. The purpose of information systems is to model the real world. Hence information systems must be able to deal with uncertainty.*” In this thesis, we present a novel ACF scheme that is applicable to data with

imperfections. This Dempster Shafer theory based Collaborative Filtering algorithm, which we refer to as CoFiDS, can more accurately represent the imperfections inherent in user preferences and propagate them throughout the entire decision making process via the use of Dempster-Shafer (DS) belief theoretic notions.

CoFiDS opens up the door for a new breed of ACF schemes capable of working with more general and richer user preference ratings, thus widening the applicability of ACF to more advanced and sensitive problem domains, such as medical expert support systems, homeland security and surveillance, etc. Since it provides “soft” predictions (as opposed to “crisp” or “hard” decisions), CoFiDS enables the end-user to make a decision with the full knowledge of the impact that data imperfections may have had on the predictions.

This chapter is organized as follows. First, we will briefly explain CF and related research work. Our motivation and goals towards the work presented in this thesis are detailed next. We conclude the chapter by presenting an outline of the rest of the chapters.

1.1 Collaborative Filtering and Related Work

CF is a method of making predictions on item preferences of a user by collecting similar preferences from many peers, based on the assumption that *those who agreed in the past tend to agree again in the future*. One can identify this as an automation of the “word of mouth” process [6] people use in day-to-day decision making tasks — a process in which the opinions of peers are used as an aid to making a decision.

CF systems were first introduced to tackle the issues where Content-Based Filtering (CBF) failed to perform well. CBF is essentially another information filtering technique that uses machine analysis of the contents of items being analyzed to make predictions. CF and CBF are both Recommender Systems; but they differ in the

following aspect: CF does not process the contents of items like CBF; instead, CF processes user's preferences of items for filtering.

Early CF systems required manual specification of predictive relationships or explicit routing of preferred items. This method of CF is sometimes referred to as *Active Collaborative Filtering* [7], e.g., the **Tapestry System** [4]. ACF automates all tasks with the exception of the collection process of historical user preference data.

ACF possesses some attractive features that have inevitably made these algorithms very popular both in academia and industry. Some of the key advantages of ACF, as pointed out in [8], are its ability to

- filter information of any type of content;
- filter based on complex and hard to represent preferences; and
- receive serendipitous recommendations.

With all these features and advantages, ACF has easily found many applications, especially in the area of e-commerce during past few years. These applications include, recommending books, audio CDs, movies, vacations, restaurants, etc. In fact, ACF plays a key role in some of the major e-commerce successes, such as amazon.com [9], Musicmatch.com, barnesandnoble.com, half.ebay.com, iLike.com, netflix.com, and TiVo.com.

Widespread adoption in such e-commerce applications [10], and the increased attention that researchers in both industry and academia are beginning to pay towards addressing the various issues of such applications [11, 12], have resulted in significant improvements in ACF technology.

Many issues related to ACF have been addressed by research work done during the past few years. Improving the accuracy of predictions provided by these algorithms has been one of the main areas addressed by many researchers [13, 14, 5]. Novel

approaches to ACF [15, 16, 17, 18], that improve upon different aspects of ACF, have been proposed. More advanced systems that combine ACF with other recommendation methods [19, 20, 21] have also been proposed to exploit the advantages of individual systems, especially combining the advantages of CBF with ACF. Work has also been done on understanding and explaining various ACF methods and their predictions [22].

With all these developments, ACF has been applied to a diverse set of domains that are not limited to the previously mentioned applications. Such applications include, recommendations of restaurants [23], ski mountaineering [24], dating [25], online recruitment support services [26], peer-to-peer file sharing [27], locating experts in an organization [28], etc.

It is clear that ACF is advancing towards an era where advanced and more general classes of user preferences need to be handled. Applications such as dating, restaurant recommendations, etc., indeed require ACF algorithms capable of handling such preferences. Researchers have already understood the importance of accurately modeling the user preferences [29] for extracting better performance from ACF systems.

1.2 Motivation and Goals

Current rate of adoption of ACF to different application domains has created a growing demand for more accurate and advanced ACF systems with improved understanding and interpretation capabilities on their predictions. This essentially calls for systems capable of handling more general classes of user preference models. We believe that it is of utmost importance to have in place mechanisms that appropriately model the imperfections inherited in user preferences and then propagate these throughout the entire filtering process. To quote [30], “... *one should take the uncertainty into account, trading the loss of elegance and simplicity for more accurate modeling.*”

To better illustrate the importance of ACF systems that are endowed with better mechanisms for modeling and propagating user preference imperfections, consider a medical expert who is involved in HIV treatment using highly active antiretroviral (HAART) therapy. HAART therapy has been shown to decrease the mortality and morbidity in HIV patients at all stages of infection [31]. In HAART therapy, patients are administered drug combinations, commonly referred to as *drug cocktails*; the choice of a particular drug cocktail is usually based upon the recommendation of the Department of Health and Human Services (DHHS) [32], experience of the physician [33], and results of clinical trials [34]. Unfortunately, the DHHS recommendations and clinical trials are both based on large well controlled clinical trials with anti HIV agents using a study population of HIV subjects which may not reflect the “real world scenario.” Hence, physicians at times rely on other factors that are also known to influence a patient’s response to drugs, and a physician may make use of previous successes/failures to aid in determining future drug administration decisions. This basically gives rise to the typical ACF scenario.

Unlike in traditional ACF algorithms, in this example scenario, one cannot expect the user ratings — in this case, drug response effectiveness — to be “hard” (or “crisp” or “perfect”). Indeed, the qualitative nature of the subjectivity that is inherent in a physician’s rating surely calls for an algorithm that can model data imperfections more effectively. This is especially the case when a rating is generated as a collective decision of a team of physicians. For instance, in this HAART therapy scenario, it is more likely that a particular drug cocktail is rated as, “*Good* with a 70% level of confidence,” or that the physician concludes that the drug cocktail is “definitely not *Poor* but more evidence is needed to discern further.” How can such ratings be adequately captured? The ACF algorithm should be capable of accommodating and propagating such data imperfections and terminate its decision-making process

by enabling the end-user — in this case, the physician — to make a decision with the full recognition and understanding of the data imperfections inherent in the user ratings.

Lack of techniques to appropriately model and accommodate the data imperfections in user preferences, one is compelled to make various “assumptions” and “interpolations” to avoid the difficulties associated with such data. But such a strategy can severely impair the integrity of the decision-making process and yield inferences that are not trustworthy [35]. It is indeed difficult to justify the use of such ACF systems in more advanced and critical applications. Lack of better techniques for modelling and accommodating data imperfections throughout the decision-making process is in fact a significant hindrance to the application of various methodologies developed in computer engineering/science domains.

Traditional ACF algorithms must also grapple with two common problems.

Data Sparsity The sparsity associated with the user preferences. i.e. only a very small subset of total user-item space is rated

Cold-Start This refers to the difficulties associated with making predictions when either new users or new items being introduced into the system.

Data sparsity directly manifests in ACF algorithms as,

- decreased prediction accuracy because of errors introduced by other users who are wrongly identified as neighbors; and
- inability to make predictions because of the unavailability of ratings from true neighbors.

Again, these difficulties associated with data sparsity and cold-start are further exacerbated by the presence of data imperfections.

Thus, having the motivation to tackle the above problems, we seek an advanced ACF system capable of modeling and working with a richer class of user preferences. Such an effective ACF methodology then needs to have mechanisms in place to address the following issues:

1. How do we represent or model user preferences?
2. How can this model be used for extracting useful knowledge and making reliable predictions that are robust against data imperfections?
3. How can the prediction accuracy be improved?
4. How can the common ACF problems of data sparsity and cold-start be addressed in this new setup?

The use of a Dempster-Shafer (DS) theoretic notions provides a very convenient framework for modeling several types of data imperfections that are commonly encountered: missing data, incomplete data, ambiguities generated from lack of evidence to discern among a set of hypotheses, and of course probabilistic uncertainties. Probabilistic approaches requiring one to make initial assumptions on the model, such as independence, identically-distributed assumptions, etc., can generate misleading predictions when the actual models are different. But, on the other hand DS models are significantly more robust to such modeling errors (see [36] and references therein). Such a methodology is exactly what is needed in the present context where one may not be able to justify the typical assumptions required for a probabilistic approach.

However, DS theoretic models draw substantial criticism because of their potentially prohibitive computational burden. So, it is essential that algorithms utilizing DS theoretic methods have in place effective mechanisms to mitigate this computational effort.

“Soft” or belief theoretic predictions that CoFiDS provides are new to both ACF community and researchers, and there are no accepted measures for evaluating such predictions. Thus we’ll have to come-up with mechanisms to overcome the difficulties in performance evaluations and comparisons to exiting algorithms.

1.3 Organization of Presentation

This thesis is organized as follows: Chapter 2 provides the preliminary information for the developments in later chapters followed by an overview of CoFiDS Collaborative Filtering engine in Chapter 3. User preference modeling are introduced in Chapter 4 with data models and machinery required for later computations. Similarity Computation and Neighborhood Selection stages are discussed in Chapter 5. CoFiDS predictions and notions associated with decision making with DS-theoretic predictions are detailed in Chapter 6, followed by a discussion on evaluation measures in Chapter 7. Experiments done on CoFiDS and two other existing algorithms and their results are presented in Chapter 8. We conclude this presentation with some remarks on future research in Chapter 9.

CHAPTER 2

Preliminaries

In this chapter, after first introducing the basic notions of ACF, we discuss different approaches available for knowledge discovery from imperfect data. Three main classes of ACF namely, *memory-based*, *model-based* and *hybrid* ACF systems are briefly explained. The main tools used for knowledge discovery from imperfect data include probability theory, possibility theory and DS theory. Each of these methods contributes a distinct methodology for addressing problems in its domain to provide a more accurate and low cost solution as compared to traditional solutions that could handle only clean data. We conclude the chapter with our contributions.

2.1 Automated Collaborative Filtering

ACF is essentially a strategy for predicting the rating that a user might allocate to an unrated item based on aggregated ratings of “similar” users in a historical database. ACF systems can be classified into three broad categories:

- 1. Memory-Based Systems:** Entire database of user preferences is used for predictions.
- 2. Model-Based Systems:** Preferences are first used to learn a “prediction model” which is then used for predictions.

3. Hybrid Systems: Combination of two or more recommender techniques, usually ACF combined with CBF

ACF possesses some unique challenges which are inherited from the nature of the filtering problem:

Data Sparsity: Problem of the *ratings matrix* being very sparse, i.e., only a very small number of entries are filled.

Cold-Start: Inability of the system to provide recommendations when a new user or item is introduced.

Gray-Sheep: Existence of users who are consistently different from others.

Reduced Coverage: Inability of the system to provide recommendations due to the number of ratings being very small compared to the large number of available items.

Scalability: Prohibitive computational requirements when applying ACF to huge datasets.

Synonymy: Tendency of same or very similar items to occupying different entries in the database.

2.1.1 User Preferences and Ratings Matrix

A user's preference on a given item is usually referred to as either *user preference* or *rating* of that item. These ratings are collected and stored in a database. This historical database of user preferences is usually viewed as a matrix which is referred to as the *ratings matrix*, where the rows and columns represent users and items respectively. Table 2.1.1 shows an example of a simple ratings matrix. Here, the

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1			4			3			1
u_2		2		4						5
u_3	4		3		2				3	
u_4		2		1		3		1		
u_5			3				1	2		
u_6	5			1	5		5		3	4

Table 2.1: A Simple Ratings Matrix

ratings space — the set from which user ratings are drawn — is a 5-star ratings system. The blank entries represent the user-item pairs that have not been rated.

An item that is rated by multiple users is referred to as being *co-rated* for those users, e.g., i_1 is co-rated by users u_1 and u_3 . We refer to a user who has rated multiple items as a *co-rated user* for those items, e.g., u_1 is a co-rated user for items i_4 and i_7 .

2.1.2 Memory-Based ACF Systems

Entire or sample of the ratings database is used by memory-based ACF algorithms for making predictions. These algorithms first locate a small subset of users referred to as the *neighborhood* via some distance metric, which estimates the differences in their ratings [6]. Users in the neighborhood are referred to as *neighbors*. For a given user, predictions are then obtained by taking a weighted average of the ratings of his/her neighbors on a given item.

Similarity Computation

Core of the memory-based ACF systems lies in locating a “good set of neighbors” who have similar preferences to the *active user* — the user to whom the current predictions are done. Thus, similarity computation is a very critical step in this method of CF. Two broad variants of memory-based ACF systems can be identified based on the similarity computation method:

User-Based: Similarity between users are calculated using their co-rated items

Item-Based: Similarity between items are calculated based on their co-rated users [37].

New algorithms have been proposed using a combination of both methods [16]. There are many different methods used by ACF community for similarity computation, each technique having its own strengths and weaknesses.

Correlation based methods: Similarity is computed based on the correlation coefficient of the co-rated entities (items or users), e.g., *Pearson correlation* and *Spearman rank correlation*.

Cosine based methods: Here, co-rated entities are viewed as two vectors and the cosine angle between these two vectors are used as a measure of similarity, e.g., *vector cosine* and *adjusted cosine*.

Conditional probability based methods: This method is not widely used, but is used to estimate the similarity of two items based on a conditional probability based matrix, e.g., similarity of items i and j via $P(i|j)/P(j)$

Neighborhood Selection

Neighborhood selection becomes a trivial task once the similarities are computed. Active user's neighborhood is usually selected as the K -nearest neighbors — K or less most similar users — to that user. The value of K is often left as an optimization parameter. A small caveat of selecting K -nearest neighbors (KNN) is the possibility of selecting a set of users having very low similarity to the active user. This problem can be overcome by setting a minimum similarity threshold for neighbors first, and then selecting K or less nearest neighbors [38]. Not all algorithms use a subset of

users (or items if item-based) for predictions. Exceptions include the ACF algorithms proposed by Nakamura, et al. [16].

Predictions

This is the most crucial step in all types of ACF algorithms. Predictions are usually given as either

- a weighted sum of neighbors ratings in user-based methods, or
- a simple weighted average in item-based methods.

Recommendations

This is often referred to as *top-N recommendations*. Top- N recommendation [5] problem is to identify a set of N items that are of best interest to a given user, e.g., in `amazon.com` recommendations. Once the ACF algorithm predicts the ratings for items that are not rated by the active user, top N items with highest ratings are selected as top- N recommendations. Top- N recommendation methodologies are different for item-based and user-based ACF variants. Item-based algorithms are shown to be more robust against scalability issues when compared to their user-based counterparts.

2.1.3 Model Based Systems

These algorithms work by learning models for user ratings. In this method, prediction for a given item of the active user is estimated by his/her own ratings of the other items. Models for user are learned using different machine learning techniques, giving rise to different types of model based ACF systems. A few popular techniques used for learning models are Bayesian networks, clustering techniques, neural network classifiers, association rules, dependency networks, etc.

2.1.4 Hybrid Systems

Hybrid systems are essentially recommender systems combining two or more algorithms. Often ACF is combined with CBF. Hybrid systems can be carefully designed to exploit the strengths and minimize the weaknesses of individual algorithms. Indeed, it has been shown empirically that hybrid systems outperform classical ACF algorithms [20, 21].

2.2 Knowledge Discovery in the Presence of Imperfect Data

Three basic frameworks are popularly used when the data available are imperfect.

2.2.1 Bayesian Probability Theory

Methods based on classical probability theory are well established and have earned wide popularity [39]. In this theory, one of many mutually exclusive hypotheses (possible events) are tested against the gathered evidence. A Bayesian probabilistic model consists of the triple (Ω, \mathcal{F}, P) where Ω is the sample space, \mathcal{F} is a σ -algebra of subsets of Ω and P is a non-negative mapping of \mathcal{F} into the interval $[0, 1]$. It possesses the following properties:

Axiom 1. $\Omega \in \mathcal{F}$ with $P(\Omega) = 1$.

Axiom 2. If $A \in \mathcal{F}$, then $\bar{A} \in \mathcal{F}$, where \bar{A} denotes the complement of A .

Axiom 3. For pairwise disjoint $\{A_n\}$, $n \geq 1$,

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n). \quad (2.1)$$

Each proposition $A \in \mathcal{F}$ is associated with the probability $P(A)$. Note that,

$$P(A) + P(\bar{A}) = 1, \forall A \in \mathcal{F}. \quad (2.2)$$

Bayes' theorem is employed to compute the probability of a hypothesis, given some observation event. Consider a collection of hypotheses $\{H_i\}_{i=1,\overline{n}}$, $H_i \in \mathcal{F}$ and suppose $P(E) > 0$ for an observation event E . Conditional probability of the occurrence of H_i provided the evidence E is given by

$$P(H_i|E) = \frac{P(E|H_i) P(H_i)}{P(E)}, \quad (2.3)$$

and

$$P(E) = \sum_{i=1}^N P(E|H_i) P(H_i). \quad (2.4)$$

The quantities $P(H_i)$ and $P(E|H)$ are termed *a-priori* probabilities since they represent statements that can be made prior to knowing the true subject of any observation. $P(E|H)$ is also referred as the *likelihood* of E given H . The conditional probabilities are calculated using above quantities along with the probability of observation event, the *evidence* $P(E)$. These conditional probabilities are then combined using the generalized Bayesian inference formula [40] based on the assumption that event E is conditionally independent w.r.t. H_i :

$$P(H_i|E_1 \cap \dots \cap E_k) = \frac{P(H_i) P(E_1|H_i) \dots P(E_k|H_i)}{\sum_{n=1}^N P(H_n) [P(E_1|H_n) \dots P(E_k|H_n)]} \quad (2.5)$$

to obtain a-posteriori probabilities with respect to the totality of events E_1, \dots, E_k . A suitable decision logic is utilized to arrive at a decision based on these final probabilities. *Maximum a posteriori (MAP)* and *maximum likelihood (ML)* methods are widely used as Bayesian decision rules [41, 42, 43].

In a Bayesian model, the knowledge one has about a proposition A determines explicitly the knowledge one has regarding its complement. We cannot refrain from assigning probability numbers to events in \bar{A} that we are not certain of. Therefore,

it is incapable of representing the ignorance we may have regarding the events in \bar{A} . Moreover, the difficulties in defining the priors $P(H_i)$ when sufficient information is not available, and the requirement that the competing hypotheses must be mutually exclusive, are other disadvantages of Bayesian probability theory [40].

2.2.2 Possibility Theory

Possibility measures introduced by Zadah [44] are closely associated with fuzzy sets and measures [45]. It considers a body of knowledge represented as subsets of a reference set \mathcal{S} . When Ω denotes the power set, i.e., $\Omega \equiv 2^{\mathcal{S}}$, confidence functions that map the elements of Ω into the interval $[0,1]$ is defined as $C : \Omega \mapsto [0, 1]$. These confidence functions are monotonic in the sense that $A \subseteq B \implies C(A) \leq C(B)$. This may be interpreted as follows: if an event A implies a second event B , then there is at least as much confidence in the occurrence of event B as in the occurrence of the event A . Consequences of this monotonicity are that

$$C(A \cup B) \geq \max[C(A), C(B)] \text{ and } C(A \cap B) \leq \min[C(A), C(B)]. \quad (2.6)$$

The limiting case $C(A \cup B) = \max[C(A), C(B)]$ defines what are referred to as *possibility measures* [46]. Suppose $E \in \Omega$ is such that $C(E) = 1$. Possibility measure Π is defined as $\Pi(A) = 1$ if $A \cap E \neq \emptyset$, and 0 otherwise. We interpret $\Pi(A) = 1$ as A is possible. Also note that $\Pi(A \cup \bar{A}) = \Pi(\mathcal{S}) = 1$ and $\max[\Pi(A), \Pi(\bar{A})] = 1$. Using this, A and \bar{A} can be interpreted as two contradictory events, i.e., at least one is possible. However, one being possible does not prevent the other being possible too. The notion of $C(A \cup B) = \max[C(A), C(B)]$ seems to be consistent with possibility in the real world, i.e., occurrence of $A \cup B$ requires only the easiest event (most possible event) of the two to happen.

Similarly, using the other limiting case $C(A \cap B) = \min[C(A), C(B)]$, necessity measure N is defined to interpret that if an event is necessary, its contrary is im-

possible. Conversely, if an event is possible, its contrary is absolutely not necessary. Uncertainty of events can be characterized by these possibility and necessity functions thus weakening axiom 2.2 of the Bayesian framework.

This framework can be used effectively to represent imperfections associated with data. Fuzzy membership functions defining various relationships among fuzzy sets serve as possibility distribution functions. Approaches based on fuzzy reasoning can represent the vagueness of information. Probability theory only allows us to represent the chance of extremes (occurrence or non-occurrence) of an event while possibility theory could extend our view over “to what extent would the event be possible ?” and “to what extent would the event be necessary ?.” In situations where such vagueness of information needs to be represented, this formalism offers advantages over probability theory [47, 48].

The disadvantages of this framework include increased number of computations compared to other methods. Also there is a potential difficulty in generating suitable membership functions corresponding to the fuzzy sets.

2.2.3 DS Theory

Let $\Theta = \{\theta_1, \dots, \theta_L\}$ be a finite set of mutually exclusive and exhaustive propositions about some problem domain. It signifies the corresponding “scope of expertise” and is referred to as its *frame of discernment (FoD)* [49]. A proposition θ_i , referred to as a *singleton*, represents the lowest level of discernible information in this FoD. Elements in 2^Θ , the power set of Θ , form all propositions of interest. A proposition that is not a singleton is referred to as a *composite*, e.g., (θ_1, θ_2) . Henceforth, the term “proposition” is used to denote both singletons and composites.

Cardinality of set A is denoted by $|A|$. The set $A \setminus B$ denotes all singletons in $A \subseteq \Theta$ that are not included in $B \subseteq \Theta$, viz., $A \setminus B = \{\theta_i \in \Theta : \theta_i \in A, \theta_i \notin B\}$; \bar{A} denotes $\Theta \setminus A$.

Basic Notions

Definition 1 (Basic Probability Assignment (BPA)) *The mapping $m : 2^\Theta \mapsto [0, 1]$ is a basic probability assignment (BPA) or mass structure for the FoD Θ if (i) $m(\emptyset) = 0$; and (ii) $\sum_{A \subseteq \Theta} m(A) = 1$. ■*

The mass of a proposition is free to move into its individual singletons. This is how DS theory allows one to model the notion of *ignorance*. For example, complete lack of evidence can be conveniently captured via the *vacuous BPA*: $m(A) = 0, \forall A \subset \Theta$ and $m(\Theta) = 1.0$. A proposition that possesses a nonzero mass is referred to as a *focal element*; the set of focal elements is the *core* and is denoted by \mathcal{F} . The triple $\{\Theta, \mathcal{F}, m\}$ is referred to as the *body of evidence (BoE)*; the number of focal elements in this BoE is $|\mathcal{F}|$.

Definition 2 (Belief, Plausibility) *Given a BoE $\{\Theta, \mathcal{F}, m\}$ and $A \subseteq \Theta$, (i) $Bl : 2^\Theta \mapsto [0, 1]$ where $Bl(A) = \sum_{B \subseteq A} m(B)$ is the belief of A ; and (ii) $Pl : 2^\Theta \mapsto [0, 1]$ where $Pl(A) = 1 - Bl(\bar{A})$ is the plausibility of A . ■*

So, while $m(A)$ measures the support assigned to proposition A *only*, the belief assigned to A takes into account the supports for all proper subsets of A as well; $Bl(A)$ represents the total support that can move into A without any ambiguity. $Pl(A)$ represents the extent to which one finds A plausible. When the core contains only singletons, the BPA, belief and plausibility all reduce to probability.

These DS theoretic notions allow one to represent a wide variety of data imperfections with ease [50, 35]. See Table 2.2.3.

A probability distribution $\Pr(\bullet)$ such that $Bl(A) \leq \Pr(A) \leq Pl(A), \forall A \subseteq \Theta$, is said to be *compatible* with the underlying BPA $m(\bullet)$. An example of such a probability distribution is the *pignistic probability distribution* $Bp(\bullet)$ [51]

$$Bp(\theta_i) = \sum_{\theta_i \in A \subseteq \Theta} m(A)/|A|. \quad (2.7)$$

Type of Imperfection	\mathbf{r}_{ik}		Remarks
	Proposition	m_{ik}	
Hard (Perfect, Crisp)	θ_2	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 BoE	Θ	1.0	Missing or unknown entry
Belief theoretic	$\sum_{A \subseteq \Theta} m_{ik}(A) = 1.0$		Most general

Table 2.2: DS Theoretic Models of Various Types of Data Imperfections

Evidence Combination

The evidence of two “independent” BoEs could be “pooled” to form a single BoE via

Definition 3 (Dempster’s Rule of Combination (DRC)) *Suppose the two BoEs $\{\Theta, \mathcal{F}_i, m_i\}$, $i = \overline{1, 2}$, span the same FoD Θ . Then, if*

$$K \equiv 1 - \sum_{\substack{C \in \mathcal{F}_1, D \in \mathcal{F}_2 \\ C \cap D = \emptyset}} m_1(C) m_2(D) \neq 0,$$

the DRC generates the BPA $m(\bullet) : 2^\Theta \mapsto [0, 1]$ where

$$m(A) = \sum_{\substack{C \in \mathcal{F}_1, D \in \mathcal{F}_2 \\ C \cap D = A}} m_1(C) m_2(D) \div K, \forall A \subseteq \Theta.$$

This combination operation is denoted as $m = m_1 \oplus m_2$. ■

The operation \oplus is both associative and commutative thus enabling the combination of multiple BoEs with ease [49]. A variation of the DRC that accounts for

evidence reliability is

$$m(A) = (m_1^{(disc)} \oplus m_2^{(disc)})(A),$$

$$\text{where } m_i^{(disc)}(A) = \begin{cases} d_i m_i(A), & \text{for } A \subset \Theta; \\ (1 - d_i) + d_i m_i(\Theta), & \text{for } A = \Theta. \end{cases} \quad (2.8)$$

Here, $d_i \in [0, 1]$ is referred to as a *discounting factor* [49].

2.3 Contribution

In the previous chapter, we pointed out the importance of advanced ACF schemes that are capable of modeling and working with a richer class of user preferences. There, we raised several questions that should be addressed by such a methodology. To rephrase,

1. How can we model a wider class of user preferences?
2. How can this model be used for extracting useful knowledge and making reliable predictions that are robust against data imperfections?
3. How can the prediction accuracy be improved?
4. How can the common ACF problems of data sparsity and cold-start be addressed in this new setup?

Our contribution is a new breed of ACF algorithms that effectively address the above questions via the incorporation of several novel and unique concepts. First, we pioneer in introducing ACF algorithms applicable to domains with data imperfections. We also introduce an ACF methodology that is capable of incorporating domain expertise and background data to aid the prediction task in unified manner.

2.3.1 Approach for Handling of Data Imperfections

We utilize a data model that allows one to accommodate imperfections and then propagate these throughout the decision-making process via the integration of DS belief theoretic notions into ACF. As we have already mentioned, due to this DS theoretic framework, we refer to our automated Collaborative Filtering algorithm as CoFiDS.

We prefer to use DS theory because of its ability to conveniently represent a wide variety of data imperfections. Such imperfections include, probabilistic uncertainties, qualitative aspects of evidence, evidence ambiguities, missing information, etc. Apart from the convenience it offers, DS models are better able to capture partial or incomplete knowledge. Table 2.2.3 shows different types of imperfections that can be captured by DS theoretic models.

Our decision to use DS theory, is justified by the wide variety of applications of DS theoretic methods to domains possessing data imperfections [35, 52, 53]. In these applications, integrity of the decision making process and robustness against modeling errors are critical issues. Here, the modeling errors could be result of lack of information on the underlying probability distributions. Example applications include battlefield target tracking, situation awareness, etc.

2.3.2 Incorporation of Contextual Information

As we have already mentioned, only a subset of user population are used in general for prediction purposes in memory-based ACF Systems [6]. Thus, identifying truly similar users is of paramount importance towards more accurate predictions.

However, the challenge one encounters is the sparsity of the ratings matrix. In practice these matrices are extremely sparse. e.g. Data Sparsity is about 93.7% in the *MovieLens* dataset [54] — a widely used, benchmark CF dataset. For instance,

consider the ratings matrix associated with the HAART therapy scenario we introduced in Chapter 1. It would definitely be extremely sparse because, the different number of drug cocktails that may have been prescribed to each patient would be significantly small compared to the number of available drug cocktails. The number of drug cocktails that have actually been “rated” — the effectiveness of drugs evaluated — could very well be even a smaller fraction.

In such a situation, the number of co-rated items that are rated by any given two users would be extremely small. Thus any identified similarity among users that is based upon co-rated items may not be a true reflection of how similar the group members are. For an illustration, take the example ratings matrix in Table 2.1.1. Let user u_5 be the *active user*. Co-rated users would be users $\{u_1, u_3, u_4, u_6\}$. But, all these users have only co-rated a single item out of 10 items in total. In such a situation, one can not justify the accuracy of predictions made by an ACF system — utilizing similarities based on co-rated entities (items or users) only.

The inferences made on such similarity computations would be less reliable and prediction accuracy would be low. But, somehow it could turn out that the prediction accuracy is reasonably better in a practical dataset, as it is indeed the case in most practical ACF systems in use today. However, in these applications, e.g. a Movie Recommender — many users tend to rate a common subset of items. Thus, the users are indeed compared on a common set and their predictions are again made onto the same set of items as well. However, this problem is not captured by the algorithm evaluation methods and measures in existence. For instance, take the N-fold cross validation method used for algorithm testing. Here, two independent subsets of user-item pairs are chosen for training and testing. Unfortunately, the testing and training sets being subsets of the “commonly rated set” — as we have pointed out earlier, all computations and predictions are done onto the common subset. This, actually

hides the effects of wrong neighbors and usually provide a good prediction accuracy in evaluations. But, one can not assure reliability of such predictions. These can manifest as misleading recommendations made by such systems. Such experiences can actually be seen by the opinions of frustrated users who use commercial recommender systems for their decision making tasks, such as “*movie selections.*”

Since our motive is to arrive at an ACF system applicable to advanced and critical domains, reliability of predictions are of paramount importance. Thus, we focus on methodologies enabling us to select a true neighbor set even at a cost of increased computational requirements.

It has been shown that one may generate more accurate predictions by making use of metadata of items and users and/or domain expertise (or background knowledge) [55, 56, 57]. Such methods usually use metadata and domain expertise to generate separate predictions, which are then fused with predictions generated by ACF algorithm. Although one may obtain higher prediction accuracy using such strategies, the problems caused by data sparsity still remains.

DS theoretic framework that forms the basis of CoFiDS allow it to address data sparsity in a unique and elegant manner. To begin with, one may simply replace each unrated entry of the ratings matrix by a *vacuous mass structure*, the DS theoretic model of a missing value. CoFiDS however go substantially further by exploiting contextual information to “narrow” the uncertainty that a vacuous mass structure may otherwise harbor. In essence, CoFiDS enables one to “fill in” or populate each unrated item in the ratings matrix *prior to carrying out the ACF task*. This strategy makes the computed similarity among users more reliable. For example, in the HAART therapy scenario, CoFiDS allows one to integrate physician expertise on domain, user and drug metadata (e.g., patient age, drug compliance, co-morbidity, etc.) into the ACF algorithm from the very outset thus resulting in improved predictions.

This strategy of exploiting contextual information prior to the ACF task also addresses the cold-start problem in an elegant manner. In the HAART therapy scenario, drug response entries corresponding to a new patient can now be populated with the DS theoretic models of the relevant contextual information; the same applies to a new drug being introduced.

CHAPTER 3

CoFiDS Overview

The ACF algorithm CoFiDS presented in this thesis can be seen as a generalization of the memory-based ACF algorithms via the incorporation of DS theoretic notions. We believe that CoFiDS presents a novel and pioneering ACF algorithm due to the following reasons:

- It constitutes a new breed of ACF systems capable of handling more general classes of user preferences.
- It is a coherent and consistent method for exploiting contextual information for improved predictions.
- It provides soft predictions enabling the end-user to make a more reliable decision with the full knowledge of the underlying data imperfections.

But, in spirit, CoFiDS is similar to classical memory-based ACF algorithms. The generalized user ratings matrix that we refer to as the *DS-ratings matrix* represents more general classes of user preferences via DS theoretic BoEs spanning the ratings space. The similarity matrix is then generated on a user-user basis. CoFiDS predictions are made by fusing user ratings from a selected neighborhood which, as usual, is a subset of user space. As far as its predictions are concerned, CoFiDS differs

significantly from its traditional counterparts. Indeed, CoFiDS gives its predictions as DS theoretic BoEs.

3.1 User Preferences

In CoFiDS, each user rating is represented as a BoE spanning the ratings space, allowing it to represent a wide range of user preferences (e.g., uncertain, partial, and ambiguous data).

DS theoretic representations of the ratings in domains where user preferences are hard, can easily be obtained by simply assigning the total mass to the given proposition. For instance, consider the user ratings matrix in Table 3.1. User u_1 's rating "2 for sure" on item i_1 can be modeled as $m(\{2\})=1$. An unrated item can be represented via the vacuous BPA $m(\{1, 2, 3, 4, 5\}) = 1.0$. This type of "N-star" rating systems are widely used in existing recommender systems.

	i_1	i_2	i_3	i_4	i_5
u_1	2 for sure			4 for 75%	
u_2		2 or below		4	
u_3	3 or 4		less than 3		not 1 or 2
u_4		2		2 if not 1	
u_5			3		
u_6	4 or above			1	5 for 95%

Table 3.1: A simple ratings matrix allowing more flexibility in user preferences. Ratings space is a "5-star" system, i.e., users are allowed to pick ratings from $\{1, 2, 3, 4, 5\}$.

However, these recommenders do not allow the flexibility that is required for the other types of user preferences in Table 3.1. In domains where user preferences are forced to be hard, which is the case in almost all the recommender systems up to date, a user may have difficulty in picking, or may be unwilling to pick, a single label as his/her proper rating. Thus, the expressed rating is imperfect to a certain degree. Such simple data imperfections can be very conveniently modeled using DS theoretic notions.

For example, user u_6 's rating on item i_5 in Table 3.1 can be captured via the BPA $\{m(5), m(1, 2, 3, 4, 5)\} = \{0.95, 0.05\}$. This capability allows CoFiDS to perform its processing on user's real preferences, taking into consideration the inherent data imperfections, thus leading to more accurate predictions.

DS theoretic models allow CoFiDS to represent user preferences in sensitive and critical application domains. This idea is best explained with an example. Consider the HAART therapy scenario we introduced in Chapter 1. The physician may generate and utilize a ratings matrix with each row and column corresponding to a patient and a drug cocktail, respectively. Each ratings matrix entry would then indicate the effectiveness of the drug cocktail when administered to the corresponding patient. For example, the physician may use elements from the set $\Theta_{\text{pref}} = \{Excellent, Good, Fair, Poor\}$ for rating a drug response. Unlike in traditional ACF algorithms, in this example scenario, one cannot expect the user ratings — in this case, drug response effectiveness — to be “hard” (or “crisp” or “perfect”). This is especially the case when a rating is generated as a collective decision of a team of physicians. For instance, it is more likely that a particular drug cocktail is rated as, “*Good* with a 70% level of confidence,” or that the physician concludes that the drug cocktail is “definitely not *Poor* but more evidence is needed to discern further.” The BPAs $\{m(Good), m(\Theta_{\text{pref}})\} = \{7/10, 3/10\}$ and $m(Excellent, Good\ Fair) = 1.0$ would elegantly capture those two ratings respectively.

CoFiDS's capability to handle this type of user preferences without making any assumptions or interpolations, allows the deployment of recommender systems with more user friendly rating schemes. Allowing simple but flexible preference specifications such as those shown in Table 3.1 would be convenient for users and indeed will result in more accurate predictions, and hence recommendations.

CoFiDS also introduces a unique methodology for estimating the user preferences for unrated items using contextual information from the very outset, prior to any ACF task. This enables one to exploit domain expertise or background information or user and/or item metadata to reduce the ambiguity in the preferences of unrated items. For instance, consider item i_4 in the Ratings Matrix shown in Table 3.1. Assume that the domain expert is certain that “no user will rate the item i_4 below ‘3-stars’ !” In such a situation, he/she may use this information to replace the ratings for all users who have not rated that item, i.e., ratings of users u_3 and u_5 in this example. This reduces the ambiguity in all those ratings to that of BPA $m(\{3, 4, 5\}) = 1.0$. When such information is not available on the user preferences, in Chapter 4, we propose a technique that can make use of the available ratings.

3.2 User Similarity and Neighborhood

Similarity matrix is generated on a user-user similarity basis. For any two given users, user-user similarity is usually computed using only the co-rated items — items that are rated by both the users in common. The ratings matrix being very sparse, the number of co-rated items used for similarity evaluation of two given users would be extremely small. Thus, any such identified similarity among users that is based upon co-rated items may not be a true reflection of their actual similarity. With our intention of applying CoFiDS to more sensitive and critical application domains, CoFiDS approach for similarity computation differs from its peer memory-based algorithms.

As we have already mentioned, CoFiDS uses contextual information from the very outset to reduce the ambiguity of the user preferences for items that are originally not rated by users. This essentially “fills in” the DS-ratings Matrix. It then extends user preference ratings of items — which can be viewed as “*intra-item*” preferences into “*inter-item*” preferences representing the user’s preference on all items as a whole.

CoFiDS then compares users based on inter-item preferences which we refer to as *User-BoEs*. With this setup, user-user similarities computed will be more reliable than those based on co-rated items only, but at a cost of increased computational requirements.

CoFiDS uses K nearest neighbors (KNN), based on their user-user similarity for predictions. K is kept as an optimization parameter as usual. For a given active user, once the item to be predicted is specified, the traditional method of selecting a neighborhood is to select the K most similar users, who have actually rated the given item. The obvious weaknesses in this method of neighborhood selection manifests as two problems exacerbated by the sparseness of the ratings matrix.

1. Possibility of neighborhood set being empty due to unavailability of rated users for the selected item. If so, ACF algorithm will then fail to make a prediction.
2. Possibility of the users who have actually rated the given item being dissimilar to active user, i.e. similarity is significantly lower.

CoFiDS handles this challenge as follows. First, it sets a lower bound or a threshold on similarity for neighborhood selection. This avoids the possibility of capturing dissimilar users, but increases the chances of neighborhood set being empty. In such a situation, since the DS-ratings matrix is already “filled out” using contextual information, CoFiDS selects a second-tier K -neighborhood of users discarding the constraint on “*item-ratedness*”, i.e., the users in neighborhood are not required to have rated the item. But, indeed they should satisfy the lower bound on their similarity to the active user. This strategy along with contextual information incorporation technique elegantly tackles the problem of cold-start as well. Similarity computations and neighborhood selection procedures are elaborated in Chapter 5.

3.3 Predictions and Decision Making

CoFiDS prediction stage closely resembles that of traditional memory-based ACFs. Once the neighborhood for a given prediction request is chosen, CoFiDS generates the prediction by fusing the item preferences of those neighbors. Since the DS-ratings matrix has already been “filled up”, entry corresponding to the requested item of active user would probably be non-vacuous — not completely ambiguous. Thus, CoFiDS indeed fuses this information with the prediction obtained using neighbors for improved prediction accuracy.

CoFiDS predictions are essentially “soft” information. This enables the end-user to make decisions with full recognition and understanding of the reliability measures of the generated predictions. If one wishes to obtain a hard decision, “*pignistic transformation*” or “*maxBL with non-overlapping intervals*” strategies can be used. But, these methods degrade the information content of a rating that is expressed as a BoE. Table 3.3 shows some example predictions generated by CoFiDS (from experiments on *MovieLens*) along with the single label predictions that one obtains with the pignistic transformation and the maxBL strategy.

Note the following facts on hard decisions arrived at via the pignistic transformation and the maxBL strategy. These examples provide evidence on the importance of “soft predictions” for better understanding of the predictions and eventual decision making process.

- Decisions arrived at for user-item pairs (72, 550) and (19, 211) are uncontroversial.
- For user-item pair (116, 758), while the maxBL strategy captures the indecision that is apparent in the CoFiDS prediction, the pignistic transformation does not.

User-Item Pair	True Rating	CoFiDS Prediction		Single Label Decision	
		Proposition	Mass	Pignistic	MaxBL
(72, 550)	4	3	0.129	4	4
		4	0.870		
		(3, 4)	0.001		
(19, 211)	4	3	0.047	4	4
		4	0.953		
(2, 251)	5	4	0.505	4	4
		5	0.490		
		(4, 5)	0.005		
(116, 758)	1	1	0.305	2	(1, 2)
		2	0.436		
		(1, 2)	0.224		
		(3, 4)	0.035		

Table 3.2: Examples of CoFiDS Predictions

- The decision for user-item pair (2, 251) illustrates the difficulty one would have in attempting to capture the richer information content in a DS theoretic BoE with a single label decision. Although both pignistic transformation and the maxBL strategy favor a “4” rating, the CoFiDS prediction is clearly not very decisive between the “4” and “5” (true) ratings.

Chapter 6 elaborates the technical details of CoFiDS prediction making process.

CHAPTER 4

User Preference Modeling

We elaborated upon the strengths of CoFiDS in modeling different types of user preferences in Section 3.1 of Chapter 3. In this chapter, we introduce DS theoretic user preference modeling techniques and elaborate upon one such simple model that can be used to represent a wide variety of data imperfections where the actual preferences are forced to be hard by the system. Then, we develop the user-BoE by extending user's preferences on individual items to the user preference on all items as a whole. First, we introduce the notation that will be used hereafter.

4.1 Nomenclature

Let $U = \{U_1, U_2, \dots, U_M\}$ and $I = \{I_1, I_2, \dots, I_N\}$ denote exhaustive sets of M users and N items, respectively. We assume that a user allocates his rating to an item via an element θ_l from the finite, rank-ordered set of L labels denoted by $\Theta_{\text{pref}} = \{\theta_1, \theta_2, \dots, \theta_L\}$, where $\theta_j < \theta_\ell$ whenever $j < \ell$. A user rating is identified as a mapping $f_R : \mathbf{U} \times \mathbf{I} \mapsto \Theta_{\text{pref}} : U_i \times I_k \mapsto r_{ik}$, where $r_{ik} \in \Theta_{\text{pref}}$ denotes the rating that user U_i allocates to item I_k ; if I_k has not been rated by U_i , we use $r_{ik} = \emptyset$. Ratings matrix is then given by the $M \times N$ matrix created as $R = \{r_{ik}\}$. Let $\mathbf{r}_{ik} = \{\Theta_{\text{pref}}, \mathcal{F}_{ik}, m_{ik}\}$ denote the dual of the rating r_{ik} in DS theoretic domain. Let the $M \times N$ matrix \mathbf{R}

denote the DS ratings matrix, which is then given by $\mathbf{R} = \{\mathbf{r}_{ik}\}$. For $i = \overline{1, M}$ and $k = \overline{1, N}$, we also introduce

$$R_i^{(user)} = \{I_j : r_{ij} \neq \emptyset\}; \quad \overline{R_i^{(user)}} = \mathbf{I} \setminus R_i^{(user)} \quad (4.1)$$

$$R_k^{(item)} = \{U_i : r_{ik} \neq \emptyset\}; \quad \overline{R_k^{(item)}} = \mathbf{U} \setminus R_k^{(item)} \quad (4.2)$$

Thus, $R_i^{(user)}$ and $R_k^{(item)}$ denote the items rated by user U_i , and users who have rated item I_k , with $\overline{R_i^{(user)}}$ and $\overline{R_k^{(item)}}$ denoting their complements respectively.

4.2 DS Modeling Functions

As we have already mentioned in Chapter 3, CoFiDS view each user rating as a BoE spanning the FoD $\Theta_{\text{pref}} = \{\theta_1, \dots, \theta_L\}$, where the singletons of Θ_{pref} (i.e., θ_i , $i = \overline{1, L}$) represent the smallest discernible levels in user ratings. The *DS modeling function* defined next, maps the given user ratings to DS theoretic domain.

Definition 4 (DS Modeling Function \mathbf{f}_R) *Let the mapping,*

$$\mathbf{f}_R : r_{ik} \mapsto \mathbf{r}_{ik} = \{\Theta_{\text{pref}}, \mathcal{F}_{ik}, m_{ik}\}$$

be defined as the DS modeling function. This generates the BoE \mathbf{r}_{ik} capturing the rating r_{ik} , for all $r_{ik} \in R$ ■

How should one select an appropriate DS modeling function that captures the explicit and implicit user preferences while accommodating the associated imperfections? Unfortunately, this question is far from being trivial and one cannot expect to arrive at a “universally applicable” DS modeling function that caters to all problem domains. We believe that a domain expert should carefully design an appropriate mapping taking into account the characteristics of the domain as well as of users and items.

In the following two sections, we provide some guidelines and insight into designing proper DS modeling functions. First, we discuss the case where users have actually rated the items, i.e. $r_{ik} \neq \emptyset$. The case where users have not explicitly rated the items, is explained next, where we use an elegant methodology to reduce the ambiguity of the ratings in unrated items.

4.2.1 Rated Items

CoFiDS is applicable with any user preference that can be modeled via a DS theoretic BPA. One can identify two broad categories of application domains based on the type of user preference ratings that they allow:

1. **Domains with “Forced” Hard Preference Ratings** Typical ACF domains fall into this category. e.g., movie recommender systems with an “N-star” ratings space.
2. **“Soft” Preference Ratings** Sensitive and critical ACF application domains require this type of user preference handling. For instance, the HAART therapy scenario falls into this category.

Domains with ‘Forced’ Hard Preference Ratings

In most of the recommender systems in use today, users are forced to rate the items via “hard” or “crisp” ratings. This type of preferences can be very easily modeled with DS notions. (See the example shown in Section 3.1 of Chapter 3). In this type of domain, it may first appear that user ratings are devoid of any imperfections. However, as pointed out in [16], the ratings assignment process itself often possesses a level of uncertainty. For example, a user may have difficulty in picking, or may be unwilling to pick, a single label as the proper rating. A DS theoretic model that can capture the user uncertainty in a wide variety of such scenarios is the following:

Definition 5 (Simple DS Modeling Function) Suppose the rating that the user U_i allocates to item I_k is $r_{ik} = \theta_\ell \in \Theta_{\text{pref}}$. The mapping $\mathbf{f}_R : r_{ik} \mapsto \mathbf{r}_{ik} : \Theta_{\text{pref}} \mapsto \{\Theta_{\text{pref}}, \mathcal{F}_{ik}, m_{ik}\}$ where

$$m_{ik}(A) = \begin{cases} \alpha_{ik}(1 - \sigma_{ik}), & \text{when } A = \theta_\ell; \\ \alpha_{ik}\sigma_{ik}, & \text{when } A = B; \\ 1 - \alpha_{ik}, & \text{when } A = \Theta_{\text{pref}}; \\ 0, & \text{otherwise,} \end{cases}$$

where

$$B = \begin{cases} (\theta_1, \theta_2), & \text{if } \ell = 1; \\ (\theta_{L-1}, \theta_L), & \text{if } \ell = L; \\ (\theta_{\ell-1}, \theta_\ell, \theta_{\ell+1}), & \text{otherwise,} \end{cases}$$

is referred to as the Simple DS modeling function (S-DS Modeling Function). Here, the real-valued parameters $\alpha_{ik} \in [0, 1]$ and $\sigma_{ik} \in [0, 1]$ are referred to as the *trust_factor* and *dispersion_factor* corresponding to the rating r_{ik} , respectively. ■

The *trust_factor* and *dispersion_factor* in combination control the DS theoretic mass assigned to the user given rating. Thus, a wide variety of user preferences can be represented by carefully selecting the two parameters, *trust_factor* and *dispersion_factor*.

- *trust_factor*: quantifies how likely the user assigned rating reflects the user's true perception. The value $\alpha_{ik} = 0$ represents the case when the user's rating is completely untrustworthy; therefore it is modeled via the vacuous BoE.
- *dispersion_factor*: quantifies how likely the user assigned rating would span a larger set. The value $\sigma_{ik} = 0$ represents the case when the user assigned rating is allocated a DS theoretic mass (provided that $\alpha_{ik} \neq 0$).

For instance, consider the ACF algorithms proposed by Nakamura and Abe in [16] where weighted majority voting strategies have produced significant prediction performance improvements compared to correlation based methods. By allowing a ± 1 tolerance on user ratings when calculating similarities, these algorithms accommodate a certain level of uncertainty in user rating assignment. We may easily capture this scenario via the S-DS modeling function with the parameters $\{\alpha_{ik}, \sigma_{ik}\} = \{1, 1\}$.

One’s choice of *trust_factor* and *dispersion_factor* would be domain and dataset dependent. Depending on the available evidence and the complexity of the process, one may utilize *user-wide*, *item-wide*, or *system-wide* constants for these parameters. They can also be used as a means of capturing the “significance” of a particular rating towards the overall ACF prediction process. To elaborate, consider a scenario where most users allocate a similar rating for a particular item (e.g., most users in the *MovieLens* dataset [54] give a higher rating for the movie *Titanic* [38]). Then that rating would play a less significant role in the CF prediction process. One can use a smaller item-wide constant value for α_{ik} in such a scenario.

Domains with “Soft” Preference Ratings

Some domains may contain user preferences that are in the “soft” format — preferences that are not “hard” or “crisp,” e.g., ratings of the HAART therapy scenario introduced in Section 1.2 of Chapter 1. In this case, an *identity map* can be chosen as the DS modeling function. But, one can still design a somewhat complicated and flexible DS modeling function — taking into account the domain characteristics and available evidence for further improvements.

4.2.2 Unrated Items

An unrated entry in ratings matrix can be mapped to DS-ratings matrix as the vacuous BoE. Although one could simply proceed with this obvious DS modeling function, CoFiDS employs a more elaborate mapping. This strategy adopted by CoFiDS allows unrated items to be modeled with a reduced degree of ambiguity, which would otherwise be completely ambiguous. This approach,

- constitutes a method whereby the ratings matrix could be completely populated prior to the application of ACF,
- has the ability to combine information from multiple sources taking into account their reliability and significance, and
- provides an elegant solution to difficulties associated with data sparsity and cold-start.

To better explain how CoFiDS incorporates contextual information into ACF, consider our HAART therapy scenario introduced in Chapter 1, where a physician is interested in predicting the therapeutic response rating of a drug cocktail that has not yet been administered to a particular patient. Patient’s *Drug-Compliance*, *Initial-Viral-Load*, and *Age* are all criteria that are known to have a significant impact on the therapeutic response of a drug cocktail. Such domain expertise can be considered *Concepts* for grouping patients — each concept gives a criterion based on which the patients may be grouped, e.g., the concept *Drug-Compliance* may have the following groups: *Drug-Compliance.High*, *Drug-Compliance.Medium*, and *Drug-Compliance.Low*. Patients belonging to a group is expected to possess a similar drug responses to a given drug. Note that, the groups corresponding to a given concept need not partition the user space, i.e., a user may belong to one or more groups from the same concept.

Suppose we are interested in incorporating such contextual information to populate the hitherto unrated entry $\mathbf{r}_{ik} \in \mathbf{R}$ corresponding to patient u_i and drug cocktail

i_k . The notion being advocated by CoFiDS for this purpose is to combine or fuse the effectiveness rating that each group in which the patient u_i is member allocates to drug cocktail i_k as a whole. This fusion operation is carried out in two stages:

1. **Group Level** fuse the *group preference* of each group which patient u_i belongs to and generate a *concept preference*; and
2. **Concept Level** use the *concept preference* of all grouping concepts and generate the overall *contextual preference*.

In an attempt to formalize these notions, we make the following observation. Although the above discussion related to the HAART therapy scenario concentrated on user-based concepts, completely analogous notions exist for item-based concepts as well. For example, the physician can very well group the drug cocktails based on the item-based concept *Class_of_Drugs*. Realizing that the development of these analogous notions follows the same pattern, we concentrate only on user-based concepts. In any case, the experiments we carry out in Section 8 in Chapter 8 utilize an item-based concept for generating contextual information.

In the development of contextual preference BoEs, we consider only *one* concept for notational simplicity; if the need arises to refer to multiple concepts, we will have to attach a subscript/superscript i to differentiate among them. Unless it becomes essential, we will not incorporate this lest the notation becomes even more cumbersome than perhaps it already is. So, we proceed as follows.

Note. At this point, all the entries of DS-ratings matrix corresponding to $r_{ik} \neq \emptyset$ has been populated via an appropriate DS modeling function.

Nomenclature

Let us identify the Q number of groups belonging to the “generic” concept *Concept* as $\{Concept.Group_1, \dots, Concept.Group_Q\}$. We identify the groups to which a user

belongs via the mapping $f_C : U \mapsto \{Concept.Group_1, \dots, Concept.Group_Q\}$; this mapping is referred to as the *grouping function*.

Group Preference BoE

We capture how the group members belonging to the group $Concept.Group_j$ would, as a whole, rate the item I_k via a DS theoretic BPA. If information regarding the group preferences of each item is available, e.g., “*therapeutic response of the group Drug_Compliance.Low is ‘poor’ for all items,*” one may use this information directly in a DS theoretic setting; otherwise, one may use those users within a given group who have already rated item I_k .

Definition 6 (Group Preference BoE) *The group preference BPA is $m_k^{(Group_j)} : 2^\Theta \mapsto [0, 1]$, where*

$$m_k^{(Group_j)} = \bigoplus_{\substack{i: U_i \in Concept.Group_j; \\ I_k \in R_i}} m_{ik}.$$

The corresponding BoE

$$\mathbf{BoE}_k^{(Group_j)} = \{\Theta_{\text{pref}}, \mathcal{F}_k^{(Group_j)}, m_k^{(Group_j)}\}$$

is referred to as the group preference BoE. ■

Concept Preference BoE

The concept preference BoE corresponding to user U_i and item I_k is then obtained by combining or fusing these group preference BoEs.

Definition 7 (Concept Preference BoE) *The concept preference BPA is $m_{ik}^{(Concept)} : 2^\Theta \mapsto [0, 1]$, where*

$$m_{ik}^{(Concept)} = \bigoplus_{\substack{j: Concept.Group_j \\ \in f_C(U_i)}} m_k^{(Group_j)}.$$

The corresponding BoE

$$\mathbf{BoE}_{ik}^{(Concept)} = \{\Theta_{\text{pref}}, \mathcal{F}_{ik}^{(Concept)}, m_{ik}^{(Concept)}\}$$

is referred to as the concept preference BoE. ■

Contextual Preference BoE

The overall contextual preference BoE corresponding to user U_i and item I_k is then obtained by fusing all the concept preference BoEs.

Definition 8 (Contextual Preference BoE) The contextual preference BPA is $m_{ik}^{(Context)}$: $2^\Theta \mapsto [0, 1]$, where

$$m_{ik}^{(Context)} = \bigoplus_{\text{All Concepts}} m_{ik}^{(Concept)}.$$

The corresponding BoE

$$\mathbf{BoE}_{ik}^{(Context)} = \{\Theta_{\text{pref}}, \mathcal{F}_{ik}^{(Context)}, m_{ik}^{(Context)}\}$$

is referred to as the contextual preference BoE. ■

At this point, CoFiDS modifies the DS ratings matrix \mathbf{R} such that each unrated entry is replaced by its corresponding contextual preference BoE, i.e., $\mathbf{r}_{ik} \equiv \mathbf{BoE}_{ik}^{(Context)}$ when matrix element $r_{ik} = \emptyset$. This is the ratings matrix that we would employ from now onwards.

Taking ‘‘Reliability and Significance of The Information Sources’’ into account

In the fusion operations being carried out in Definitions 6,7 and 8, one may employ a discounting factor to discount each constituent BoE prior to application of the DRC. This may be particularly relevant in an application such as our HAART therapy scenario. For example, if one concept, say *Age*, is known to have less of an impact

on the drug response than the other concepts, a discounting factor can be used to accommodate this fact.

Once the DS-ratings matrix is generated via use of appropriate DS modeling functions, we may now generate the user-BoE.

4.3 User-BoE

While the BoE \mathbf{r}_{ik} can be considered an “intra-item” BoE that captures the preference that a user has towards a single item, to capture the preference that the user has towards *all* items as a whole, one requires an appropriately constructed “inter-item” BoE defined over the cross-product space of $\Theta \equiv \Theta_{\text{pref}} \times \dots \times \Theta_{\text{pref}}$ (N times) $\equiv \prod_{j=1}^N \Theta_{\text{pref}}$. To proceed, we need to introduce the following notion:

Definition 9 (Cylindrical Extension) *Consider the focal element $A \in \mathcal{F}_{ik}$ extracted from the BoE \mathbf{r}_{ik} . Its cylindrical extension to the cross-product FoD Θ is*

$$\text{cyl}_{\Theta}(A) = \left[\Theta_1 \quad \dots \quad \Theta_{i-1} \quad A \quad \Theta_{i+1} \quad \dots \quad \Theta_N \right],$$

where $\Theta_i = \Theta_{\text{pref}}, \forall i = \overline{1, N}$. ■

We may then show that [35]

Lemma 1 *The mapping $M_{ik} : 2^{\Theta} \mapsto [0, 1]$ where*

$$M_{ik}(B) = \begin{cases} m_{ik}(A), & \text{for } B = \text{cyl}_{\Theta}(A); \\ 0, & \text{otherwise.} \end{cases}$$

generates a valid BPA defined on the FoD Θ . The corresponding BoE is referred to as the user-BoE generated by extending the BoE \mathbf{r}_{ik} . ■

This leads us to

Definition 10 (User-BoE) For user u_i , consider the BoEs $M_{ik}(\bullet)$, $k = \overline{1, N}$, generated by extending the BoEs \mathbf{r}_{ik} , $i = \overline{1, N}$, respectively. Then, the BPA $M_i : 2^\Theta \mapsto [0, 1]$ where

$$M_i = \bigoplus_{k=1}^N M_{ik},$$

is referred to as the user-BPA of user u_i ; the corresponding BoE $\{\Theta, \mathcal{F}_i, M_i\}$ is referred to as the user-BoE. ■

4.3.1 An Important Result on User-BoEs

We utilize the following important result in similarity computation stage.

Claim 1 Consider the i -th user's user-BPA M_i (defined over the cross-product FoD Θ) and the ratings BPAs m_{ik} , $k = \overline{1, N}$, each defined over the FoD Θ_{pref} . Then, the pignistic probability of the singleton $\prod_{k=1}^N \theta^{(i_k)} \equiv \theta^{(i_1)} \times \dots \times \theta^{(i_N)} \in \Theta$ is

$$Bp_i \left(\prod_{k=1}^N \theta^{(i_k)} \right) = \prod_{k=1}^N Bp_{ik}(\theta^{(i_k)}),$$

where $\theta^{(i_k)} \in \Theta_{\text{pref}}$, $j = \overline{1, N}$. Here, $Bp_i(\bullet)$ and $Bp_{ik}(\bullet)$ refer to user u_i 's pignistic probability distributions corresponding to its user-BoE and ratings BoEs, respectively.

□

Proof: From (2.7), note that

$$\begin{aligned}
\text{Bp}_i\left(\prod_{k=1}^N \theta^{(i_k)}\right) &= \sum_{\theta^{(i_k)} \in A_k \subseteq \Theta_{\text{pref}}} \frac{M_i(A_1 \times \cdots \times A_N)}{|A_1| \cdots |A_N|} \\
&= \sum_{\theta^{(i_k)} \in A_k \subseteq \Theta_{\text{pref}}} \prod_{k=1}^N \frac{m_{i_k}(A_k)}{|A_k|} \\
&= \prod_{k=1}^N \sum_{\theta^{(i_k)} \in A_k \subseteq \Theta_{\text{pref}}} \frac{m_{i_k}(A_k)}{|A_k|} \\
&= \prod_{k=1}^N \text{Bp}_{i_k}(\theta^{(i_k)}).
\end{aligned}$$

■

CHAPTER 5

Similarity and Neighborhoods

We have elaborated upon the importance of computing user similarities based on their preferences over all items. A user-BoE defines a user’s preference over all items. Thus, we can employ a distance metric defined on the cross-product FoD $\Theta (\equiv \prod_{j=1}^N \Theta_{\text{pref}})$ to calculate the “distance” between two users and then use it to identify the similarity among users. But, in general, DS theoretic models demand extra computational power compared to other related methods. If one wish to work on the cross-product space, computations can quickly become untractable. These huge computational requirements — sometimes prohibitive — have to be tackled very efficiently and carefully, making the problem at hand tractable, but still not destroying the rich information content, a DS theoretic model provides. We proceed as follows.

5.1 Distance Between User-BoEs

If a distance measure between two probability mass functions (p.m.f.s) is available, via the application of the pignistic transformation in (2.7) [51], one may use it as a distance measure between two BoEs. For our purposes, we use the distance measure introduced recently in [58] mainly because of its numerous desirable properties. We

combine these properties with the result we obtained in last chapter in Claim 1 to obtain significant improvements over computational requirements.

Definition 11 (Distance Measure Between User-BoEs) *The distance between the two user-BPAs M_i and M_j defined over the same cross-product FoD Θ is*

$$D(M_i, M_j) = CD(Bp_i, Bp_j),$$

where Bp_i and Bp_j denote the pignistic probability transformations corresponding to M_i and M_j , respectively, and $CD(\bullet, \bullet)$ refers to the Chan-Darwiche (CD) distance measure [58]:

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

■

We then obtain the following important result on distances measured between user-BoEs. This result implies that one can obtain user distances by summing the distances among individual items over all items.

Claim 2 *The distance between the two user-BPAs M_i and M_j is*

$$D(M_i, M_j) = \sum_{k=1}^N CD(Bp_{ik}, Bp_{jk}),$$

where Bp_{ik} and Bp_{jk} refer to the pignistic probability distributions corresponding to the ratings BPAs of users u_i and u_j , respectively. □

Proof: Consider the singleton $\theta = \theta^{(\ell_1)} \times \dots \times \theta^{(\ell_N)} \equiv \prod_{k=1}^N \theta^{(\ell_k)} \in \Theta$. Use

Claim 1:

$$\begin{aligned} \ln \max_{\theta \in \Theta} \frac{\text{Bp}_j(\theta)}{\text{Bp}_i(\theta)} &= \ln \max_{\ell_k=1,L} \frac{\prod_{k=1}^N \text{Bp}_j(\theta^{(\ell_k)})}{\prod_{k=1}^N \text{Bp}_i(\theta^{(\ell_k)})} \\ &= \max_{\ell_k=1,L} \ln \prod_{k=1}^N \frac{\text{Bp}_j(\theta^{(\ell_k)})}{\text{Bp}_i(\theta^{(\ell_k)})} \\ &= \max_{\ell_k=1,L} \sum_{k=1}^N \ln \frac{\text{Bp}_j(\theta^{(\ell_k)})}{\text{Bp}_i(\theta^{(\ell_k)})} \\ &= \sum_{k=1}^N \ln \max_{\ell_k=1,L} \frac{\text{Bp}_j(\theta^{(\ell_k)})}{\text{Bp}_i(\theta^{(\ell_k)})}. \end{aligned}$$

Similarly, we have

$$\ln \min_{\theta \in \Theta} \frac{\text{Bp}_j(\theta)}{\text{Bp}_i(\theta)} = \sum_{k=1}^N \ln \min_{\ell_k=1,L} \frac{\text{Bp}_j(\theta^{(\ell_k)})}{\text{Bp}_i(\theta^{(\ell_k)})}.$$

Therefore,

$$\begin{aligned} CD(\text{Bp}_i, \text{Bp}_j) &= \sum_{k=1}^N \left[\ln \max_{\ell_k=1,L} \frac{\text{Bp}_j(\theta^{(\ell_k)})}{\text{Bp}_i(\theta^{(\ell_k)})} - \ln \min_{\ell_k=1,L} \frac{\text{Bp}_j(\theta^{(\ell_k)})}{\text{Bp}_i(\theta^{(\ell_k)})} \right] \\ &= \sum_{k=1}^N CD(\text{Bp}_{i_k}, \text{Bp}_{j_k}). \end{aligned}$$

This completes the proof. ■

Thus, one may now use distances between user preference ratings BoEs defined over Θ_{pref} , instead of directly computing the distance between the two user-BoEs defined over the cross-product FoD Θ . The associated reduction in computational overhead is from $\mathcal{O}^M C_2 L^N$ to $\mathcal{O}^M C_2 L N$, or by a fraction of $\mathcal{O}(L^{N-1}/N)$.

Computational Complexity Reduction

Let's compare the computational requirements for the two approaches. First, let's define the set, $\Theta_{\text{BetP}} = \{\theta \mid \theta = \theta^{(\ell_1)} \times \dots \times \theta^{(\ell_N)}, \theta^{(\ell_k)} \in \Theta_{\text{pref}}\}$. This set exhausts all possible singletons θ , that are to be used, if one wishes to use $D(M_i, M_j) =$

$CD(\text{Bp}_i, \text{Bp}_j)$ approach. Here, we make a formal assumption that the pignistic probabilities $\text{Bp}_j(\theta)$ and $\text{Bp}_j(\theta^{(l_k)})$ are readily available for all j, k, l . Moreover, we spread-out the computational resource requirements for \min , \max and \ln over $\frac{\text{Bp}_j(\bullet)}{\text{Bp}_j(\bullet)}$ computations, and take that as the unit measure.

Let's compute the distance between two user-BPAs M_i and M_j , via $D(M_i, M_j) = CD(\text{Bp}_i, \text{Bp}_j)$. In this approach, we need to evaluate $\ln \max_{\theta \in \Theta_{\text{BetP}}} \frac{\text{Bp}_j(\theta)}{\text{Bp}_i(\theta)}$ and $\ln \min_{\theta \in \Theta_{\text{BetP}}} \frac{\text{Bp}_j(\theta)}{\text{Bp}_i(\theta)}$ for all such $\theta \in \Theta_{\text{BetP}}$. So, with $|\Theta_{\text{BetP}}| = L^N$, there are L^N of $\frac{\text{Bp}_j(\bullet)}{\text{Bp}_i(\bullet)}$ computations for each user pair. Since, the distance is to be calculated for each user pair, we get ${}^{|U|}C_2 = {}^M C_2$ possibilities, thus giving rise to a total number of ${}^M C_2 L^N$ computations. Hence, the computational complexity for this approach is in $\mathcal{O}({}^M C_2 L^N)$.

Now, let's compute the distance via $D(M_i, M_j) = \sum_{k=1}^N CD(\text{Bp}_{ik}, \text{Bp}_{jk})$. Thus, there are N computations of $CD(\text{Bp}_{ik}, \text{Bp}_{jk})$. Now, each such computation requires evaluation of $\ln \max_{\theta^{(l_k)} \in \Theta_{\text{Pref}}} \frac{\text{Bp}_j(\theta^{(l_k)})}{\text{Bp}_i(\theta^{(l_k)})}$ and $\ln \min_{\theta^{(l_k)} \in \Theta_{\text{Pref}}} \frac{\text{Bp}_j(\theta^{(l_k)})}{\text{Bp}_i(\theta^{(l_k)})}$ for each $\theta^{(l_k)} \in \Theta_{\text{Ppref}}$, thus with $|\Theta_{\text{Ppref}}| = L$ we have a total of LN computations of $\frac{\text{Bp}_j(\bullet)}{\text{Bp}_i(\bullet)}$ for each user pair. So, for all pairs of users we get a total of ${}^M C_2 LN$ computations, hence the computational complexity is in $\mathcal{O}({}^M C_2 LN)$.

Hence, the use of Claim 2 results in an optimization with an order of $\mathcal{O}(L^{N-1}/N)$.

5.2 User-User Similarity

We quantify the similarity between two users via

Definition 12 (User-User Similarity) *Consider a monotonically decreasing function $\psi : [0, \infty] \mapsto [0, 1]$ satisfying $\psi(0) = 1$ and $\psi(\infty) = 0$. Then, with respect to $\psi(\bullet)$, $s_{ij} = \psi(D(M_i, M_j))$ is referred to as the user-user similarity between users U_i and U_j .*

■

For $\psi(\bullet)$, one may simply use $\psi(x) = e^{-\gamma x}$, where $\gamma \in (0, \infty)$ is a domain specific constant. The $M \times M$ user-user similarity matrix is then generated as $S = \{s_{ij}\}$. This transformation allows one to view the similarities in the conventional $[0, 1]$ interval. Indeed, CoFiDS uses the same similarity values for discounting neighbor rating BoEs.

5.3 User Neighborhood

As we have mentioned earlier, to overcome the weaknesses involved with KNN strategy, CoFiDS uses *K-nearest with minimum similarity thresholding technique* as proposed in [38]. In a situation, where adequate amount of rated users satisfying given constraints can not be obtained, CoFiDS makes use of the evidence gathered from fusing contextual information. We proceed as follows.

Definition 13 (Neighborhood Set) *The neighborhood set \mathbf{Nbhd}_{ik} of user $U_i \in \mathbf{U}$ for prediction of item $I_k \in \mathbf{I}$ is the largest set that satisfies the following: for given parameters τ , K and K_{min} ,*

$$\mathbf{Nbhd}_{ik} = \begin{cases} \mathbf{Nbhd}^{(primary)}, & \text{if } |\mathbf{Nbhd}^{(primary)}| < K_{min} \\ \mathbf{Nbhd}^{(secondary)}, & \text{otherwise} \end{cases}$$

where,

$$\begin{aligned} \mathbf{Nbhd}^{(primary)} &= \left\{ U_j \in R_k^{(user)} \mid s_{ij} \geq \max_{\forall U_l \notin \mathbf{Nbhd}^{(primary)}} \{\tau, s_{il}\} \right\} \text{ and,} \\ \mathbf{Nbhd}^{(secondary)} &= \left\{ U_j \in \mathbf{U} \mid s_{ij} \geq \max_{\forall U_l \notin \mathbf{Nbhd}^{(secondary)}} \{\tau, s_{il}\} \right\} \\ &\text{with } |\mathbf{Nbhd}^{(primary)}| \leq K, \quad |\mathbf{Nbhd}^{(secondary)}| \leq K. \end{aligned}$$

■

So, \mathbf{Nbhd}_{ik} may be chosen via the following strategy:

1. Use a similarity threshold τ , and select users from $R_k^{(user)}$ — the users who have rated item I_k , those who meet the minimum similarity threshold τ with U_i ; then
2. Apply KNN and select at most K users having the highest similarity with U_i from this user set. Denote the resulting set as $\mathbf{Nbhd}^{(primary)}$; then
3. If the number of users in $\mathbf{Nbhd}^{(primary)}$ is larger than or equal to K_{min} , then choose it as \mathbf{Nbhd}_{ik} . If not,
4. Use a similarity threshold τ , and select users from \mathbf{U} — the total user space, those who meet the minimum similarity threshold τ with U_i ; then
5. Apply KNN and select at most K users having the highest similarity with U_i from this user set. The resulting set — $\mathbf{Nbhd}^{(secondary)}$, gives the \mathbf{Nbhd}_{ik} .

CHAPTER 6

Prediction and Decision Making

As we have already discussed, prediction generation is the most crucial step in any recommender system. ACF predictions are usually generated by fusing only the evidence gathered from the neighbors. CoFiDS slightly differs in this aspect. It has already used contextual information to populate the ratings of all users. Moreover, CoFiDS has filled up the active user's item on which the current prediction is to be done as well. Thus, CoFiDS fuses this information with the gathered evidence from neighbors to make the final prediction. We proceed as follows.

6.1 Prediction

Definition 14 *CoFiDS represents the prediction of the unrated item i_k of the active user u_i as the BoE $\hat{\mathbf{r}}_{ik} = \{\Theta_{\text{pref}}, \hat{\mathcal{F}}_{ik}, \hat{m}_{ik}\}$, where*

$$\hat{m}_{ik} = m_{ik}^{(\text{Nbhd})} \oplus m_{ik}.$$

Here, $m_{ik}^{(\text{Nbhd})}$ is the BPA corresponding to the neighborhood prediction BoE

$$\mathbf{BoE}_{ik}^{(\text{Nbhd})} = \{\Theta_{\text{pref}}, \mathcal{F}_{ik}^{(\text{Nbhd})}, m_{ik}^{(\text{Nbhd})}\},$$

where

$$m_{ik}^{(\text{Nbhd})} = \bigoplus_{\forall U_j \in \text{Nbhd}_{ik}} m_{jk}^{(\text{disc})},$$

with

$$m_{jk}^{(\text{disc})}(A) = \begin{cases} s_{ij}m_{jk}(A), & \text{for } A \subset \Theta_{\text{pref}}; \\ (1 - s_{ij}) + s_{ij}m_{jk}(\Theta_{\text{pref}}), & \text{for } A = \Theta_{\text{pref}}. \end{cases}$$

Remark. Since the similarities are captured via the user-user similarity, in Definition 14 CoFiDS utilizes this similarity as a discounting factor to “discount” the ratings BoEs of the neighbors prior to fusion. Higher the similarity of a neighbor to active user, more weight is given to his actual rating. When, the similarity between two users degrades, DS theoretic mass is pushed towards Θ_{pref} , lowering the contribution to fused BPA.

6.2 Decision Making

The DS theoretic ratings prediction that we get from CoFiDS provides much more flexibility to the decision-maker that other ACF schemes may not provide. Not only does this flexibility allow one to make a decision that better caters to the application domain requirements, it also provides information regarding the confidence associated with the ratings prediction.

For a “hard” decision on a *singleton* classification, one may use the pignistic probability in (2.7) and pick a singleton as *the* preference label. If *one* preference label (a singleton *or* a composite) is desired, one can use the *maximum belief with non-overlapping interval strategy (maxBL)* [59]. This involves picking the singleton preference label whose belief is greater than the plausibility of any other singleton; if such a preference label does not exist, one decides in favor of the composite preference label constituted of the singleton label that has the maximum belief and those singletons that have a higher plausibility. The pignistic transformation may be utilized to establish a rank ordering of the constituent singletons of this maxBL classifier [35, 60].

CHAPTER 7

Evaluation Matrices

Performance evaluation has always been a problem for CF researches. With CoFiDS providing “soft” predictions, this task has become even more difficult. Thus, we will stick to the widely used measures. In this section, we describe how an algorithm producing predictions with richer information contents, such as CoFiDS can be evaluated and compared to other algorithms along different dimensions. Before proceeding, recall the following notation regarding the rating that user U_i allocates to item I_k :

7.1 Nomenclature

User ratings are allocated via the label set $\Theta_{\text{pref}} = \{\theta_1, \dots, \theta_L\}$. The true rating is given by $r_{ik} \in \Theta_{\text{pref}}$. Let \hat{r}_{ik} denote a “crisp” prediction, whereas CoFiDS prediction is given by $\hat{\mathbf{r}}_{ik}$. Let $D^{(\text{Test})}$ denote the testing dataset.

7.2 Performance Measures for “Crisp” Databases

Here, the dataset is “crisp.” Thus, the widely used *Mean Absolute Error (MAE)* [61] can be used for performance evaluation. *MAE* is a direct indication of how different the true and predicted ratings are. It is defined as,

Definition 15 (Mean Absolute Error for a given label θ_j - $MAE(\theta_j)$)

$$MAE(\theta_j) = \frac{1}{|D_j|} \sum_{(i,k) \in D_j} |r_{ik} - \hat{r}_{ik}|, \quad (7.1)$$

where

$$D_j = \bigcup \left\{ (i, k) \in D^{(\text{Test})} \mid r_{ik} = \theta_j \in \Theta_{\text{pref}} \right\}, \quad (7.2)$$

where $MAE(\theta_j)$ is referred to as the MAE corresponding to the rating θ_j of the testing set $D^{(\text{Test})}$. ■

$MAE(\theta_j)$ identifies the prediction error in user-item pairs whose true rating is $\theta_j \in \Theta_{\text{pref}}$. The overall MAE for $D^{(\text{Test})}$ can be obtained as,

Definition 16 (Overall Mean Absolute Error - MAE)

$$MAE = \frac{1}{|D^{(\text{Test})}|} \sum_{(i,k) \in D^{(\text{Test})}} |r_{ik} - \hat{r}_{ik}| \quad (7.3)$$

$$= \frac{1}{|D^{(\text{Test})}|} \sum_{j=1}^L MAE(\theta_j) \cdot |D_j| \quad (7.4)$$

■

But, note that the MAE can only be used if the predictions are “crisp.” Thus, one can use pignistic transformation on DS-theoretic predictions to obtain “hard” decisions in order to apply MAE . But, this transformed prediction do not contain the richer information content, that the original belief theoretic predictor had.

Memory-based ACF algorithms are usually not viewed as a classification task. But, if one wish to evaluate the algorithm as a classification task, traditional *precision*, *recall* and other related measures can be used. For “soft” decisions, one can use the DS theoretic measures *DS-Precision*, *DS-Recall*, and *DS-Accuracy* proposed in [35].

Definition 17 (DS Performance Measures [35])

$$DS\text{-Precision}(\theta_j) = \frac{TP(\theta_j)}{TP(\theta_j) + FP(\theta_j)} ; \quad (7.5)$$

$$DS\text{-Recall}(\theta_j) = \frac{TP(\theta_j)}{TP(\theta_j) + FN(\theta_j)} ; \quad (7.6)$$

$$DS\text{-Accuracy} = \frac{\sum_{\theta_j \in \Theta_{\text{pref}}} TP(\theta_j)}{\sum_{\theta_j \in \Theta_{\text{pref}}} |D_j|}, \quad (7.7)$$

where

$$TP(\theta_j) = \sum_{(i,k) \in D_j} \widehat{Bp}_{ik}(\theta_j);$$

$$FP(\theta_j) = \sum_{\substack{(i,k) \in D_\ell; \\ j \neq \ell}} \widehat{Bp}_{ik}(\theta_j);$$

$$FN(\theta_j) = \sum_{\substack{(i,k) \in D_j; \\ j \neq \ell}} \widehat{Bp}_{ik}(\theta_\ell).$$

Here, $\widehat{Bp}_{ik}(\bullet)$ refers to the pignistic probability corresponding to the DS theoretic BPA $\widehat{m}_{ik}(\bullet)$. ■

Inspired by these DS theoretic measures and the traditional definitions, for the purpose at hand, we also introduce the following measures:

Definition 18 (DS- F_β Measure)

$$DS\text{-}F_\beta(\theta_j) = \frac{(\beta^2 + 1) \cdot DS\text{-Precision}(\theta_j) \cdot DS\text{-Recall}(\theta_j)}{\beta^2 \cdot DS\text{-Precision}(\theta_j) + DS\text{-Recall}(\theta_j)} \quad (7.8)$$

where $DS\text{-}F_\beta(\theta_j)$ defines the DS theoretic counterpart for F_β measure. ■

Definition 19 (DS Theoretic Mean Absolute Error DS-MAE)

$$DS\text{-MAE}(\theta_j) = \frac{1}{|D_j|} \sum_{\substack{(i,k) \in D_j; \\ \theta_\ell \in \Theta_{\text{pref}}}} \widehat{Bp}_{ik}(\theta_\ell) \cdot |r_{ik} - \text{rank}(\theta_\ell)| \quad (7.9)$$

where $DS\text{-MAE}$ defines the DS theoretic counterpart of the MAE measure. ■

$DS\text{-MAE}$ measure is capable of taking into account the mass that is assigned to preference labels in computing the error between predicted and actual ratings.

7.3 Performance Measures for “Soft” Databases

If the given dataset is “soft” to begin with, none of the above defined measures can not be used. But, in this case, both given and predicted ratings are both “soft.” Thus, one can make a comparison using a distance measure defined over belief functions. Inspired by the work done in [62], we define the following measure.

Definition 20 (Type-1 DS Prediction Error - $DS-PE1$)

$$DS-PE1 = \frac{1}{|D^{(test)}|} \sum_{(i,k) \in D^{(test)}} BD(\hat{\mathbf{r}}_{ik}, \tilde{\mathbf{r}}_{ik}), \quad (7.10)$$

where $BD(\bullet, \bullet)$ is given by,

$$BD(\vec{m}_i, \vec{m}_j) = \sqrt{\frac{1}{2}(\vec{m}_i - \vec{m}_j)^T D (\vec{m}_i - \vec{m}_j)}$$

where D is a $(2^{\Theta_{\text{pref}}} \times 2^{\Theta_{\text{pref}}})$ -dimensional matrix with $d[i, j] = |A \cap B|/|A \cup B|$, and $A, B \in 2^{\Theta_{\text{pref}}}$ with $|\emptyset \cap \emptyset|/|\emptyset \cup \emptyset| = 0$. ■

We define another performance measure,

Definition 21 (Type-2 DS Prediction Error - $DS-PE2$)

$$DS-PE2 = \frac{1}{|D^{(test)}|} \sum_{(i,k) \in D^{(test)}} \|Bp(\hat{\mathbf{r}}_{ik}) - Bp(\tilde{\mathbf{r}}_{ik})\|, \quad (7.11)$$

where $Bp(\bullet)$ and $\|\bullet\|$ denote the pignistic probability distribution and the euclidean norm respectively. ■

Note that both $DS-PE1$ and $DS-PE2$ is bounded both from above and below by 1 and 0 respectively. One could use KL-Divergence instead of euclidean norm in $DS-PE2$. But, in that case the error is unbounded and one has to overcome issues that may arise from not having the identical support on predictions and actual preferences.

CHAPTER 8

Experiments and Results

In this chapter we apply CoFiDS to a popular ACF problem of *movie recommendations*. Our intention is to study its behavior on different conditions and to evaluate and compare to existing algorithms. The behavior of the proposed simple DS modeling function was studied in a setup where, it was used to model the uncertainties in a typical movie recommendation domain.

First, we apply CoFiDS to a benchmark ACF dataset, *MovieLens* — a movie recommendation dataset [54] widely used by ACF researchers for validation and comparison purposes. Strength of CoFiDS is best illustrated on a “soft” dataset. But, due to the unavailability of such a dataset, CoFiDS is evaluated on a synthetic dataset which we refer to as *DS–MovieLens*.

8.1 Datasets

8.1.1 *MovieLens*

MovieLens consists of 100,000 ratings of 943 users on 1682 movies. Ratings are given via the integers 1 – 5, with 1 being the worst and 5 being excellent. Thus, $\Theta_{\text{pref}} = \{1, 2, 3, 4, 5\}$. In addition, *MovieLens* contains *Genre* data on all 1682

movies along with other user metadata such as *Age*, *Gender* and *Occupation*, and item metadata such as *Title*, *IMDb_URL*, etc.

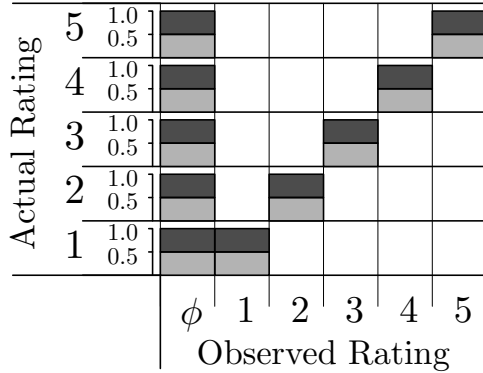
8.1.2 *DS–MovieLens*

DS–Movielens is generated using *MovieLens* itself, thus making it identical to original *MovieLens* dataset in database properties, such as number of users, number of items, rating characteristics such as data sparsity etc. CF datasets are unique in the sense that they carry hidden user-user, item-item and item-user relationships. We believe that these relationships lay the foundation for ACF, and the procedure we adopted is actually capable of preserving such relationships in the transformation of *MovieLens* to *DS–MovieLens*. In fact, this is indirectly proven, by the fact that the *LKLD–MovieLens* dataset we obtained being identical to *MovieLens*.

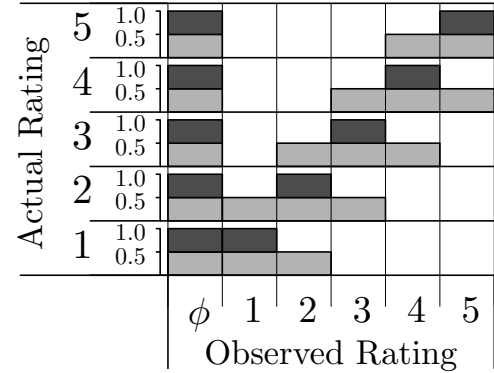
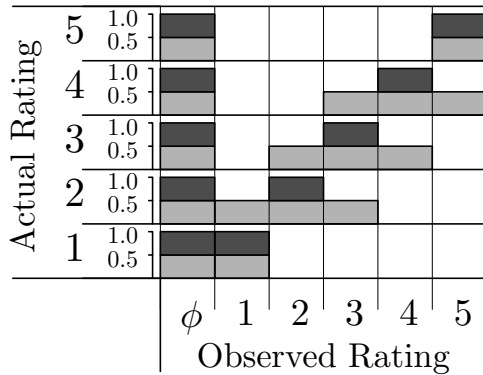
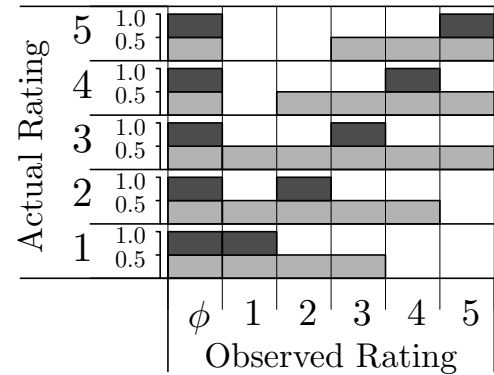
A user rating for a given movie in *MovieLens* can be identified as an observation based on which, we have to deduce the actual preference. If the user has not rated a movie, preference is completely ambiguous. If the user has actually rated it, his actual preference may or may not be interrupted by his disposition or any other related cause which manifests as a noise in his real preference. We utilize this fact, and employ a partial probability model with the evidential reasoning based approach referred to as “*powerset method*” in [63] to generate a “soft” dataset. Powerset method has previously been used for generating “soft” datasets in [35].

Thus the *Observed_Rating* in *MovieLens* could actually come from either user’s actual preference or from noise in his/her rating. This is modeled via two equally likely uniform probability distributions as shown by light and dark regions respectively in Figures 8.1(a), 8.1(b), 8.1(c) and 8.1(d), where the dark distribution denote the user’s actual rating, and light distribution denote his uncertainty associated with it.

Different users or user dispositions give rise to different levels of noise in ratings. These different user dispositions can be modeled via appropriate partial probability distributions.



(a) Zero Tolerance User

(b) ± 1 Tolerance User(c) End-weighted ± 1 Tolerance User(d) ± 2 Tolerance UserFigure 8.1: Partial Probability Models used in generating *DS-MovieLens*.

To explain the power set approach, let us identify the gray and black distributions via 0 and 1, respectively. Then, the state of nature can be considered to be in one of $2^5 = 32$ states. Suppose a ± 1 tolerance user has allocated an *Observed_Rating* of 2 in *MovieLens*. Then, if the state of nature is $\{1, x, 1, x, x\}$ – i.e., the generating distributions are black for *Actual_Rating* = $\{1, 3\}$ and it is either gray or black for the other ratings – then the only “feasible” *Actual_Rating* that could have generated *Observed_Rating* = 2 is in fact *Actual_Rating* = 2; if

the state of nature is $\{0, x, 0, x, x\}$ – i.e., the generating distributions are gray for $Actual_Rating = 2 = \{1, 3\}$ and it is either gray or black for the other ratings – then the feasible true ratings are $Actual_Rating = \{1, 2, 3\}$. In this manner, one may complete the “Feasible True Rating” column in Table 8.1.

State of Nature $\{1, 2, 3, 4, 5\}$	Feasible States $Actual_Rating$	<i>MovieLens</i> dataset		
		<i>DS</i>	<i>PR</i>	<i>LKLD</i>
(1,x,1,x,x)	{2}	0.4286	{1} = 0.1548	
(0,x,1,x,x)	(1,2)	0.2143	{2} = 0.6904	{2}
(1,x,0,x,x)	(2,3)	0.2143	{3} = 0.1548	
(0,x,0,x,x)	(1,2,3)	0.1428		

Table 8.1: $Observed_Rating = 2$ of a ± 1 user. Generating $DS-MovieLens$, $PR-MovieLens$ and $LKLD-MovieLens$ from *MovieLens*

The set of feasible true ratings of any other MovieLens rating corresponding to an arbitrary user “disposition” can be obtained similarly. For more details on this mechanism, see [35] and [63]. So, we generate $DS-MovieLens$ as follows.

1. Observe a rating given by user u_i for movie i_k
2. Randomly pick one of the four models in Figure 8.1 with probability $\{p, (1 - p)/3, (1 - p)/3, (1 - p)/3\}$.
3. Obtain the feasible states and evidential masses from the selected model for the observed reading
4. Assign the read BoE as corresponding entry in $DS-MovieLens$. viz., r_{ik}
5. Repeat for all entries in *MovieLens*

8.1.3 $PR-MovieLens$

Probabilistic dataset of $DS-MovieLens$ is obtained via pignistic transformation.

8.1.4 *LKLD–MovieLens*

Applying the likelihood criterion to *PR–MovieLens* one can obtain the *LKLD–MovieLens*. Due to the symmetry in the user “disposition” profiles and properties of powerset method, *LKLD–MovieLens* dataset turns out to be identical to *MovieLens* dataset. This indeed verifies our initial contention of preserving the user-user, item-item and user-item relationships in the *MovieLens*, when generating the soft dataset.

8.2 Other ACF algorithms for Comparison

For comparison purposes, we re-implemented the following ACF algorithms as well:

CORR : This refers to the correlation-based ACF algorithm in [38]. These results were taken as the baseline since it can be considered the most widely used and well known ACF system.

NA : This refers to the recent ACF strategy proposed by Nakamura and Abe in [16]. They suggest three variants based on user-user and item-item similarities, and a combination of the two; we refer to these as *u–NA*, *i–NA*, and *c–NA* respectively. These algorithms attracted our attention because they enable one to accommodate the ignorance inherent in user ratings. Indeed, the results in [16] demonstrate a significant improvement over correlation-based methods.

We apply CoFiDS, CORR and NA to *MovieLens*. But, one cannot directly apply either CORR or NA to *DS–MovieLens*. If one wishes to apply the widely used *CORR* to such a dataset, the trivial approach is to obtain a crisp dataset via pignistic transformation first, and then apply the algorithms to the crisp “ratings.” Thus, for comparison purposes we obtain such a “crisp” ratings as follows.

1. Pick the corresponding entry in *PR–MovieLens*. Then,
2. Weigh each rating $\theta_\ell \in \Theta_{\text{pref}}$ by it’s probability
3. Sum the weighted ratings to obtain the “crisp” value, which is in general a non-integer.

CORR is then applied to those “crisp” ratings for comparison with CoFiDS. But, on the other hand, *NA* requires the ratings to be integer valued. Generating an integer valued dataset from a “soft” dataset severely damages the information content. Thus, we do not test *NA* on *DS–MovieLens*.

8.3 User Preference Modeling

8.3.1 Experiments on “Crisp” Dataset

DS Modeling Function

We have extensively tested CoFiDS using the *S-DS* modeling function in Definition 5. The model parameters — trust factors and dispersion factors — were all set to be system-wide constants to reduce extra computational cost and due to the unavailability of relevant supplementary information. Thus, we have $\{\alpha_{ik}, \sigma_{ik}\} \equiv \{\alpha, \sigma\}$, $\forall i, k$ throughout all experiments.

Group Preference BoE

The only concept we consider for generating item-based contextual information is the *Genre* information. The other concepts one may consider are *Cast* and *Director* (both item-based concepts), *Age* (a user-based concept), etc.

Recalling the nomenclature and the development in Section 4.2.2, for *Genre* information, we identify the following:

$$\begin{aligned} \text{Concept} &:= \text{Genre}; \\ \text{Groups} &:= \{\text{Genre.Group}_1, \dots, \text{Genre.Group}_Q\}, \end{aligned} \tag{8.1}$$

where the concept groups could be *Drama*, *Thrillers*, *Romance*, etc.

For generating the group preference BoEs, we capture how movies belonging to a particular genre would, as a whole, be rated by a given user U_i . One can design a movie recommender system where users are allowed to express their genre preferences explicitly. These then can be captured via a DS theoretic model. With no such information available in *MovieLens*, we used Definition 6 with those movies that have already been rated by user U_i to estimate the genre preferences. No discounting was incorporated.

Concept Preference BoE

We followed Definition 7 with no discounting.

Contextual Preference BoE

We followed Definition 8 with no discounting. Note that, if additional concepts are being utilized, not all concepts may contribute equally to user preferences. For example, the concept *Director* may contribute differently than the concept *Cast*. These difference should be accommodated via discounting.

8.3.2 Experiment on “Soft” Dataset

We simply take the given ratings as the user preference BoE without additional modeling. i.e. The DS modeling function is simply an identity map. Genre information was used as in the above case.

8.4 Experimental Technique

We employed the experimental technique suggested in [38] to generate the training and testing datasets for both experiments:

- **Testing Set:** 10% of users were randomly selected; for each user in this set, ratings for 5 randomly selected movies were withheld. These user-item pairs constituted the testing set.

- **Training Set:** The remaining user-item pairs constituted the training set.

This process was repeated 10 times thus yielding 10 splits of *MovieLens*, viz., the testing sets $D_\ell^{(\text{Test})}$ and training sets $D_\ell^{(\text{Train})} = \text{MovieLens} \setminus D_\ell^{(\text{Test})}$ (or $D_\ell^{(\text{Train})} = \text{DS-MovieLens} \setminus D_\ell^{(\text{Test})}$ for *DS-MovieLens*), for $\ell = \overline{1, 10}$. These datasets were generated prior to conducting the experiments and the same 10 splits were used throughout the entire process of experiments. Results provided are the averaged values for all the 10 datasets. For determining user-user similarity, we used $\gamma = 10^{-4}$ throughout the experiments.

8.5 Comparison

8.5.1 Comparison on “Crisp Dataset”

For a fair comparison, we interpret (a) CoFiDS predictions as “hard,” (b) retain them as “soft” and interpret the CORR and NA predictions as “soft.” We use Overall MAE (and DS-MAE in (b)) as the main comparison criterion and tune all the algorithms (*refer Section 7*) on that.

Interpreting CoFiDS Predictions as “Hard” Decisions

We used the pignistic transformation to generate “hard” decisions from the CoFiDS predictions. For a movie recommendation scenario, one is particularly interested in

precision as well. *e.g. what fraction of total predictions are accurate ?*. Thus, We also compared the algorithms via *Precision*.

Retaining CoFiDS Predictions as “Soft” Decisions

The predictions of CORR are not necessarily integer-valued. Hence, to interpret a CORR prediction \hat{r}_{ik} as a “soft” decision, we allocated the following DS theoretic BPA:

$$\hat{m}_{ik}(A) = \begin{cases} \lceil \hat{r}_{ik} - \hat{r}_{ik}; & \text{for } A = \lfloor \hat{r}_{ik} \text{ when } \hat{r}_{ik} \notin \Theta_{\text{pref}}; \\ \hat{r}_{ik} - \lfloor \hat{r}_{ik}; & \text{for } A = \lceil \hat{r}_{ik} \text{ when } \hat{r}_{ik} \notin \Theta_{\text{pref}}; \\ 1, & \text{for } A = \hat{r}_{ik} \text{ when } \hat{r}_{ik} \in \Theta_{\text{pref}}; \\ 0, & \text{otherwise.} \end{cases} \quad (8.2)$$

where $\lfloor \hat{r}_{ik}$ and $\lceil \hat{r}_{ik}$ denote the highest integer ratings not more than and the lowest integer rating not less than the CORR prediction \hat{r}_{ik} , respectively.

As an example, suppose the ratings belong to $\Theta_{\text{pref}} = \{1, 2, 3, 4, 5\}$. We interpret a CORR prediction of 3.3 as the “Bayesian” statement, “The rating is 3 with 70% confidence, and it is 4 with 30% confidence;” (8.2) corresponds well with this typical interpretation of a CORR prediction.

It is difficult to interpret the NA predictions as “soft” because they are necessarily integer-valued. Therefore, NA predictions were not included in this comparison. Now that all predictions are in “soft” form and “actual ratings” are “crisp” one can use the proposed DS-MAE measure and other measures in [35].

8.5.2 Comparison on *DS–MovieLens*

Here the actual preferences are “soft” information. Thus, we use the measure *DS–PE1* as the main criterion for performance evaluation. CoFiDS predictions are indeed “soft,” but CORR predictions are not. Thus we convert the CORR predic-

tions to “soft” via the model in Definition 8.2. Again, we show the comparison on pignistic probability transformed predictions via $DS-PE2$. We observe that CoFiDS outperforms $CORR$ in this case as well.

8.6 Results

8.6.1 Experiments on *MovieLens*

Interpreting CoFiDS Predictions as “Hard” Decisions

Algorithm	Metric	True Rating					Overall
		1	2	3	4	5	MAE Mean \pm Var%
CoFiDS	MAE	2.0080	1.2506	0.6383	0.2443	1.0504	0.7374
	Precision	0.6364	0.2731	0.3576	0.4113	0.6827	$\pm 0.13\%$
	Recall	0.0254	0.1215	0.3675	0.7736	0.0696	
	F_1	0.0488	0.1682	0.3625	0.5371	0.1263	
CORR	MAE	1.8143	1.2370	0.8220	0.6144	0.9020	0.8725
	Precision	0.2957	0.2211	0.3158	0.3994	0.4158	$\pm 0.45\%$
	Recall	0.1283	0.2038	0.3254	0.4800	0.3442	
	F_1	0.1789	0.2121	0.3205	0.4360	0.3766	
u-NA	MAE	2.0217	1.2094	0.6450	0.3418	1.1071	0.7838
	Precision	0.0000	0.2632	0.3428	0.3953	0.3962	$\pm 0.25\%$
	Recall	0.0000	0.1740	0.3710	0.6902	0.0548	
	F_1	0.0000	0.2095	0.3563	0.5027	0.0963	
i-NA	MAE	2.1776	1.5170	0.8259	0.4995	0.8641	0.8783
	Precision	0.0000	0.1691	0.3627	0.3849	0.3373	$\pm 0.18\%$
	Recall	0.0000	0.1187	0.3179	0.5464	0.3048	
	F_1	0.0000	0.1395	0.3388	0.4516	0.3202	
c-NA	MAE	1.9485	1.2514	0.7233	0.4745	0.9248	0.8115
	Precision	0.0000	0.2435	0.3589	0.4000	0.3863	$\pm 0.10\%$
	Recall	0.0000	0.1938	0.3676	0.5717	0.2506	
	F_1	0.0000	0.2158	0.3632	0.4707	0.3040	

Table 8.2: Performance Comparison with “Hard” Decisions

Table 8.2 compares CoFiDS with CORR and NA using four performance measures; the last column depicts the overall MAE along with its variance (as a percentage). For

each ACF algorithm, the configuration that yields the best overall MAE was used for the comparison carried out in Table 8.2 [61]. For CoFiDS, we used $\{\alpha, \sigma\} = \{0.9, 2/3\}$.

Remark. Coverage, which measures the percentage of items for which the recommendation system can make predictions, tends to be lower for CORR when it is tuned for a lower MAE; both NA and CoFiDS however provide almost 100% coverage. So, for a fairer comparison, for CORR, we used a configuration that minimizes MAE while having a 90% level of coverage.

Bold values in Table 8.2 indicate the best performance measures in each category. Even with the rather simple DS modeling function and the coarse system-wide parameters, CoFiDS shows a significant improvement over CORR and NA. Bear in mind that this is in spite of the fact that conversion to a “hard” decision cannot exploit the full strength of the DS theoretic basis of CoFiDS.

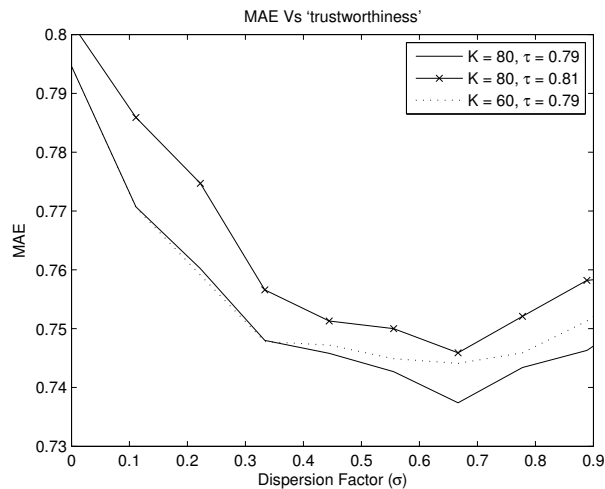


Figure 8.2: Variation of MAE of CoFiDS with dispersion factor σ for several combinations of $\{K, \tau\}$. Here, CoFiDS $\alpha = 0.9$.

With $\alpha = 0.9$, which corresponds to a 90% trust for each user rating, Fig. 8.2 depicts the variation of MAE of CoFiDS with the dispersion factor σ for several combinations of $\{K, \tau\}$.

One is particularly interested on the “Effect of Neighborhood Size and Similarity Threshold” on performance. Thus, with all other parameters held constant, Figure 8.3 and Figure 8.4 shows the variation of MAE of CoFiDS with neighborhood size K and similarity threshold τ respectively.

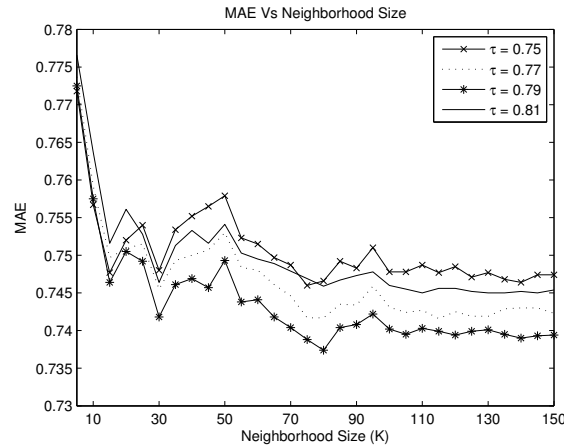


Figure 8.3: Variation of MAE of CoFiDS with neighborhood size K .

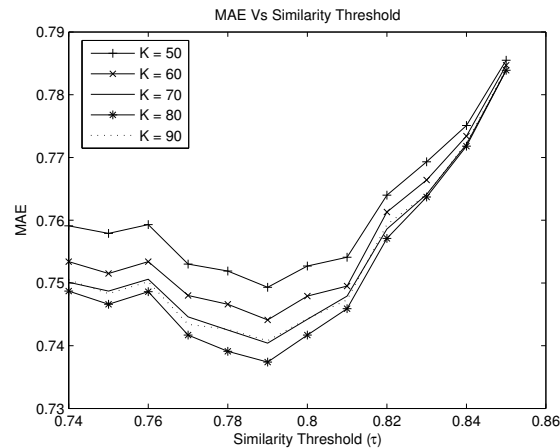


Figure 8.4: Variation of MAE of CoFiDS with similarity threshold τ .

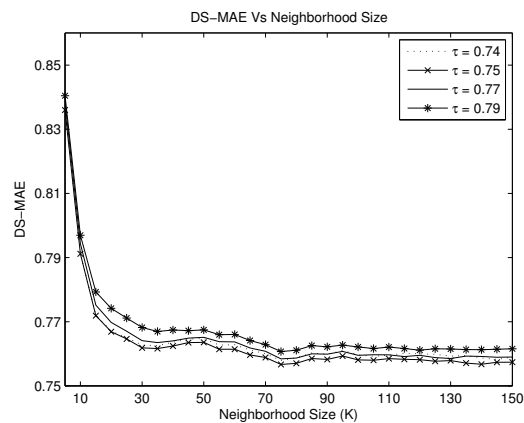
Retaining CoFiDS Predictions as “Soft” Decisions

Here we compare the algorithms DS theoretic versions of the performance measures. Table 8.3 compares CoFiDS with CORR using the DS theoretic versions of

Algorithm	DS -Metric	True Rating					Overall MAE Mean
		1	2	3	4	5	
		<hr/>					
CoFiDS	MAE	2.0119	1.2892	0.6570	0.2581	1.0590	0.7567
	Precision	0.5302	0.2773	0.3427	0.4030	0.6210	
	Recall	0.0353	0.1129	0.3529	0.7595	0.0660	
	F_1	0.0661	0.1605	0.3478	0.5266	0.1192	
<hr/>							
CORR	MAE	1.8109	1.2537	0.8581	0.6544	0.8907	0.8951
	Precision	0.2861	0.2110	0.3083	0.3994	0.3941	
	Recall	0.1618	0.2009	0.3128	0.4471	0.3678	
	F_1	0.2067	0.2059	0.3105	0.4219	0.3805	

Table 8.3: Performance Comparison with “Soft” Decisions

the performance measures that were used in Table 8.2. For each ACF algorithm, the configuration that yields the best overall DS-MAE was used for the comparison carried out in Table 8.3. We continued to use same $\{\alpha, \sigma\}$ pairs. Again, bold values in Table 8.3 indicate the best performance measures in each category.

Figure 8.5: Variation of DS-MAE of CoFiDS with neighborhood size K .

The effects of neighborhood size and similarity threshold on performance are drawn in Figure 8.5 and Figure 8.6 respectively.

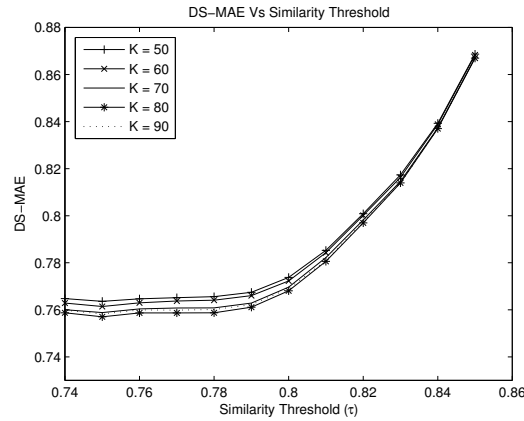


Figure 8.6: Variation of DS-MAE of CoFiDS with similarity threshold τ .

8.6.2 Evaluations on *DS – MovieLens*

User actual ratings are “soft” in the *DS–MovieLens* dataset. Thus, the performance comparison is done based on the *DS–PE1* and *DS–PE2* as in 20 and 21.

Algorithm	Metric	Zero tolerance user selection probability p				
		0.1	0.3	0.5	0.7	0.9
CoFiDS	<i>DS-PE1</i>	0.4498	0.4903	0.5220	0.5496	0.5566
	<i>DS-PE2</i>	0.5657	0.6265	0.6818	0.7361	0.7669
CORR	<i>DS-PE1</i>	0.5513	0.5779	0.5804	0.5906	0.6125
	<i>DS-PE2</i>	0.6946	0.7529	0.7740	0.8060	0.8550

Table 8.4: Performance Comparison on *DS–MovieLens*

CORR predictions were converted back to “soft” form via 8.2. The comparison is carried out in Table 8.4 for several different values of p — probability with which the zero tolerance user was selected; the other 3 user profiles were selected with equal probability.

Figures 8.7 and 8.8 show plots of variations of *DS–PE1* and *DS–PE2* of CoFiDS

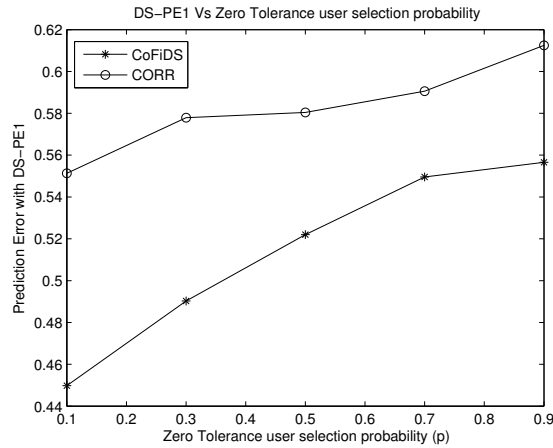


Figure 8.7: CoFiDS with $\{\alpha, \sigma\} = \{0.9, 2/3\}$: $DS-PE1$ versus p .

and CORR with the zero tolerance user selection probability p respectively. Error in CORR is always higher than that of CoFiDS, even though both of them follow a similar trend.

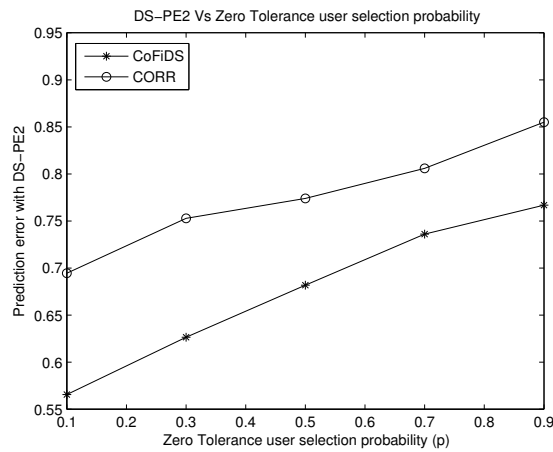


Figure 8.8: CoFiDS with $\{\alpha, \sigma\} = \{0.9, 2/3\}$: $DS-PE2$ versus p .

Figure 8.9 shows how $DS-PE1$ varies with the changing neighborhood size K . Here $p = 0.1$ is fixed at 0.1. This is typical for a “Error Vs K ” plot, where the error decreases with increasing neighborhood size and then at a certain point it starts increasing again.

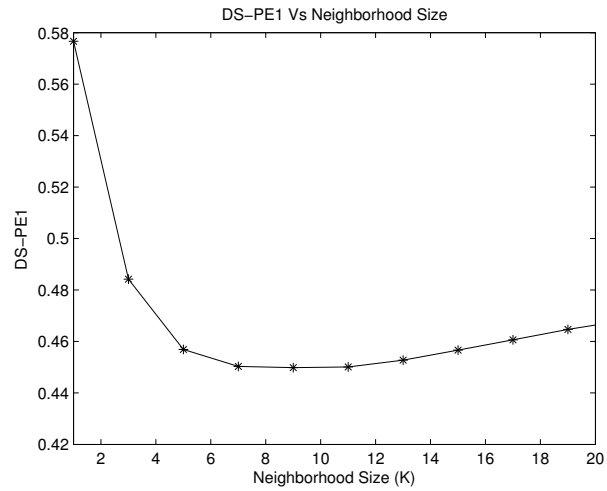


Figure 8.9: CoFiDS with $\{\alpha, \sigma\} = \{0.9, 2/3\}$: $DS-PE1$ versus K when $p = 0.1$.

CHAPTER 9

Conclusion and Future Research

9.1 Conclusion

Most of the information systems in existence today are based on the assumption that the information is perfect. These methods are not capable of accurately modeling and working with imperfect, real-world data. This has resulted in technologies that possess elegant models but do not represent reality and produce unreliable outputs.

DS theory has been in the focus of many researchers working in different disciplines, and it has proven to produce better and reliable results in critical and important applications when in the presence of data imperfections. Few attempts have been made to apply DS theory to information systems for representing various types of data imperfections in rule mining and classification problems.

The most important contribution of the work presented in this thesis is to introduce DS theoretic notions for use in recommender systems, in particular, for ACF. We utilized a DS theoretic data model that allows one to accommodate imperfections and then propagate these throughout the decision-making process, thus producing more reliable decisions. We have shown how different user preferences could be captured via simple DS theoretic models.

In the last two decades, ACF has gained a tremendous significance among other competing recommender systems as one of the most successful recommendation strategies. ACF has been applied to various problem domains, especially to e-commerce applications. The nature of ACF, difficulties in evaluating and understanding the generated predictions, and lack of proper methods for working with imperfect data, have indeed restricted these algorithms to much simpler and low-risk application domains. On the other hand, the ACF's strength of being able to work with extremely sparse, different types of information, make it an ideal candidate for critical and sensitive applications, such as medical expert support systems, homeland security and surveillance etc. However, before ACF's utility in such applications, it is imperative that techniques are put in place that enable ACF to perform in the presence of various types of data imperfections that are inherent in most sensitive applications.

The ACF algorithm presented in this thesis, CoFiDS, has the ability to conveniently represent a wide variety of data imperfections, such as probabilistic uncertainties, qualitative aspects of evidence, evidence ambiguities, missing information, etc., using DS theoretic notions. Apart from the convenience it offers in representation, DS theoretic models are better in capturing partial or incomplete knowledge, and are more robust against modeling errors. These models are based on a solid mathematical foundation. Moreover, CoFiDS being able to provide "soft" predictions, enables the end-user to make decisions with full recognition and understanding of the reliability of the generated predictions. Thus, the work we present here widens the applicability of ACF, and make it possible to apply ACF to critical and sensitive domains with confidence.

In ACF, most of the existing algorithms use only co-rated items (or users) for comparing users (or items). This approach has the weakness of generating similarities based on a small subset of total items (or users) space, thus has a risk of generating

misleading neighbors. In most of these application domains, contextual information is available which can be very useful in improving predictions. Indeed, there are various algorithms making separate predictions using these information and combining them at a latter stage with ACF predictions. But, we have proposed a unified method to fuse contextual information from the very outset and integrate them into the same ACF framework. This method is more consistent and allows the ACF algorithm to make use of extra information to identify neighbors with improved accuracy.

Even though the use of DS theoretic notions allow significant improvements, such as ability to work with imperfections, robustness against modeling errors and “transparent predictions;” it requires extensive computational requirements. These are prohibitively large when one works on the cross-product space, especially with the larger dimensional datasets used in ACF. Thus, in our similarity computation stage — where we have to work on cross-product space, the user-BoEs are compared after projecting them to probability space. We have obtained significant computational complexity reductions via this conversion along with an attractive distance metric defined for probability mass functions.

ACF performance evaluations have always been a subtle issue due to weaknesses in existing measures, and various other reasons related to recommendations. Now that the proposed method output being in “soft” format, this problem is further complicated. There are no other ACF algorithms providing predictions as belief functions, neither in a “soft” format. So, no direct comparisons are possible. So, we took different directions for comparison with existing algorithms, by transforming “soft” predictions to “crisp” and vice versa. We have proposed new basic measures to be used with algorithms such CoFiDS, providing rich predictions. Method presented in this thesis was extensively tested on a benchmark dataset comparing to existing popular algorithms. There are no benchmark “soft” datasets in use by ACF commu-

nity. So, we have created a synthetic dataset using popular and widely used powerset method, for evaluation of CoFiDS on a pure “soft” data domain. On all these datasets, CoFiDS outperforms correlation based method [38] and the method proposed by Nakamura and Abe [16].

9.2 Future Research

ACF is being used extensively in e-commerce applications today. Thus, most of the current work is mainly focused on improving performance in existing algorithms and new application development related to these areas. Even though, there has been few attempts in different directions, less attention has been paid on how to accommodate and handle imperfections that could potentially improve the quality of the recommendation process. On the other hand, research in uncertainty reasoning domain has shown significant progress with the development of many tools and methodologies to represent and manipulate imperfections in information. But there have been only a few attempts to couple reasoning under the presence of imperfection to information systems.

This lack of methodologies and datasets have hindered our ability for a fair comparison of CoFiDS to existing algorithms. This ignorance has manifested as, no developed or accepted measures for comparison of ACF algorithms capable of producing predictions with such richer information contents. Development of such meaningful measures for comparison of CF predictions, is an important research problem to be undertaken.

CoFiDS uses DRC for evidence combination. Some critics have shown that DCR has some profound weaknesses when combining BoEs with conflicting evidence. Even though CoFiDS combines only users that have been found to be similar, there can be instances where such a combination of conflicting evidences occur. This combination

of evidence of neighbors with conflicting evidence, is another important area for ACF research, especially in the presence of data imperfections. In particular, CoFiDS has to be extensively checked using different combination rules proposed in DS literature.

Existing recommenders only allow users to rate the items via a crisp rating space. Now, with more advanced ACF algorithms such as CoFiDS, one can model more richer user preferences for accurate and improved predictions. Thus, the development of sophisticated but user-friendly interfaces allowing users to indicate their actual preference, would be a good research topic to be undertaken.

CoFiDS predictions are in “soft” format. This allows the end-user to make the final decision him/herself, with the full recognition of reliability of it’s outputs. But, a commercial recommender system should provide a user-friendly interface. Showing the resulting mass structure to the end-users will not be the ideal method one should undertake. Thus, methodologies should be developed for interpreting the resulting mass structures and presenting them in a more user-friendly, easy to understand manner, especially if one wish to apply these technologies to domain where end-users are not very competent in evidence theory.

Bibliography

- [1] M. Smithson, *Ignorance and Uncertainty: Emerging Paradigms*. Springer-Verlag, 1989.
- [2] A. Motro, “Sources of uncertainty in information systems,” in *Proc. Second Workshop Uncertainty Management and Information Systems: From Needs to Solutions, Catalina, Calif.*, vol. 2032, July 1993, pp. 266–279.
- [3] D. Billsus and M. J. Pazzani, “Learning collaborative information filters,” in *Proc. International Conference on Machine Learning (ICML’98)*. San Francisco, CA: Morgan Kaufmann, 1998, pp. 46–54.
- [4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.
- [5] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, June 2005.
- [6] U. Shardanand and P. Maes, “Social information filtering: algorithms for automating “word of mouth”,” in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 1995, pp. 210–217.
- [7] D. Maltz and K. Ehrlich, “Pointing the way: active collaborative filtering,” in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 1995, pp. 202–209.
- [8] J. L. Herlocker, “Understanding and improving automated collaborative filtering systems,” Ph.D. dissertation, University of Minnesota, 2000.
- [9] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.
- [10] J. B. Schafer, J. A. Konstan, and J. Riedl, “E-Commerce recommender applications,” *Data Mining and Knowledge Discovery*, vol. 5, no. 1/2, pp. 115–152, 2001.

- [11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proc. ACM Conference on Electronic Commerce*, Minneapolis, MN, 2000, pp. 158–167.
- [12] D. M. Fleder and K. Hosanagar. (2007) Blockbuster culture's next rise or fall: the impact of recommender systems on sales diversity. Rockville, MD. [Online]. Available: <http://ssrn.com/paper=955984>
- [13] P. Symeonidis, A. Nanopoulos, A. Papadopoulos, and Y. Manolopoulos, "Collaborative filtering process in a whole new light," in *Proc. International Database Engineering and Applications Symposium (IDEAS'06)*, 2006, pp. 29–36.
- [14] M. Kunaver, T. Pozrl, M. Pogacnik, and J. Tasic, "Optimisation of combined collaborative recommender systems," *International Journal of Electronics and Communications*, vol. 61, no. 7, pp. 433–443, July 2007.
- [15] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 56–69, Jan. 2004.
- [16] A. Nakamura and N. Abe, "Collaborative filtering using weighted majority prediction algorithms," in *Proc. International Conference on Machine Learning (ICML'98)*. San Francisco, CA: Morgan Kaufmann, 1998, pp. 395–403.
- [17] P. Li and S. Yamada, "A movie recommender system based on inductive learning," in *Proc. IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, Dec. 2004, pp. 318–323.
- [18] S. Kuwata and N. Ueda, "One-shot collaborative filtering," in *Proc. IEEE Symposium on Computational Intelligence and Data Mining (CIDM'07)*, Mar./Apr. 2007, pp. 300–307.
- [19] C. Christakou, L. Lefakis, S. Vrettos, and A. Stafylopatis, "A movie recommender system based on semi-supervised clustering," in *Proc. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 2, Nov. 2005, pp. 897–903.
- [20] J. Salter and N. Antonopoulos, "CinemaScreen recommender agent: combining collaborative and content-based filtering," *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 35–41, Jan./Feb. 2006.
- [21] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proc. National Conference on Artificial Intelligence*, July 2002, pp. 187–192.
- [22] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proc. ACM Conference on Computer Supported Cooperative Work*, 2000, pp. 241–250.

- [23] H.-W. Tung and V.-W. Soo, "A personalized restaurant recommender agent for mobile e-service," in *Proc. IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, 2004, pp. 259–262.
- [24] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *Proc. ACM Symposium on Applied Computing*, Santa Fe, NM, 2005, pp. 1589–1593.
- [25] L. Brozovsky and V. Petricek, "Recommender system for online dating service," in *Proc. Znalosti 2007 Conference*, Ostrava, Czech Republic, 2007.
- [26] P. De Meo, G. Quattrone, G. Terracina, and D. Ursino, "An XML-based multiagent system for supporting online recruitment services," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, no. 4, pp. 464–480, July 2007.
- [27] J. Wang, J. Pouwelse, R. L. Lagendijk, and M. J. T. Reinders, "Distributed collaborative filtering for peer-to-peer file sharing systems," in *Proc. ACM Symposium on Applied Computing (SAC'06)*, 2006, pp. 1026–1030.
- [28] E. Aimeur, F. S. M. Onana, and A. Saleman, "HELP: A recommender system to locate expertise in organizational memories," in *Proc. IEEE/ACS International Conference on Computer Systems and Applications (AICCSA'07)*, May 2007, pp. 866–874.
- [29] S. Y. Jung, J.-H. Hong, and T.-S. Kim, "A statistical model for user preference," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 834–843, June 2005.
- [30] S. Parsons, "Current approaches to handling imperfect information in data and knowledge bases," *Knowledge and Data Engineering*, vol. 8, no. 3, pp. 353–372, 1996.
- [31] F. J. Palella, K. M. Delaney, A. C. Moorman, M. O. Loveless, J. Fuhrer, G. A. Satten, D. J. Aschman, and S. D. Holmberg, "Declining morbidity and mortality among patients with advanced human immunodeficiency virus infection. HIV outpatient study investigators," *The New England Journal of Medicine*, vol. 338, no. 13, pp. 853–860, Mar. 1998.
- [32] AIDSinfo. U.S. Department of Health and Human Services. Rockville, MD. [Online]. Available: <http://www.aidsinfo.nih.gov/>
- [33] D. T. Jayaweera, M. A. Kolber, M. Brill, T. Tanner, and R. Campo, et al, "Effectiveness and tolerability of a once-daily amprenavir/ritonavir-containing highly active antiretroviral therapy regimen in antiretroviral-naive patients at risk for non-adherence: 48-week results after 24 weeks of directly observed therapy," *HIV Medicine*, vol. 5, no. 5, pp. 364–370, Sept. 2004.

- [34] J. Gathe, D. A. Cooper, C. Farthing, D. Jayaweera, and D. Norris, et al, “Efficacy of the protease inhibitors tipranavir plus ritonavir in treatment-experienced patients: 24-week analysis from the RESIST-1 trial,” *Clinical Infectious Diseases*, vol. 43, no. 10, pp. 1337–1346, Oct. 2006.
- [35] K. K. R. G. 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 Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 37, no. 6, pp. 1446–1459, Dec. 2007, to appear.
- [36] P. Smets, “Practical uses of belief functions,” in *Proc. Conference on Uncertainty in Artificial Intelligence*, 1991, pp. 612–621.
- [37] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in *World Wide Web*, 2001, pp. 285–295.
- [38] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Proc. Annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 230–237.
- [39] P. Cheeseman, “In defense of probability,” *Int. Conf. on Artificial Intelligence*, vol. 189, no. 6, pp. 1002–1009, 1985.
- [40] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*. Norwood, MA: Artech House Inc, 1992.
- [41] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
- [42] M. Schervish, *Theory of Statistics*. New York, NY: Springer-verlag, 1995.
- [43] P. Varshney, *Distributed Detection and Data Fusion*. New York, NY: Springer-verlag, 1997.
- [44] L. Zadah, “Fuzzy sets as a basis for a theory of possibility,” *Fuzzy sets and systems*, vol. 1, no. 3-28, 1978.
- [45] D. Dubois and H. Prade, “An introduction to possibilistic and fuzzy logics,” in *Non-Standard Logics for Automated Reasoning*. London: Academic Press, 1988, pp. 287–326.
- [46] Dubois and Prade, *Possibility Theory : An Approach to Computerized Processing of Uncertainty*. New York: Planum Press, 1988.
- [47] G. Cooman, D. Ruan, and E. Kerre, *Foundations and Applications of Possibility Theory*, ser. Advances in Fuzzy Systems - Applications and Theory, R. D. Cooman G and K. E, Eds. Singapore: World Scientific, 1995, vol. 8.

- [48] D. Dubois and H. Prade, “Possibility theory as a basis for qualitative decision theory,” in *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI 95)*, Montreal, August 1995, pp. 1924–1930.
- [49] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.
- [50] P. Vannoorenberghe, “On aggregating belief decision trees,” *Information Fusion*, vol. 5, no. 3, pp. 179–188, Sept. 2004.
- [51] P. Smets, “Practical uses of belief functions,” in *Proc. Conference on Uncertainty in Artificial Intelligence (UAI’99)*, K. B. Laskey and H. Prade, Eds. San Francisco, CA: Morgan Kaufmann, 1999, pp. 612–621.
- [52] F. Delmotte and P. Smets, “Target identification based on the transferable belief model interpretation of Dempster-Shafer model,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 4, pp. 457–471, July 2004.
- [53] D. A. Dewasurendra, P. H. Bauer, and K. Premaratne, “Evidence filtering,” *IEEE Transactions on Signal Processing*, 2007, to appear.
- [54] GroupLens Research. (2007) MovieLens data sets. Department of Computer Science and Engineering, University of Minnesota. Twin Cities, MN. [Online]. Available: <http://www.grouplens.org/taxonomy/term/14>
- [55] M. G. Vozalis and K. G. Margaritis, “Using SVD and demographic data for the enhancement of generalized collaborative filtering,” *Information Sciences: An International Journal*, vol. 177, no. 15, pp. 3017–3037, Aug. 2007.
- [56] J. Cho, K. Kwon, and Y. Park, “Collaborative filtering using dual information sources,” *Intelligent Systems*, vol. 22, no. 3, pp. 30–38, May/June 2007.
- [57] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, “Combining collaborative filtering with personal agents for better recommendations,” in *Proc. National Conference on Artificial Intelligence and Innovative Applications of Artificial Intelligence*, 1999, pp. 439–446.
- [58] H. Chan and A. Darwiche, “A distance measure for bounding probabilistic belief change,” *International Journal of Approximate Reasoning*, vol. 38, no. 2, pp. 149–174, Feb. 2005.
- [59] I. Bloch, “Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account,” *Pattern Recognition Letters*, vol. 17, no. 8, pp. 905–919, July 1996.
- [60] E. Hullermeier and J. Furnkranz, “Learning label preferences: Ranking error versus position error,” in *Advances in Intelligent Data Analysis VI, Proc. International Symposium on Intelligent Data Analysis (IDA’05)*, ser. Lecture Notes in

Computer Science, A. F. Famili, J. N. kok, J. M. Pena, A. Siebes, and A. Feelders, Eds. Berlin, Germany: Springer-Verlag, 2005, vol. 3646, pp. 180–191.

- [61] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, Jan. 2004.
- [62] A. L. Jousselme, D. Grenier, and E. Bosse, “A new distance between two bodies of evidence,” *Information 2*, 2001.
- [63] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.