Université de Tunis

Institut Supérieur de Gestion

MASTER RECHERCHE

Spécialité

Sciences et Techniques de l'Informatique Dcisionnelle (STID)

Option

Informatique et Gestion de la Connaissance (IGC)

TOWARDS A NEW POSSIBILISTIC COLLABORATIVE FILTERING APPROACH

Manel Slokom

Nahla Ben Amor	Professeur, ISG Tunis	Directeur du mémoire
Raouia Ayachi	Maitre assistante, ESSEC Tunis	Co-directeur

2014-2015

Laboratoire/Unité de recherche: LARODEC

Acknowledgments

In the end of this work, I would like to express my sincere thanks to my advisor Pr. Nahla Ben Amor for her guidance and help throughout my studies and for valuable advice.

I offer my deep gratitude to my master thesis adviser Mme Raouia Ayachi for her willingness and determination to supervise this work through her suggestions for improvement and her continuous encouragement and precious comments.

Thank you for your endless patience and valuable remarks on chapter drafts and for your great efforts to explain things clearly and smoothly. May this modest work satisfy you and express my gratitude to the help and advice you have given me.

I am also grateful to all the professor of *Institut Supérieur de Gestion de Tunis* for providing me the background that i need for study and research.

I would like to thank all my family, especially my dear father, my beloved mother, my sisters and my brother for giving me strength and motivation, for creating a pleasant environment at home and for supporting me during difficult periods. I dedicate this work to them because it would not have been possible without their support.

Last and not least, I would also like to thank my friends especially my dear Thierno Kanté who helped me a lot in finalizing this project within the limited time frame.

Contents

Introduction **1**

1 Basics on recommendation and possibility theory framework **3**

1.1 Introduction 3

1.2 Recommender systems 3

1.2.1 Notations and definitions 5

1.2.2 Recommendation approaches 5

1.2.3 Collaborative filtering approach 6

1.3 Possibility theory 14

1.3.1 Notations 14

1.3.2 Possibility distribution 14

1.3.3 Inconsistency 16

1.3.4 Possibility and necessity measures 16

1.4 Conclusion 17

2 Preferences-based recommendation **18**

2.1 Introduction 18

2.2 Certain preferences-based recommendation 18

2.3 Uncertain preferences-based recommendation 21

2.4 Conclusion 23

3 New possibilistic item-based collaborative filtering recommender **24**

3.1 Introduction 24

3.2 Proposed possibilistic CF recommender 24

3.2.1 Preferences representation 26

3.2.2 Similarity computation 27

3.2.3 Prediction and recommendation 30

3.3 Conclusion 31

4 Experimental study **32**

4.1 Introduction 32

4.2 Experimental protocol 32

4.2.1 Data sets 32

4.2.2 Evaluation metrics 33

4.2.3 Implementation 35

4.3 Experimental results 35

4.4 Conclusion 38

Conclusion **39**

References **41**

List of Figures

1.1 The recommender system process (Bobadilla, Ortega, Hernando, & Gutiérrez, 2013). 4

1.2 The user to user collaborative filtering. 7

1.3 The item to item collaborative filtering. 7

2.1 The preference Order (L. Yu & Yang, 2008). 19

2.2 (a) a utility function, (b) a corresponding preference relation (Brun et al., 2010). 20

2.3 The components of PMCF (K. Yu et al., 2004). 22

2.4 The datasets after and before a clustering step (Xiang et al., 2013). 23

3.1 The possibilistic item-based collaborative filtering process: ΠCF. 25

3.2 Isolation of the co-rated items and similarity computation. 28

4.1 A capture screen showing the Top-K recommendation on the MovieLens data. 36

4.2 Evaluation of classification accuracy metrics: Precision, recall and F-measure. 37

List of Tables

1.1 The users' ratings matrix 9

3.1 User-item preferences matrix 26

4.1 Classification of the possible result of a recommendation of an item to a user. 34

4.2 The MAE and NMAE accuracy 36

4.3 The precision, recall and F-measure of the two approaches 37

List of Algorithms

3.1 Similarity computation 29
3.2 Prediction and recommendation 30

Introduction

Recommender systems have emerged in the past several years as an efficient tool to deliver users with more intelligent and proactive information service. Such systems apply knowledge discovery techniques to personalize recommendations provided for each user. Recommendation systems use several algorithms to help us sort through the masses of information to find the ”good product” in a very personalized way. They recommend products or services that well fit to the learned users' preferences and needs. These systems generally combine, on one hand, information extracted from users' profiles and social interactions, and on the other hand, machine learning techniques that are used to predict the user's ratings or preferences.

Several types of recommenders have been proposed that can be categorized into three major categories (Ricci, Rokach, Shapira, & Kantor, 2011) namely *content-based filtering*, *collaborative filtering*, and *hybrid approaches*. In this work, we focus on the most popular approach (Ekstrand, Riedl, & Konstan, 2011), the collaborative filtering, which predicts the user's interest for a given item based on a collection of users profiles. Collaborative filtering algorithms are divided into two main categories, namely *memory*-based and *model*-based. The first category focuses on the entire collection of previously rated items, while the model-based one uses a model learned from the collection of ratings to make predictions. In this work, we are in particular interested in memory-based approaches based on user-item matrix representing users preferences.

Collaborative filtering gives good results in a certain context, in which ratings provided by users are known with certainty. This does not reflect the reality, which is related to uncertainty and imprecision by nature. Consequently, the recommendation results are deeply affected if uncertainty is not considered. So, a good recommender should be able to suggest items even when information about ratings are imperfect. To improve the recommender's accuracy, recent few researches have introduced uncertainty in the recommendation process. In fact, authors in (Samia, Allel, & Aicha, 2014) modeled uncertain users preferences by means of fuzzy set theory. They handled evolution of preferences by taking into account the temporal dynamics of users preferences. Also, other research works (K. Yu, Schwaighofer, Tresp, Xu, & Kriegel, 2004) have studied a probabilistic memory-based collaborative filtering (*PMCF*) framework. In *PMCF*, authors used a generative probabilistic density to model preference profiles. Then, a posterior distribution of user's preferences is to predict an active user's preferences. In addition, authors (Price & Messinger, 2005) have proposed a new approach for recommender systems that optimize decision making. This approach produces on the one hand a diversity of alternatives in the recommendation set by applying a specific formulation ”*expected utility*” and on the other hand, covers the uncertainty over possible user preferences using *probability distributions*. More recently, another research (Xiang, Guisheng, Long, & Yongjin, 2013) proposed a

new method of CF based on uncertain user interests clusters where uncertainty appears in the form of a trustworthy degree to measure the rationality of clustering algorithm results. However to the best of our knowledge, no research studied a purely uncertain recommender system dealing with uncertain ratings as an input.

In this report, we will develop a new collaborative filtering recommender under uncertainty, which uses the possibility theory framework in order to cope with the uncertainty that may pervade users ratings. In fact, our approach, denoted by ΠCF, is based on three steps, namely:

1. *preferences representation* handles users preferences using possibility distributions.
2. *similarity computation* calculates the similarity between items using a possibilitic similarity measure, namely Information Affinity.
3. *prediction and recommendation* predicts the estimated preference of a target user towards an item he has never seen before. Then, it generates a list of top K items for the target user using most similar items.

This report is organized as follows: Chapter 1 briefly presents basic concepts related to recommender system and the possibility theory framework. Chapter 2 is dedicated to preferences-based research works. Chapter 3 details our proposed possibilistic collaborative filtering approach ΠCF. Finally, chapter 4 presents experimental results evaluating the performance of our proposed approach.

Chapter 1

Basics on recommendation and possibility theory framework

1.1 Introduction

Recommendation plays an increasingly important role in our daily lives. Recommender systems may be helpful for users that are choosing between a large number of items and aren't able to browse information about all available items (Cvengroš, 2011). Thus recommender systems give good results when ratings provided by users are known with certainty. But this does not reflect the reality, which is related to uncertainty and imprecision by nature. Consequently, the recommendation results are deeply affected if uncertainty is not considered. This uncertainty that exists is not studied in previous recommender systems researches. Ignoring uncertainty puts the modeling in a less realistic setting, and the resulting model does not precisely represent the reality (Zenebe & Norcio, 2009).

This chapter will be devoted to basics on both recommendation and possibility theory. Section 1.2 presents recommendation systems and focuses in particular on collaborative filtering approach and Section 1.3 gives an overview on possibility theory framework.

1.2 Recommender systems

Recommender systems (RSs) are computer-based techniques used to reduce information overload and provide recommendations of items (e.g., books, songs, etc.) based on expectations and tastes of users. RSs are very helpful for indecisive users, who are not able to browse the whole information of items and choose the most appropriate ones. This results in a decrease of the search time and an increasing of user's satisfaction (Brun, Hamad, Buffet, & Boyer, 2010).

In this section, we will first give some notations and definitions relative to recommender systems. Then, we will briefly present existing recommendation approaches with a particular focus on collaborative filtering. Figure 1.1

illustrates the recommender system process.

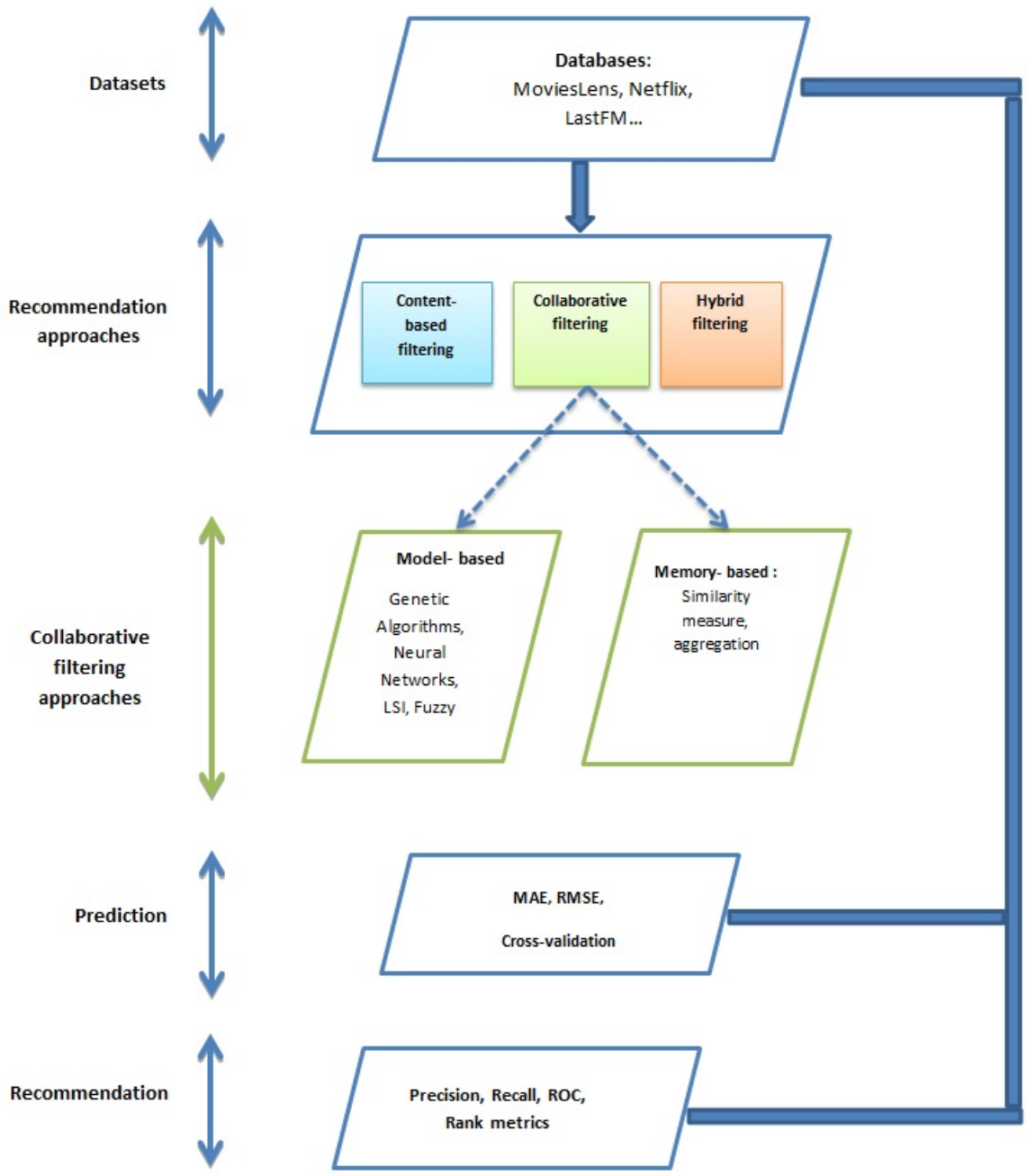


Figure 1.1: The recommender system process (Bobadilla, Ortega, Hernando, & Gutiérrez, 2013).

1.2.1 Notations and definitions

- $U = \{u_1, u_2, ..., u_m, ..., u_M\}$ denotes the set of users in the system where M is the number of distinct users.
- $I = \{i_1, i_2, ..., i_n, ..., i_N\}$ denotes the set of items in the system where N is the number of distinct items.
- *Target user:* The user for whom the task is to find an item suggestion.
- *Target item:* The current item for which we would like to predict user's preference.
- *User profile:* a set of user's preferences.

1.2.2 Recommendation approaches

Based on the information used to perform recommendation, recommender systems generally fall into three categories: **i)** Content-based filtering, **ii)** Collaborative filtering and **iii)** Hybrid filtering.

Content-based filtering (CBF): learns to recommend items that are similar to the ones that the user liked in the past (Ricci et al., 2011). For example, in a web-based E-commerce RS, if the user purchased some fiction films in the past, the RS will probably recommend a recent fiction film that he has not yet purchased on this website.
Content-based filtering analyzes items rated by a user, and builds a user's profile based on the features of the rated objects. The profile is a structured representation of user interests, adopted to recommend new interesting items. The recommendation process basically consists in matching the attributes of the user profile and the attributes of a content object. The result is a relevance judgment that represents user's level of interest (Lops, De Gemmis, & Semeraro, 2011).

Collaborative filtering (CF): is the process of filtering or evaluating items using the opinions of other people. Intuitively, CF assumes that if user X and Y rate n items similarly or have similar behaviour (e.g. buying, watching, listening), they will rate or act on other items similarly. Therefore, CF analyzes relationships between users and interdependencies among products, in order to identify new user-item associations (Koren & Sill, 2013).

The major difference between CF and content-based recommender systems is that CF only uses the user-item ratings data to make predictions and recommendations, while content-based recommender systems rely on the features of users and items for predictions.

Hybrid filtering: A hybrid recommender system combines different recommender systems using different hybridization strategies, for example (Burke, 2002):

- *Weighted:* The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
- *Switching:* The system switches between recommendation techniques depending on the current situation.
- *Mixed:* Recommendations from several different recommenders are presented at the same time.
- *Cascade:* One recommender refines the recommendations given by another.

In this work, we will focus on collaborative filtering as it is the most widely implemented technology (Nair & Kelkar, 2013). It works well with complex objects and it proves an explainable result which is an important aspect in recommender systems.

1.2.3 Collaborative filtering approach

Collaborative Filtering approach (CF) is the most successful recommendation technique to date (Koren & Sill, 2013). The basic idea of CF-based algorithms is to provide item recommendations or predictions based on the opinions of other users.
The goal of collaborative filtering algorithm is to predict the preferences of one user, referred to as the target user, based on the preferences of a group of users. The key idea is that the target user, will prefer those items that his neighbors preferred in the past. Intuitively, CF assumes that if user X and Y rate n items similarly or have similar behavior (e.g. buying, watching, listening), they will rate or act on other items similarly. Therefore, CF analyzes relationships between users and inter-dependencies among products, in order to identify new user-item associations. Collaborative filtering first analyze the user-item preferences matrix to identify relations between different items, then use this relations to compute prediction which is a numerical value expressing the predicted likeliness of item for the target user. Finally generate a list of K items, that the target user will like the most, also known as *Top-K recommendation*.

Collaborative filtering algorithms are divided into two main categories, namely memory-based and model-based algorithms:

Model-based collaborative filtering groups together different users in the training database into a small number of classes based on their rating patterns. In order to predict the ratings of a test user on a particular item, we can simply categorize the test user into one of the predefined user classes and use the predicted class as the prediction for the test user (Jin, Si, Zhai, & Callan, 2003).
Consequently, there exist several model-based recommendation methods such as bayesian classifiers (Park, Hong, & Cho, 2007), neural networks (Roh, Oh, & Han, 2003), fuzzy systems (Yager, 2003), genetic algorithms (Linqi & Li, 2008) have been investigated.

1. *Bayesian collaborative filtering algorithms:* use a naïve bayes strategy to make prediction for CF tasks (Su & Khoshgoftaar, 2009). Assuming the features are independent given the class, the probability of a certain class can be computed, and then the class with the highest probability will be classified as the predicted class (Park et al., 2007).

2. *Clustering collaborative filtering algorithms:* A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The measurement of the similarity between objects is determined using metrics such as *Minkowski distance* and *Pearson correlation*. Clustering methods can be classified into three categories (Su & Khoshgoftaar, 2009):

    - Partitioning methods: a commonly used partitioning method is *k-means*.
    - Density-based clustering methods: search for dense clusters of users or items.
    - Hierarchical clustering methods: create a hierarchical decomposition of users or items using some criterion.

Memory-based collaborative filtering makes predictions based on the entire collection of previously user's rated items. Typical memory-based approaches are (Ekstrand et al., 2011):

1. *User to user collaborative filtering*, also known as user-based CF, consists in exploiting past ratings users, whose behaviors are similar to the one of the target user, to predict preferences (As shown in figure 1.2 below).

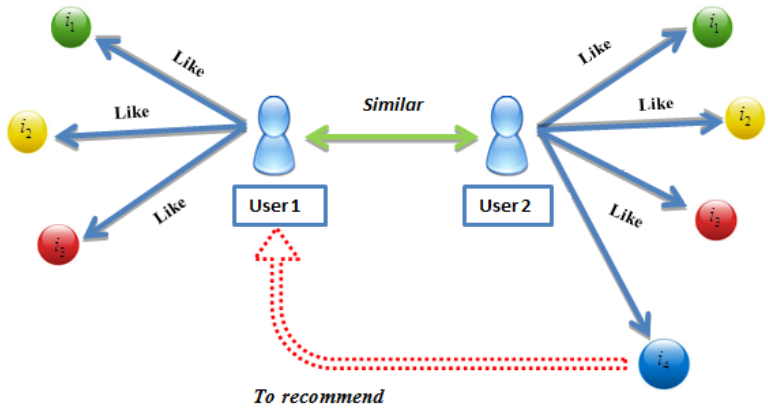


Figure 1.2: The user to user collaborative filtering.

2. *Item to item collaborative filtering*, also called item-based CF, computes how similar a set of items the target user has rated, to the target item i and then selects K most similar items. (As shown in figure 1.3 below).

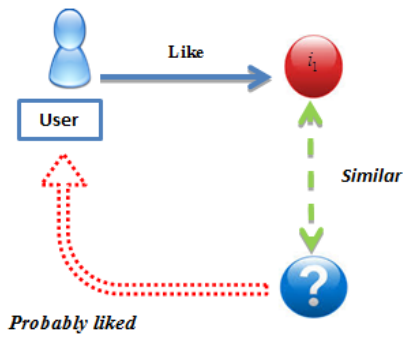


Figure 1.3: The item to item collaborative filtering.

In our work, we are in particular interested on the most widely used recommendation method, namely memory-based collaborative filtering approach (Brun et al., 2010) . It executes the following tasks to generate recommendations for a target user:

1. Calculate the similarity, S, which reflects distance correlation between two users or items. For item-based CF algorithms, the basic idea is to work on the users who have both rated items i and j and then apply a

similarity computation to determine the similarity $S(i, j)$ between the two co-rated items of the users (Sarwar, Karypis, Konstan, & Riedl, 2001). For a user-based CF, we calculate the similarity $S(u, u')$ between users u and u' who have both rated the same items.
There are many different methods to compute similarity between users or items, we cite in particular:

- *Pearson correlation*: computes the extent to which two users are linearly related to each other. It is only based on items rated by both u and $u\prime$, formally:

$$S(u, u') = \frac{\sum_{i \in I'} (r_{u,i} - \overline{r}_u)(r_{u',i} - \overline{r}_{u'})}{\sqrt{\sum_{i \in I'} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{i \in I'} (r_{u',i} - \overline{r}_{u'})^2}} \tag{1.1}$$

  where I' is the set of items that both u and u' have rated, $r_{u,i}$ is the rating of the user u for the item i and $\overline{r}_u$ is the average rating of the co-rated items of the user u.

- *Cosine-based similarity:* is computed by considering each user as a vector of users' ratings and measuring the cosine of the angle formed by these vectors. Formally:

$$S(u, u') = \cos(u, u') = \frac{\overrightarrow{A} \bullet \overrightarrow{B}}{\|\overrightarrow{A}\| * \|\overrightarrow{B}\|} \tag{1.2}$$

  where A and B correspond to vectors of u and u', respectively and $\bullet$ denotes the dot product.

2. Produce a prediction for the target user by taking the weighted average of all the ratings of the user or item on a certain item or user.

- *Weighted sum of others' ratings:* To make prediction for the target user $u\prime$ on a certain item i ,we take the weighted average of all the ratings on that item according to this formula:

$$P_{u',i} = \overline{r}_{u'} + \frac{\sum_{u \in U} (r_{u,i} - \overline{r}_u) * S(u, u')}{\sum_{u \in U} |S(u, u')|} \tag{1.3}$$

  where $\overline{r}_u$ and $\overline{r}_{u'}$ are the average ratings for the user u' and user u on all other rated items, and $S(u, u')$ is the similarity between the user u' and user u.

- *Simple weighted average:* For item-based prediction, we use the simple weighted average to predict the rating, $P_{u,i}$, for user u on item i. Formally:

$$P_{u,i} = \frac{\sum_{j \in N} r_{u,j} * S(i, j)}{\sum_{j \in N} |S(i, j)|} \tag{1.4}$$

  where the summations are over all other rated items $j \in N$ for user u, $S(i, j)$ is the similarity between items i and j, $r_{u,j}$ is the rating for user u on item j.

3. Top-K recommendations is to generate a set of K top-ranked items that will be of interest to a certain user.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4	4	2	1	2	?	?
u_2	3	?	?	?	5	1	?	?
u_3	3	?	?	3	2	2	?	3
u_4	4	?	?	2	1	1	2	4
u_5	1	1	?	?	?	?	?	1
u_6	?	1	?	?	1	1	?	1
u_a	?	?	4	3	?	1	?	5

Table 1.1: The users' ratings matrix

Example 1.1. *Let us consider the user-item matrix of Table* 1.1 *composed of seven users* $\{u_1, u_2, u_3, u_4, u_5, u_6, u_a\}$ *where* u_a *is the active user and eight items* $\{i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8\}$*. Our aim is to recommend to the active user* u_a *the most preferred item not yet used. To this end, we should compute all of* $P_{u_a,i_1}, P_{u_a,i_2}, P_{u_a,i_5}$ *and* P_{u_a,i_7} *using Equation* (1.1) *as follows:*

$$S(u_a, u_1) = \frac{(r_{u_a,i_3}-\bar{r}_{u_a})(r_{u_1,i_3}-\bar{r}_{u_1})+(r_{u_a,i_4}-\bar{r}_{u_a})(r_{u_1,i_4}-\bar{r}_{u_1})+(r_{u_a,i_6}-\bar{r}_{u_a})(r_{u_1,i_6}-\bar{r}_{u_1})}{\sqrt{(r_{u_a,i_3}-\bar{r}_{u_a})^2+(r_{u_a,i_4}-\bar{r}_{u_a})^2+(r_{u_a,i_6}-\bar{r}_{u_a})^2} * \sqrt{(r_{u_1,i_3}-\bar{r}_{u_1})^2+(r_{u_1,i_4}-\bar{r}_{u_1})^2+(r_{u_1,i_6}-\bar{r}_{u_1})^2}}$$

$$= \frac{(4-3.25)(4-2.6)+(3-3.25)(2-2.6)+(1-3.25)(2-2.6)}{\sqrt{(4-3.25)^2+(3-3.25)^2+(1-3.25)^2} * \sqrt{(4-2.6)^2+(2-2.6)^2+(2-2.6)^2}}$$

$$= \mathbf{0.655}$$

$$S(u_a, u_2) = \frac{(r_{u_a,i_6}-\bar{r}_{u_a})(r_{u_2,i_6}-\bar{r}_{u_2})}{\sqrt{(r_{u_a,i_6}-\bar{r}_{u_a})^2} * \sqrt{(r_{u_2,i_6}-\bar{r}_{u_2})^2}}$$

$$= \frac{(1-3.25)(1-3)}{\sqrt{(1-3.25)^2}\sqrt{(1-3)^2}}$$

$$= \mathbf{1}$$

$$S(u_a, u_3) = \frac{(r_{u_a,i_4}-\bar{r}_{u_a})(r_{u_3,i_4}-\bar{r}_{u_3})+(r_{u_a,i_6}-\bar{r}_{u_a})(r_{u_3,i_6}-\bar{r}_{u_3})+(r_{u_a,i_8}-\bar{r}_{u_a})(r_{u_3,i_8}-\bar{r}_{u_3})}{\sqrt{(r_{u_a,i_4}-\bar{r}_{u_a})^2+(r_{u_a,i_6}-\bar{r}_{u_a})^2+(r_{u_a,i_8}-\bar{r}_{u_a})^2} * \sqrt{(r_{u_3,i_4}-\bar{r}_{u_3})^2+(r_{u_3,i_6}-\bar{r}_{u_3})^2+(r_{u_3,i_8}-\bar{r}_{u_3})^2}}$$

$$= \frac{(3-3.25)(3-3.4)+(1-3.25)(2-3.4)+(5-3.25)(3-3.4)}{\sqrt{(3-3.25)^2+(1-3.25)^2+(5-3.25)^2} * \sqrt{(3-3.4)^2+(3-3.4)^2+(3-3.4)^2}}$$

$$= \mathbf{0.586}$$

$$S(u_a, u_4) = \frac{(r_{u_a,i_4}-\bar{r}_{u_a})(r_{u_4,i_4}-\bar{r}_{u_4})+(r_{u_a,i_6}-\bar{r}_{u_a})(r_{u_4,i_6}-\bar{r}_{u_4})+(r_{u_a,i_8}-\bar{r}_{u_a})(r_{u_4,i_8}-\bar{r}_{u_4})}{\sqrt{(r_{u_a,i_4}-\bar{r}_{u_a})^2+(r_{u_a,i_6}-\bar{r}_{u_a})^2+(r_{u_a,i_8}-\bar{r}_{u_a})^2} * \sqrt{(r_{u_4,i_4}-\bar{r}_{u_4})^2+(r_{u_4,i_6}-\bar{r}_{u_4})^2+(r_{u_4,i_8}-\bar{r}_{u_4})^2}}$$

$$= \frac{(3-3.25)(2-2.33)+(1-2.33)(2-3.4)+(5-3.25)(4-2.33)}{\sqrt{(3-3.25)^2+(1-3.25)^2+(5-3.25)^2} * \sqrt{(2-2.33)^2+(1-2.33)^2+(4-2.33)^2}}$$

$$= \mathbf{0.9967 \simeq 1}$$

$$S(u_a, u_5) = \frac{(r_{u_a,i_8}-\bar{r}_{u_a})(r_{u_5,i_8}-\bar{r}_{u_5})}{\sqrt{(r_{u_a,i_8}-\bar{r}_{u_a})^2} * \sqrt{(r_{u_5,i_8}-\bar{r}_{u_5})^2}}$$

$$= \frac{(5-3.25)(1-1)}{\sqrt{(5-3.25)^2}\sqrt{(1-1)^2}}$$

$$= \mathbf{0}$$

$$S(u_a, u_6) = \frac{(r_{u_a,i_6}-\bar{r}_{u_a})(r_{u_6,i_4}-\bar{r}_{u_6})+(r_{u_a,i_8}-\bar{r}_{u_a})(r_{u_6,i_8}-\bar{r}_{u_6})}{\sqrt{(r_{u_a,i_6}-\bar{r}_{u_a})^2+(r_{u_a,i_8}-\bar{r}_{u_a})^2} * \sqrt{(r_{u_6,i_6}-\bar{r}_{u_6})^2+(r_{u_6,i_8}-\bar{r}_{u_6})^2}}$$

$$= \frac{(1-3.25)(1-1)+(5-3.25)(1-1)}{\sqrt{(1-3.25)^2+(5-3.25)^2} * \sqrt{(1-1)^2+(1-1)^2}}$$

$$= \mathbf{0}$$

We will only consider the most similar users to u_a, namely u_1, u_2, u_3 and u_4 . *The predicted preference is therefore computed using Equation (1.3) as follows:*

$$P_{u_a,i_1} = \bar{r}_{u_a} + \frac{(r_{u_2,i_1}-\bar{r}_{u_2})*S(u_a,u_2)+(r_{u_3,i_1}-\bar{r}_{u_3})*S(u_a,u_3)+(r_{u_4,i_1}-\bar{r}_{u_4})*S(u_a,u_4)}{|S(u_a,u_2)|+|S(u_a,u_3)|+|S(u_a,u_4)|}$$

$$= 3.25 + \frac{(3-3)*1(3-3.4)*0.586+(4-2.33)*0.9967}{1+0.586+0.9967}$$

$$= \mathbf{2.685}$$

$$P_{u_a,i_2} = \bar{r}_{u_a} + \frac{(r_{u_1,i_2}-\bar{r}_{u_1})*S(u_a,u_1)}{|S(u_a,u_1)|}$$

$$= 3.25 + \frac{(4-2.6)*0.655}{0.655}$$

$$= \mathbf{4.65}$$

Also, we compute P_{u_a,i_5} and we obtain:

$$P_{u_a,i_5} = \bar{r}_{u_a} + \frac{(r_{u_1,i_5}-\bar{r}_{u_1})*S(u_a,u_1)+(r_{u_2,i_5}-\bar{r}_{u_2})*S(u_a,u_2)+(r_{u_3,i_5}-\bar{r}_{u_3})*S(u_a,u_3)+(r_{u_4,i_5}-\bar{r}_{u_4})*S(u_a,u_4)}{|S(u_a,u_1)|+|S(u_a,u_2)|+|S(u_a,u_3)|+|S(u_a,u_4)|}$$

$$= 3.25 + \frac{(1-2.6)*0.655+(5-3)*1+(2-3.4)*0.586+(1-2.33)*0.9967}{0.586+0.9967+0.655+1}$$

$= \mathbf{2.684}$

With the same manner, we compute P_{u_a,i_7} and we obtain:

$$P_{u_a,i_7} = \bar{r}_{u_a} + \frac{(r_{u_4,i_7} - \bar{r}_{u_4}) * S(u_a,u_4)}{|S(u_a,u_4)|}$$

$$= 3.25 + \frac{(2-2.33)*1}{1}$$

$$= \mathbf{2.92}$$

As $P_{u_a,i_2} = 4.65 > P_{u_a,i_7} = 2.92 > P_{u_a,i_1} = 2.685 > P_{u_a,i_5} = 2.684$, then item i_2 is more preferred than i_7 which in turn more preferred than i_1 and *i_5 and consequently i_2 will be in the top one recommendation list of u_a.*

Example 1.2. *Let us consider the same ratings matrix of Table* 1.1. *We want to compute user u_a's prediction for items i_1, i_2 and i_5 in order to recommend the most close item to user's taste. So, we need to find the most similar items for each of i_1, i_2 and i_5 based on Pearson Correlation of Equation* (1.1) *as follows:*

$$S(i_1,i_4) = \frac{(r_{u_3,i_4} - \bar{r}_{i_4})(r_{u_3,i_1} - \bar{r}_{i_1}) + (r_{u_4,i_1} - \bar{r}_{i_1})(r_{u_4,i_4} - \bar{r}_{i_4})}{\sqrt{(r_{u_3,i_4} - \bar{r}_{i_4})^2 + (r_{u_4,i_4} - \bar{r}_{i_4})^2} * \sqrt{(r_{u_3,i_1} - \bar{r}_{i_1})^2 + (r_{u_4,i_1} - \bar{r}_{i_1})^2}}$$

$$= \frac{(3-2.75)(3-2.5)+(4-2.75)(2-2.5)}{\sqrt{(3-2.75)^2+(4-2.75)^2}\sqrt{(3-2.5)^2+(2-2.5)^2}}$$

$$= -0.555$$

$$S(i_1,i_6) = \frac{(r_{u_2,i_1} - \bar{r}_{i_1})(r_{u_2,i_6} - \bar{r}_{i_6}) + (r_{u_3,i_1} - \bar{r}_{i_1})(r_{u_3,i_6} - \bar{r}_{i_6}) + (r_{u_4,i_1} - \bar{r}_{i_1})(r_{u_4,i_6} - \bar{r}_{i_6})}{\sqrt{(r_{u_2,i_1} - \bar{r}_{i_1})^2 + (r_{u_3,i_1} - \bar{r}_{i_1})^2 + (r_{u_4,i_1} - \bar{r}_{i_1})^2} * \sqrt{(r_{u_2,i_6} - \bar{r}_{i_6})^2 + (r_{u_3,i_6} - \bar{r}_{i_6})^2 + (r_{u_4,i_8} - \bar{r}_{i_8})^2}}$$

$$= \frac{(3-2.75)(1-1.33)+(3-2.75)(2-1.33)+(4-2.75)(1-1.33)}{\sqrt{(3-2.75)^2+(3-2.75)^2+(4-2.75)^2}\sqrt{(1-1.33)^2+(2-1.33)^2+(1-1.33)^2}}$$

$$= -0.309$$

$$S(i_1,i_8) = \frac{(r_{u_3,i_1} - \bar{r}_{i_1})(r_{u_3,i_8} - \bar{r}_{i_8}) + (r_{u_4,i_1} - \bar{r}_{i_1})(r_{u_4,i_8} - \bar{r}_{i_8}) + (r_{u_5,i_1} - \bar{r}_{i_1})(r_{u_5,i_8} - \bar{r}_{i_8})}{\sqrt{(r_{u_3,i_1} - \bar{r}_{i_1})^2 + (r_{u_4,i_1} - \bar{r}_{i_1})^2 + (r_{u_5,i_1} - \bar{r}_{i_1})^2} * \sqrt{(r_{u_3,i_8} - \bar{r}_{i_8})^2 + (r_{u_4,i_8} - \bar{r}_{i_8})^2 + (r_{u_5,i_8} - \bar{r}_{i_8})^2}}$$

$$= \frac{(3-2.75)(3-2.8)+(4-2.75)(4-2.8)+(1-2.75)(1-2.8)}{\sqrt{(3-2.75)^2+(4-2.75)^2+(1-2.75)^2}\sqrt{(3-2.8)^2+(4-2.8)^2+(1-2.8)^2}}$$

$$= \mathbf{1}$$

We will consider the most similar item to i_1 namely is i_8. The predicted preference P_{u_a,i_1} is therefore computed based on Equation (1.4) *as follows:*

$$P_{u_a,i_1} = \frac{r_{u_a,i_6} * S(i_1,i_8)}{|S(i_1,i_8)|}$$

$$= \frac{5*1}{1}$$

$$= \mathbf{5}$$

in the same way, we calculate:

$$S(i_2,i_3) = \frac{(r_{u_1,i_2}-\bar{r}_{i_2})(r_{u_1,i_3}-\bar{r}_{i_3})}{\sqrt{(r_{u_1,i_2}-\bar{r}_{i_2})^2}\sqrt{(r_{u_1,i_3}-\bar{r}_{i_3})^2}}$$

$$= \frac{(4-2)(4-4)}{\sqrt{(4-2)^2}\sqrt{(4-4)^2}}$$

$$= 0$$

$$S(i_2,i_4) = \frac{(r_{u_1,i_2}-\bar{r}_{i_2})(r_{u_1,i_4}-\bar{r}_{i_4})}{\sqrt{(r_{u_1,i_2}-\bar{r}_{i_2})^2}\sqrt{(r_{u_1,i_4}-\bar{r}_{i_4})^2}}$$

$$= \frac{(4-2)(2-2.5)}{\sqrt{(4-2)^2}\sqrt{(2-2.5)^2}}$$

$$= -1$$

$$S(i_2,i_6) = \frac{(r_{u_1,i_2}-\bar{r}_{i_2})(r_{u_1,i_6}-\bar{r}_{i_6})+(r_{u_6,i_2}-\bar{r}_{i_2})(r_{u_6,i_6}-\bar{r}_{i_6})}{\sqrt{(r_{u_1,i_2}-\bar{r}_{i_2})^2+(r_{u_6,i_2}-\bar{r}_{i_2})^2}\sqrt{(r_{u_1,i_6}-\bar{r}_{i_6})^2+(r_{u_6,i_6}-\bar{r}_{i_6})^2}}$$

$$= \frac{(4-2)(2-1.33)+(1-2)(1-1.33)}{\sqrt{(4-2)^2+(1-2)^2}\sqrt{(2-1.33)^2+(1-1.33)^2}}$$

$$= \mathbf{1}$$

$$S(i_2,i_8) = \frac{(r_{u_5,i_2}-\bar{r}_{i_2})(r_{u_5,i_8}-\bar{r}_{i_8})+(r_{u_6,i_2}-\bar{r}_{i_2})(r_{u_6,i_8}-\bar{r}_{i_8})}{\sqrt{(r_{u_5,i_2}-\bar{r}_{i_2})^2+(r_{u_6,i_2}-\bar{r}_{i_2})^2}\sqrt{(r_{u_5,i_8}-\bar{r}_{i_8})^2+(r_{u_6,i_8}-\bar{r}_{i_8})^2}}$$

$$= \frac{(1-2)(1-2.8)+(1-2)(1-2.8)}{\sqrt{(1-2)^2+(1-2)^2}\sqrt{(1-2.8)^2+(1-2.8)^2}}$$

$$= \mathbf{1}$$

We will consider the most similar item to i_2 namely are i_6 and i_8. The predicted preference P_{u_a,i_2} is therefore computed as follows:

$$P_{u_a,i_2} = \frac{r_{u_a,i_6} * S(i_2,i_6) + r_{u_a,i_8} * S(i_2,i_8)}{|S(i_2,i_6)| + |S(i_2,i_8)|}$$

$$= \frac{(1*1)+(5*1)}{2}$$

$$= \mathbf{3}$$

Also, with the same manner, we calculate:

$$S(i_3,i_5) = \frac{(r_{u_1,i_3} - \bar{r}_{i_3})(r_{u_1,i_5} - \bar{r}_{i_5})}{\sqrt{(r_{u_1,i_3} - \bar{r}_{i_3})^2}\sqrt{(r_{u_1,i_3} - \bar{r}_{i_3})^2}}$$

$$= \frac{(4-2)(4-4)}{\sqrt{(4-2)^2}\sqrt{(4-4)^2}}$$

$$= 0$$

$$S(i_5,i_4) = \frac{(r_{u_1,i_4} - \bar{r}_{i_4})(r_{u_1,i_5} - \bar{r}_{i_5}) + (r_{u_3,i_4} - \bar{r}_{i_4})(r_{u_3,i_5} - \bar{r}_{i_5}) + (r_{u_4,i_4} - \bar{r}_{i_4})(r_{u_5,i_5} - \bar{r}_{i_5})}{\sqrt{(r_{u_1,i_4} - \bar{r}_{i_4})^2 + (r_{u_3,i_4} - \bar{r}_{i_4})^2 + (r_{u_4,i_4} - \bar{r}_{i_4})^2}\sqrt{(r_{u_1,i_5} - \bar{r}_{i_5})^2 + (r_{u_3,i_5} - \bar{r}_{i_5})^2 + (r_{u_5,i_5} - \bar{r}_{i_5})^2}}$$

$$= \frac{(2-2.5)(1-2)+(3-2.5)(2-2)+(2-2.5)(1-2)}{\sqrt{(2-2.5)^2+(3-2.5)^2+(2-2.5)^2}\sqrt{(1-2)^2+(2-2)^2+(1-2)^2}}$$

$$= \mathbf{0.819}$$

$$S(i_5,i_6) = \frac{(r_{u_1,i_5} - \bar{r}_{i_5})(r_{u_1,i_6} - \bar{r}_{i_6}) + (r_{u_2,i_5} - \bar{r}_{i_5})(r_{u_2,i_6} - \bar{r}_{i_6}) + (r_{u_3,i_5} - \bar{r}_{i_5})(r_{u_3,i_6} - \bar{r}_{i_6}) + (r_{u_4,i_5} - \bar{r}_{i_5})(r_{u_4,i_6} - \bar{r}_{i_6})}{\sqrt{(r_{u_1,i_5} - \bar{r}_{i_5})^2 + (r_{u_2,i_5} - \bar{r}_{i_5})^2 + (r_{u_3,i_5} - \bar{r}_{i_5})^2 + (r_{u_4,i_5} - \bar{r}_{i_5})^2}\sqrt{(r_{u_1,i_6} - \bar{r}_{i_6})^2 + (r_{u_2,i_6} - \bar{r}_{i_6})^2 + (r_{u_3,i_6} - \bar{r}_{i_6})^2 + (r_{u_4,i_6} - \bar{r}_{i_6})^2}}$$

$$= \frac{(1-2)(2-1.33)+(5-2)(1-1.33)+(2-2)(2-1.33)+(1-2)(1-1.33)}{\sqrt{(1-2)^2+(5-2)^2+(2-2)^2+(1-2)^2}\sqrt{(2-1.33)^2+(1-1.33)^2+(2-1.33)^2+(1-1.33)^2}}$$

$$= -0.360$$

$$S(i_5,i_8) = \frac{(r_{u_3,i_5} - \bar{r}_{i_5})(r_{u_3,i_8} - \bar{r}_{i_8}) + (r_{u_4,i_5} - \bar{r}_{i_5})(r_{u_4,i_8} - \bar{r}_{i_8}) + (r_{u_6,i_5} - \bar{r}_{i_5})(r_{u_6,i_8} - \bar{r}_{i_8})}{\sqrt{(r_{u_3,i_5} - \bar{r}_{i_5})^2 + (r_{u_4,i_5} - \bar{r}_{i_5})^2 + (r_{u_6,i_5} - \bar{r}_{i_5})^2}\sqrt{(r_{u_3,i_8} - \bar{r}_{i_8})^2 + (r_{u_4,i_8} - \bar{r}_{i_8})^2 + (r_{u_6,i_8} - \bar{r}_{i_8})^2}}$$

$$= \frac{(2-2)(3-2.8)+(1-2)(4-2.8)+(1-2)+(1-2.8)}{\sqrt{(2-2)^2+(1-2)^2+(1-2)^2}\sqrt{(3-2.8)^2+(4-2.8)^2+(1-2.8)^2}}$$

$$= \mathbf{0.195}$$

This means that the most similar item to i_5 *are* i_4 and i_8. *The prediction* P_{u_a,i_5} *is therefore computed as follows:*

$$P_{u_a,i_5} = \frac{r_{u_a,i_4} * S(i_5,i_4) + r_{u_a,i_8} * S(i_5,i_8)}{|S(i_5,i_4)| + |S(i_5,i_8)|}$$

$= \frac{(3*0.819)+(5*0.195)}{0.819+0.195}$

$= \mathbf{3.384}$

$\Rightarrow$ *Item i_1 has the highest predicted preference, then item i_5, and finally item i_2. Consequently the top list recommendation for user u_a* will be as follow:

$u_a, i_1, P_{u_a,i_1} = 5$
$u_a, i_5, P_{u_a,i_5} = 3.384$
$u_a, i_2, P_{u_a,i_2} = 3$

1.3 Possibility theory

The probability theory, which dates from the 17th century, is a classical theory that enables representing and quantifying uncertain information. It has been involved in several real world areas. However, such theory, that does not consider the situation of *total ignorance*, can be only used when the expert provides precise numerical values. This situation is not always feasible, which has motivated the development of alternative uncertainty frameworks.

In order to deal with uncertain and imprecise data, several non classical theories have been proposed, such as fuzzy sets theory (Zadeh, 1999), belief functions theory (Smets & Kennes, 1994), possibility theory (Dubois & Prade, 1998), etc. We are, in particular interested in possibility theory introduced at first by Zadeh (Zadeh, 1999) and then developed by Dubois and Prade (Dubois & Prade, 2011). It offers a natural and simple tool to handle imperfect information. It represents an appropriate framework for experts to express their partial beliefs in a much more flexible way than within the probability theory framework. In what follows, we present possibility theory concepts (for more details see (Dubois & Prade, 1998)).

1.3.1 Notations

- Ω the universe of discourse. $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$.
- The power set 2^Ω is the set of all subsets of Ω. This power set includes obviously the empty set $\varnothing$ and the universe of discourse Ω.
- ω is an element of Ω ($\omega \in \Omega$) and by A an element of the power set ($A \in 2^\Omega$ or $A \subseteq \Omega$). $\cap$ and $\cup$ denote, respectively, the intersection and union operations.

1.3.2 Possibility distribution

The basic building block in the possibility theory is the concept of *possibility distribution* π, which corresponds to a function associating to each element ω_i from the universe of discourse Ω a value to a bounded and linearly ordered valuation set (L,<). Contrary to the standard probability theory, the possibilistic scale could be interpreted in twofold: a *numerical interpretation* when values have a real sense (L=[0,1]) and an *ordinal* one ($>_\pi$) when values only reflect a total pre-order between the different states of the world.

Intuitively, in the qualitative way, $\geq_\pi$ corresponds to a plausibility relation on Ω which enables us to express that some situations are more plausible than others.

Example 1.3. The plausibility relation relative to the possibility distribution given by the arbiter of Example 1. is as follows: $T1 = win >_\pi T1 = equalize >_\pi T1 = lose$.

In the qualitative setting, the possibility distribution, denoted by π_Q, , is equipped by a finite and totally ordered scale denoted by $\mathcal{L}= \{a_0 = 1, a_1, ..., a_n, a_{n+1} = 0\}$ such that $a_0 > a_1 > ... > a_{n+1}$.

Example 1.4. *Let us continue with the same example. The possibility distribution can be represented qualitatively by the arbiter as follows:*
$\pi_Q(T1 = win) = a_0 = 1$,
$\pi_Q(T1 = equalize) = a_1 = 0.7$,
$\pi_Q(T1 = lose) = a_2 = 0.2$.

We can deduce that T1= win is more plausible than T1= equalize, which is in its turn more plausible than T1= lose since $a_0 = 1 > a_1 = 0.7 > a_2 = 0.2$

In this work, we will focus on the numerical interpretation. The degree $\pi(\omega)$ represents the compatibility of ω with available pieces of information. By convention, $\pi(\omega) = 1$ means that ω is totally possible and $\pi(\omega) = 0$ means that ω is an impossible state. If $\pi(\omega) > \pi(\omega')$, this means that ω is preferred to ω'.
In the possibility theory framework, there are two extreme cases:

- *Complete knowledge*: $\exists\ \omega_0, \pi(\omega_0) = 1$ and $\pi(\omega) = 0\ \forall \omega \neq \omega_0$ (only ω_0 is possible).
- *Total ignorance*: $\forall \omega \in \Omega, \pi(\omega) = 1$ (all states are possible).

A possibility distribution π is said to be *normalized* if there exists at least one totally possible state. Formally:

$$\exists\ \omega \in \Omega, \pi(\omega) = 1 \tag{1.5}$$

Example 1.3. *Let us consider a handball match. Each team can win, lose or equalize. Then, the universe of discourse related to the match can be defined as follows:* $\Omega = \{win, equalize, lose\}$. *Assuming that the arbiter gives his point of view regarding the game result for team 1, in the form of a possibility distribution , denoted by* $\pi(T1)$ *and defined as follows:*
$\pi(T1 = win) = 1$,
$\pi(T1 = equalize) = 0.7$,
$\pi(T1 = lose) = 0.2$,
$\pi(T1 = win) = 1$ *means that it is fully possible for team 1 to win the game.*

The possibility distribution given by the arbiter is normalized since max(1, 0.7, 0.2)=1.

1.3.3 Inconsistency

In possibility theory, the inconsistency is measured by the degree of conflict between uncertain information. Formally:

$$Inc(\pi) = 1 - max_{\omega \in \Omega} \{\pi(\omega)\} \tag{1.6}$$

In this case, π is considered as *sub-normalized*, otherwise, π is said to be *normalized* (i.e. $max_{\omega \in \Omega}\ \pi(\omega) = \pi(\omega_i) = 1$). It is clear that, for normalized π, $max_{\omega \in \Omega}\ \pi(\omega) = 1$, hence $Inc(\pi) = 0$. The measure *Inc* is very useful in computing the conflict between two distributions π_1 and π_2 given by $Inc(\pi_1, \pi_2) = Inc(\pi_1 \wedge \pi_2)$, where $\wedge$ is a conjunctive t-norm operator. For simplicity, we take the *minimum* conjunctive ($\wedge$) operator. Obviously, when $\pi_1 \wedge \pi_2$ gives a sub-normalized possibility distribution, it indicates that there is a conflict between π_1 and π_2. On the other hand, $\pi_1 \wedge \pi_2$ is normalized, there is no conflict and hence $Inc(\pi_1, \pi_2) = 0$.

Example 1.4. *Let $\pi_1[1, 0.2, 0.5, 0]$ and $\pi_2[0.8, 0, 0.3, 1]$ be two possibility distributions. We take the minimum as the conjunctive operator, we obtain: $Inc(\pi_1, \pi_2) = Inc([0.8, 0, 0.3, 0]) = 1 - 0.8 = 0.2$. Thus, the two sources are inconsistent with each other.*

1.3.4 Possibility and necessity measures

Contrary to probability theory which only uses one measure, namely the probability measure P, possibility theory uses two dual measures: the possibility (plausibility) measure Π and the necessity (certainty) measure N.

Possibility measure

Given a possibility distribution π, we can define a mapping grading the *possibility measure* of any subset $\phi \subseteq \Omega$ by:

$$\Pi(\phi) = max_{\omega \in \phi}\ \pi(\omega) \tag{1.7}$$

$\Pi(\phi)$ is called the possibility degree of ϕ, it corresponds to to the possibility degree to have one of the models of ϕ as the real world. This measure evaluates at which level ϕ is **consistent** with our knowledge represented by π.

Necessity measure

The dual of the possibility measure is the necessity measure defined by $\forall \phi \subseteq \Omega$:

$$N(\phi) = 1 - \Pi(\neg\phi) = min_{\omega \notin \phi}\ (1 - \pi(\omega)) \tag{1.8}$$

$N(\phi)$ is called the necessity degree of ϕ. It corresponds to the certainty degree associated with ϕ. This measure evaluates at which level ϕ is **certainly** implied by our knowledge represented by π.

Example 1.5. *Let us consider the possibility distribution π given in Example 1 and $\phi = \{T1 = win\}$. $\Pi(\phi) = max\{1\} = 1$, which means that it is fully possible that Team 1 wins the game but we are not certain about this fact. Hence, the certainty degree is given by: $N(\phi) = 1 - \Pi(\neg\phi) = 1 - max\{0.7, 0.2\} = 0.3$.*

1.4 Conclusion

This chapter presented two main concepts for recommendation systems, more especially item-based and user-based collaborative filtering approach, and possibility theory framework. Next chapter is devoted to preferences-based research works related to our proposed attempt.

Chapter 2

Preferences-based recommendation

2.1 Introduction

Recommender systems are systems that provide users with recommendation of items and information that help them to decide which items to procure or look based on the individual customer preferences. Recommendation systems are generally consisting of background data such as historical data consisting of *preferences* of items before the recommendation begins. In recent years, RSs have attracted a considerable amount of research attention resulting in a large variety of approaches (Ekstrand et al., 2011; Koren & Sill, 2013). Among this relatively recent approaches, it has already been proven the benefits of using collaborative filtering (Sarwar et al., 2001) preferences over absolute ratings, to perform more accurate predictions for users. Although, in everyday life, rating items is not such a natural mechanism, in contrast, there exists some kind of uncertainty behind using preferences. Yet, uncertainty in users' ratings is so pervasive, that can't be ignored. This uncertainty that inherently exists is not studied in previous recommender systems research and ignoring this type of uncertainty puts the modeling in a less realistic setting, and the resulting model does not precisely represent the reality (Zenebe & Norcio, 2009).

This chapter is organized as follows: Section 2.2 presents certain preferences-based recommendation and Section 2.3 is dedicated to the uncertainty aspect of preferences-based recommendation.

2.2 Certain preferences-based recommendation

Many researches have been proposed to deal with preferences under recommendation. According to most collaborative filtering, researches can be categorized into two classes:

- ***Handling preference order:***

  There have been some works that use preference orders instead of actual ratings for recommender systems. According to the standard CF approach (Ricci et al., 2011), first, the user inputs his/her preferences to the

system. The system then searches for other people whose preferences are similar to those of the target user. Next, the system recommends to the user the items that those people prefer. *Nantonac collaborative filtering* (Kamishima, 2003) has been proposed in the same context of CF using a new representation of the user's preferences. In fact, author adopted the *order responses* by the ranking method to represent preferences. Obviously, the system in this framework shows a set of items X_i, to the user i, and the user sorts these items according to his / her preferences, then, the sorted sequences are denoted by $O_i = x^1 \succ x^2 \succ ... \succ x^{|X_i|}$ indicating that user i prefers item x^1 to item x^2. The rank, $r(O_i, x^j)$ indicates the *position* of the item x^j in the order O_i. For example, the *order* $O_i = x^1 \succ x^3 \succ x^2$, $r(O_i, x^1) = 1$, $r(O_i, x^3) = 2$ and $r(O_i, x^2) = 3$. Consequently, the task of nantonac CF is estimating the items that the target user is expected to prefer. In the same context, in (L. Yu & Yang, 2008), authors proposed an improved collaborative filtering algorithm based on a preference order of items. In fact, in this CF recommendation, the most important step is to *obtain a preference order* that can be gotten by several methods:

1. *Explicit preference order by user*, when number of items is few, users can directly give the preference order according to their interests. For example, as shown in Figure 2.1, there is three items, A, B and C:

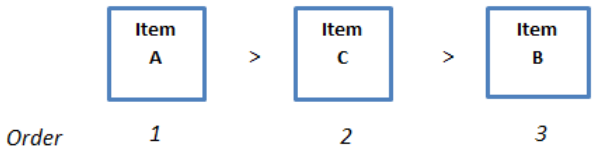


Figure 2.1: The preference Order (L. Yu & Yang, 2008).

   - Order number of item A in sequence is 1: $r(O, A) = 1$.
   - Order number of item B in sequence is 1: $r(O, B) = 3$.
   - Order number of item C in sequence is 1: $r(O, C) = 2$.

2. *Implicit preference order based on transaction data*, in this case, preference order can be gotten according to the cost of product or to the quality of transaction on product.
3. *Implicit preference order based on web mining*, Web mining is an important method to get the users' preference by analyzing browsing pattern. for example, preference for product can be measured by time of browsing web pages.

Once preference order of items is acquired, collaborative filtering algorithm executes the remaining steps as the traditional one, and finally gives recommendation result.

In addition, Jin et al. (Jin et al., 2003) handled separately users preferences and the users rating. More specifically, for each user, they built two separate models, namely:

- *A preference model* capturing which items are preferred by the user.
- *A rating model* capturing how the user would rate an item given the preference information.

Hence, they proposed at first a memory-based probabilistic approach, which decouples the rating and the preference of a user. Then a model-based bayesian approach can be used to predict the rating for a new user by combining the prediction given by all the models.

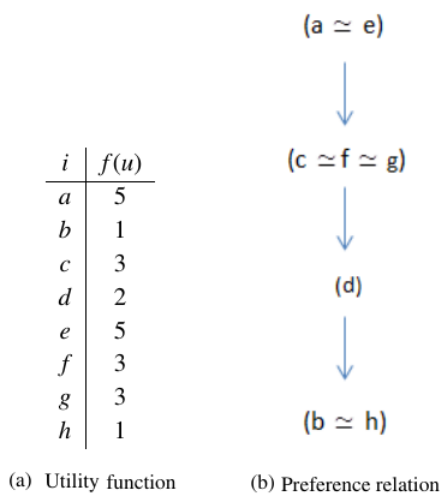
More recently, in (de Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010) authors have presented a novel CF idea to improve the predictions of the system by increasing the available information in the datasets. The objective is to use all possible preferences to improve recommendations made with little information. So, the purpose of second-hand information is when a similar user has not rated the target item then they will guess his/her preferences using the available information.

- ***Handling preference relation:***

Recently, there have been active researches on *preference relations*. In fact, in (Brun et al., 2010) authors are interested in the collaborative filtering approach. In this framework, the user is not asked to vote for resources but to express a qualitative interest about the resources he/she has already seen. For example, the user will say " I prefer resource j to resource i". Formally, authors defined a preference relation as a binary relation $i \preceq j$ where:

  - " j is strictly preferred to i" is noted $i \prec j$.
  - " i and j are equivalent" or " the user does not mind between i and j" is noted $i \simeq j$.
  - $i ? j$ the user does not know.

However, these preferences are partially known. Consequently, some other preferences are missing and CF aims to guess it. In this work, authors presented a new approach to compute recommendations using preference relations instead of using ratings (utilities). They proposed to replace utilities by their qualitative counterpart: *preference relations*. Figure 2.2 shows an example of representing preference relation where: items a and e are equivalent and are more preferred than c, f and g which are in turn equivalent and more preferred than item d which in its turn preferred than items b and h.



i	$f(u)$
a	5
b	1
c	3
d	2
e	5
f	3
g	3
h	1

(a) Utility function (b) Preference relation

Figure 2.2: (a) a utility function, (b) a corresponding preference relation (Brun et al., 2010).

In (Sarwar et al., 2001), authors analyzed different item-based recommendations, the main idea is to analyze the user-item representation preferences matrix to identify relations between different items and then use these relations to compute the prediction score for a given user-item.

Another CF framework based on matrix factorization is proposed in (Desarkar, Saxena, & Sarkar, 2012) uses preference relations to solve the problem of personalized recommendation. First of all, in rating based recommender systems, preferences of the users are of the form $\pi(u, i, j)$. It indicates that for the ordered item pair (i, j), the strength of user u's preference relation is $\pi(u, i, j)$. Then, a matrix factorization is used to learn the user and item features. The goal is to develop a factorization model that is able to predict the users' preference relations for different item pairs. Finally, once the user and item features are computed, the system can generate recommendations. Intuitively, if for a particular user u, an item i is predicted to be better than many others items, then i can be recommended to u. Also, in (Hu, Koren, & Volinsky, 2008) authors studied collaborative filtering on datasets with implicit feedback, which is a very common situation. The main idea is that implicit user observations should be transformed into two paired magnitudes *preferences* and *confidence levels*. In other words, for each user-item pair, they derived from the input data an estimate to whether the user would like or dislike the item ("preference") and couple this estimate with a confidence level. So, this preference-confidence serves a key role in analyzing implicit feedback.
In (Liu & Yang, 2008), authors have proposed a new CF algorithm for ranking items based on the preferences of similar users, so, they have modeled a user's preference relation function denoted by $\Psi(i, j) > 0$ stating that item i is more preferable to item j for a user u. The magnitude of this preference relation function $|\Psi(i, j)|$ indicates the strength of preference and a value of zero means that there is no preference relation between the two items.
In (Yuan, Huang, & Zhong, 2013), a framework based on similarity measures on (user, item, tag) from qualitative and quantitative perspective has been developed. The qualitative measure makes use of the *preference structure relation*, and the quantitative measure makes use of *reflection* on (user, item, tag). Then, the k nearest neighbors and reverse k' nearest neighbors are used to generate recommendations.

These methods only deal with certain preferences derived from datasets. Collaborative filtering gives good results in a certain context, in which ratings provided by users are known with certainty (Liu & Yang, 2008). But this does not reflect the reality, which is related to uncertainty and imprecision by nature. Consequently, the recommendation results are deeply affected if uncertainty is not considered. For these reasons, new studies have considered this uncertainty in order to improve the recommender's accuracy. This will be the focus of the following section.

2.3 Uncertain preferences-based recommendation

Research efforts that address the representation of user behavior and information about items under uncertainty are limited. These researches focus on covering uncertainty over user preferences. A first attempt has been established in (Price & Messinger, 2005). In fact, authors have proposed an approach for recommender systems that optimize decision making. This approach selects the alternative set that:

- maximizes the expected valuation of the user's choice: *Expected Utility*
- covers the uncertainty over user preferences: *Probability distribution*

Intuitively, authors developed a specific formulation called *maximization expected max (MEM)* that produces a diversity of alternatives in the recommendation set and covers the uncertainty over possible user preferences. This approach is limited since it is not obvious that an optimal set of MEM algorithm is treatable because a naïve version of a MEM set optimizer must enumerate all $k-$element subsets of the n alternatives.

In addition, authors in (K. Yu et al., 2004) have studied a probabilistic memory-based collaborative filtering (*PMCF*) framework similar to the classical memory-based CF approach. A schematic of the components of *PMCF* is shown in Figure 2.3. The *PMCF* framework is based essentially on the two main phases:

- *The interactively learning individual user profiles step:* relates to the new user problem in case the available information is insufficient and executes different steps:
  1. Present informative items to the user for preference.
  2. User rates items which are familiar to him.
  3. Search profile space for similar profile.
  4. Present probabilistic prediction results to user: First of all a generative probabilistic model in which the preference of a target user are generated based on a probability density to model preference profiles. Then, calculate the posterior density of the active user's preferences on not yet rated items in order to estimate user ratings. Finally,predictions are made by combining the predictions based on other users weighted by their degree of agreement with the target user.
- *The incrementally constructing a compact profile space step:* allows to select a small subset, called the profile space from a database of user ratings. The selection procedure is derived from the probabilistic framework and ensures that the small profile space leads to predictions that are as accurate as predictions made by using the whole database of user preferences.

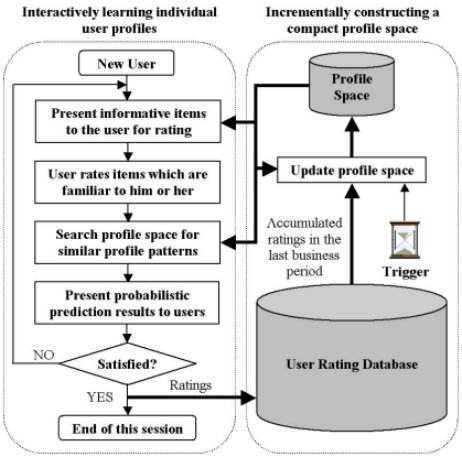


Figure 2.3: The components of PMCF (K. Yu et al., 2004).

More recently, authors in (Xiang et al., 2013) proposed a new method of CF based on uncertain user interests clusters. In fact, authors have proposed an architecture of a Collaborative Filtering Recommender System to adapt uncertain users' evolving interests based on the following steps:

1. Define uncertain interest.
2. Solve the uncertain feature using a clustering algorithm.
3. Compute the between-class entropy of any two clusters and get stable classes (see Figure 2.4).
4. Define a trustworthy degree of uncertain interests to measure the rationality of clustering algorithm results.

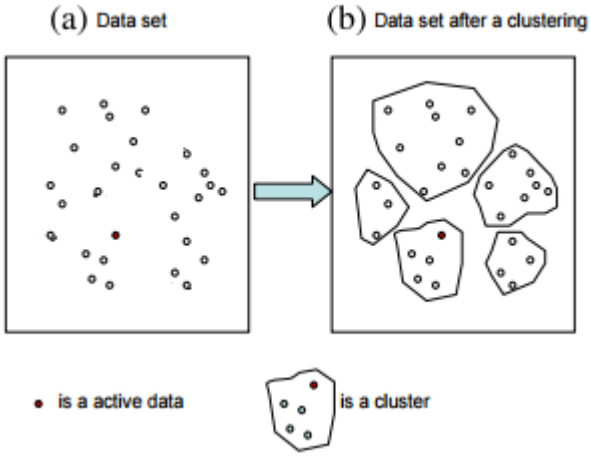


Figure 2.4: The datasets after and before a clustering step (Xiang et al., 2013).

A recent work in (Samia et al., 2014) handles uncertainty in users' preferences where authors have emphasized on the presence of uncertainty's hidden through fuzzy preferences' analysis . For example, a recommender system recommends to user u an item such as *Tassili* which is a restaurent that is renowned according to similar users' past reviews for its good food. But, after consuming the service, user u is not satisfied because the food wasn't that good as expected. Therefore, he/she dismiss the system because it becomes less trustworthy. So, they expressed preferences in multi-criteria fuzzy ratings through linguistic terms, then, preference relations are deduced and quantified with a preference intensity degree that expresses to which extent is more preferable than another.

2.4 Conclusion

This chapter surveys previous works of preferences based recommendation. According to collaborative filtering approaches, researches can be categorized into *preferences orders* (Jin et al., 2003; L. Yu & Yang, 2008) handling preferences over a quantitative way, and a qualitative way by using *preferences relations* (Brun et al., 2010; Desarkar et al., 2012). However, almost of these works only deal with certain preferences. The few other works that attempt to take into account the uncertainty aspect of preferences have not used a purely uncertain algorithm. Based on uncertain preferences, we will propose in the next chapter a novel recommendation approach for uncertain preferences.

Chapter 3

New possibilistic item-based collaborative filtering recommender

3.1 Introduction

Representing the uncertain aspect of users' preferences is crucial in recommender systems. There exist few recent researches that modeled uncertain users preferences by means of fuzzy set theory (Yager, 2003; Zenebe & Norcio, 2009; Samia et al., 2014). Another research works (K. Yu et al., 2004) studied a probabilistic memory-based collaborative filtering (*PMCF*) framework using a generative probabilistic density to model preference profiles. Then, a posterior distribution of user's preferences is to predict an active user's preferences. However, no research studied a purely uncertain recommender system dealing with uncertain rating as input. To this end, we propose a new possibilistic item-based CF recommender based on a possibilistic representation of preferences, similarity computation and prediction and recommendation. Section 3.2 will detail the different steps of the proposed new possibilistic CF recommender.

3.2 Proposed possibilistic CF recommender

Our aim in this work is to take into consideration the uncertain aspect of users preferences under a possibilistic framework. To ensure this task, uncertain preferences should be at first represented using possibility distributions, then a purely possibilistic similarity measure should be used to compute the most similar items. Finally, the unknown preference degree of a target item by a target user is predicted by averaging the preferences of other similar items rated by this target user.

We propose a possibilistic item-based CF approach, denoted by ΠCF, based on three steps, namely, *preferences representation*, *similarity computation* and *prediction and recommendation generation*. The whole process of the proposed ΠCF method is illustrated by the diagram of Figure 3.1. The first step consists in building the user-item matrix under a possibilistic framework. The second step applies a possibilistic similarity measure to calculate the

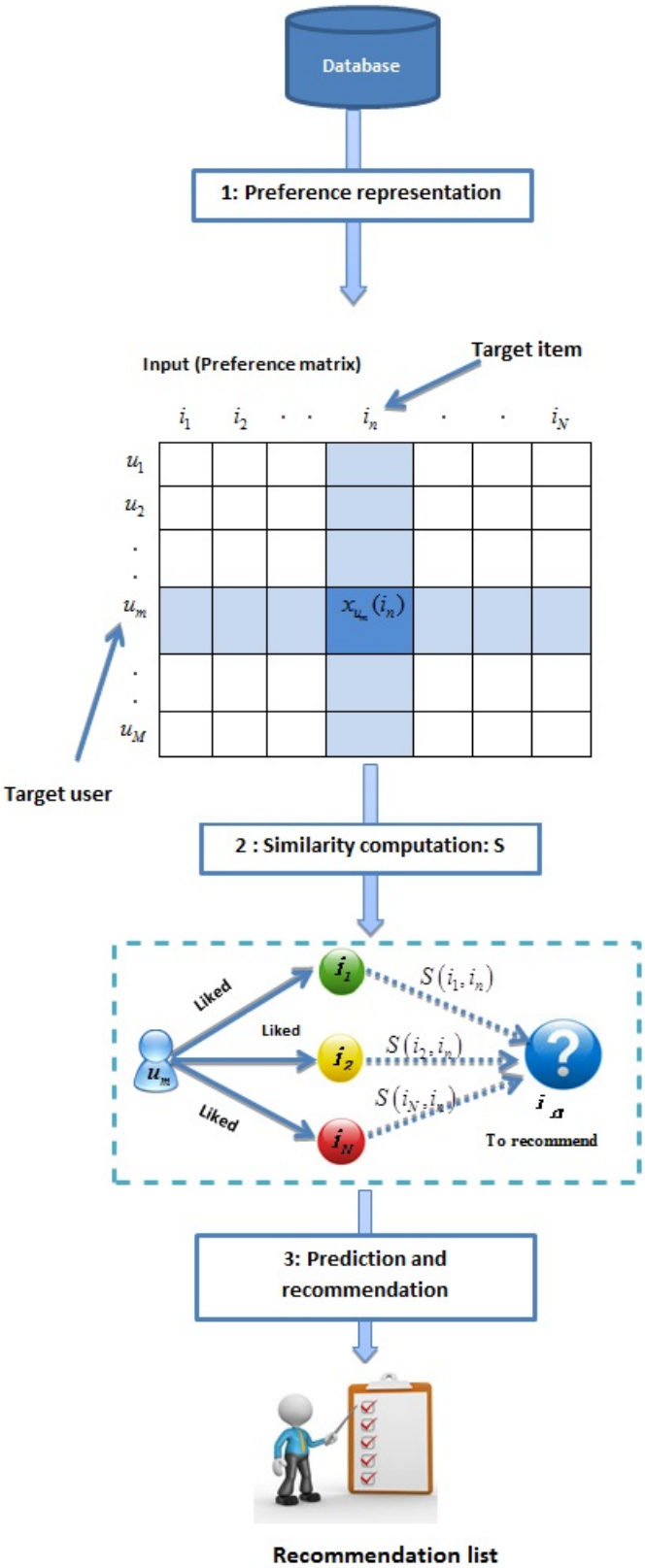


Figure 3.1: The possibilistic item-based collaborative filtering process: ΠCF.

similarity between items. The last step consists in recommending the top K most likely items that can interest the target user. Section 3.2.1 presents the preferences representation step.

3.2.1 Preferences representation

The representation of users preferences is a primordial step in our approach. In fact, each user should provide his preferences about items as a possibility distribution where each degree corresponds to a satisfaction degree for an item i. When only one item is fully satisfactory by a user and all remaining items are not satisfactory at all, we deal with the extreme case of complete knowledge. While when all items are satisfactory by a user, the total ignorance case is tackled. Based on the simplicity of the possibility theory framework, we present users preferences using possibility distributions described in Definition 1.

Definition 1. *Let's consider a list of* $\boldsymbol{M}$ *users* $U = \{u_1, .., u_m, .., u_M\}$ *and a list of* $\boldsymbol{N}$ *items* $I = \{i_1, .., i_n, .., i_N\}$. *P denotes a possibilistic user-item matrix where each entry* $x_{u_m}(i_n)$ *corresponds to the preference degree of item* i_n *provided by user* u_m. *Each row in P is a user profile and represents user's items preferences degrees. In* Π*CF, each user* u_m *should provide its preferences using a possibility distribution. Formally:*

$\pi_{u_m} : I \rightarrow [0, 1]$, *where* $\pi_{u_m}(i_n)$ *denotes the degree of satisfaction of an item* i_n *by* u_m *such that:*

- $\pi_{u_m}(i_n) = 1$, *the item* i_n *is fully satisfactory.*
- $0 < \pi_{u_m}(i_n) < 1$, *the item* i_n *is somewhat satisfactory.*
- $\pi_{u_m}(i_n) = 0$, *the item* i_n *is not satisfactory at all.*

$\pi(i_n)$ *expresses the preference degrees assigned to item* i_n *by all users.*

Example 3.1. *Table 3.1 represents an example of a possibilistic user-item matrix composed of five users* $\{u_1, u_2, u_3, u_4, u_5\}$ *and four items* $\{i_1, i_2, i_3, i_4\}$, *representing users preferences of items using possibility distributions.*

	i_1	i_2	i_3	i_4
u_1	$\pi_{u_1}(i_1) = 1$	$\pi_{u_1}(i_2) = 0.5$	$\pi_{u_1}(i_3) = 0.3$	$\pi_{u_1}(i_4) = 1$
u_2	$\pi_{u_2}(i_1) = 1$	$\pi_{u_2}(i_2) =?$	$\pi_{u_2}(i_3) = 0$	$\pi_{u_2}(i_4) =?$
u_3	$\pi_{u_3}(i_1) =?$	$\pi_{u_3}(i_2) = 1$	$\pi_{u_3}(i_3) = 0.3$	$\pi_{u_3}(i_4) = 0.7$
u_4	$\pi_{u_4}(i_1) = 0$	$\pi_{u_4}(i_2) = 1$	$\pi_{u_4}(i_3) =?$	$\pi_{u_4}(i_4) = 0.7$
u_5	$\pi_{u_5}(i_1) = 0.2$	$\pi_{u_5}(i_2) = 0$	$\pi_{u_5}(i_3) = 1$	$\pi_{u_5}(i_4) = 0.3$

Table 3.1: User-item preferences matrix

For instance, item i_1 *is fully satisfactory for user* u_1 *and not satisfactory at all for user* u_4, *while ? for* u_3 *means an unknown preference degree.*

3.2.2 Similarity computation

One critical step in the ΠCF approach is to compute the similarity between items and then select the most similar ones. These latter contribute more to predicting the target item preference degree. The basic idea in similarity computation between two items i_1 and i_2 is to first isolate the users who have both rated these items and then apply a similarity measure (as depicted by Figure 3.2). In the possibilistic framework, we will use a recent similarity measure for the comparison of uncertain information represented by possibility distributions, the so-called *information affinity* (Jenhani, Benferhat, & Elouedi, 2010). In fact, Information Affinity is proposed to overcome the weaknesses of the existing possibilistic similarity measures like information closeness, Sangüesa et al. distance and information divergence since they do not satisfy all the properties discussed in (Jenhani et al., 2010) such as:

- Non-negativity: $s(\pi_1, \pi_2) \geqslant 0$.
- Symmetry: $s(\pi_1, \pi_2) = s(\pi_2, \pi_1)$.
- Upper bound and non-degeneracy: $\forall \pi_i, s(\pi_i, \pi_i) = 1$ and $\forall \pi_i, \pi_j, s(\pi_i, \pi_i) \leq 1$. etc.

Therefore, information affinity chooses to combine two important criteria namely distance and inconsistency. This combination is justified by the fact that a distance measure taken alone does not always decide which is the closest distribution. Intuitively, information affinity takes into account the classical informative distance, e.g. Manhattan or Euclidean which evaluates the difference between two normalized possibility distributions and the inconsistency measure which evaluates the conflict between the possibility distributions.

Definition 2. **Information Affinity similarity measure:** *Let $\pi(i_n)$ and $\pi(i_{n'})$ be two possibility distributions representing respectively, the preference degrees associated to items i_n and $i_{n'}$. Information Affinity denoted by, $Aff(\pi(i_n), \pi(i_{n'}))$ is defined as follows:*

$$Aff(\pi(i_n), \pi(i_{n'})) = 1 - \frac{\kappa * d(\pi(i_n), \pi(i_{n'})) + \lambda * Inc(\pi(i_n), \pi(i_{n'}))}{\kappa + \lambda} \tag{3.1}$$

Where $\kappa > 0$ and $\lambda > 0$. d, denotes normalized metric distances between $\pi(i_n)$ and $\pi(i_{n'})$:

- *Euclidean distance:*

$$d(\pi(i_n), \pi(i_{n'})) = \frac{1}{M}\sqrt{\sum_{i=1}^{M}(\pi(i_n) - \pi(i_{n'}))^2} \tag{3.2}$$

- *Manhattan distance:*

$$d(\pi(i_n), \pi(i_{n'})) = \frac{1}{M}\sum_{i=1}^{M}|\pi(i_n) - \pi(i_{n'})| \tag{3.3}$$

$Inc(\pi(i_n) \wedge \pi(i_{n'}))$ denotes the degree of conflict between the two preference degrees (see Equation (1.6)) where $\wedge$ is taken as the product or min conjunctive operators. Intuitively, the use of min operator instead of the product operator gives less importance to the inconsistency degree, since $Inc(\pi(i_n) * \pi(i_{n'})) > Inc(min(\pi(i_n), \pi(i_{n'})))$.

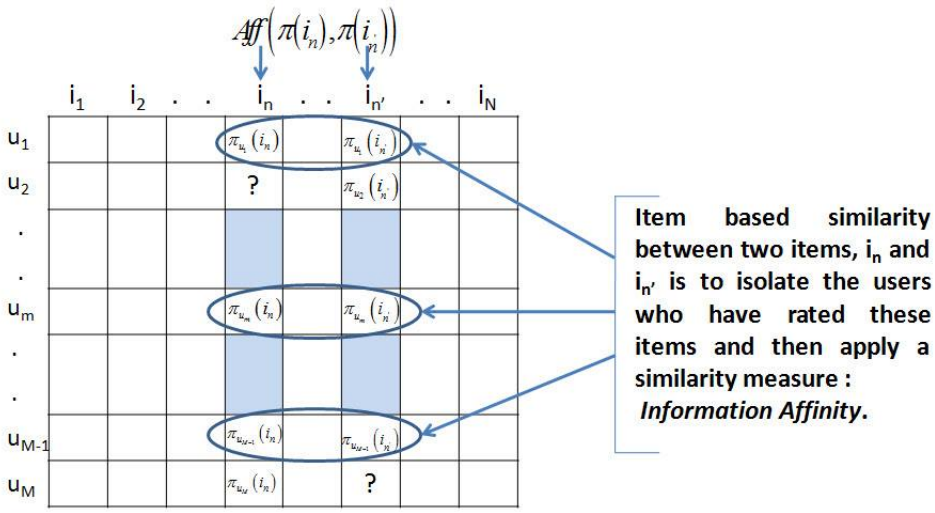


Figure 3.2: Isolation of the co-rated items and similarity computation.

The choice of parameters κ and λ depends on the problem under study. Generally, one gives the same importance to distance and inconsistency (i.e. $\kappa = \lambda$) when assessing the similarity between two possibilistic pieces of evidence. However, if someone wants to give more importance to distance than to inconsistency, he should set $\kappa > \lambda$ and vice-versa.

Algorithm 3.1 outlines the ΠCF similarity computation pseudo-code where M is a set of users and N is a set of items. P denotes a possibilistic user-item matrix. We denote by Aff a value determining how similar two items are to each other in order to generate recommendations and $LstAff$ a list of items similarities.

MANHATTAN-Distance, INCONSISTENCY and AFFINITY are the key functions of ΠCF similarity computation algorithm.

- MANHATTAN-Distance($\pi(i_n), \pi(i_{n'})$): calculates the distance between two items $\pi(i_n)$ and $\pi(i_{n'})$.
- INCONSISTENCY ($\pi(i_n), \pi(i_{n'})$): calculates the degree of conflict between preferences $\pi(i_n)$ and $\pi(i_{n'})$.
- AFFINITY($\pi(i_n), \pi(i_{n'})$): calculates similarities between items i_n and $i_{n'}$ based on the computation of MANHATTAN-Distance and INCONSISTENCY measures.

Algorithm 3.1 Similarity computation

```
input: Preferences matrix: P;
output: List of similarity measures between target item and other items;
forall u_m ∈ U do                          /* Manhattan distance */
forall i_n ∈ I do
   MANHATTAN-Distance (π(i_n), π(i_n'));
end for
end for
forall u_m ∈ U do                          /* Inconsistency */
forall i_n ∈ I do
   INCONSISTENCY (π(i_n), π(i_n'));
end for
end for
forall u_m ∈ U do                          /* Information Affinity measure */
forall i_n ∈ I do
   Aff ← AFFINITY (π(i_n), π(i_n'));
   LstAff.add(Aff);
end for
end for
```

Example 3.2. *Considering the preference matrix of Table* 3.1. *We aim to find the most similar items for* i_2 *using the Information Affinity similarity measure. By using the Manhattan distance,* $\wedge$ *as the minimum conjunctive operator and* $\kappa = \lambda = 1$. *we obtain:*

- $Aff(\pi_{i_2}, \pi_{i_1}) = 1 - \frac{d(\pi_{i_2},\pi_{i_1}) * Inc(\pi_{i_2},\pi_{i_1})}{2}$, *where:*
  - $d(\pi_{i_2}, \pi_{i_1}) = \frac{1}{3}(0.5 + 1 + 0.2) = 0.567$
  - $Inc(\pi_{i_2}, \pi_{i_1}) = 1 - max\{0.5; 0; 0\} = 0.5$

  $\Rightarrow Aff(\pi_{i_2}, \pi_{i_1}) = 1 - \frac{0.5+0.567}{2} = \mathbf{0.467}$
- $Aff(\pi_{i_2}, \pi_{i_3}) = 1 - \frac{d(\pi_{i_2},\pi_{i_3}) * Inc(\pi_{i_2},\pi_{i_3})}{2}$, *where:*
  - $d(\pi_{i_2}, \pi_{i_3}) = \frac{1}{3}(0.2 + 0.7 + 1) = 0.633$
  - $Inc(\pi_{i_2}, \pi_{i_3}) = 1 - max\{0.3; 0.3; 0\} = 0.7$

  $\Rightarrow Aff(\pi_{i_2}, \pi_{i_3}) = 1 - \frac{0.633+0.7}{2} = \mathbf{0.334}$
- $Aff(\pi_{i_2}, \pi_{i_4}) = 1 - \frac{d(\pi_{i_2},\pi_{i_4}) * Inc(\pi_{i_2},\pi_{i_4})}{2}$, *where:*
  - $d(\pi_{i_2}, \pi_{i_4}) = \frac{1}{4}(0.5 + 0.3 + 0.3 + 0.3) = 0.35$
  - $Inc(\pi_{i_2}, \pi_{i_4}) = 1 - max\{0.5; 0.7; 0.7; 0\} = 0.3$

  $\Rightarrow Aff(\pi_{i_2}, \pi_{i_4}) = 1 - \frac{0.35+0.3}{2} = \mathbf{0.675}$

As $Aff(\pi_{i_2}, \pi_{i_4}) > Aff(\pi_{i_2}, \pi_{i_1}) > Aff(\pi_{i_2}, \pi_{i_3})$, *we will consider the most similar items to* i_2, *namely* i_4 *then* i_1 *and finally* i_3.

3.2.3 Prediction and recommendation

The most important step in a collaborative filtering system is to generate the output interface in terms of prediction. Once we isolate the set of most similar items based on the *Information Affinity* similarity measure, the next step is to look into the target users preferences and use the weighted sum technique to obtain prediction as expressed in equation (3.4). $\widehat{\pi}_{u_m}(i_n)$ represents the prediction on item i_n for user u_m, it is computed by picking the k most similar items to the target item i_n. Formally:

$$\widehat{\pi}_{u_m}(i_n) = \frac{\sum_{Allsimilaritems} Aff(\pi(i_n), \pi(i_{n'})) * \pi_{u_m}(i_{n'})}{\sum_{Allsimilaritems} |Aff(\pi(i_n), \pi(i_{n'}))|} \tag{3.4}$$

Once unknown items are predicted, the recommendation step consists in choosing one or more items from a set of alternatives and sorting them in a descending order. In this context, there are various ways to present recommendations to the user either by offering (choosing) the best item, or by presenting the top-K items as a recommendation list, or by classifying items into categories, i.e.*'highly recommended'* , *'fairly recommended'* and *'not recommended'*.

Algorithm 3.2 presents the prediction and recommendation step. Intuitively, to predict a rating for an item a user has not seen before, the algorithm takes as input a list of similarity measures namely $lstAff$, then, by applying the prediction formula, it generates a prediction list $LstPrd$ sorted in a descending order. Intuitively, PREDICTION and Top-K Recommendation are the key functions of this algorithm:

- PREDICTION($\pi_{u_m}(i_n)$): calculates the preference of u_m about item i_n.
- Top-K Recommendation: identifies a set of K items that will be of interest.

Algorithm 3.2 Prediction and recommendation

```
input: LstAff (List of similarity measures between target item and other items);
output: Recommendation list;
LstAff, Lstpred: List
forall LstAff ∈ LstAff do                    /* Prediction */
   π̂_{u_m}(i_n) ←PREDICTION (π_{u_m}(i_n));
   LstPrd.add(π̂_{u_m}(i_n));
end for
forall π̂_{u_m}(i_n) ∈ LstPrd do              /* Top-K Recommendation */
   LstPrd.Sort(π̂_{u_m}(i_n));
end for
```

Example 3.3 *Let us consider the preferences matrix of Table* 3.1. *We want to compute the u_2's prediction for item i_2 and i_4 in order to recommend the most close item to user's taste. We have previously, $Aff(\pi(i_2), \pi(i_1)) = 0.467$, $Aff(\pi(i_2), \pi(i_3)) = 0.334$ and $Aff(\pi(i_2), \pi(i_4)) = 0.675$. Based on this, the prediction $\widehat{\pi}_{u_2}(i_2)$ is therefore computed as follows:*

$$\widehat{\pi}_{u_2}(i_2) = \frac{Aff(\pi(i_2),\pi(i_1))*\pi_{u_2}(i_1)+Aff(\pi(i_2),\pi(i_3))*\pi_{u_2}(i_3)+Aff(\pi(i_2),\pi(i_4))*\pi_{u_2}(i_4)}{Aff(\pi(i_2),\pi(i_1))+Aff(\pi(i_2),\pi(i_3))+Aff(\pi(i_2),\pi(i_4))}$$

$\Rightarrow \widehat{\pi}_{u_2}(i_2) = \mathbf{0.316}$

In the same manner, we select the two most similar items to i_4:

- $Aff(\pi_{i_4}, \pi_{i_1}) = 1 - \frac{d(\pi_{i_4}, \pi_{i_1}) * Inc(\pi_{i_4}, \pi_{i_1})}{2}$, *where:*
  - $d(\pi_{i_4}, \pi_{i_1}) = \frac{1}{3}(0 + 0.7 + 0.1) = 0.266$
  - $Inc(\pi_{i_4}, \pi_{i_1}) = 1 - max\{1; 0; 0.2\} = 0$

  $\Rightarrow Aff(\pi_{i_4}, \pi_{i_1}) = 1 - \frac{0.266+0}{2} = \mathbf{0.867}$
- $Aff(\pi_{i_4}, \pi_{i_2}) = 1 - \frac{d(\pi_{i_4}, \pi_{i_2}) * Inc(\pi_{i_4}, \pi_{i_2})}{2}$, *where:*
  - $d(\pi_{i_4}, \pi_{i_2}) = \frac{1}{4}(0.5 + 0.3 + 0.3 + 0.3) = 0.35$
  - $Inc(\pi_{i_4}, \pi_{i_2}) = 1 - max\{0.5; 0.7; 0.7; 0\} = 0.3$

  $\Rightarrow Aff(\pi_{i_4}, \pi_{i_2}) = 1 - \frac{0.35+0.3}{2} = \mathbf{0.675}$
- $Aff(\pi_{i_4}, \pi_{i_3}) = 1 - \frac{d(\pi_{i_4}, \pi_{i_3}) * Inc(\pi_{i_4}, \pi_{i_3})}{2}$, *where:*
  - $d(\pi_{i_4}, \pi_{i_3}) = \frac{1}{3}(0.7 + 0.4 + 0.7) = 0.6$
  - $Inc(\pi_{i_4}, \pi_{i_3}) = 1 - max\{0.3; 0.3; 0.3\} = 0.7$

  $\Rightarrow Aff(\pi_{i_4}, \pi_{i_3}) = 1 - \frac{0.6+0.7}{2} = \mathbf{0.35}$

Finally, we compute the prediction value of user u_2 for item i_4 as follows:

$$\widehat{\pi}_{u_2}(i_4) = \frac{Aff(\pi(i_4),\pi(i_1)) * \pi_{u_2}(i_1) + Aff(\pi(i_4),\pi(i_3)) * \pi_{u_2}(i_3) + Aff(\pi(i_4),\pi(i_2)) * \pi_{u_2}(i_2)}{Aff(\pi(i_4),\pi(i_1)) + Aff(\pi(i_4),\pi(i_2)) + Aff(\pi(i_4),\pi(i_3))}$$

$\Rightarrow \widehat{\pi}_{u_2}(i_4) = \mathbf{0.712}$

Item i_4 has the highest predicted preference and consequently it will be in the top one recommendation list for user u_2.

3.3 Conclusion

In this chapter, we presented the three steps of our proposed approach. The first step consists in building user-item preferences matrix. In the next step, a possibilistic similarity computation measure is applied to measure the weight (similarity) between items in order to generate the recommendation list of the final step. Next chapter provides an experimental study to evaluate our proposed approach comparing it with the traditional collaborative filtering approach.

Chapter 4

Experimental study

4.1 Introduction

In this chapter we present the experimentation results relative to our possibilistic collaborative filtering approach ΠCF comparing with traditional item-based collaborative filtering method. This chapter is composed of two Sections. Section 4.2 details the experimental protocol used in the implementation process of our approach, and Section 4.3 presents the experimental results evaluating the effectiveness of the recommendation based of proposed approach.

4.2 Experimental protocol

All our experiments were implemented using java language and compiled in Netbeans framework. We ran all our experiments on a windows 7 based PC with intel Core $i3$ processor having a speed of 1.7 GHz and 4 GB of Ram. This section describes our experimental data, then it presents the evaluation metrics that will be used to evaluate this experiment. Finally, it presents the procedure that we have followed to implement the ΠCF approach.

4.2.1 Data sets

To evaluate the effectiveness of ΠCF, we choose to work on the well-known recommendation data sets movieLens available through the movieLens[1] website. MovieLens is a free service provided by GroupLens [2] Research at the University of Minnesota. It has three available data sets: one with $100K$ ratings of movies, another with $1M$ ratings, and a third containing $10M$ applying tags to movies. This data set helps users to find movies they will certainly like. It is made up of a set of users preferences about movies. These preferences are user-provided star ratings, from 1

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

(dislike) to 5 (like) stars. In our approach, we will convert these preferences into possibility degrees between 0 (not satisfied) and 1 (fully satisfied) by dividing the rating by 5 then getting a normalized distribution between [0, 1]. The dataset contains in total 100.000 ratings collected by 943 users on 1682 movies, from 19-09-1997 to 22-04-1998. Each user has rated at least 20 movies. The dataset is divided into 2 parts, 80% of the data is used to train the recommender system (the training set) and 20% are used to evaluate the approach (the test set). MovieLens data are represented as a sequence of events in the following way:

- user u_1 rates movie i_1 with 1,
- user u_1 rates movie i_3 with 0.5,
- user u_2 rates movie i_1 with 0.6, etc.

In order to evaluate both of the effectiveness and efficiency of our proposed ΠCF approach we compare its performance to the traditional item-based collaborative filtering using Pearson correlation similarity measure available from Apache mahout library in Java[3].

4.2.2 Evaluation metrics

In order to evaluate the performance of recommender systems, several metrics have been proposed. According to (Cremonesi, Turrin, Lentini, & Matteucci, 2008) the evaluation metrics can be classified into three categories: *Predictive accuracy metrics*, *classification accuracy metrics* and *rank accuracy metrics*. We introduce the commonly used CF metrics of each class.

1. **Predictive accuracy metrics:** measure how much the prediction p_i is close to the true numerical rating r_i expressed by the user. The evaluation can be done only for items that have been rated.

   - *Mean Absolute Error (MAE)* takes the mean of the absolute difference between each prediction and preference degree for all held-out preference degrees of users in the testing set. The lower the *MAE* the more accurately the recommendation engine predicts user ratings. Formally:

   $$MAE = \frac{\sum_{u_m,i_n} |\widehat{\pi}_{u_m}(i_n) - \pi_{u_m}(i_n)|}{N}, \tag{4.1}$$

   where N is the total number of preferences over all users, $\widehat{\pi}_{u_m}(i_n)$ is the predicted preference degree for user u_m on item i_n, and $\pi_{u_m}(i_n)$ is the actual preference.

   - *Normalized Mean Absolute Error (NMAE)* normalizes *MAE* to express errors as percentages:

   $$NMAE = \frac{MAE}{\pi_{max} - \pi_{min}}, \tag{4.2}$$

   where π_{max} and π_{min} are the upper and lower bounds of the preferences.

2. **Classification accuracy metrics:** evaluate how predictions help the active user in distinguishing good items from bad items. Therefore, it is useful in finding if the active user will like or not the current item. With classification metrics recommendation can be classified as:

[3]https://mahout.apache.org

- *True positive (TP)*: an interesting item is recommended to the user.
- *True negative (TN)*: an uninteresting item is not recommended to the user.
- *False negative (FN)*: an interesting item is not recommended to the user.
- *False positive (FP)*: an uninteresting item is recommended to the user.

Precision and recall are the most popular metrics in the classification accuracy metrics. They are computed from a 2×2 table, such as the one shown in Table 4.1.

Precision: is used to evaluate the validity of a given recommendation list and is defined as the ratio of all recommended items that are used. In fact, if an algorithm has a measured precision of 80%, then the user can expect that, on average, 8 out of every 10 movies returned to the user will be used. A perfect precision score of 1.0 means that every item recommended in the list was good.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

Recall: computes the ratio of all used items that were recommended for active user relative to the total number of the objects actually collected. A perfect recall score of 1.0 means that all good recommended items were suggested in the list.

$$Recall = \frac{TP}{TP + TN} \tag{4.4}$$

Consequently, we exploit preference degrees in the whole recommendation process, to thus measure, among the user's preferences, the number of items that are evaluated as used and / or the number of items that are evaluated as not used by the recommender.

F-measure: considers both precision and recall measures of the test to compute the score. We interpret it as a weighted average of the precision and recall, where the best F-measure has its value at 1 and worst score at the value 0. It is obtained combining both the precision and recall measures and indicates an overall utility of the recommendation list.

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{4.5}$$

Table 4.1 shows the classification of recommendation of an item to a user where N is the number of items in the database.

	Recommended	Not recommended	Total
Used	TP	TN	TP+TN
Not used	FP	FN	FP+FN
Total	TP+FP	TN+FN	N

Table 4.1: Classification of the possible result of a recommendation of an item to a user.

We will use as our choice for predictive accuracy metrics the MAE ans NMAE measures and for classification accuracy metrics we choose the *Precision*, *Recall* and *F-measure* of evaluation metrics to report prediction experiments because they are the most commonly used in information retrieval (Herlocker, Konstan, Terveen, & Riedl, 2004).

4.2.3 Implementation

Our goal is to recommend top K movies for the target user. In what follows, we present the implementation protocol for each step:

1. *Preferences representation step:* The idea is to create a user-item matrix from the user preference data and then to predict the missing entries by finding patterns from the user preference information.

   In this step, the moviesLens ratings file is the most interesting since it's the main input to our recommendation where each line has the format:

   UserID::MovieID::Preferences::Timestamp

   where

   - UsersIDs are integers.
   - MovieIDs are integers.
   - Ratings are in [0,1]. Mathematically, we convert the user rating from a certain space where preferences are in [1, 5] to an uncertain space in [0,1], by dividing each user rating by 5.

   The predicted preference computations are obtained from the training set and the evaluation of the efficiency of recommended items is performed by the testing set.

2. *Similarity computation step:* In this stage, we compute the number of times each pair of items occurs together. In order to evaluate both of the effectiveness and efficiency of our proposed ΠCF approach we compare its information affinity similarity measure to the Standard Collaborative Filtering (denoted SCF for short) using Pearson correlation-based similarity and Vector cosine-based similarity measures as described in chapter 1 (Equation (1.1) and (1.2)). For each similarity measures, we implement the different algorithms to compute the neighborhood. The implementation is based on Mahout Library tools, which provide an open source java package for recommendation task.

3. *Prediction and recommendation step:* Once we know how similar the items are, we can then predict the target user's preference towards a subset of other items he/she has not seen before. Intuitively, we want to predict a rating for an item a user has not seen before based on the information gathered from other users. Consequently, to do this we look at all items that are similar to the unrated item, then we multiply the user column vector with each item row vector. The sum creates a rating for each item relative to the user. We can then select the top K most highly rated items to recommend to the user.

4.3 Experimental results

We perform our experiments by computing the MAE, NMAE, precision, recall and F-measure of the generated recommendations using ΠCF and item-based CF. We study the behavior of our approach using the possibility measure information affinity and standard ones pearson correlation and vector cosine. The obtained results are summarized in Tables 4.2 and 4.3. Figure 4.1 shows a capture screen of the Top-K recommendation on the MovieLens data.

Figure 4.1: A capture screen showing the Top-K recommendation on the MovieLens data.

Prediction: As shown in Table 4.2, the mean error is higher when using the traditional item-based CF approach for both of Pearson similarity (equal to 0.83) and Cosine similarity (equal to 0.82) measures. Thus, it can be observed from the results that the user-average error for information affinity computation (equal to 0.149) has a clear advantage, as the MAE is significantly lower in this case. Similarly, the NMAE information affinity measure has the lowest value (equal to 0.186) despite those of pearson similarity (equal to 0.207) and cosine similarity (equal to 0.205).

Approach	Similarity measure	MAE	NMAE
ΠCF	*Information affinity*	**0.149**	**0.186**
SCF	*Pearson similarity*	0.83	0.207
	Cosine similarity	0.82	0.205

Table 4.2: The MAE and NMAE accuracy

Recommendation: We exploit preference degrees in the whole recommendation process, we thus measure, among the user's preferences, the number of items that are evaluated as relevant and / or irrelevant by the recommender. These measures are computed twice: when using traditional item-based CF approach and our approach ΠCF. The obtained results are summarized in Table 4.3.

Approach	Classification accuracy measure	
SCF	*Precision*	0.098
	Recall	0.106
	F-measure	0.102
ΠCF	*Precision*	**0.433**
	Recall	**0.787**
	F-measure	**0.538**

Table 4.3: The precision, recall and F-measure of the two approaches

From Table 4.3, results show that our approach outperforms SCF in terms of precision, recall and F-measures. In fact, we pinpoint that the precision of our approach is equal to 0.433, which is higher than that of SCF (equal to 0.098). Similarly, ΠCF's recall and F measure are largely higher than SCF's ones. Intuitively, the precision value equal to 0.433 means that on average 5 of 10 recommendations are good . Recall is 0.787 so on average about 8 of 10 are good recommendations among those top recommended. In addition to precision and recall, the F-measure value 0.538 indicates an overall utility of the recommendation list compared to SCF's one as shown in Figure 4.2.

This is explained by the fact that the *possibility theory*, especially the use of information affinity similarity measure has a considerable effect on the quality of the recommendation rather than the traditional absolute ratings. This confirms that considering uncertainty for the beginning of the process has a great impact on recommendation results.

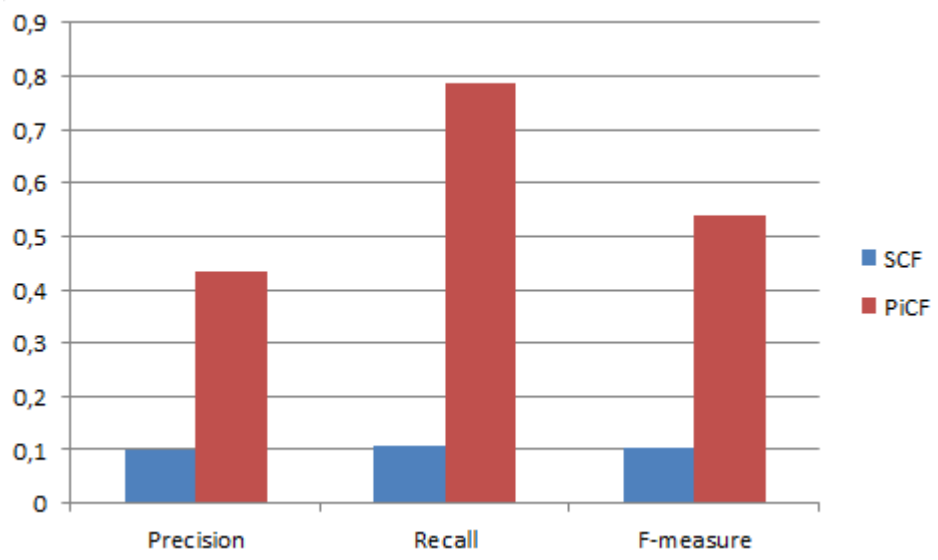


Figure 4.2: Evaluation of classification accuracy metrics: Precision, recall and F-measure.

4.4 Conclusion

The experimental study provided in this chapter, shows that our proposed ΠCF approach gives motivating results comparing to traditional item-based collaborative filtering approach. Consequently, the use of information affinity measure shows a significant performance, giving higher results in terms of MAE, NMAE, precision, recall and F-measure comparing to those of SCF.

Conclusion

Recommender systems are proving to be a useful tool for generating recommendations. Their evolution has accompanied the evolution of the web. They represent a powerful method for enabling users to filter through large information and product spaces. Consequently, they have been attacking the interest of researchers during the last decade.
With the aim to enhance the accuracy and the performance of the existing recommendations, many attempts have incorporated the recommendation with certain absolute ratings. In fact, in general uncertainty occurs whenever information pertaining to a situation is incomplete, contradictory or fluctuating. Intuitively, uncertainty can't be ignored in real word problems, but there is almost no research work addressing this issue in the recommender systems framework, especially that relates to users ratings preferences.
Representing the uncertain aspect of users' preferences is crucial in recommendation systems but it seems harder to ensure with the current recommendation methods, because most of them rely on certain absolute ratings, no research studied a purely uncertain recommender system dealing with uncertain ratings as an input.

In this work, we have proposed a new collaborative filtering recommender under uncertainty, which uses the possibility theory framework in order to cope with the uncertainty that may pervade users ratings. We initially modeled the uncertain aspects of user preferences in a user-item matrix. Then, we computed the similarity between items using a purely possibilitic similarity measure, namely Information Affinity .Therefore, we estimated preference of each target user towards an item he/she has never seen before. Finally, we generated the top K most similar items for the target user.

The experimental results presented in this report are very promising and are proving the validity of these kind of uncertain relations and the benefits of their use to improve the accuracy of recommendations system. In fact, we showed that the prediction and recommendation performance of ΠCF are clearly superior to those of the traditional method. This confirms that considering uncertainty for the beginning of the process has a great impact on recommendation results. Consequently, we have shown that using a purely uncertain user-item preferences matrix in the context of recommender systems presents a significant improvement on recommendation, therefore, it outperforms the traditional collaborative filtering one. Our proposed approach shows that uncertainty is an ubiquitous aspect in building recommender systems and taking into account such aspect predicts more accurate items. In addition, our ΠCF is able to overcome the problem of sparsity data corresponding to the lack of rating information which seems even harder to handle with the existing standard collaborative filtering recommendation techniques and which are crucial for effective recommendation.

As a future work, we will address preference relations using the qualitative aspect of the possibility theory

framework. It includes comparing the robustness and stability of uncertain preference relations instead of absolute ratings and exploring a purely possibilistic qualitative similarity measure to compute the similarity between preference relations.

References

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132.

Brun, A., Hamad, A., Buffet, O., & Boyer, A. (2010). Towards preference relations in recommender systems. In *Workshop on preference learning, european conference on machine learning and principle and practice of knowledge discovery in databases (ecml-pkdd 2010).*

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), 331–370.

Cremonesi, P., Turrin, R., Lentini, E., & Matteucci, M. (2008). An evaluation methodology for collaborative recommender systems. In *Automated solutions for cross media content and multi-channel distribution, 2008. axmedis'08. international conference on* (pp. 224–231).

Cvengroš, P. (2011). *Universal recommender system* (Unpublished doctoral dissertation).

de Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Using second-hand information in collaborative recommender systems. *Soft Computing*, *14*(8), 785–798.

Desarkar, M. S., Saxena, R., & Sarkar, S. (2012). Preference relation based matrix factorization for recommender systems. In *User modeling, adaptation, and personalization* (pp. 63–75). Springer.

Dubois, D., & Prade, H. (1998). Possibility theory: qualitative and quantitative aspects. In *Quantified representation of uncertainty and imprecision* (pp. 169–226). Springer.

Dubois, D., & Prade, H. (2011). Possibility theory and its applications: Where do we stand. *Mathware and Soft Computing*, *18*(1), 18–31.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, *4*(2), 81–173.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1),

5–53.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Data mining, 2008. icdm'08. eighth ieee international conference on* (pp. 263–272).

Jenhani, I., Benferhat, S., & Elouedi, Z. (2010). Possibilistic similarity measures. In *Foundations of reasoning under uncertainty* (pp. 99–123). Springer.

Jin, R., Si, L., Zhai, C., & Callan, J. (2003). Collaborative filtering with decoupled models for preferences and ratings. In *Proceedings of the twelfth international conference on information and knowledge management* (pp. 309–316).

Kamishima, T. (2003). Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining* (pp. 583–588).

Koren, Y., & Sill, J. (2013). Collaborative filtering on ordinal user feedback. In *Proceedings of the twenty-third international joint conference on artificial intelligence* (pp. 3022–3026).

Linqi, G., & Li, C. (2008). Hybrid personalized recommended model based on genetic algorithm. In *Wireless communications, networking and mobile computing, 2008. wicom'08. 4th international conference on* (pp. 1–4).

Liu, N. N., & Yang, Q. (2008). Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st annual international acm sigir conference on research and development in information retrieval* (pp. 83–90).

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook* (pp. 73–105). Springer.

Nair, R., & Kelkar, K. (2013). Implementation of item and content based collaborative filtering techniques based on ratings average for recommender systems. *International Journal of Computer Applications (0975-8887)*, *65*(24).

Park, M.-H., Hong, J.-H., & Cho, S.-B. (2007). Location-based recommendation system using bayesian users preference model in mobile devices. In *Ubiquitous intelligence and computing* (pp. 1130–1139). Springer.

Price, R., & Messinger, P. R. (2005). Optimal recommendation sets: Covering uncertainty over user preferences. In *Aaai* (Vol. 10, p. 5).

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2011). *Recommender systems handbook* (Vol. 1). Springer.

Roh, T. H., Oh, K. J., & Han, I. (2003). The collaborative filtering recommendation based on som cluster-indexing cbr. *Expert Systems with Applications*, *25*(3), 413–423.

Samia, B., Allel, H., & Aicha, A. M. (2014). Handling preferences under uncertainty in recommender systems. In *Fuzzy systems (fuzz-ieee), 2014 ieee international conference on* (pp. 2262–2269).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on world wide web* (pp. 285–295).

Smets, P., & Kennes, R. (1994). The transferable belief model. *Artificial intelligence*, *66*(2), 191–234.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*, 4.

Xiang, C., Guisheng, Y., Long, Z., & Yongjin, K. (2013). Method of collaborative filtering based on uncertain user interests cluster. *Journal of computers*, *8*(1), 186–193.

Yager, R. R. (2003). Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems*, *136*(2), 133–149.

Yu, K., Schwaighofer, A., Tresp, V., Xu, X., & Kriegel, H.-P. (2004). Probabilistic memory-based collaborative filtering. *Knowledge and Data Engineering, IEEE Transactions on*, *16*(1), 56–69.

Yu, L., & Yang, X. (2008). Collaborative filtering recommendation based on preference order. In *Research and practical issues of enterprise information systems ii* (pp. 1567–1573). Springer.

Yuan, X., Huang, J.-j., & Zhong, N. (2013). Preference structure and similarity measure in tag-based recommender systems. In *Active media technology* (pp. 193–202). Springer.

Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, *100*, 9–34.

Zenebe, A., & Norcio, A. F. (2009). Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems. *Fuzzy Sets and Systems*, *160*(1), 76–94.