Incorporating User Rating Credibility in Recommender Systems

By

Naime Ranjbar Kermany

A thesis submitted to Macquarie University
for the degree of
Master of Research
Department of Computing
May 2019

MACQUARIE UNIVERSITY
FACULTY OF SCIENCE

Typeset in LaTeX 2_ε.

Declaration

I certify that the work in this thesis entitled INCORPORATING USER RATING CREDIBILITY IN RECOMMENDER SYSTEMS has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree to any other university or institution other than Macquarie University. I also certify that the thesis is an original piece of research and it has been written by myself. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Naime Ranjbar Kermany

Dedication

This thesis is dedicated to my **parents**
for their love, endless support,
and encouragement.

Acknowledgements

I would like to gratefully thank the many people who have helped, encouraged, and supported me through this one year spent in finishing this work. The first person to thank is my supervisor, Prof. Jian Yang, for her guidance, friendly manner, patience, and knowledge throughout this work. She has always been there to listen and give advice.

I would also like to thank Dr. Weiliang Zhao for all his immediate support, inspiration, encouragement and suggestions.

In addition, I would like to acknowledge the Macquarie University for providing me with the scholarship. Thanks extended to the Higher Degree Research department and the School of science and Engineering.

Last but not the least, I would like to thank my family and friends for their support throughout my candidature.

List of Publications

- Naime Ranjbar Kermany, Weiliang Zhao, Jian Yang, and Jia Wu; *Incorporating User Rating Credibility in Recommender Systems*. (submitted to IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING)
- Naime Ranjbar Kermany, Weiliang Zhao, Jian Yang, and Jia Wu; *ECFRIURC: Enhancing Collaborative Filtering Recommendations by Incorporating User Rating Credibility*. (submitted to IJCAI 19)

Abstract

There has been a lot of research efforts aimed at improving the recommendation accuracy with Collaborative Filtering (CF); yet there is still a lack of investigation into the integration of CF algorithms with the analysis of users' rating behaviors. Considering that by incorporating the rating credibility, the impact of the ratings given by neighbors with low credibility should be decreased. In this work, we develop an integrated solution for CF recommendation by incorporating the credibility of users' ratings, demographic information of the people, and ontological semantics of items. The demographic information of users and ontological semantics of items are used in the similarity measurement of users/items to alleviate the issues of sparsity and cold-start in CF algorithms. To our knowledge, this is the first time an integration of the rating credibility, demographic information of users, and ontological semantics of items is created in order to improve the performance of CF recommendation system. Experiments are conducted on the real-world datasets of MovieLens and Yahoo!Movie. Comparing with baseline methods, the experimental results show that the proposed approach significantly improves the quality of recommendation in terms of accuracy, *precision*, *recall*, *F-measure*, and standard deviation of the errors.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Problem Statement and Motivation

Quite often we see "- People who watched this movie also watched ... - According to your recent browsing history you may like ... - Because you bought this product you may also buy ...". These common quotes on e-commerce websites have been successful to draw users' attention with the aim of providing recommendations. In fact, with the increasing number of choices on the internet, recommender systems play a significant role in helping users to find their desired items [1]. Recommender Systems are able to predict the ratings and generate personalized recommendations for users through various type of approaches (see Section 2.1). A widely-used approach in recommender systems is collaborative filtering (CF) [1].

CF has been widely adopted due to its good performance, flexible implementation, and the

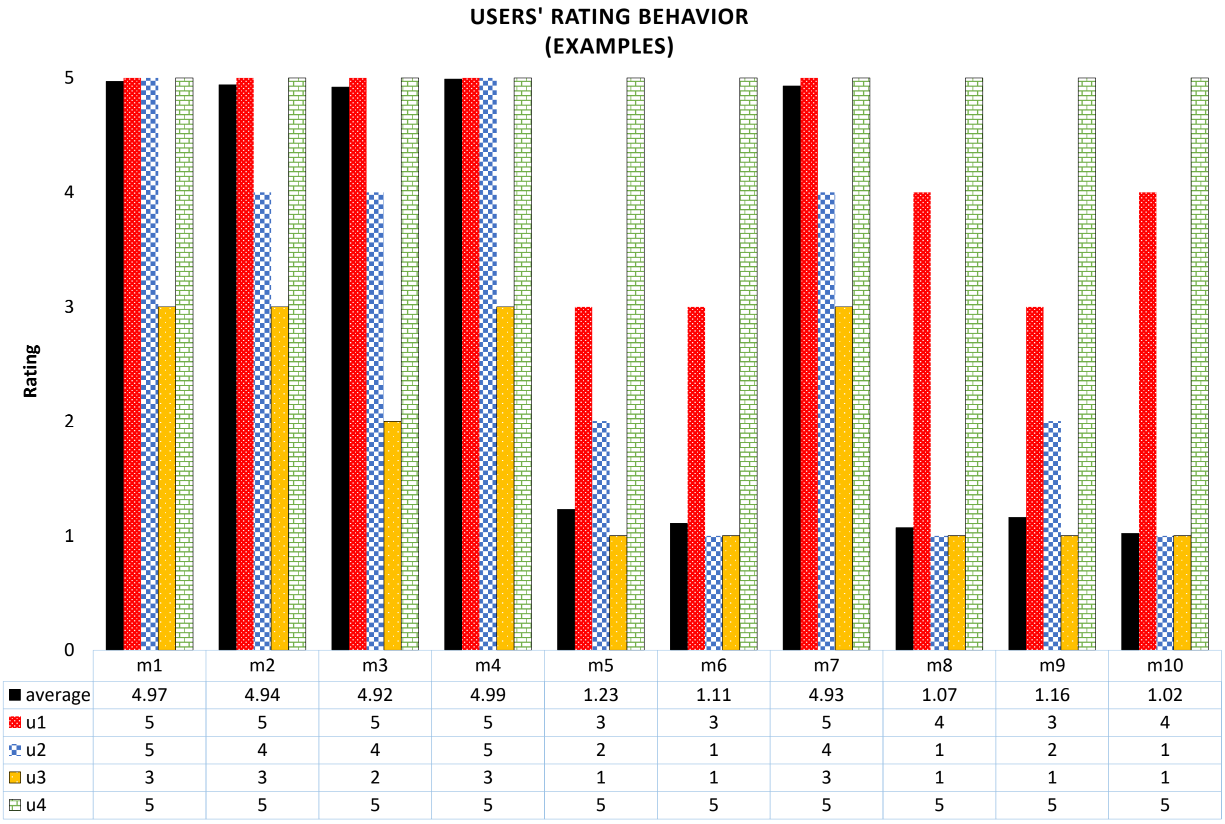


FIGURE 1.1: Simple-to-Understand Examples of Various Users' Rating Behavior

capability to cover the latent relations between users and items [1, 2]. However, one big issue that has not been touched upon is *how users' rating behavior shall be treated*, taken as they are, ignored, or if weighted, how? In recommender systems, users' rating scores vary, i.e., some users' scores are widely distributed while others fall in a small range. Existing CF recommendation approaches largely ignore such differences. In the CF approach, all users' ratings are treated as they are, and inappropriate and inaccurate ratings unavoidably affect the recommendation results. For instance, the user who gives the same score (e.g. 5.0) to all the items is not worth taking seriously [3]. Figure 1.1 shows some simple-to-understand examples of various users' rating behavior. Assume "$average$" as the average of ratings by people and experts (e.g. IMDB ratings) and $u1$, $u2$, $u3$, and $u4$ as four users of the system who have given ratings to 10 movies. These movies are the most popular movies ($m1$, $m2$, $m3$, $m4$, $m7$) and the least popular movies ($m5$, $m6$, $m8$, $m9$, $m10$) to ensure simplicity of the example. As it can be seen, $u1$ is a generous user as she/he always has given top ratings to the movies, even the least popular ones; $u2$ is a person whose ratings are mostly close to the $average$, referred as a credible user; $u3$ is a strict user as she/he always has given low ratings to the movies, even the most popular ones; and $u4$ always has given 5.0 (the top score) to all movies, that shows his/her ratings should not be considered as serious as a credible user. Thus, one important challenge is *how users' credibility/reliability can be measured according to their rating behavior*.

While the credibility of users' ratings plays a significant role, the quantity of ratings given by the users also affect the accuracy of a CF recommender system. The CF approach suffers from the lack of data/ratings resulting in two major problems, namely the sparsity and cold-start. The cold-Start problem occurs when the system does not have the knowledge to accurately recommend existing items to a new user [4]. As for the sparsity problem, it happens when the number of available ratings is much smaller than the number of possible ratings [1]. These issues also have a serious effect on the quality and accuracy of the recommendations.

In this work, we have designed an integrated CF recommendation system by incorporating the credibility of users' ratings, the demographic information of users and the ontological semantics of items. Thus, measuring the user's credibility and addressing the sparsity and cold-start problems associated with CF recommendation systems at the same time in situations where all these problems are present. To the best knowledge of the authors, there is no other integrated solution to answer all of the above problems simultaneously. It is also for the first time that credibility of users' ratings has been incorporated to CF approach to optimize the neighbors and to study its effect on the performance of CF recommendation. Our system is further described in the following section.

1.2 Solution and Contribution

As part of our solution, in this work, we developed an algorithm to measure the credibility of users by taking both rating behavior of users and their rating distributions into consideration because we expect that by incorporating the user's rating credibility, the accuracy and quality of CF recommendation will be improved. This is because the recommendation accuracy of the CF method relies highly on the neighbors for a target user. Thus, the credibility values incorporated in

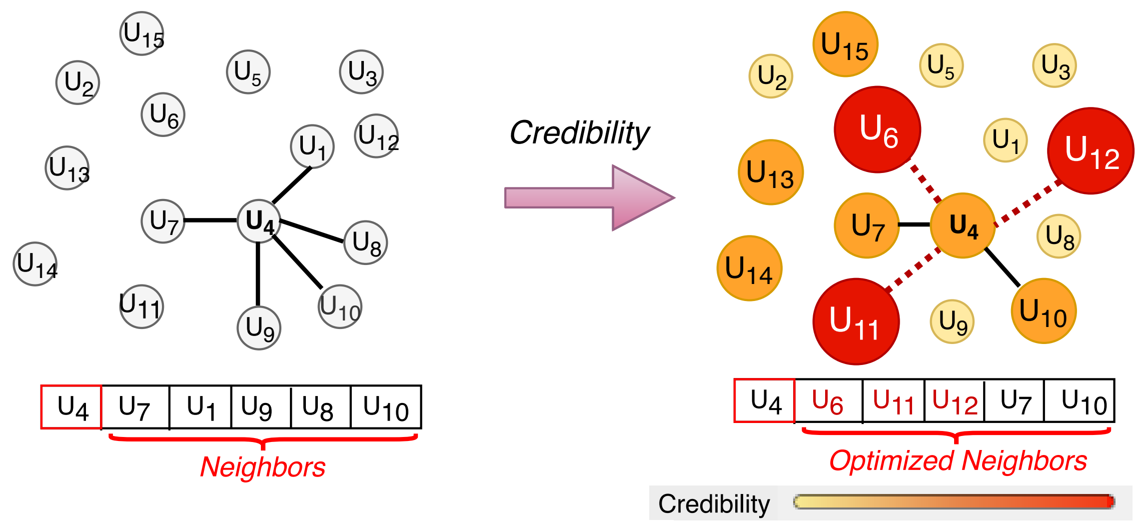


FIGURE 1.2: An Example of Neighbor Selection With and Without Considering Credibility When Number of Neighbors = 5; each circle show a user, and the size and color of circles show the degree of credibility

the CF algorithm helps to decrease the impact of the ratings given by neighbors with low credibility.

Figure 1.2 shows an example of selecting top-5 neighbors with and without considering credibility; where each circle shows a user, and the size and color of circles show the degree of credibility for users. From Figure 1.2, it can be observed that the top-5 neighbors of $u4$ are {u7,u1,u9,u8,u10}. However, after considering the credibility of users, $u1$, $u9$, and $u8$ are not selected as the neighbors any more, while $u6$, $u11$, and $U12$ are now selected as neighbors. Thus, the optimized neighbors for $u4$ are {u6,u11,u12,u7,u10} due to the credibility of users.

In addition, we also take into account the ontological semantics of items and the demographic information of users in the recommendation. This additional information provides an external level of knowledge for the system to address the sparsity and cold-start problems and accordingly help to improve the accuracy of CF recommendation.

Three issues are to be addressed in this work: (1) how the credibility of each user is calculated and incorporated into the recommendation system; (2) how the semantic and demographic similarities are aggregated with the CF similarities; and (3) how (1) and (2) are integrated. We carry out experiments using three real-world datasets, MovieLens 100K, MovieLens 1M and Yahoo!Movie, and compare our results with several existing studies. The contributions of this work are summarized as follows:

- developing functions to measure the similarity for items and users based on the ontological semantics of items and demographic information of users;

- proposing a weighting method to measure the credibility of users according to their rating behavior;
- building a unified CF recommendation solution by incorporating the credibility of users and the similarities of items and users;
- carrying out a set of experiments against real-world datasets from MovieLens and Yahoo!Movie to show the significant improvement of recommendation quality in terms of *accuracy*, *precision*, *recall*, and *F-measure* compared with baseline methods.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2, recommender system and its various techniques will be discussed briefly. Then, a literature review of CF approach, neighbor-based CF approach, the users' credibility, ontology, and demographic information are represented. In Chapter 3, we will present the methods used for the proposed unified CF recommendation solution. In Chapter 4, we will first explain the datasets and evaluation metrics. Next, the various baselines will be explained separately. Finally, the results and discussion will be presented. Chapter 5 will provide a conclusion and discussion of potential future work.

2

Literature Studies and Related Work

2.1 Recommender Systems

Nowadays, e-commerce has become increasingly popular in providing a broad range of products/services through the Internet and online web pages. However, offering more items does not necessarily infer that people will have the perceptive aptitude or enough time to consider them all as alternatives. In fact, it is shown that as the number of items increases a threshold, information overloading takes over, initiating a sequence of negative effects to users. A crucial issue for a person who intends to purchase online is *how to efficiently choose the desired products/services among the pool of options* [1]. This motivates the study of "recommender systems", which have emerged as a solution to address the information overload problem [1]. These systems provide personalized recommendations so that users will receive the most relevant recommendations according to their preferences [1]. Recommender systems are being increasingly applied in various domain applications such as the recommendation of movies [5–10], friends [11–13], Web pages [14, 15], music [16–19], books [20–22], tourism [23–25], documents [26, 27], and news [28–30].

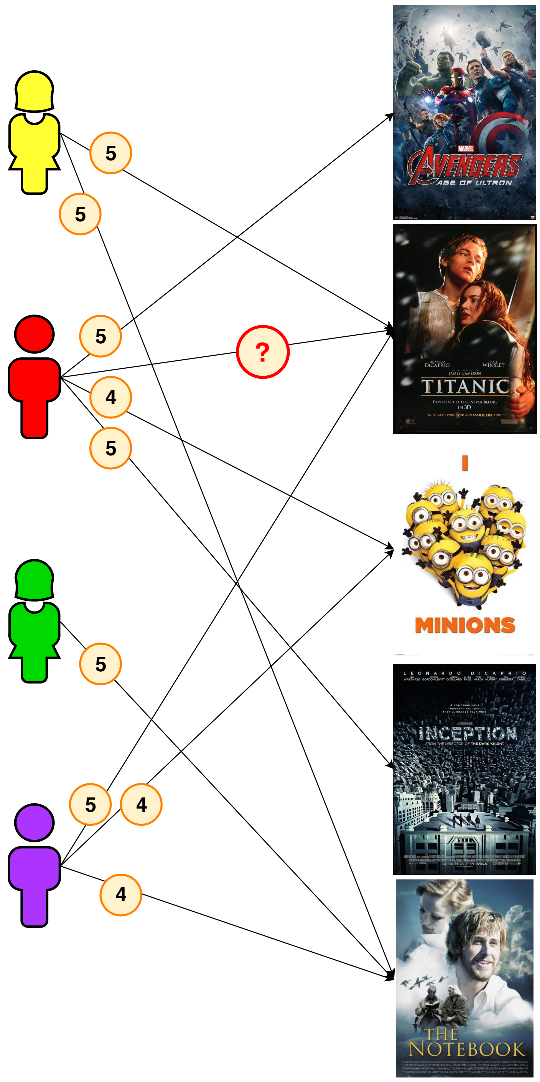


FIGURE 2.1: An Example of a Movie Recommender System with four Users and five Movies

As an example, we will discuss a simple movie recommender system with four users, who have rated five movies. Figure 2.1 shows a user-movie graph, where each user's rating for each movie is shown on the edges of the graph. The main goal of the recommender system is to accurately predict the rating of those movies for which a user has not watched yet (with no rating value at all). Here, the recommender system tries to predict the rating of user 2 on the movie *Titanic* that is shown with a question mark in Figure 2.1. Assuming a 5-star scale for ratings, if the recommender system predicts this rating to be higher than 4.0, then, movie *Titanic* could be recommended to the target user 2.

Recommender systems are employed as algorithms to recommend the most desirable items to a target user [31]. To do so, the items which are worth suggesting must be recommended based on a prediction of items' utility values. In a mathematical definition, a utility function ($R : U \times I \rightarrow R_o$) investigates whether the recommending item ($i \in I$) is suitable for the user or not ($u \in U$); "U" and "I" represent a set of users and items respectively; R_o is the overall rating which is normally an integer in a bounded interval [6]. Recommender system predicts the utility function $R(u, i)$ for user u on item i, and then recommend a top-K list of the items with the largest utility values [1]. To do so, various recommendation techniques have been introduced, as shown in Figure 2.2.

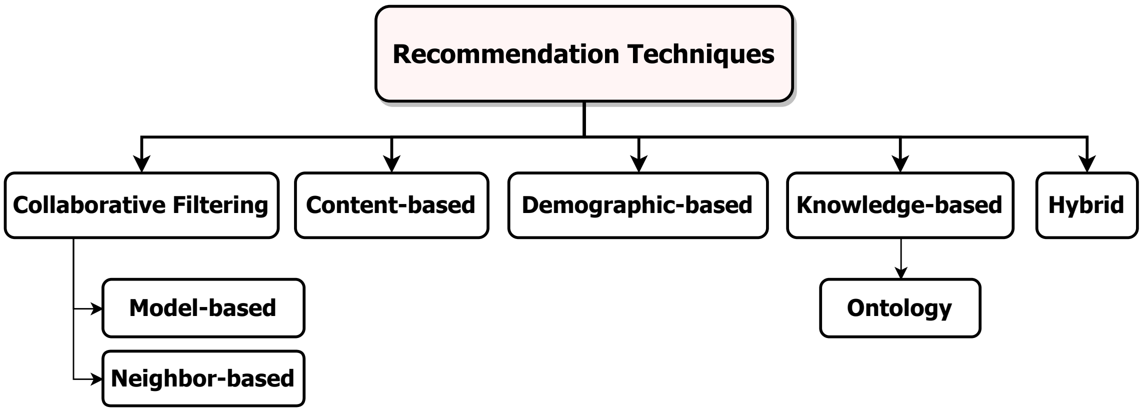


FIGURE 2.2: Various Techniques of Recommender Systems

- Collaborative Filtering: Collaborative Filtering (CF) recommender systems suggest items based on the preferences of a collection of users with similar tastes [1]. The CF approach is divided into two categories [32]: model-based CF and neighbor-based CF (or memory-based CF). Model-based approaches try to learn a predictive model from the ratings and use this model to predict unknown ratings [1]. Neighbor-based approaches predict unknown ratings based on the similarity values between users (user-based CF) or items (item-based CF) [33].

- Content-Based: Content-Based (CB) recommender systems generate recommendations explicitly with the use of the content and features of the items [1].

- Demographic-based: Demographic-Based (DB) recommender systems provide recommendations based on the demographics of users such as age and gender [1].
- Knowledge-based: Knowledge-Based (KB) recommender systems suggest items to target users based on domain knowledge about how items meet users' preferences [1]. The most well-known knowledge domain is *ontology*.
- Hybrid recommender systems: Hybrid recommender systems are based on the combination of any two or more of the above-mentioned techniques [1].

Regardless of which technique is used to provide recommendations, selecting the accurate items to be recommended is technically challenging in terms of which user is more credible and what additional information is useful to enhance the performance of recommender systems. In the following, we are going to discuss the CF approach and its problems with more detail.

2.2 Collaborative Filtering Recommender System

In recommender systems, the user's preferences can be revealed from their interaction explicitly (e.g. from ratings for items) or implicitly (e.g. from user activities and behaviors). In Collaborative Filtering approaches, the interactions between users and items are formed explicitly as a rating matrix $R^{m \times n}$, where m is the number of users and n is the number of items. Most of the elements in a rating matrix are zero. The elements with non-zero values are the known interactions for those users and items. The benefit of forming the interactions as a matrix is that algebraic techniques can easily be applied to it [34]. The CF personalized recommendation is based on the prediction

	Avengers	Titanic	Minions	Inception	The Notebook
$User_1$	0	5	0	0	5
$User_2$	5	?	4	5	0
$User_3$	0	0	0	0	5
$User_4$	0	5	4	0	5

FIGURE 2.3: User-Item Rating Matrix Representation for the Movie Recommender System Example

of the zero elements of the rating matrix. Our example of Figure 2.1 can be represented with a user-item rating matrix, which is shown in Figure 2.3.

2.2.1 General Problems in Collaborative Filtering

Although CF recommender systems have been applied in many applications, they face common and challenges of "the lack of data/information" [1]. A good recommender system needs sufficient information about the interest of users in items to generate proper recommendations. The lack of data would result in well-known challenges in a recommender system, namely *sparsity* and *cold start*.

- Sparsity: The number of users and items in e-commerce recommendation systems are quite large [1]. Even users that are very active would rate only a few of the total number of items. Similarly items, even though popular, have been rated by only a few of the total number of users in the system. Therefore, we are likely to have a sparse rating matrix with a large number of rows and columns, most of which are zeros. This problem, referred to as sparsity, is one of the main technical limitations of recommender systems. In a sparse ratings matrix, a recommender system is not able to find appropriate neighbors and fails to provide proper recommendations.

  Sparsity is defined as a measure of the density of available ratings, Eq. 2.1.

$$Sparsity = 1 - \frac{number\_of\_available\_ratings}{number\_of\_all\_possible\_ratings} \tag{2.1}$$

- Cold start: The cold-start issue is similarly divided to *new user* and *new item* problems [35].
  - New user: This arises when a new user is created or when an existing user has not contributed sufficient explicit data to satisfy the recommender system. This lack of data results in recommendations that are not in accordance with the user's taste. To remedy the new-user problem, some systems like MovieLens ask users to provide preferences on sign-up [36].
  - New item: When a new item is added to the system, it has no relations to other products and users. This lack of data is especially difficult for CF, that utilizes the

relations between items and users [36]. As a result, the recommender system will be unable to find an appropriate neighborhood for a new item and would be unlikely to suggest one.

These issues can result in serious impacts on the accuracy of a CF recommendation as it relies heavily on the quantity of ratings given by the users. In this thesis, we particularly work on improving the accuracy and quality of neighbor-based CF recommender systems.

2.3 Neighbor-based Collaborative Filtering

Neighbor-based CF is a method of high significance among recommender systems, with advantages of simplicity and good performance [37]. According to Figure 2.4, the main steps in the neighbor-

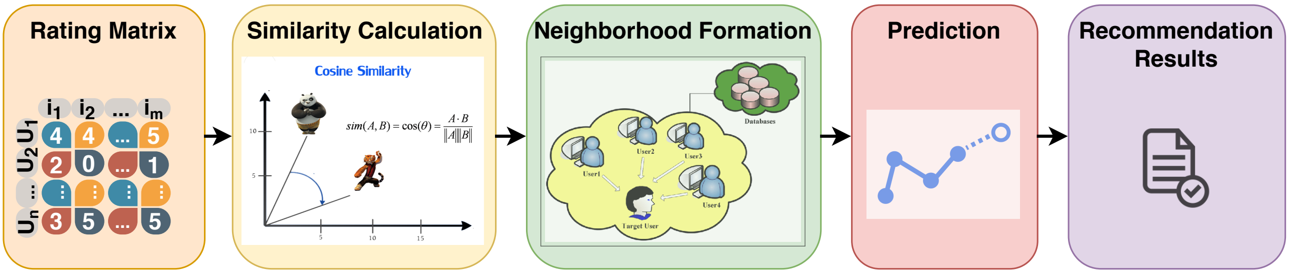


FIGURE 2.4: Neighbor-based CF Approach Procedure

based CF approach are (1) finding the similarity values between users or items, (2) selecting the most-similar users or items as the neighbors, (3) predicting the ratings of users for items that they have not yet rated, and (4) recommending the items with the highest predicted ratings [1]. To do so, the system tries to find the relationship and similarities between users or items through using similarity calculations (Section 2.3.1). The network of similar users (or items) form a neighborhood for the active user (or item) (Section 2.3.2). This neighborhood will be used to provide recommendations. The neighbor-based CF approach can be divided into two classes, namely user-based and item-based. In the user-based approach, the recommendation is provided based on the users whom are more similar to the active user (similarity between users). In the item-based approach, the items are recommended based on similar items that she/he previously liked (similarity between items) [33]. Compared to the individual approaches, the combination of user-based CF and item-based CF results in more-accurate recommendations [38]. Similarity calculation and neighbor selection are discussed in the following, as they are the key concerns in the neighbor-based CF method.

2.3.1 Similarity

The similarity is a measure of how much alike two objects/data are. In neighbor-based CF recommender systems, the similarity between users or items are calculated according to the historical scores they have given to items. Two users are similar if they have shown similar tastes about items. Two items are similar if they have been given similar ratings by users. The cosine similarity and the Pearson correlation coefficient are the two most popular measurement methods of similarity [1]. The cosine similarity uses the cosine of the angle between two vectors [39] while the Pearson correlation coefficient uses the linear correlation between two variables [40]. According to the literature, cosine similarity has shown better results on mean-centered ratings [41] and top-K recommendations [42]. However, these similarity measurements only consider the set of attributes in common between two vectors. The Jaccard metric measures the similarity between two sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets [43]. An integration of the cosine similarity or the Pearson correlation coefficient with the Jaccard metric has been proposed by Candillier et al. [44]. They used the Jaccard metric to give priority to the users/items that share more attributes. Other studies used this approach and investigated the performance of their recommendation system when employing the Jaccard metric in the similarity calculation [5, 6, 45]. It should be noted that all the similarity measures serve as a basis of selecting appropriate neighbors for recommendations [46].

2.3.2 Neighbors

As a principle in CF, if a user has a higher degree of similarity with his/her neighbors, his/her preferences are more important in the recommendation [46]. There have been a lot of efforts to improve the accuracy of similarities by optimizing the neighborhood of users. Bobadilla et al. [47] improved recommendation results using some modified similarity functions that took into account the importance of items and users. Choi et al. [48] introduced a new similarity calculation method by selecting different neighbors for individual users. Moradi and Ahmadian [49] proposed a reliability-based method to detect and correct unreliable ratings using the fusion of the similarities and the trust reports. Polatidis and Georgiadis [50] proposed a multi-level CF approach to assign a higher similarity score to a pair of users if their similarity or the number of commonly rated items surpasses a fix number. Zhang et al. [51] represented a two-layer neighbor-selection method for CF recommender systems to select the most trustworthy neighbors. They first considered the number of commonly rated items between a target user and his/her potential neighbors, and then set zero scores to potential neighbors who are not helpful for the recommendation process. Liji et al. [52] introduced a dynamic clustering method to fill the rating matrix based on item genres and scoring time. Jing et al. [53] introduced a new similarity calculation between users using the interest vector and the rating matrix of users with less impact. The user interest vector is made from integrating the movies' features and the users' ratings.

All these existing neighbor-selection approaches have not considered user credibility according to their different rating behavior.

2.4 User Credibility

Trust information can be explicitly collected from users' reviews/opinion or implicitly inferred from users' rating behavior [54]. Due to the high computational costs and performance drawbacks of text analysis, many researchers tried to consider rating behavior rather than deep natural reviews and opinions. This brings the focus on detecting review/opinion dishonesty or spammers from the users' rating behavior [55, 56]; Lim et al. [55] used ratings to identify and model spammers' behavior. Fang et al. [56] applied users' rating behavior to distinguish honest users from dishonest ones.

The users' rating behavior has also been used in social recommendation; Yang et al. [57] created a category-specific circle of friends to infer trust values based on users' rating behaviors in each category. Guo et al. [54] introduced a social-trust-based method through merging the ratings of a user's credible neighbors, to represent his/her preferences. Recently, Zhang et al. [3] proposed a social recommendation method that exploits the credibility of users' ratings to enhance the robustness of recommender systems. However, their work did not cover anything on neighbor-selection optimization. A summary of the related works is shown in Table 2.1.

TABLE 2.1: Summary of Research Work using Credibility of Users in various context

Reference	Year	Context	Dataset
Lim et al. [55]	2010	Spammer detection	Amazon.com
Fang et al. [56]	2014	Review-dishonesty detection	Epinions, Flixster, FilmTrust
Yang et al. [57]	2012	Social recommendation	Epinions
Guo et al. [54]	2014	Social recommendation	FilmTrust, Flixster, Epinions
Zhang et al. [3]	2017	Social recommendation	Flixster

External levels of information such as ontology and demographic information can also be used to further improve the accuracy of the neighbor-based CF recommendation method.

2.5 Ontology

Ontology was defined by Gruber in 1993 [58] as an “explicit specification of a conceptualization”. Ontology has been applied to model the structure of a system according to the relationships which emerge from its observation. It holds a set of concepts, namely entities, attributes and properties between the objects, along with their definitions and relations [59]. A domain ontology represents concepts which belong to a specific domain, such as movies or music. The domain ontology can be represented through several languages such as Resource Description Framework (RDF) and Ontology Web Language (OWL) [60].

Domain ontology is widely used in recommender systems due to the efficacy of ontology as a method of knowledge representation [60]. Ontology in recommender systems is made up of the semantic information of items, including the attributes of items, the relationships among items and the relationship between meta-information and items [7]. Many research efforts have been made using ontology in recommender systems (see Table 2.2). Daramola et al. [61] developed a recommender system using semantic information for tourism services. Shambour and Lu [62, 63] developed a recommendation method using the semantic similarity between pair of items. Kermany and Alizadeh [6] proposed a recommendation technique using ontological semantics of movies in

TABLE 2.2: Summary of Research Work Using Ontology in Recommender Systems

Reference	**Year**	**Attributes**	**Dataset**
Daramola et al. [61]	2009	Weather temperature, traffic, crime rate, scenery, status	Tourism data
Shambour and Lu [62]	2011	Movie genre	Movielens 100K
Shambour and Lu [63]	2012	Movie genre	Movielens 100K, Yahoo! Webscope R4
Kermany and Alizadeh [6]	2017	Movie genre	Yahoo!Movies multi-criteria datasets
Alhijawi et al. [64]	2018	Movie genre, actors, director	HetRec 2011: MovieLens + IMDb
Nilashi et al. [7]	2018	Movie genre	MovieLens 1M, Yahoo! Webscope R4
Martinez-Garcia et al. [65]	2018	Leisure, sports	Touristic activities in Tarragona

a multi-criteria fuzzy context. Alhijawi et al. [64] proposed a method using semantic information about the item to calculate the semantic similarity between users. Nilashi et al. [7] represented a model using the movies ontology domain to find semantic relations among movies to enhance the quality and accuracy of recommendation. Martinez-Garcia et al. [65] proposed a tourism recommendation using semantic attributes such as cultural and leisure activities available in the city, and also sports that may have been done in the city.

Ontology-based recommender systems have often been applied to tackle the lack-of-data problem as they rely on ontology domain knowledge instead of ratings [1]. These systems deal with the characteristics of items to find the semantic relationships among them [6, 7, 63, 64]. The semantic information of items provides additional information to reduce the cold-start and sparsity problems by letting the recommender do prediction based on these additional sources of knowledge. Moreover, the use of ontology in recommender systems can provide further advantages including the dynamic contextualization of users' tastes/interests in specific domains and the guarantee of interoperability of system resources [66].

2.6 Demographic Information

Demographic information is some Socioeconomic characteristics of a population expressed statistically, such as age, occupation, gender, education level, income level, location, marital status, language, religion and so on.

A demographic recommendation is based on the assumption that users with similar demographic information will have similar tastes in selecting items [1]. The system therefore recommends items to users based on their demographic information as extracted from their profiles [1]. Demographic information plays a significant role in detecting groups of users with similar tastes to improve the accuracy of recommendations [6]. As can be seen from Figure 2.5, the main steps of a

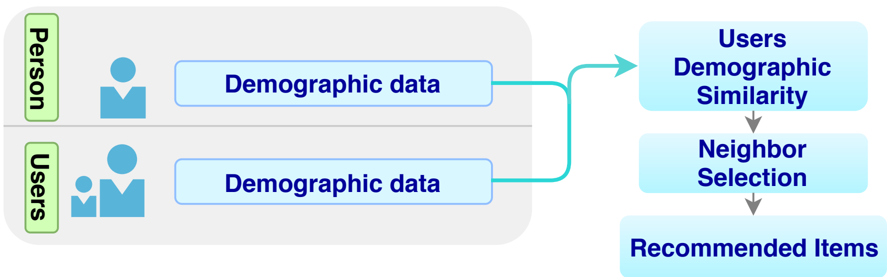


FIGURE 2.5: Demographic-based approach

demographic-based recommender system are: (1) calculating the similarity between pairs of users based on their demographic data, (2) selecting neighbors based on their similarity values [67], and (3) recommending items that neighbors have liked. Many studies have been done on demographic information and improving the recommendation result (see Table 2.3). Bi et al. [68] gave a solution to the issue of inferring users' demographic features such as religion, gender, age and political view from their search queries. Zhao et al. [69] tried to match users' demographic information derived from their public profiles with item information learned from micro-blogs and reviews to use in recommender systems. Zhao et al. [70] used demographic information of both users and items extracted from social media for item recommendation. Kermany and Alizadeh [6] applied users' demographic information such as age and gender to obtain a new similarity measure among users. Lastly, Subramaniyaswamy et al. [71] used demographic information of the user along with travel sequences, actions, motivations, and opinions to exploit user preferences to

TABLE 2.3: Summary of Research Work Using Demographic Information in Recommender Systems

Reference	**Year**	**Attributes**	**Dataset**
Bi et al. [68]	2013	Gender, age, religion, political view	MyPersonality: Facebook
Zhao et al. [69]	2014	Gender, career	Sina Weibo: phone, camera, laptop
Zhao et al. [70]	2016	Gender, age, education, marital status, career	Sina Weibo: phone, camera, laptop
Kermany and Alizadeh [6]	2017	Age, gender	Yahoo!Movies multi-criteria datasets
Subramaniyaswamy et al. [71]	2018	Gender, age, travel type, travel experience	Climate-based, food information, and user datasets

provide recommendations.

Demographic information has also been applied as additional information to deal with the lack-of-data/information problem. Demographic-based recommender systems do not experience the new-user problem as they do not need an initial list of ratings from a new user to make recommendations [1]. When new users enter a system and they have zero or few ratings, their demographic information will be used to find their neighbors to generate recommendations. Demographic information can also be applied as an additional source of information to categorize users into different groups based on their demographics rather than the ratings. This categorization of users can be useful when we have a sparse rating matrix [6]. Similarly, demographic information can be used instead of asking the user to vote items on sign-up to avoid user dissatisfaction [1].

3

An Integrated Method for Recommendation

In this work, we propose a solution to improve the accuracy of recommendation by incorporating the rating credibility of users with the ontological semantics of items and the demographic information of users. We are using movie recommendation as an example, however the method can be applied in other areas as well. The ontological semantics of movies and demographic information of users are employed to alleviate issues of sparsity and cold start. The user credibility provides a criterion for selecting neighbors with more-reliable ratings. To the best knowledge of authors, there is no work done on an integration of the rating credibility, demographic information of users, and ontological semantics of items to further improve the quality of CF recommendation system. It also should be noted that, credibility has been incorporated to the CF recommendation for the first time to optimize the neighbors and to study its influence on the performance of CF recommendation. For simplicity, we name this incorporation of credibility, ontological semantics of

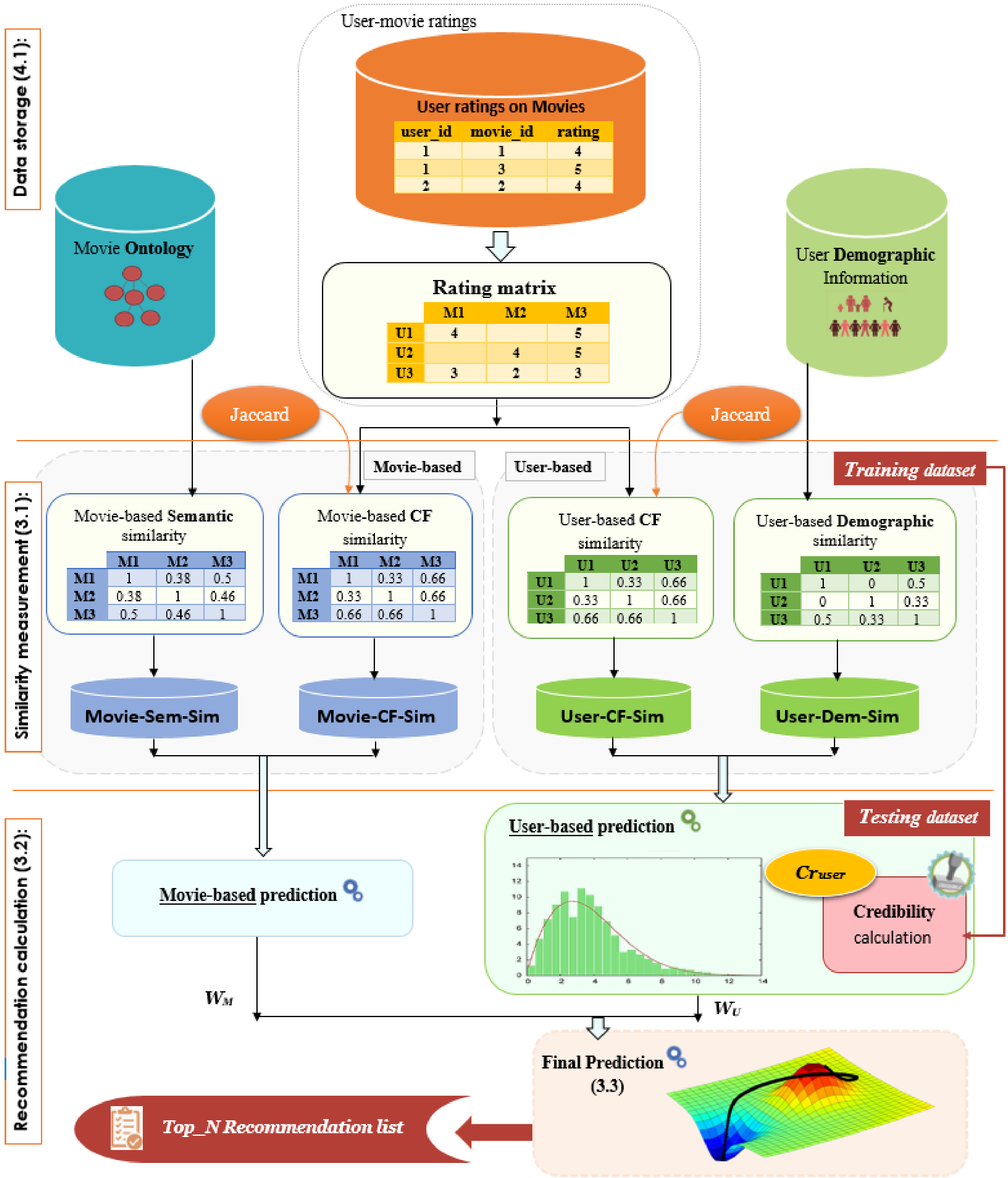


FIGURE 3.1: Framework of our Recommendation Solution

movies, and demographic information of users as the CrSemDemCF method. Figure 3.1 shows the main components of the framework of our proposed integrated recommendation solution including the data storage, similarity measurement, and recommendation calculation. The movie ontology, user-movie ratings and user demographic information are stored in the databases. The training data is used for the similarity measurements and credibility calculations while the testing data is used for the predictions. The details of similarity calculation and recommendation calculation will be presented in the following subsections.

3.1 Similarity Measurement

Figure 3.1 shows four kinds of similarity measurements including: $Movie\text{-}CF\text{-}Sim$ (movie-based CF similarity), $User\text{-}CF\text{-}Sim$ (user-based CF similarity), $Movie\text{-}Sem\text{-}Sim$ (movie-based semantic similarity), and $User\text{-}Dem\text{-}Sim$ (user-based demographic similarity). $User\text{-}CF\text{-}Sim$ and $Movie\text{-}CF\text{-}Sim$ measure the similarity between a pair of users or movies respectively according to their rating analysis with the CF algorithm. $Movie\text{-}Sem\text{-}Sim$ measures the semantic similarity of two movies. $User\text{-}Dem\text{-}Sim$ measures the demographic similarity of two users.

3.1.1 CF Similarity Measures ($User\text{-}CF\text{-}Sim$ and $Movie\text{-}CF\text{-}Sim$)

The combination of cosine similarity and the Jaccard metric is used to measure the similarity between a pair of users or movies [44]. The cosine similarity measurement only considers users or movies that have common ratings. The Jaccard metric is the number of the common ratings between two users or movies divided by the total number of ratings related to these two users or movies. The Jaccard metric is employed to adjust the cosine measurement to make the similarity more reliable [44].

Let $A = (a_1, a_2, \cdots, a_n)$ and $B = (b_1, b_2, \cdots, b_n)$ be the two vectors of ratings for two users or two movies extracted from the rating matrix. The similarity is calculated as

$$Jaccard\_Sim(A, B) = Sim(A, B) * \frac{|A \cap B|}{|A \cup B|} \tag{3.1}$$

where $Sim(A, B)$ is the cosine similarity [1] between the ratings for two users or two movies A and B, and is calculated as

$$Sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.2}$$

Note that both $User\text{-}CF\text{-}Sim$ and $Movie\text{-}CF\text{-}Sim$ should be calculated using Eq. 3.1.

3.1.2 Semantic Similarity ($Movie\text{-}Sem\text{-}Sim$)

$Movie\text{-}Sem\text{-}Sim$ measures the similarity between a pair of movies according to their semantic relations based on the ontology of movies including genre types such as $action$, $adventure$, and $animation$. Each movie is associated with a number of genre types, $Movie\text{-}Sem\text{-}Sim$ is calculated using Eq. 3.3 as a binary Jaccard similarity [44].

For a movie ontology with k genre types, each movie has a binary vector (T_m=($t_{m,1}$, $t_{m,2}$, $\cdots$, $t_{m,k}$)), where $t_{m,g} = 1$ if movie m has the genre g and $t_{m,g} = 0$ if movie m does not have the genre g ($g = 1, \cdots, k$).

$$ISemSim(A,B) = \frac{O_{11}}{O_{01} + O_{10} + O_{11}} \tag{3.3}$$

where O_{11} , O_{01} and O_{10} are the numbers of genre types when ($t_{A,g}$= 1 and $t_{B,g} = 1$), ($t_{A,g} = 0$ and $t_{B,g} = 1$) and ($t_{A,g} = 1$ and $t_{B,g} = 0$) respectively. Algorithm 1 shows the procedure of semantic similarity measurement.

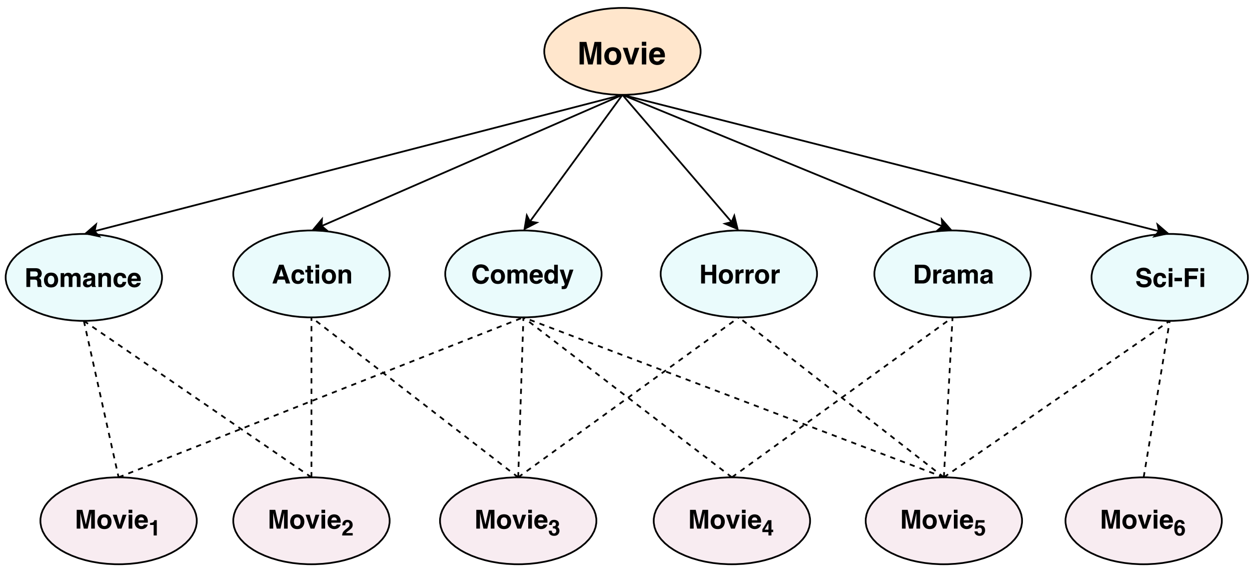


FIGURE 3.2: An Example of tree-structured Movie Ontology

As an example, consider $Movie_2$ and $Movie_5$ from the tree-structure example in Figure 3.2. It can be seen that there is no semantic similarity between $Movie_2$ and $Movie_5$ as they do not

Algorithm 1 Semantic Similarity Measurement

Input: $m \times k$ $Ont\text{-}Matrix$ with m movies in rows and k features in columns; the features here are "genre types".

procedure $m \times k$ $Ont\text{-}Matrix$
    for $movie_i$ `from 1 to` m **do**
        for $movie_j$ `from 1 to` m **do**
            $O_{01} \longleftarrow$ number of cells where $movie_i$ is 0 and $movie_j$ is 1
            $O_{10} \longleftarrow$ number of cells where $movie_i$ is 1 and $movie_j$ is 0
            $O_{11} \longleftarrow$ number of cells where $movie_i$ is 1 and $movie_j$ is 1
            if $(O_{01} + O_{10} + O_{11}) == 0$ **then**
                $ISemSim\ (movie_i\ , movie_j) \longleftarrow 0$
            else
                $ISemSim\ (movie_i\ , movie_j) \longleftarrow O_{11}/(O_{01} + O_{10} + O_{11})$
            End if
        End for
    End for
return $m \times m$ $ISemSim$ matrix (the semantic similarity between all pairs of movies)

share any common features (genre types). However, when the user shows some interest in a new movie, for example $Movie_3$, a semantic similarity will emerge, because each pair of movies shares common specific genre types, i.e. "Action" between $Movie_2$ and $Movie_3$, and "Comedy" and "Horror" between $Movie_3$ and $Movie_5$. This can be more easily understood by looking at each movie vector: $Movie_2 = (1, 1, 0, 0, 0, 0)$, $Movie_3 = (0, 1, 1, 1, 0, 0)$, and $Movie_5 = (0, 0, 1, 1, 1, 1)$, where the 0 and 1 numbers are assigned following the relationship between each movie and the genre types. Number 1 is assigned if the movie contains any of the movie genre types of Romance, Action, Comedy, Horror, Drama, and Science-Fiction (Sci-Fi) respectively; vice versa is true for assigning 0. Next, the semantic similarity between $Movie_2$ and $Movie_3$ can be calculated following Eq. 3.3, where $O_{11} = 1$ (only "Action" genre has position 1 for both movies), $O_{01} = 2$ ($Movie_2$ is position 0 and $Movie_3$ is position 1 for "Comedy" and "Horror" genre type), and$O_{10} = 1$ ($Movie_2$ is position 1 and $Movie_3$ is position 0 for "Romance" genre type), and so the semantic similarity measure between $Movie_2$ and $Movie_3$ will be $1/(2+1+1) = 0.25$. Similarly, the movie-based semantic similarity measure between $Movie_3$ and $Movie_5$ is $2/(2+1+2) = 0.4$ and between $Movie_2$ and $Movie_5$ is $0/(4+2+0) = 0$.

3.1.3 Demographic Similarity (*User-Dem-Sim*)

User-Dem-Sim is the similarity between two users based on demographic features. The intuition here is that users with the same demographic features are most likely to have the same interests. The vector $D_u = (d_1, d_2, \cdots, d_p)$ holds the values of demographic features (d_i, $i \in 1, 2, \cdots, p$) such as *age group*, *gender* and *occupation group* for user u. All values are 0 or 1, where '1' means the user has the feature and '0' means the user does not have the feature. *User-Dem-Sim* is calculated using Eq. 3.3.

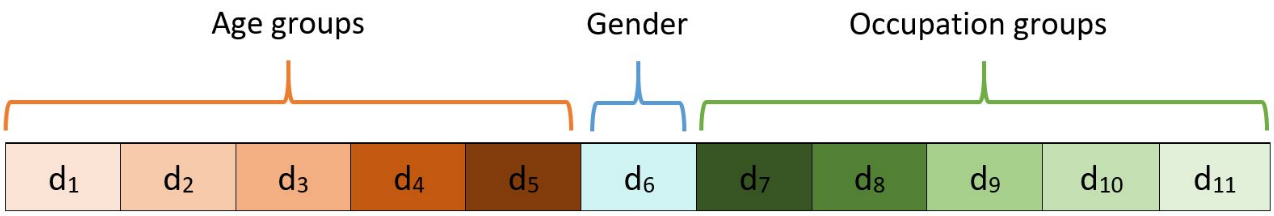


FIGURE 3.3: Demographic Vector Representation

Assume "*age-group*", "*gender*" and "*occupation-group*" as the demographic features of users in a system. Also assume there are five age groups (kid, teenage, young, middle-aged, old) and five occupation groups (student, manager, homemaker, engineer, educator), as shown in Figure 3.3. Thus, the demographic features vector for each user is represented as $(d_1, d_2, \cdots, d_5, d_6, d_7, d_8, \cdots, d_{11})$, where $d_1, d_2, \cdots, d_5$ are related to the age group, d_6 is related to the gender ("0" for female and "1" for male) and $d_7, d_8, \cdots, d_{11}$ are related to the occupation group. If (0,0,1,0,0,0,1,0,0,0,0), (0,0,0,1,0,1,0,1,0,0,0), and (0,0,1,0,0,1,1,0,0,0,0) hold the demographic features of $user_1$, $user_2$, and $user_3$ respectively, $user_1$ is a young female student, $user_2$ is a middle-aged male manager, and $user_3$ is a young male student. According to Eq. 3.3, the user-based demographic similarity measure between $user_1$ and $user_2$ is $0/(2+3+0) = 0$, between $user_2$ and $user_3$ is $1/(2+2+1)$ $= 0.2$, and between $user_1$ and $user_3$ is $2/(0+1+2) = 0.66$. Demographic features can help the recommenders to suggest items to new users, with no or only a few ratings, based on similar users whose demographic features are mostly the same.

3.2 Movie-based and User-based Rating Prediction

To generate recommendations for a target user, at first we should predict the ratings of those movies that the user has not watched yet. Movies with high prediction scores will be recommended to the user. Figure 3.1 shows the framework of the recommendation solution. The final recommendation is obtained by integrating the movie-based rating prediction $H\widetilde{r}_{Movie}$ and the user-based rating prediction $H\widetilde{r}_{User}$. More details will be described in the following subsections.

3.2.1 Movie-based Rating Prediction ($H\widetilde{r}_{Movie}$)

In order to calculate $H\widetilde{r}_{Movie}$, we need to calculate $\widetilde{r}_{ICF}$ (movie-based CF prediction) with the $Movie$-CF-Sim similarity as the input, and to calculate $\widetilde{r}_{ISem}$ (movie-based semantic prediction) with $Movie$-Sem-Sim similarity as the input. Both $\widetilde{r}_{ICF}$ and $\widetilde{r}_{ISem}$ are calculated using Eq. 3.4 [1]:

$$\widetilde{r}(u,m) = \widetilde{r}_m + \frac{\sum_{y=1}^{k}(r_{u,y} - \widetilde{r}_y) * Sim(m,y)}{\sum_{y=1}^{k} Sim(m,y)} \tag{3.4}$$

where $\widetilde{r}(u,m)$ is the predicted rating from user u on movie m. It could be $\widetilde{r}_{ICF}$ or $\widetilde{r}_{ISem}$ depending on whether the input is $Movie$-CF-Sim or $Movie$-Sem-Sim as $Sim(m,y)$. $\widetilde{r}_m$ is the average rating for movie m; k is the number of neighbors; $r_{u,y}$ is the rating of user u on movie y; and $\widetilde{r}_y$ is the average rating for movie y.

Note that, for each prediction calculation, K-Nearest-Neighbors (KNN) is applied to select the top-K most-similar movies as the neighbors of the target movies.

The two predictions, $\widetilde{r}_{ICF}$ and $\widetilde{r}_{ISem}$, are integrated with Eq. 3.5 to calculate the movie prediction $H\widetilde{r}_{Movie}$ [62].

$$H\widetilde{r}_{Movie} = \begin{cases} 0 & \text{if } \widetilde{r}_{ICF} = 0 \ \& \ \widetilde{r}_{ISem} = 0 \\ \widetilde{r}_{ICF} & \text{if } \widetilde{r}_{ISem} = 0 \\ \widetilde{r}_{ISem} & \text{if } \widetilde{r}_{ICF} = 0 \\ (2 * \widetilde{r}_{ICF} * \widetilde{r}_{ISem})/(\widetilde{r}_{ICF} + \widetilde{r}_{ISem}) & \text{otherwise} \end{cases} \tag{3.5}$$

3.2.2 User-based Rating Prediction ($H\widetilde{r}_{User}$)

$H\widetilde{r}_{User}$ is calculated using Eq. 3.5 with integrating $\widetilde{r}_{UDem}$ (user-based demographic prediction) and $\widetilde{r}^{Cr}_{UCF}$ (user-credibility-based CF prediction). There are three steps in the calculation of the user prediction value $H\widetilde{r}_{User}$:

- **Step 1:** $\widetilde{r}_{UDem}$ is calculated using Eq. 3.4 with $User\text{-}Dem\text{-}Sim$ similarity as the input.
- **Step 2:** The neighbors are selected based on the credibility of users through the proposed Algorithm 2. $\widetilde{r}^{Cr}_{UCF}$ is calculated using Eq. 3.4 with neighbors' ratings and their $User\text{-}CF\text{-}Sim$ similarity to the target user as the input. The credibility measurements will be described in detail later.
- **Step 3:** $H\widetilde{r}_{User}$ is calculated by integrating the two predictions $\widetilde{r}^{Cr}_{UCF}$ and $\widetilde{r}_{UDem}$ with Eq. 3.5.

Algorithm 2 Credibility-based KNN

Input: $N \times N$ $User\text{-}CF\text{-}Sim$ matrix ($Jaccard\_Sim$) with N users in rows and columns; $1 \times N$ $Credibility$ values.

procedure $1 \times N$ CREDIBILITY-BASED-KNN
  for `each row in` $User\text{-}CF\text{-}Sim$ `matrix` **do**
    $W\text{-}Cr\text{-}Sim_{user}(row_i) \longleftarrow Jaccard\_Sim(row_i) * Credibility$
    Sort each row in $W\text{-}Cr\text{-}Sim_{user}$ **decreasingly**
  End for
return *top-K* similar users to be used in prediction formula

Credibility: According to our solution, each user has a credibility value. As mentioned before, credibility has been incorporated to the CF recommendation for the first time to optimize the neighbors and to study its influence on the performance of CF recommendation. These credibility values are calculated according to the user's ratings history.

In a rating system, some users may give ratings with little variation, while others may give widely distributed ratings to items. $Credibility$ is measured for each user with a value between 0 and 1. The $Credibility$ of each user can be calculated according to his or her rating behavior and its deviation from the total distribution of all ratings.

In order to calculate each user's credibility, we compute the distribution of all ratings and adjust each user's rating based on that. The Absolute Error (AE) metric that represents the difference between the user's ratings and the Weibull curve is calculated. A user with a smaller AE has a higher credibility.

The 2-parameter Weibull distribution is applied for this purpose because of its benefit of providing sensibly accurate fitting [72]. The Weibull Probability Density Function (pdf) is defined as follows:

$$f(t) = \frac{\beta}{\eta}(\frac{t-\gamma}{\eta})^{\beta-1}e^{-(\frac{t-\gamma}{\eta})^{\beta}} \qquad f(t) \geqslant 0, \;\; t \geqslant \gamma, \;\; \beta > 0, \;\; \eta > 0, \;\; -\infty < \gamma < +\infty \tag{3.6}$$

where β is a shape parameter; η is a scale parameter; and γ is a location parameter. Now, if we set $\gamma = 0$, the 2-parameter Weibull pdf is defined as follows:

$$f(t) = \frac{\beta}{\eta}(\frac{t}{\eta})^{\beta-1}e^{-(\frac{t}{\eta})^{\beta}} \tag{3.7}$$

"Scipy.stats.weibull_min.fit(data, floc=0)" in python can be used to estimate the parameters, where *data* is an array of ratings, and $floc = 0$ keeps the location fixed at zero.

For a specific user, the credibility of a user is calculated as

$$Cr_{user} = e^{-\alpha(AE_{user}-AE_{min})} \tag{3.8}$$

where AE_{min} is a minimum value of AE, which will be discussed below, and α is a regulating parameter, and

$$AE_{user} = \sum_{i=1}^{N} |adj\_r_{u_i} - r_{u_i}| \tag{3.9}$$

where the adjusted ratings and the real ratings are denoted by $adj\_r_{u_i}$ and r_{u_i} respectively, and N is the total number of available ratings. Note that the adjusted ratings must fall in the value domain. For this purpose, we make the formulas:

$$\begin{cases} adj\_r_{u_i} = r_{u_i} * \frac{\widetilde{r}_{u_i}}{C_{dist}} & \text{if } \widetilde{r}_{u_i} > C_{dist} \\ adj\_r_{u_i} = max_r - (max_r - r_{u_i}) * \frac{\widetilde{r}_{u_i}}{C_{dist}} & \text{if } \widetilde{r}_{u_i} \leq C_{dist} \end{cases} \tag{3.10}$$

where $\widetilde{r}_{u_i}$ and C_{dist} are the mean ratings for u_i and the center of the Weibull distribution for all ratings. Accordingly, the credibility values for all users are calculated using the proposed Algorithm 3.

Algorithm 3 Credibility calculation

Input: $N \times M$ rating matrix with N users and M movies; r_{ij} is the rating of the ith user for the jth movie; C_{dist} is the center of the Weibull distribution.

```
procedure CREDIBILITY(user_i)
    for each row (1 × M) in rating matrix do
        if r̃_{u_i} > C_dist then
            adj_r_{u_i} = r_{u_i} * r̃_{u_i} / C_dist
        else if r̃_{u_i} ≤ C_dist then
            adj_r_{u_i} = max_r − (max_r − r_{u_i}) * r̃_{u_i} / C_dist
        End if
        AE_user ← Σ_{i=1}^{v} |adj_r_{u_i} − r_{u_i}|
        Cr_user ← e^{−α(AE_user − AE_min)}
    End for
return Credibility values of all users
```

3.3 Final Prediction and Recommendation

The final prediction ($\widetilde{r}_{final}(u,m)$) of user u and movie m, as defined in Eq. 3.11, is a fusion of user-based prediction ($H\widetilde{r}_{User}$) and movie-based prediction ($H\widetilde{r}_{Movie}$) in an optimization weighting scheme. With Eq. 3.11, the individual and optimized weights will be determined for each single user and movie.

$$\widetilde{r}_{final}(u,m) = W_{user}^{best}(u) * H\widetilde{r}_{User}(u,m) + W_{movie}^{best}(m) * H\widetilde{r}_{Movie}(u,m) \tag{3.11}$$

where $\widetilde{r}_{final}(u,m)$ represents the final prediction for user u on movie m; W_{user}^{best} and W_{movie}^{best} are the best personalized prediction weights for each user and movie respectively. We set $W_{user}^{best}(u)$ = 1 - $W_{movie}^{best}(m)$, in order to ensure that the predicted ratings remain within the allowed range. Generally, the best weights are the ones that solve the following scalar minimization to a high precision:

$$w = \underset{w}{\text{argmin}}\, f(x - wh), \quad h = \triangledown f(x), \quad w_0 = 0,$$
$$w_{i+1} = f(x - w_i h) - \alpha_i \frac{d}{dw} f(x - wh)\bigg|_{w=w_i}, \quad i = 1, 2,$$

where α_i is the step size that ensures convergence, f is the interest function, h is the current search direction, and x is the current point.

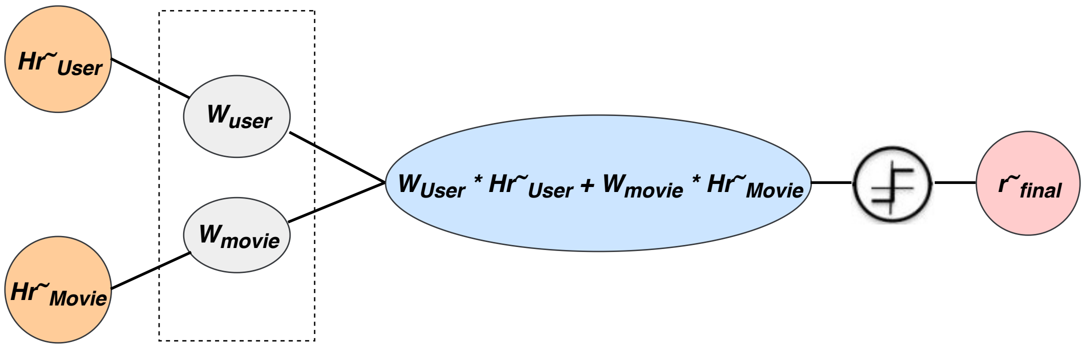


FIGURE 3.4: One-Layer Neural Network for Final Prediction

According to Figure 3.4, we implement a one-layer neural network (gradient descent) to efficiently evaluate the weights for each user-item rating pair. This is done by optimizing the

Algorithm 4 User- and item- weight optimization through one-layer neural network

Input: $epochs(=25)$, $\lambda(\in[0.001{,}0.01])$, $\alpha(\in[0.01{,}0.1])$, $W^{best}_{user}=0.5$, and $W^{best}_{movie}=$ 0.5.

```
procedure WEIGHT OPTIMIZATION THROUGH ONE-LAYER NEURAL NETWORK
    for 1 to epochs do
        for each user u do
            for each rated item m of user u do
                r~_final(u,m) ← W^best_user(u) * Hr~_User(u,m) + W^best_movie(m) * Hr~_Movie(u,m)
                e_{u,m} ← r(u,m) − r~_final(u,m)
                W^best_user(u) ← W^best_user(u) + γ.(e_{u,m} − λ.W^best_user(u))
                W^best_movie(m) ← 1 − W^best_user(u)
            End for
        End for
    End for
return W^best_user for each user u and W^best_movie for each movie m
```

Line 5: $\widetilde{r}_{final}(u,m) \leftarrow W^{best}_{user}(u) * H\widetilde{r}_{User}(u,m) + W^{best}_{movie}(m) * H\widetilde{r}_{Movie}(u,m)$

Line 6: $e_{u,m} \leftarrow r(u,m) - \widetilde{r}_{final}(u,m)$

Line 7: $W^{best}_{user}(u) \leftarrow W^{best}_{user}(u) + \gamma.(e_{u,m} - \lambda.W^{best}_{user}(u))$

Line 8: $W^{best}_{movie}(m) \leftarrow 1 - W^{best}_{user}(u)$

weighting coefficients in order to provide a minimum error in prediction [73]. The prediction error, $e_{u,m}$, is calculated using Eq. 3.12, where $r(u,m)$ and $\widetilde{r}_{final}(u,m)$ are the real and predicted ratings for user u and movie m respectively.

$$e_{u,m} = r(u,m) - \widetilde{r}_{final}(u,m) \tag{3.12}$$

The general scheme of weight optimization is shown in Algorithm 4 [38, 74]. First, the weights are initialized with values around 0.5. In each epoch, we compute a prediction $\widetilde{r}_{final}(u,m)$ using $H\widetilde{r}_{User}(u,m)$, $H\widetilde{r}_{Movie}(u,m)$ and their current weights for each user-item rating pair. Second, the prediction error, $e_{u,m}$, is computed with Eq. 3.12. This prediction error is then used to update the weights with Eq. 3.13. The parameter λ determines the size of the correcting step and α is used for regularization and to avoid over-fitting. These values are determined according to the literature and through test and trial in order to achieve more-accurate predictions.

$$\begin{aligned} W_{user}^{best}(u) &= W_{user}^{best}(u) + \lambda.(e_{u,m} - \alpha.W_{user}^{best}(u)) \\ W_{movie}^{best}(m) &= 1 - W_{user}^{best}(u) \end{aligned} \tag{3.13}$$

Thus, the final rating predictions of unknown items for a target user are obtained and the top-K ones will be recommended to the user.

4

Experiments and Discussion

This chapter includes the datasets and evaluation metrics used in this work, the various baselines for recommendation performance analysis including our proposed methods, and lastly the results and discussion of the work where our proposed methods, the baseline systems and the state-of-the-art methods are compared to find the recommender system with the best performance.

4.1 Datasets

Three datasets were used to validate the proposed solution under different users' rating behavior as shown below.

MovieLens 100K dataset: This dataset[1] is a well-known movie dataset that has been widely

[1] https://grouplens.org/datasets/movielens/100k/

used for the evaluation of CF recommender systems. This dataset consists of 100,000 ratings from 943 users on 1682 movies. The data is collected by the University of Minnesota and is associated with their online movie-recommendation system. Each user has given scores to at least 20 movies on a 5-star scale.

MovieLens 1M dataset: This dataset[2] is bigger than the previous dataset and is collected by Grouplens which consists of 1,000,209 ratings from 6040 users on 3952 movies. Each user has given scores to at least 20 movies on a 5-star scale.

Yahoo!Movie (or Yahoo! Webscope R4 dataset): This dataset[3] has been collected by the Yahoo Webscope library. This dataset consists of 221,367 ratings from 7642 users and 11,915 movies. Each user has provided ratings on a 5-star scale (1 to 5).

TABLE 4.1: The Statistics of the Experimental Datasets

Dataset	**MovieLens 100K**	**MovieLens 1M**	**Yahoo!Movie**
Users	943	6040	7,642
Movies	1,682	3952	11,915
Ratings	100,000	1,000,209	221,367

The three datasets are summarized in Table 4.1. The rating matrix is designed by having users in the rows and movies in the columns, where each element of the matrix holds the rating (r_{ij}) from the ith user on the jth movie; $r_{ij} = 0$ when there is no rating value from user U_i on movie M_j.

Other required information such as movie ontology and users' demographic information are stored in specific databases. The demographic information of users are held in a database with "age", "gender" and "occupation" features for Movielens and "age" and "gender" features for Yahoo!Movie. Similarly, the content of movies are saved in a database with 21 genre types for Movielens and 12 genre types for Yahoo!Movie.

[2]https://grouplens.org/datasets/movielens/1m/

[3]http://webscope.sandbox.yahoo.com

To evaluate our solution, each dataset is further divided into two groups of training and testing. Where, 80% of the data is used as the training dataset, and the remaining 20% is used as the test dataset with the 5-fold cross-validation method.

4.2 Evaluation Metrics

Mean Absolute Error (*MAE*) and Root Mean Square Error (*RMSE*) were used to evaluate the performance of our proposed method and existing solutions.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|p_i - r_i| \tag{4.1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|p_i - r_i|^2} \tag{4.2}$$

where r_i and p_i denote the actual rating and the predicted rating respectively, and N represents the total number of ratings. Smaller *MAE* and *RMSE* values mean that the recommendations are more accurate.

In addition to *MAE* and *RMSE*, *precision* and *recall* metrics are commonly used to measure the quality of the top-K recommendations. *Precision* shows the proportion of items that are relevant within the retrieved result. In contrast, *recall* represents the proportion of relevant items that have been retrieved.

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

$$recall = \frac{TP}{TP + FN} \tag{4.4}$$

where TP is the number of true-positive items (relevant and recommended), FP is the number of false-positive items (non-relevant and recommended), and FN is the number of false-negative items (relevant and not recommended). To differentiate the relevant and non-relevant items, we set items with an actual rating of greater than or equal to 4.0 out of 5.0 as relevant and those with a rating below 4.0 out of 5.0 as non-relevant to the user. High *precision* is obtained when an algorithm returns noticeably more relevant recommendations than non-relevant, while high *recall* means that an algorithm returns most of the relevant results.

F_1-*measure* is the harmonic mean of the *recall* and the *precision*.

$$F_1 - measure = \frac{2.precision.recall}{precision + recall} \tag{4.5}$$

The sparsity level is calculated as follows:

$$Sparsity = 1 - \frac{number\_of\_ratings}{number\_of\_users \times number\_of\_items} \tag{4.6}$$

4.3 Baselines

We carried out a set of experiments against Movielens 100k and compared our results with four existing methods, COS [52], DEUC [51], MLCF [50] and IVU [53]. These existing methods use different ways to improve the accuracy of similarity and select more-similar neighbors in different contexts.

1. MLCF [50] is a multi-level CF approach which assigns a higher similarity score to a pair of users if their similarity or the number of commonly rated items surpasses a fixed number.

2. DEUC [51] is a two-layer neighbor-selection method for CF recommender systems. It first considers the number of commonly-rated items between a target user and his/her potential neighbors, and then sets zero scores to potential neighbors who are not helpful for the recommendation process.

3. COS [52] is a dynamic clustering method to fill the rating matrix based on item genres and scoring time.

4. IVU [53] is a new similarity calculation between users using the interest vector and rating matrix of users with less impact. The user interest vector is made by integrating the movies' features and the users' ratings.

For the matter of comparison and recommendation performance analysis, we have used the MovieLens 100K, MovieLens 1M, and Yahoo!Movie datasets for all the following recommendation methods:

1. UCF [75] considers the user-based CF recommendation. The prediction is based on the similarity values between users;

2. ICF [33] considers the item-based CF recommendation. The prediction is based on the similarity values between items;

3. $UDemCF$ [6] considers the demographic information of users in the user-based CF recommendation. The demographic features of users are used to calculate a user-based similarity;

4. $ISemCF$ [6] considers the ontological semantics of items in the item-based CF recommendation. The ontological semantics of items are used to calculate an item-based similarity;

5. $SemDemCF$ [6] considers both ontological semantics of items and the demographic information of users in CF recommendation. It is a combination of $UDemCF$ and $ISemCF$;

6. UCF-Cr considers the rating credibility in the user-based CF recommendation;

7. $UDemCF$-Cr considers both demographic information of users and the rating credibility in the user-based CF recommendation;

8. $CrSemDemCF$ considers the rating credibility, the ontological semantics of movies, and the demographic information of users in the CF recommendation.

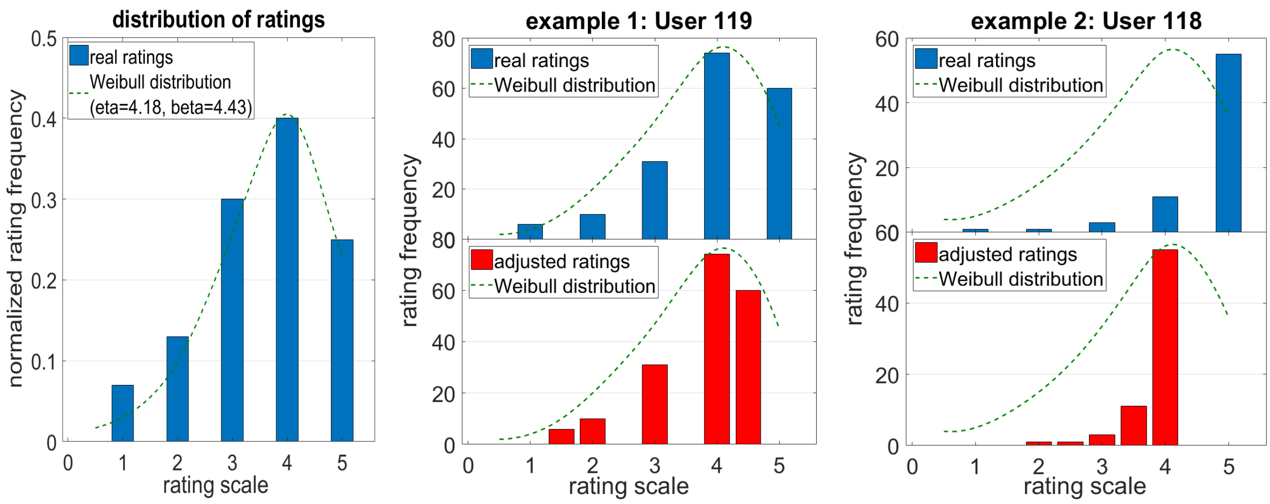
UCF, ICF, $UDemCF$, $ISemCF$, and $SemDemCF$ recommendation methods can be found in literature as well. However, the incorporation of user credibility with these methods namely, UCF-Cr, $UDemCF$-Cr, and $CrSemDemCF$ respectively are shown for the first time in this work. Where $CrSemDemCF$ method is the CF recommendation system with integration of rating credibility, ontological semantics of movies, and demographic information of users.

4.4 Results and Discussion

Following is the results and discussion for the five main analyses we performed on the baseline methods using the three datasets. The five analysis are (1) users' credibility measurement, (2) neighbor optimization, (3) *MAE* and *RMSE*, (4) standard deviation of errors, and (5) *precision*, *recall* and *F-measure*.

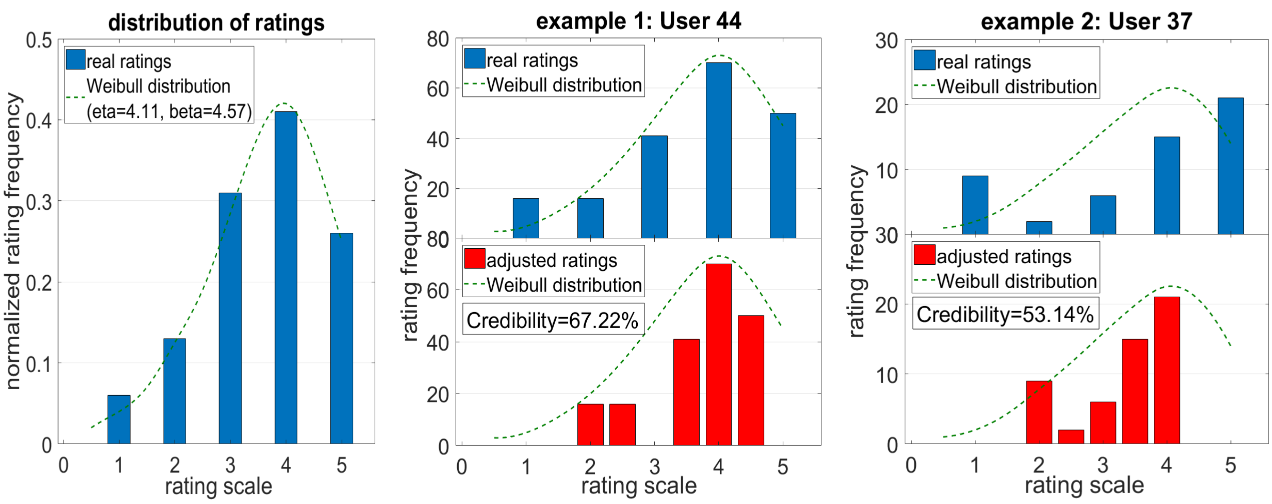
4.4.1 Users' Credibility Measurement

The users' credibility measurement is performed through the comparison with the Weibull distribution. Figure 4.1a shows the Weibull distribution (with *eta* (η) = 4.18 and *beta* (β) = 4.43) of ratings in the Movielens 100K dataset which is used as the benchmark to evaluate the credibility of users as discussed in Section 3.2.2.



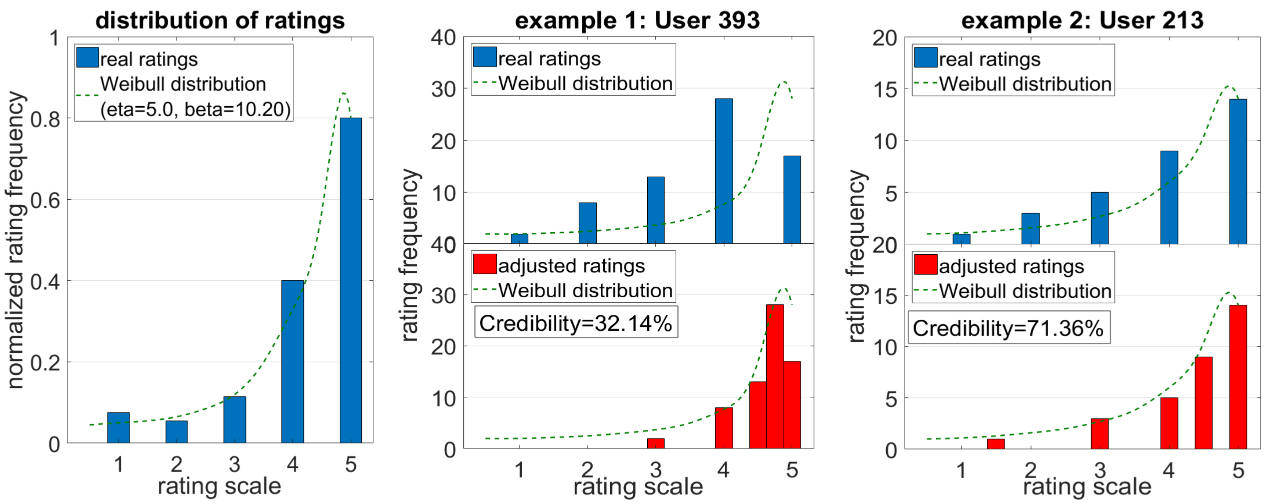
(a) (b) (c)

FIGURE 4.1: Rating Distribution of Dataset Movielens 100K and Two Examples of Rating Adjustment



(a) (b) (c)

FIGURE 4.2: Rating Distribution of Dataset Movielens 1M and Two Examples of Rating Adjustment



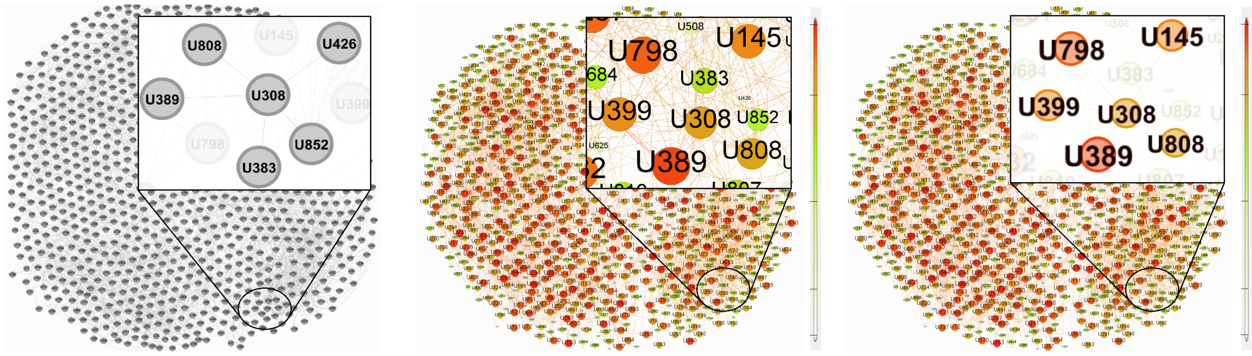
(a) (b) (c)

FIGURE 4.3: Rating Distribution of Dataset Yahoo!Movie and Two Examples of Rating Adjustment

As an example, the rating behavior and rating adjustment of users 119 and 118 are shown in Figure 4.1b and Figure 4.1c, where the top graphs (in blue) are the real ratings of users 119 and 118, and the bottom graphs (in red) are the ratings after adjustment with Eq. 3.10. Figure 4.1 shows that the behavior of the ratings given by user 119 are much more similar to the distribution (Figure 4.1a) than that of user 118. With Eq. 3.8, user 119 has a credibility value of 91.48% and user 118 has a credibility value 38.51%. Thus, user 119 is much more reliable for the recommendations than user 118 and will have the priority to be selected as a neighbor for top-K recommendations. According to the example, we can observe that the more similar is a user's rating behavior to the whole distribution of ratings, the higher the credibility value he/she has. Figure 4.2 and Figure 4.3 also show the Weibull distribution of ratings in the Movielens 1M dataset (with $\eta = 4.11$ and $\beta = 4.57$) and the Yahoo!Movie dataset (with $\eta = 5.0$ and $\beta = 10.20$) along with some examples of users' credibility measurements.

4.4.2 Neighbor Optimization

The calculated credibility values of users are then used in neighbor optimization. It significantly helps to decrease the impact of the ratings given by neighbors with low credibility. Figure 4.4 shows a graph visualization of our neighbor optimization solution on Movielens 100K dataset when the number of neighbors (K) is 5. The method is divided into three steps where (a) is the



(a) (b) (c)

FIGURE 4.4: Graph Visualization of Our Neighbor Optimization Solution on Movielens 100K with Number of Neighbors = 5 and in three steps; (a) U308 and His/Her Neighbors without Considering Credibility, (b) Users' Credibility in Different Sizes and Colors, and (c) U308 and His/Her Neighbors after Considering Credibility

first step: $user_{308}$ and his/her neighbors without considering credibility, (b) is the second step: users' credibility is calculated and the result is shown through different sizes and colors, and (c) is the last step: neighborhood optimization is performed following the results from step 2. Figure 4.4b shows different users' credibility with different sizes and colors of Nodes from low credibility in light yellow to high credibility in dark red. Comparing Figure 4.4a and Figure 4.4c, we can observe that more credible users are selected as the neighbors.

4.4.3 *MAE* and *RMSE*

In this section, we show the results of the *MAE* and *RMSE* evaluation metrics for the baseline methods using the test datasets. Figure 4.5 and Table 4.2 show the results of our solution in comparison with the baselines for Movielens 100K. We can see that *MAE* and *RMSE* have been decreased by using the combination of rating credibility, ontological semantics of movies, and demographic information of users as compared to the other methods. As a lower *MAE* or *RMSE* value shows higher accuracy of the recommender system, $CrSemDemCF$ has the highest accuracy. Figure 4.5 shows that the gap between $SemDemCF$ and $CrSemDemCF$ is larger at smaller neighborhood sizes for Movielens 100K. The reason is that if there are fewer neighbors, the effect of replacing more-credible users will be bigger.

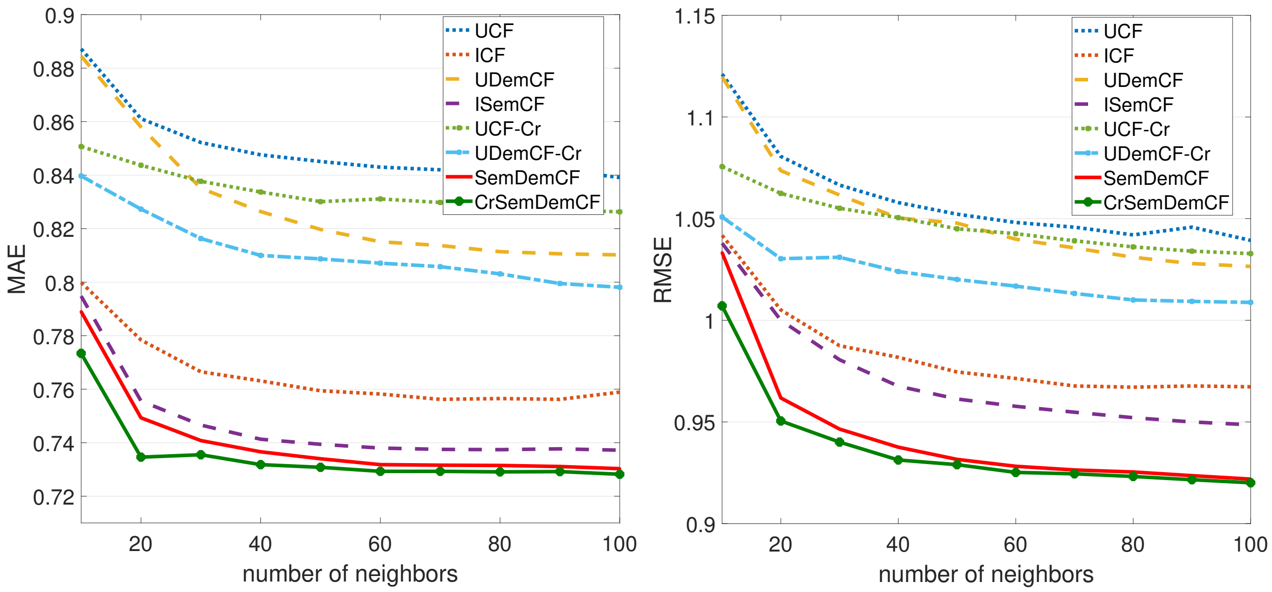


FIGURE 4.5: *MAE* and *RMSE* Values on Movielens 100K

TABLE 4.2: *MAE* and *RMSE* Values of our Solution with Various Numbers of Neighbors on Movielens 100K

Model	**Metric**	**Number of neighbors**						
		10	20	30	40	60	80	100
UCF	MAE	0.8872	0.8611	0.8522	0.8476	0.8430	0.8398	0.8391
	RMSE	1.1211	1.0806	1.0665	1.0579	1.0481	1.0420	1.0393
ICF	MAE	0.7999	0.7784	0.7665	0.7631	0.7582	0.7565	0.7589
	RMSE	1.0418	1.0051	0.9875	0.9818	0.9714	0.9671	0.9673
UDemCF	MAE	0.8845	0.8580	0.8351	0.8264	0.8151	0.8114	0.8102
	RMSE	1.1197	1.0736	1.0616	1.0503	1.0399	1.0310	1.0265
ISemCF	MAE	0.7948	0.7556	0.7466	0.7413	0.7380	0.7374	0.7372
	RMSE	1.0379	1.0000	0.9806	0.9676	0.9577	0.9521	0.9486
UCF_Cr	MAE	0.8507	0.8437	0.8377	0.8337	0.8311	0.8276	0.8263
	RMSE	1.0756	1.0624	1.0551	1.0505	1.0426	1.0361	1.0328
UDemCF_Cr	MAE	0.8397	0.8273	0.8163	0.8100	0.8071	0.8031	0.7981
	RMSE	1.0508	1.0303	1.0310	1.0240	1.0168	1.0100	1.0088
SemDemCF	MAE	0.7890	0.7493	0.7408	0.7366	0.7318	0.7315	0.7303
	RMSE	1.0333	0.9618	0.9465	0.9376	0.9282	0.9254	0.9219
*CrSemDemCF**	MAE	**0.7734**	**0.7346**	**0.7355**	**0.7318**	**0.7293**	**0.7291**	**0.7282**
	RMSE	**1.0071**	**0.9505**	**0.9401**	**0.9313**	**0.9252**	**0.9232**	**0.9201**

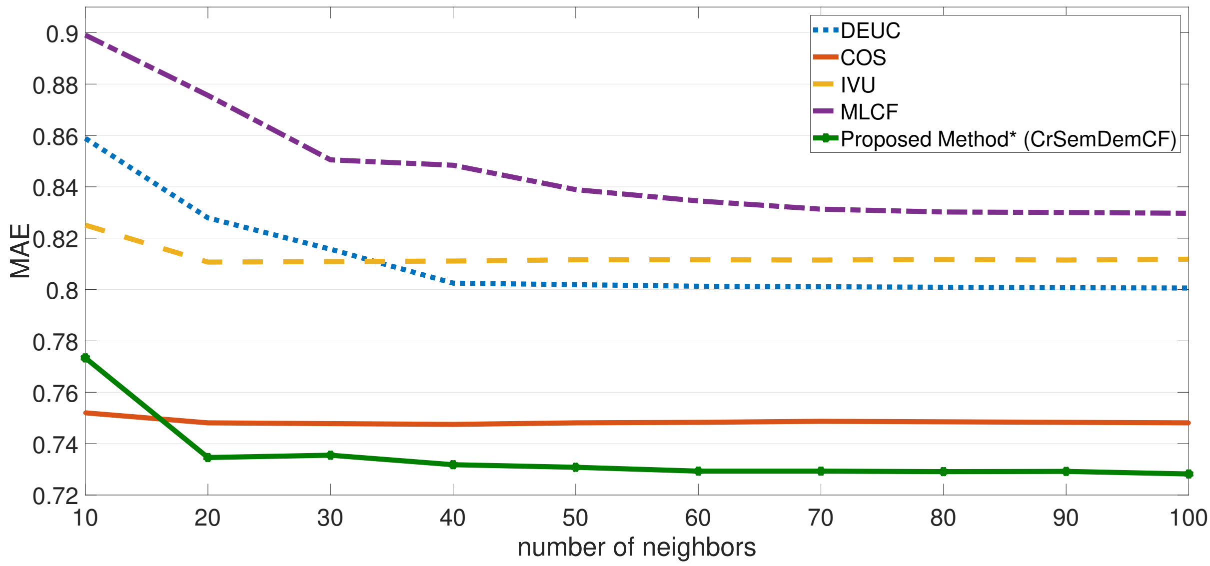


FIGURE 4.6: *MAE* Values on Movielens 100K For Our Proposed Method ($CrSemDemCF$) and Related Work

Figure 4.6 shows how our solution outperforms related studies including COS [52], DEUC [51], MLCF [50] and IVU [53] and thus results in a better recommendation. All these studies have used the MovieLens 100K dataset in their experiments, further details can be found in Section 2.3.2 . It is also evident that the proposed method outperforms the existing ones. The only section where our method has lower recommendation is in comparison with COS method and when the number of neighbors are less than 16. It should be noted that the number of neighbors is usually chosen to be between 20 and 100 for an effective analysis in existing studies. In summary, the incorporation of rating credibility, ontological semantics of movies, and demographic information of users for the CF recommendation have brought in advantages to make our solution perform better than existing methods.

In general, for a dataset with a higher sparsity level and higher variety of users' rating behavior, the incorporation of user credibility will have more effect on reducing *MAE* and *RMSE*. According to Eq. 4.6 the sparsity level for the MovieLens 100K dataset is 93.7% (sparsity level = 1 - (100,000/(943 × 1682)) = 0.937), for the MovieLens 1M dataset is 96%, and for the Yahoo!Movie dataset is 99.7%. The results shown in Figure 4.7 and Table 4.3 for the Yahoo!Movie dataset, and Figure 4.8 and Table 4.4 for Movielens 1M, support the above understanding. Comparison between $SemDemCF$ and $CrSemDemCF$ of the three datasets shows that Movielens 1M results in

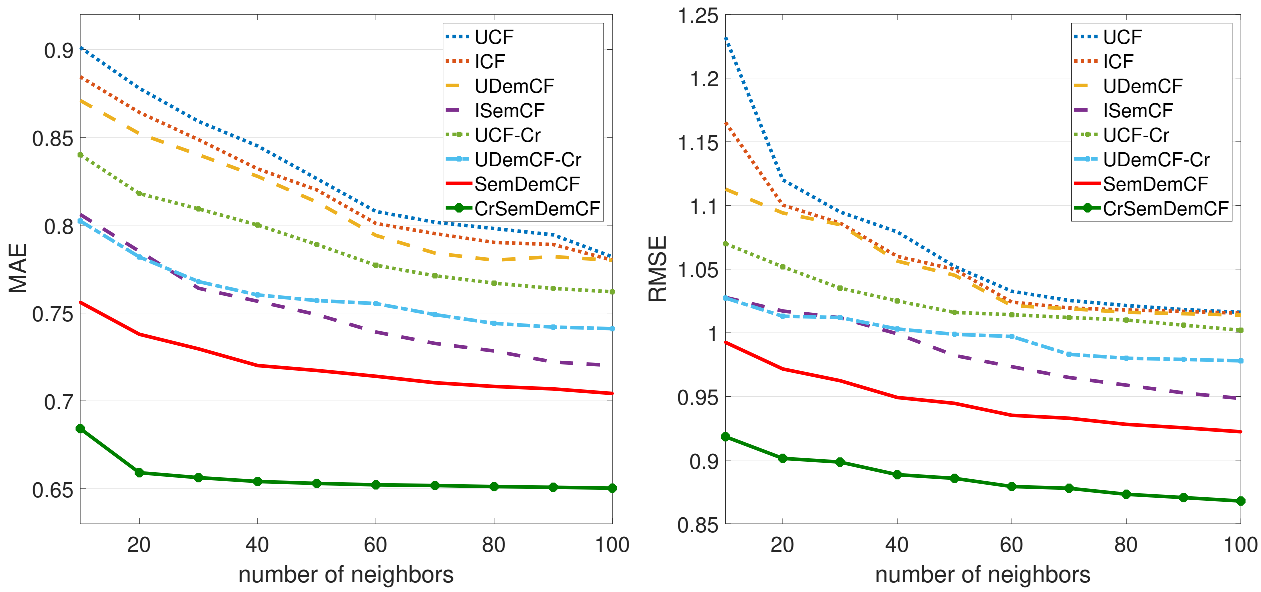


FIGURE 4.7: *MAE* and *RMSE* Values on Yahoo!Movie

TABLE 4.3: *MAE* and *RMSE* Values of our Solution with Various Numbers of Neighbors on Yahoo!Movie

Model	**Metric**	**Number of neighbors**						
		10	20	30	40	60	80	100
UCF	MAE	0.9012	0.8779	0.8591	0.8451	0.8078	0.7981	0.7820
	RMSE	1.2321	1.1201	1.0949	1.0792	1.0326	1.0214	1.0161
ICF	MAE	0.8845	0.8642	0.8487	0.8321	0.8010	0.7902	0.7801
	RMSE	1.1652	1.1003	1.0862	1.0600	1.0241	1.0179	1.0154
UDemCF	MAE	0.8710	0.8521	0.8401	0.8278	0.7942	0.7801	0.7800
	RMSE	1.1128	1.0941	1.0849	1.0563	1.0213	1.0161	1.0139
ISemCF	MAE	0.8061	0.7851	0.7641	0.7567	0.7392	0.7284	0.7201
	RMSE	1.0278	1.0171	1.0118	0.9992	0.9734	0.9589	0.9484
UCF_Cr	MAE	0.8401	0.8180	0.8093	0.8001	0.7772	0.7670	0.7621
	RMSE	1.0700	1.0518	1.0351	1.0250	1.0142	1.0100	1.0020
UDemCF_Cr	MAE	0.8024	0.7819	0.7679	0.7603	0.7554	0.7441	0.7411
	RMSE	1.0273	1.0130	1.0120	1.0030	0.9971	0.9800	0.9780
SemDemCF	MAE	0.7561	0.7379	0.7296	0.7201	0.7140	0.7082	0.7042
	RMSE	0.9925	0.9716	0.9624	0.9492	0.9352	0.9281	0.9223
*CrSemDemCF**	MAE	**0.6842**	**0.6591**	**0.6563**	**0.6541**	**0.6522**	**0.6512**	**0.6503**
	RMSE	**0.9184**	**0.9014**	**0.8985**	**0.8886**	**0.8793**	**0.8732**	**0.8679**

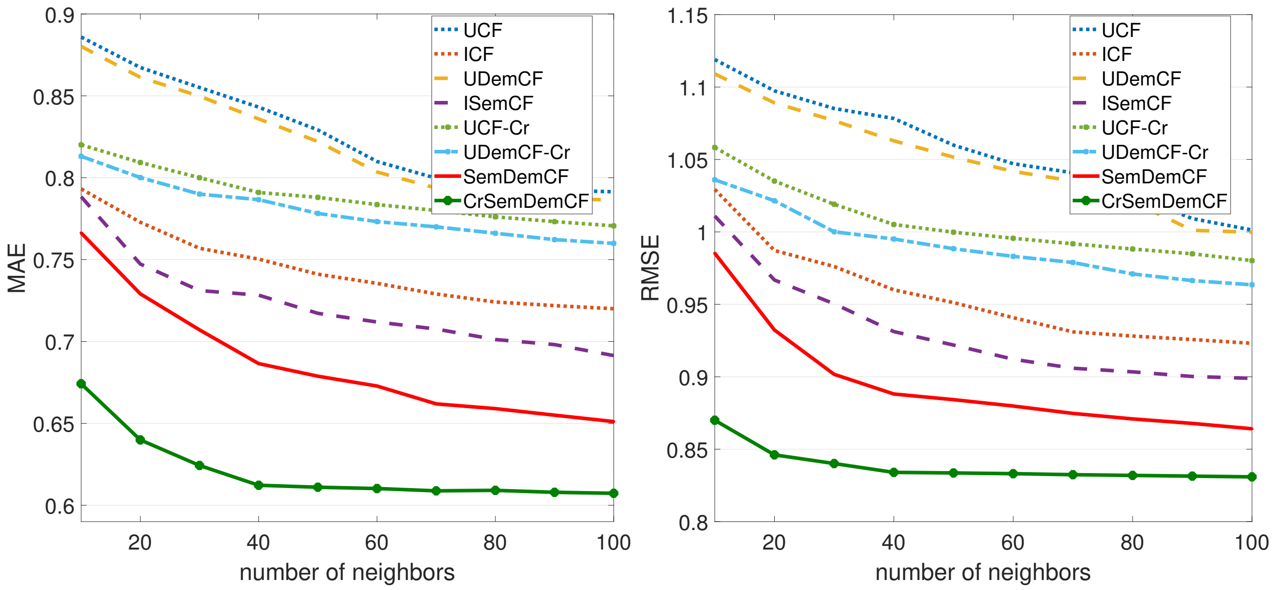


FIGURE 4.8: *MAE* and *RMSE* Values on Movielens 1M

TABLE 4.4: *MAE* and *RMSE* Values of our Solution with Various Numbers of Neighbors on Movielens 1M

Model	**Metric**	**Number of neighbors**						
		10	20	30	40	60	80	100
UCF	MAE	0.8859	0.8673	0.8551	0.8431	0.8099	0.7941	0.7914
	RMSE	1.1190	1.0974	1.0852	1.0783	1.0471	1.0256	1.0013
ICF	MAE	0.7932	0.7728	0.7569	0.7503	0.7355	0.7241	0.7200
	RMSE	1.0294	0.9872	0.9761	0.9600	0.9408	0.9281	0.9231
UDemCF	MAE	0.8802	0.8613	0.8498	0.8359	0.8035	0.7894	0.7864
	RMSE	1.1091	1.0891	1.0768	1.0629	1.0418	1.0210	0.9998
ISemCF	MAE	0.7884	0.7473	0.7311	0.7283	0.7119	0.7012	0.6914
	RMSE	1.0110	0.9668	0.9504	0.9312	0.9121	0.9034	0.8989
UCF_Cr	MAE	0.8201	0.8093	0.8000	0.7910	0.7836	0.7761	0.7706
	RMSE	1.0583	1.0351	1.0191	1.0051	0.9956	0.9882	0.9802
UDemCF_Cr	MAE	0.8131	0.8001	0.7900	0.7866	0.7732	0.7661	0.7599
	RMSE	1.0360	1.0215	1.0001	0.9951	0.9831	0.9710	0.9635
SemDemCF	MAE	0.7663	0.7291	0.7071	0.6865	0.6728	0.6590	0.6511
	RMSE	0.9853	0.9323	0.9017	0.8881	0.8798	0.8709	0.8641
*CrSemDemCF**	MAE	**0.6742**	**0.6399**	**0.6243**	**0.6122**	**0.6102**	**0.6091**	**0.6073**
	RMSE	**0.8700**	**0.8461**	**0.8401**	**0.8340**	**0.8331**	**0.8319**	**0.8309**

the most *MAE*/*RMSE* reduction, followed by Yahoo!Movie and Movielens 100K. This is true even though Yahoo!Movie has higher sparsity than Movielens 1M, due to the fact that Movielens 1M has a much larger number of users' ratings and thus much more variety in users' rating behavior.

4.4.4 Standard Deviation of Errors

In this section, we show the differences between real ratings and the predicted ones of the three datasets, test data, through $SemDemCF$ and $CrSemDemCF$ methods. This is to show the affect of adding credibility to the recommendation system. The frequency of the test dataset for ratings, predictions, and their difference (as errors) histograms with and without considering the users' credibility are shown in Figure 4.9, Figure 4.10, and Figure 4.11 for the Movielens 1M, Yahoo!Movie, and Movielens 100K datasets, respectively, when the number of neighbors is 30. In these three figures: (a) represents the frequency histogram of the real ratings; (b) and (d) show predicted ratings with and without considering credibility respectively; and (c) and (e) represent error measures with and without considering credibility respectively.

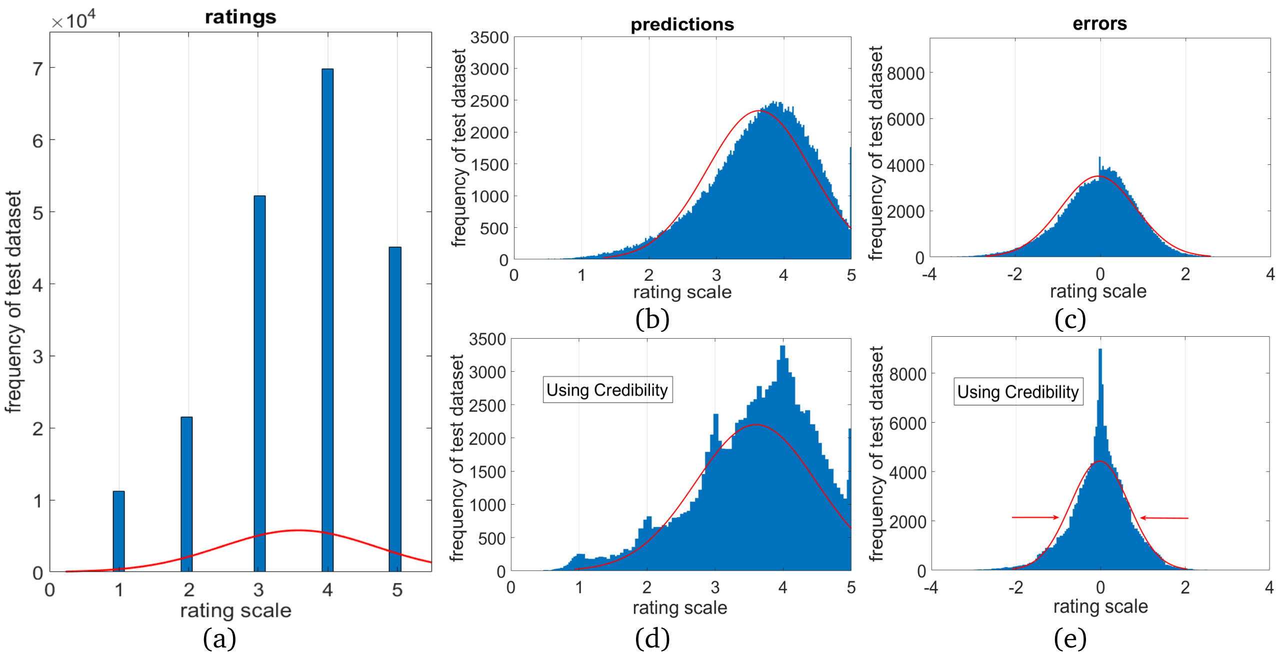


FIGURE 4.9: Frequency of test dataset for (a) real ratings, (b) predicted ratings, (c) error histogram, (d) predicted ratings with the use of credibility, and (e) error histogram with the use of credibility, for the Movielens 1M dataset with the number of neighbors = 30

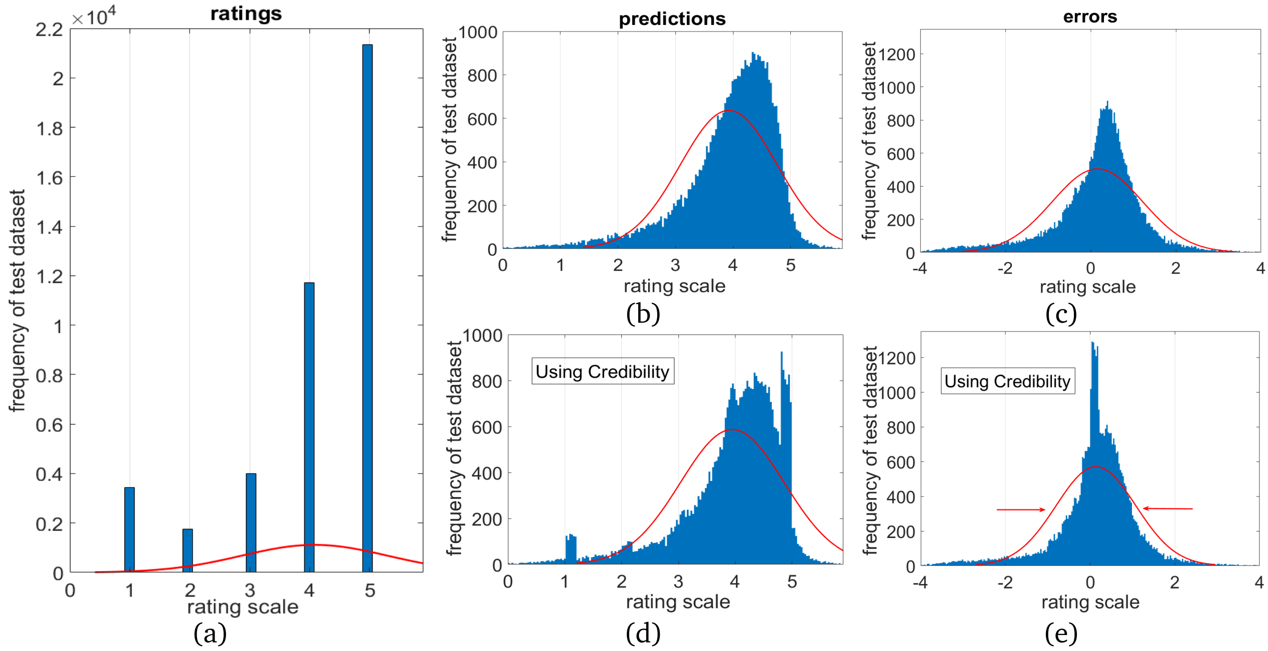


FIGURE 4.10: Frequency of test dataset for (a) real ratings, (b) predicted ratings, (c) error histogram, (d) predicted ratings with the use of credibility, and (e) error histogram with the use of credibility, for the Yahoo!Movie dataset with the number of neighbors = 30

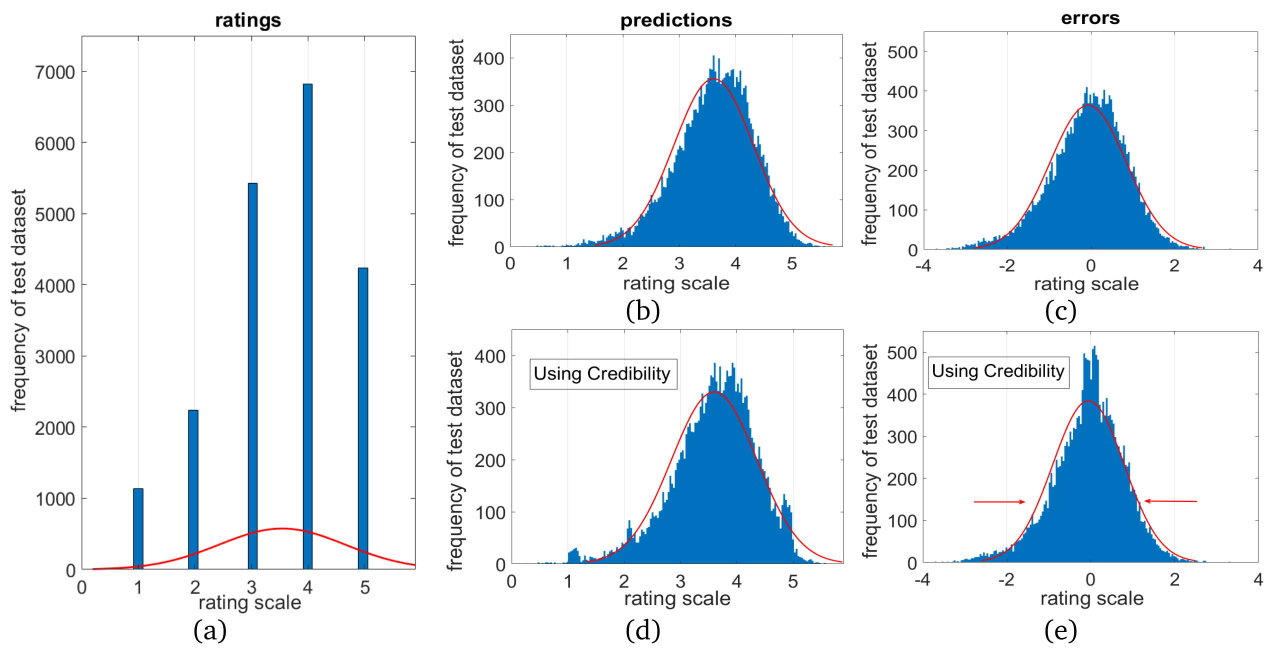


FIGURE 4.11: Frequency of test dataset for (a) real ratings, (b) predicted ratings, (c) error histogram, (d) predicted ratings with the use of credibility, and (e) error histogram with the use of credibility, for the Movielens 100K dataset with the number of neighbors = 30

From Figure 4.9 for the Movielens 1M dataset it can be seen that, similarly to the histogram of real ratings (in Figure 4.9a), the predictions' histograms (in Figure 4.9b and Figure 4.9d) also follow a left-skewed distribution. Comparing Figure 4.9b and Figure 4.9d, we can observe that Figure 4.9c has sharp peaks around the real rating scores because the user credibility is considered in the model. This results in a smaller standard deviation of errors (Figure 4.9e) as compared to the model without considering credibility (Figure 4.9c). The same behavior can be seen in Figure 4.10, and Figure 4.11 for the Yahoo!Movie and Movielens 100K datasets, respectively. This reduction of the width of the error distribution reflects the improvement of the quality of the recommendation.

4.4.5 *Precision* and *Recall* and *F-measure*

To further evaluate the quality of the proposed method, *precision*, *recall* and F_1-*measure* evaluation metrics are used. In the context of recommender systems, it is common to recommend top-K items to the target user. So the notions *precision@*K, *recall@*K and F_1-*measure@*K are used to measure the quality of the recommendations, where $K = 1, 3, 5, 7, \cdots$ is chosen to match the top-K recommendations objective. In fact, *precision@*K is the proportion of the recommended items in the top-K set that are relevant. As an example, if *precision@10* in a top-10 recommendation problem is 70%, this means that 70% of the generated recommendation are relevant to the user. Similarly, if *recall@10* is 20% in our top-10 recommendation system, this means that 20% of the total number of relevant items appear in the recommendation list. Table 4.5 and Figure 4.12 show the results *precision*, *recall* and F_1-*measure* for different Top-K, with and without considering credibility ($CrSemDemCF$ and $SemDemCF$), for Movielens 100K, Yahoo!Movie and Movielens 1M datasets, when the number of neighbors is 30. As it can be seen from Figure 4.12, the most improvement is for Movielens 1M. The *recall* of recommended items in Movielens 1M has been improved by about 20% to 30%, which means that a larger proportion of relevant items are retrieved in the top-K recommendations. The incorporation of the user credibility helps to provide significantly better recommendations.

TABLE 4.5: *Precision*, *recall* and F_1-*measure* for different Top-K, with and without considering credibility, for the Movielens 100K, Yahoo!Movie and Movielens 1M datasets with the number of neighbors = 30

Dataset	Metric	Model	@K							
			1	3	5	7	9	11	13	15
$Movielens100K$	$precision$	$SemDemCF$	81.12	74.47	69.20	64.21	59.39	55.27	51.78	48.54
		$CrSemDemCF$	86.81	75.34	69.11	65.11	59.81	55.30	52.10	48.81
	$recall$	$SemDemCF$	15.86	38.42	53.56	63.71	70.54	75.73	79.70	82.68
		$CrSemDemCF$	22.10	49.65	67.00	72.32	78.49	83.83	89.82	94.67
	$F_1 - measure$	$SemDemCF$	26.54	50.69	60.38	63.96	64.49	63.90	62.77	61.17
		$CrSemDemCF$	35.23	59.85	68.04	68.53	67.89	66.64	65.95	64.41
$Yahoo!Movie$	$precision$	$SemDemCF$	84.14	69.72	55.14	44.41	36.99	31.65	27.65	24.55
		$CrSemDemCF$	94.45	77.36	69.78	58.74	47.24	39.11	33.69	29.18
	$recall$	$SemDemCF$	37.45	75.96	88.95	93.57	95.73	96.93	97.68	98.19
		$CrSemDemCF$	48.32	88.00	94.10	97.38	98.21	98.68	98.90	99.28
	$F_1 - measure$	$SemDemCF$	51.83	72.71	68.08	60.23	53.36	47.72	43.10	39.28
		$CrSemDemCF$	63.93	82.34	80.14	73.28	63.79	56.02	50.26	45.10
$Movielens1M$	$precision$	$SemDemCF$	88.01	84.08	79.59	75.05	70.80	66.89	63.32	60.10
		$CrSemDemCF$	95.12	91.48	86.84	80.76	78.92	75.21	70.54	66.28
	$recall$	$SemDemCF$	11.75	30.88	44.46	54.20	61.36	66.90	71.19	74.75
		$CrSemDemCF$	37.11	49.82	66.61	72.12	78.51	83.82	89.28	95.21
	$F_1 - measure$	$SemDemCF$	20.73	45.17	57.05	62.95	65.74	66.89	67.03	66.34
		$CrSemDemCF$	53.39	64.51	75.39	76.20	78.71	79.28	78.81	78.15

4.4.6 Summary

This chapter provided details on the three datasets (MovieLens 100K, MovieLens 1M, and Yahoo!Movie), evaluation metrics (*MAE*, *RMSE*, standard deviation of errors, *precision*, *recall* and F_1-*measure*), and various baselines used in this work. It also included the experimental results on the performance of the recommendation systems baseline methods through the evaluation metrics using the three datasets. In summary, we observed that adding credibility of users rating to the recommendation helped improving the accuracy of the recommendation. In addition, our proposed method ($CrSemDemCF$) which is an integration of the users' rating credibility, ontological semantics of items and demography information had the best recommendation accuracy as compared to the other baseline methods and the state-of-the-art.

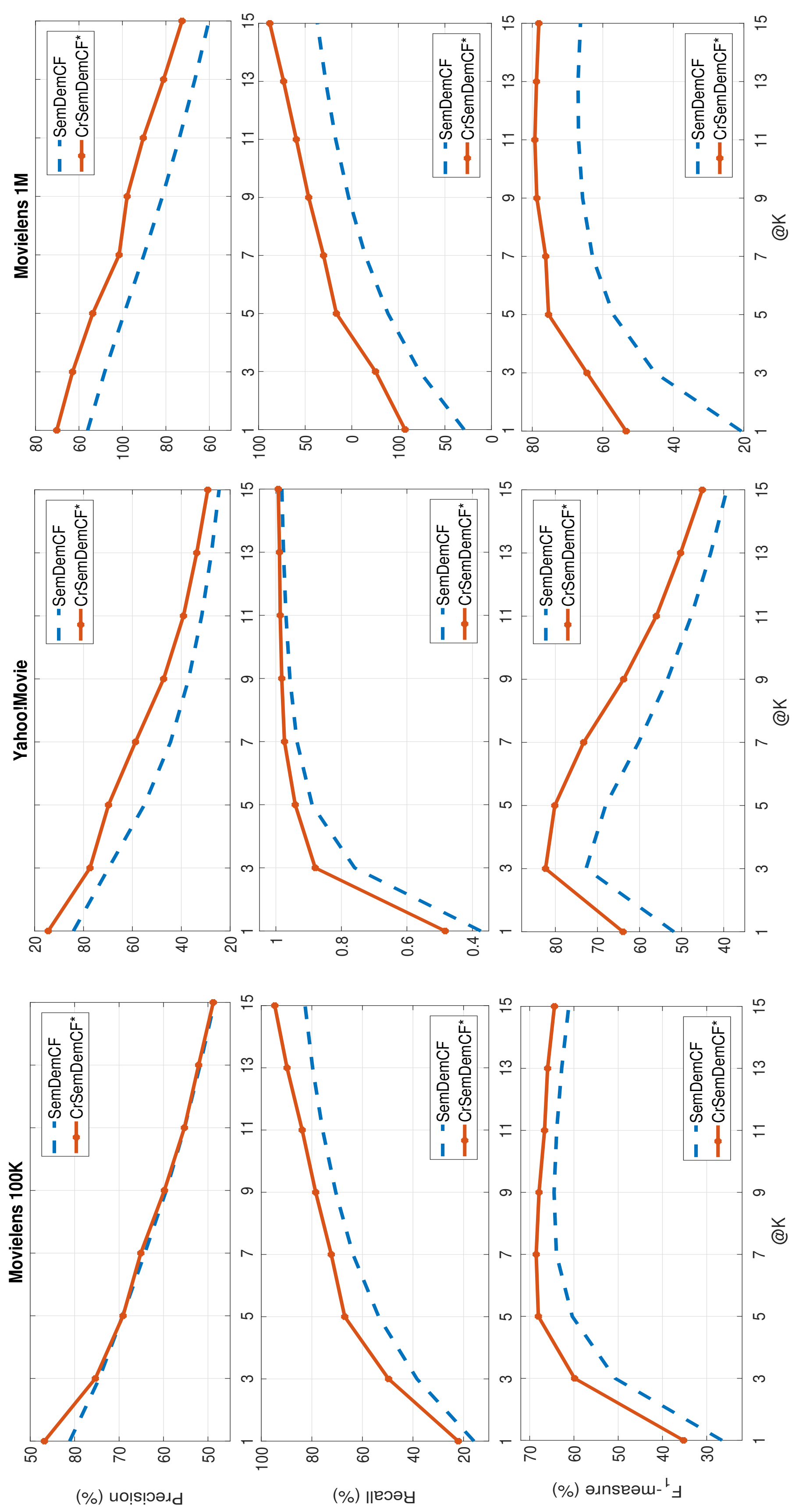


FIGURE 4.12: *Precision*, *recall* and F_1-*measure* for different Top-K, with and without considering credibility, for the Movielens 100K, Yahoo!Movie and Movielens 1M datasets with the number of neighbors = 30

5

Conclusion

The state-of-the-art CF approaches treat all users' ratings as they are, and thus the differences in the rating behavior of users are largely ignored, which further results in inaccurate ratings affecting the recommendation accuracy. In this work, the credibility values of users are evaluated based on their rating behavior as compared to the overall statistics of all ratings. The calculated credibility values are then incorporated with the CF algorithm to decrease the impact of the ratings given by neighbors with low credibility. In addition, the ontological semantics of items and the demographic information of users have been considered when measuring the similarity of items and users. The use of these additional information helped to overcome sparsity and cold start issues and accordingly improve the accuracy of recommendation. Thus, we have developed, to our best knowledge, the first integrated CF recommendation system that incorporates user credibility, the demographic information of users and the ontological semantics of items.

It is worth mentioning, in this ten-month research, we performed several experiments using real-world datasets from MovieLens and Yahoo!Movie in order to evaluate the performance of our proposed solution. The experimental results showed the recommendation accuracy improvement when user credibility was incorporated into the CF recommendation systems as compared to other methods and the state-of-the-art. The proposed approach has improved the recommendation quality in terms of accuracy, *precision*, *recall*, and F_1-*measure* significantly. The incorporation of the rating credibility also helped to reduce the standard deviation of errors between the prediction values and real ratings. It finally helped the system to provide more accurate recommendations to users.

For future work, we are going to investigate other factors that are associated with the credibility of users to further improve the prediction accuracy such as the users' reviews in social media. To do this, natural language processing techniques and sentiment analysis can be used. We will also consider the temporal features of users' ratings in the credibility measurement as users' interests may change during the time. In addition, including more level of ontology relations among items can further optimize the proposed solution.

List of Symbols

The following list is neither exhaustive nor exclusive, but may be helpful.

CF Collaborative Filtering

CB Content-Based

DB Demographic-Based

KB Knowledge-Based

$movie$-CF-Sim movie-based CF similarity

$user$-CF-Sim user-based CF similarity

$movie$-Sem-Sim movie-based semantic similarity

$user$-Dem-Sim user-based demographic similarity

KNN K Nearest-Neighbors

pdf probability density fuction

β shape parameter

η scale parameter

γ location parameter

AE Absolute Error

α regulation parameter

λ size of correcting step

MAE Mean Absolute Error

$RMSE$ Root Mean Square Error

UCF User-Based CF

ICF Item-Based CF

$UDemCF$ User-based CF with considering Demographic information of users

$ISemCF$ Item-based CF with considering ontological Semantics of Items

$SemDemCF$ CF with considering ontological Semantics of items and Demographic information of users

$UCF-Cr$ User-based CF considering users' rating Credibility

$UDemCF-Cr$ User-Based CF with considering users' rating credibility and Demographic information of users

$CrSemDemCF$ CF with considering users' rating Credibility, ontological Semantics of items, and Demographic information of users

References

[1] F. Ricci, L. Rokach, and B. Shapira. *Recommender systems: introduction and challenges*. In *Recommender systems handbook*, pp. 1–34 (Springer, 2015). 1, 2, 3, 7, 9, 10, 11, 12, 13, 16, 17, 18, 21, 25

[2] Y. Wang, J. Deng, J. Gao, and P. Zhang. *A hybrid user similarity model for collaborative filtering*. Information Sciences **418**, 102 (2017). 2

[3] Z. Zhang, W. Zhao, J. Yang, S. Nepal, C. Paris, and B. Li. *Exploiting users' rating behaviour to enhance the robustness of social recommendation*. In *International Conference on Web Information Systems Engineering*, pp. 467–475 (Springer, 2017). 2, 14, 15

[4] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, and X. Li. *Connecting social media to e-commerce: Cold-start product recommendation using microblogging information*. IEEE Transactions on Knowledge and Data Engineering **28**(5), 1147 (2016). 3

[5] M. Nilashi, O. bin Ibrahim, and N. Ithnin. *Hybrid recommendation approaches for multi-criteria collaborative filtering*. Expert Systems with Applications **41**(8), 3879 (2014). 8, 13

[6] N. R. Kermany and S. H. Alizadeh. *A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques*. Electronic Commerce Research and Applications **21**, 50 (2017). 9, 13, 15, 16, 17, 18, 35

[7] M. Nilashi, O. Ibrahim, and K. Bagherifard. *A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques*. Expert Systems with Applications **92**, 507 (2018). 15, 16

[8] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang. *Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars)*. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 193–202 (ACM, 2014).

[9] K. N. Jain, V. Kumar, P. Kumar, and T. Choudhury. *Movie recommendation system: Hybrid information filtering system*. In *Intelligent Computing and Information and Communication*, pp. 677–686 (Springer, 2018).

[10] Z. Wang, X. Yu, N. Feng, and Z. Wang. *An improved collaborative movie recommendation system using computational intelligence*. Journal of Visual Languages & Computing **25**(6), 667 (2014). 8

[11] F. Yu, N. Che, Z. Li, K. Li, and S. Jiang. *Friend recommendation considering preference coverage in location-based social networks*. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 91–105 (Springer, 2017). 8

[12] X. Ma, J. Ma, H. Li, Q. Jiang, and S. Gao. *Armor: A trust-based privacy-preserving framework for decentralized friend recommendation in online social networks*. Future Generation Computer Systems **79**, 82 (2018).

[13] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang. *Friendbook: a semantic-based friend recommendation system for social networks*. IEEE transactions on mobile computing **14**(3), 538 (2015). 8

[14] S. Zhang, S. Zhang, N. Y. Yen, and G. Zhu. *The recommendation system of micro-blog topic based on user clustering*. Mobile Networks and Applications **22**(2), 228 (2017). 8

[15] J. Xuan, X. Luo, G. Zhang, J. Lu, and Z. Xu. *Uncertainty analysis for the keyword system of web events*. IEEE Transactions on Systems, Man, and Cybernetics: Systems **46**(6), 829 (2016). 8

[16] A. Van den Oord, S. Dieleman, and B. Schrauwen. *Deep content-based music recommendation*. In *Advances in neural information processing systems*, pp. 2643–2651 (2013). 8

[17] X. Wang and Y. Wang. *Improving content-based and hybrid music recommendation using deep learning*. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 627–636 (ACM, 2014).

[18] S. Oramas, V. C. Ostuni, T. D. Noia, X. Serra, and E. D. Sciascio. *Sound and music recommendation with knowledge graphs*. ACM Transactions on Intelligent Systems and Technology (TIST) **8**(2), 21 (2017).

[19] T. Li, M. Choi, K. Fu, and L. Lin. *Music sequence prediction with mixture hidden markov models*. arXiv preprint arXiv:1809.00842 (2018). 8

[20] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett. *Novel centroid selection approaches for kmeans-clustering based recommender systems*. Information sciences **320**, 156 (2015). 8

[21] S. S. Sohail, J. Siddiqui, and R. Ali. *An owa-based ranking approach for university books recommendation*. International Journal of Intelligent Systems **33**(2), 396 (2018).

[22] P. Jomsri. *Book recommendation system for digital library based on user profiles by using association rule*. In *Innovative Computing Technology (INTECH), 2014 Fourth International Conference on*, pp. 130–134 (IEEE, 2014). 8

[23] M. Al-Hassan, H. Lu, and J. Lu. *A semantic enhanced hybrid recommendation approach: A case study of e-government tourism service recommendation system*. Decision Support Systems **72**, 97 (2015). 8

[24] R. Colomo-Palacios, F. J. García-Peñalvo, V. Stantchev, and S. Misra. *Towards a social and context-aware mobile recommendation system for tourism*. Pervasive and Mobile Computing **38**, 505 (2017).

[25] A. Dewangan and R. Chatterjee. *Tourism recommendation using machine learning approach*. In *Progress in Advanced Computing and Intelligent Engineering*, pp. 447–458 (Springer, 2018). 8

[26] M. Habibi and A. Popescu-Belis. *Keyword extraction and clustering for document recommendation in conversations*. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) **23**(4), 746 (2015). 8

[27] J. Beel, S. Langer, and B. Gipp. *Tf-iduf: a novel term-weighting scheme for user modeling based on users' personal document collections*. Proceedings of the 12th iConference. To appear in March 2017 (2017). 8

[28] L. Li, L. Zheng, F. Yang, and T. Li. *Modeling and broadening temporal user interest in personalized news recommendation*. Expert Systems with Applications **41**(7), 3168 (2014). 8

[29] S. Okura, Y. Tagami, S. Ono, and A. Tajima. *Embedding-based news recommendation for millions of users*. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1933–1942 (ACM, 2017).

[30] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li. *Drn: A deep reinforcement learning framework for news recommendation*. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 167–176 (International World Wide Web Conferences Steering Committee, 2018). 8

[31] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang. *Recommender system application developments: a survey*. Decision Support Systems **74**, 12 (2015). 9

[32] F. S. Gohari, F. S. Aliee, and H. Haghighi. *A new confidence-based recommendation approach: combining trust and certainty*. Information Sciences **422**, 21 (2018). 9

[33] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. *Item-based collaborative filtering recommendation algorithms*. In *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295 (ACM, 2001). 9, 12, 35

[34] B. Purkaystha, T. Datta, M. S. Islam, *et al.* *Rating prediction for recommendation: Constructing user profiles and item characteristics using backpropagation*. Applied Soft Computing **75**, 310 (2019). 10

[35] G. Shani and A. Gunawardana. *Evaluating recommendation systems*. In *Recommender systems handbook*, pp. 257–297 (Springer, 2011). 11

[36] R. Burke. *Hybrid web recommender systems*. In *The adaptive web*, pp. 377–408 (Springer, 2007). 11, 12

[37] M. Fu, H. Qu, D. Moges, and L. Lu. *Attention based collaborative filtering*. Neurocomputing **311**, 88 (2018). 12

[38] D. Jannach, Z. Karakaya, and F. Gedikli. *Accuracy improvements for multi-criteria recommender systems*. In *Proceedings of the 13th ACM conference on electronic commerce*, pp. 674–689 (ACM, 2012). 13, 30

[39] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. *Recommender systems survey*. Knowledge-based systems **46**, 109 (2013). 13

[40] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. *Grouplens: an open architecture for collaborative filtering of netnews*. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186 (ACM, 1994). 13

[41] M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. *Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit*. In *Proceedings of the fifth ACM conference on Recommender systems*, pp. 133–140 (ACM, 2011). 13

[42] P. Cremonesi, Y. Koren, and R. Turrin. *Performance of recommender algorithms on top-n recommendation tasks*. In *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39–46 (ACM, 2010). 13

[43] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining. 1st* (2005). 13

[44] L. Candillier, F. Meyer, and F. Fessant. *Designing specific weighted similarity measures to improve collaborative filtering systems*. In *Industrial Conference on Data Mining*, pp. 242–255 (Springer, 2008). 13, 21, 22

[45] S. Bag, S. K. Kumar, and M. K. Tiwari. *An efficient recommendation generation using relevant jaccard similarity*. Information Sciences (2019). 13

[46] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, J. J. Pazos-Arias, Y. Blanco-Fernández, and M. López-Nores. *A semantic approach to improve neighborhood formation in collaborative recommender systems*. Expert Systems with Applications **41**(17), 7776 (2014). 13

[47] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez. *Collaborative filtering based on significances*. Information Sciences **185**(1), 1 (2012). 13

[48] K. Choi and Y. Suh. *A new similarity function for selecting neighbors for each target item in collaborative filtering*. Knowledge-Based Systems **37**, 146 (2013). 13

[49] P. Moradi and S. Ahmadian. *A reliability-based recommendation method to improve trust-aware recommender systems*. Expert Systems with Applications **42**(21), 7386 (2015). 14

[50] N. Polatidis and C. K. Georgiadis. *A multi-level collaborative filtering method that improves recommendations*. Expert Systems with Applications **48**, 100 (2016). 14, 34, 40

[51] Z. Zhang, Y. Liu, Z. Jin, and R. Zhang. *A dynamic trust based two-layer neighbor selection scheme towards online recommender systems*. Neurocomputing **285**, 94 (2018). 14, 34, 40

[52] U. Liji, Y. Chai, and J. Chen. *Improved personalized recommendation based on user attributes clustering and score matrix filling*. Computer Standards & Interfaces **57**, 59 (2018). 14, 34, 40

[53] J. Li, W. Xu, W. Wan, and J. Sun. *Movie recommendation based on bridging movie feature and user interest*. Journal of computational science **26**, 128 (2018). 14, 34, 40

[54] G. Guo, J. Zhang, and D. Thalmann. *Merging trust in collaborative filtering to alleviate data sparsity and cold start*. Knowledge-Based Systems **57**, 57 (2014). 14, 15

[55] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. *Detecting product review spammers using rating behaviors*. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 939–948 (ACM, 2010). 14, 15

[56] H. Fang, J. Zhang, and N. Magnenat Thalmann. *Subjectivity grouping: learning from users' rating behavior*. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1241–1248 (International Foundation for Autonomous Agents and Multiagent Systems, 2014). 14, 15

[57] X. Yang, H. Steck, and Y. Liu. *Circle-based recommendation in online social networks*. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1267–1275 (ACM, 2012). 14, 15

[58] T. R. Gruber. *A translation approach to portable ontology specifications*. Knowledge acquisition **5**(2), 199 (1993). 15

[59] G. Antoniou and F. Van Harmelen. *A semantic web primer* (MIT press, 2004). 15

[60] J. K. Tarus, Z. Niu, and A. Yousif. *A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining*. Future Generation Computer Systems **72**, 37 (2017). 15

[61] O. Daramola, M. Adigun, and C. Ayo. *Building an ontology-based framework for tourism recommendation services*. Information and communication technologies in tourism 2009 pp. 135–147 (2009). 15, 16

[62] Q. Shambour and J. Lu. *A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations*. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pp. 71–78 (IEEE Computer Society, 2011). 15, 16, 25

[63] Q. Shambour and J. Lu. *A trust-semantic fusion-based recommendation approach for e-business applications*. Decision Support Systems **54**(1), 768 (2012). 15, 16

[64] B. Alhijawi, N. Obeid, A. Awajan, and S. Tedmori. *Improving collaborative filtering recommender systems using semantic information*. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 127–132 (IEEE, 2018). 16

[65] M. Martínez-García, A. Valls, and A. Moreno. *Inferring preferences in ontology-based recommender systems using wowa*. Journal of Intelligent Information Systems pp. 1–31 (2018). 16

[66] L. aDepartament de Llenguatges i Sistemes Informàtics. *Taking advantage of semantics in recommendation systems*. In *Artificial Intelligence Research and Development: Proceedings of*

the 13th International Conference of the Catalan Association for Artificial Intelligence, vol. 220, p. 163 (IOS Press, 2010). 16

[67] M. J. Pazzani. *A framework for collaborative, content-based and demographic filtering*. Artificial intelligence review **13**(5-6), 393 (1999). 17

[68] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel. *Inferring the demographics of search users: Social data meets search queries*. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 131–140 (ACM, 2013). 17, 18

[69] X. W. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li. *We know what you want to buy: a demographic-based system for product recommendation on microblogs*. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1935–1944 (ACM, 2014). 17, 18

[70] W. X. Zhao, S. Li, Y. He, L. Wang, J.-R. Wen, and X. Li. *Exploring demographic information in social media for product recommendation*. Knowledge and Information Systems **49**(1), 61 (2016). 17, 18

[71] V. Subramaniyaswamy, G. Manogaran, R. Logesh, V. Vijayakumar, N. Chilamkurti, D. Malathi, and N. Senthilselvan. *An ontology-driven personalized food recommendation in iot-based healthcare system*. The Journal of Supercomputing pp. 1–33 (2018). 17, 18

[72] R. B. Abernethy. *The New Weibull handbook: reliability and statistical analysis for predicting life, safety, supportability, risk, cost and warranty claims* (Dr. Robert B. Abernethy, 2004). 27

[73] Y. Koren. *Factor in the neighbors: Scalable and accurate collaborative filtering*. ACM Transactions on Knowledge Discovery from Data (TKDD) **4**(1), 1 (2010). 30

[74] Y. Koren, R. Bell, and C. Volinsky. *Matrix factorization techniques for recommender systems*. Computer (8), 30 (2009). 30

[75] G. Adomavicius and Y. Kwon. *New recommendation techniques for multicriteria rating systems*. IEEE Intelligent Systems **22**(3) (2007). 34