2019

ENHANCE NMF-BASED RECOMMENDATION SYSTEMS WITH AUXILIARY INFORMATION IMPUTATION

Fatemah Alghamedy
University of Kentucky, fal247@g.uky.edu

Digital Object Identifier: https://doi.org/10.13023/etd.2019.144

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Fatemah Alghamedy, Student

Dr. Jun Zhang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

ENHANCE NMF-BASED RECOMMENDATION SYSTEMS WITH AUXILIARY INFORMATION IMPUTATION

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By

Fatemah Alghamedy

Lexington, Kentucky

Director: Jun Zhang, Ph.D., Professor of Computer Science

Lexington, Kentucky

2019

ABSTRACT OF DISSERTATION

ENHANCE NMF-BASED RECOMMENDATION SYSTEMS WITH AUXILIARY INFORMATION IMPUTATION

This dissertation studies the factors that negatively impact the accuracy of the collaborative filtering recommendation systems based on nonnegative matrix factorization (NMF). The keystone in the recommendation system is the rating that expresses the user's opinion about an item. One of the most significant issues in the recommendation systems is the lack of ratings. This issue is called "cold-start" issue, which appears clearly with New-Users who did not rate any item and New-Items, which did not receive any rating.

The traditional recommendation systems assume that users are independent and identically distributed and ignore the connections among users whereas the recommendation actually is a social activity. This dissertation aims to enhance NMF-based recommendation systems by utilizing the imputation method and limiting the errors that are introduced in the system. External information such as trust network and item categories are incorporated into NMF-based recommendation systems through the imputation.

The proposed approaches impute various subsets of the missing ratings. The subsets are defined based on the total number of the ratings of the user or item before the imputation, such as impute the missing ratings of New-Users, New-Items, or cold-start users or items that suffer from the lack of the ratings. In addition, several factors are analyzed that affect the prediction accuracy when the imputation method is utilized with NMF-based recommendation systems. These factors include the total number of the ratings of the user or item before the imputation, the total number of imputed ratings for each user and item, the average of imputed rating values, and the value of imputed rating values. In addition, several strategies are applied to select the subset of missing ratings for the imputation that lead to increasing the prediction accuracy and limiting the imputation error. Moreover, a comparison is conducted with some popular methods that are in common with the proposed method in utilizing the imputation to handle the lack of ratings, but they differ in the source of the imputed ratings.

Experiments on different large-size datasets are conducted to examine the proposed approaches and analyze the effects of the imputation on accuracy. Users and items are divided into three groups based on the total number of the ratings before the imputation is applied and their recommendation accuracy is calculated.

The results show that the imputation enhances the recommendation system by capacitating the system to recommend items to New-Users, introduce New-Items to users, and increase the accuracy of the cold-start users and items. However, the analyzed factors play important roles in the recommendation accuracy and limit the error that is introduced from the imputation.

KEYWORDS: recommendation system, collaborative filtering, NMF, trust matrix, imputation.

Fatemah Alghamedy

April 16, 2019

ENHANCE NMF-BASED RECOMMENDATION SYSTEMS WITH AUXILIARY INFORMATION IMPUTATION

By

Fatemah Alghamedy

Jun Zhang, Ph.D.

____

Director of Dissertation

Miroslaw Truszczynski, Ph.D.

____

Director of Graduate Studies

April 16, 2019

____

Date

To my father, my mother, my husband, my daughter, my son, my sisters, and my brother.

ACKNOWLEDGMENTS

I would like to express my gratitude to the people who helped and encouraged me during my Ph.D. journey.

First of all, I would like to express the deepest appreciation to my academic advisor, Prof. Jun Zhang, for his continuous support, patience, motivation, enthusiasm, and immense knowledge. His guidance helped me throughout my time of researching and writing this dissertation.

Next, I would like to thank the committee faculty members Dr. Jerzy W. Jaromczyk (Department of Computer Science), Dr. Jinze Liu (Department of Computer Science), Dr. Sen-ching Samson Cheung (Department of Electrical and Computer Engineering), and the outside examiner Prof. Qiang Ye (Department of Mathematics) for their time and the helpful comments on my dissertation.

Also, I would like to express my special thanks to the Computer Science Department and to UK Research Computing (UKRC) for their research resources.

In addition, I would like to thank the Imam Abdulrahman bin Faisal University and Saudi Arabian Cultural Mission (SACM) for their full financial support for the entirety of my Ph.D. years.

Last but not least, I would like to thank my family for their unlimited support, especially my father and my mother, whom I wouldn't be able to reach this success without their care and support ever since I was a child. My father taught me the meaning of determination and ambition and was always there when I needed any advice or opinion, especially concerning the academic field. My mother surrounded me by her love and care and never forgot to support me and take care of me even though there were thousands of miles distance between us. A special thanks to my adorable children, Aseel and Suhail. Thank you for bearing with me on this journey, as I'm mostly busy; and above all of that, thank you for being a source of strength for me. Having you around me gave me the will to continue and to keep moving forward.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

With the emergence of E-commerce, recommendation system [53] becomes an important tool that can help both sellers and buyers. The way it helps sellers is by increasing their profits and increasing advertising items to customers. In addition, recommendation systems help buyers to find items they are looking for easily.

Recommendation systems (RS) are classified into three categories: content-based (CB), collaborative filtering (CF), and hybrid. The content-based (CB) system recommends items similar to the ones that users preferred in the past by utilizing external information, such as item descriptions and user profiles, to calculate the similarity between items or users. A CB method needs manual intervention to collect the item descriptions and user profiles, which are susceptible to errors and do not scale to the large item basis. In contrast, collaborative filtering (CF) assumes that users who agree on the items in the past agree in the future as well. CF calculates the similarity measurement between users with their previous ratings of common items. If two users have a high similarity between them, we can predict that these two users may like the same items in the future. In addition, there is no need for any external information like the CB method. However, there are some approaches that combine content-based (CB) and collaborative filtering (CF) to merge the advantages of both systems into one system and avoid each of the system's limitations [8, 12, 43, 46, 47, 63, 70].

Collaborative filtering is the most popular approach because it has higher accuracy in its results and needs fewer resources. Basically, collaborative filtering algorithms are divided into two main categories: memory-based methods and model-based methods.

Memory-based methods, also known as neighborhood-based methods, rely on the similarity measure. The similarity measurement is calculated based on common ratings, which could be common ratings between users for the same item (user-oriented CF) or common ratings between items from the same user (item-oriented CF). The user-oriented CF computes the similarity between users based on their past ratings on common items between them; such users are known as user neighbors. For each missing rating, memory-based methods predict the rating by using a past neighbor's ratings for that item. If there are no common rating items between users, similarity cannot be calculated, especially with cold-start users. Cold-start users are the users in the system who did not rate many items, e.g., fewer than five items. It is hard to find neighbors for cold-start users, thus the system cannot recommend items. A good recommendation system must have some strategies to allow cold-start users to use the system. In addition, one of the most significant issues is the rating matrix sparseness due to the fact that most users rate a small number of items, which causes the rating matrix to suffer from sparsity.

On the other hand, model-based methods have been proposed to reduce the issues with the memory-based methods. In model-based algorithms, users are modeled based on their past ratings by employing statistical and machine learning techniques to learn models then using these learned models to predict the missing ratings. There is no need to calculate the similarity and find the users' neighbors. There are several different models, such as SVD (Singular Value Decomposition) based latent factor CF [58], aspect model [21, 60], clustering methods [26], Bayesian model [82], and low-dimensional linear factor models, such as matrix factorization (MF) [51, 56, 64, 80] which is the most efficient model for very massive datasets. There are different types of matrix factorization, such as weighted nonnegative matrix factorization (WNMF) [80], maximum margin matrix factorization (MMMF) [51], and probabilistic matrix factorization (PMF) [56].

However, the model-based algorithms still suffer from the data sparsity problem and fail to address the cold-start users issue. It is irrational to rely only on the rating matrix and ignore other sources of information in the dataset that we may use to increase the accuracy of the recommendation, such as user information (gender, occupation, location, interests, etc.), item categories, and social information (relationship between users or trust and distrust list).

Traditional recommendation systems assume that users are i.i.d. (independent and identically distributed) and that they ignore the connections among users, which is insufficient because it does not reflect the real world recommendations [35, 36].

Basically, recommendation is a social activity. For example, we usually ask a friend to recommend movies to see or books to read [13]. In addition, friends in real life are more qualified to suggest good and useful recommendations than the traditional recommendation system [2]. Sinha and Swearingen showed [62] that a user prefers recommendations from friends over recommendation systems in terms of quality and usefulness even if the recommendation systems have a high novelty factor.

There are many studies that show the relationship between the users' taste and their friends' taste. Ziegler and Lausen demonstrated [83] through an empirical study of a real online community that there is similarity in the ratings between users and their friends. In addition, users who are in the same social network often have similar behaviors and tastes [42]. Singla and Richardson [61] analyzed over 10 million users on the social network MSN Instant Messenger with their related search records. They concluded that users who chat with each other are more probable to have similar interests, such as their web search and the topics they are searching for. The analysis of this large dataset revealed that friends have a tendency to give similar ratings to items [18].

There are websites designed to rate and review items by the users. Some of these websites allow for the creation of a trust network between the users. Users trust each other at the beginning because they agree with each other's ratings and reviews. We call the user that creates the trust relationship a *trustor* and the user that has been trusted a *trustee*. However, after a while, the trustee influences the trustor even on some topics that they did not agree on in the past [4]. In addition, most users participate in social networks more than rating items [9].

On the other hand, most data analysis algorithms require complete data. Imputation is one of the approaches that has been used to complete missing data through the process of replacing missing data with substituted values [33, 59]. In addition, techniques and assumptions are used to estimate missing data for the imputation process [68]. Imputation has been used in several fields, such as social surveys, industrial experiments, and medical databases [59].

There are two basic methods of imputation: single imputation and multiple imputation (MI) [7]. With single imputation, each missing value is substituted by one single value. There are several single imputation methods, such as mean imputation, hot-deck imputation, and k-Nearest Neighbor (kNN) imputation. On the other hand, multiple imputation methods generate more than one imputed data and analyze each imputed data independently, such as Bayesian multiple imputation [55].

Imputation has been used in the recommendation system to reduce rating matrix sparsity, which is one of the biggest issues in the recommendation system. Even though most recommendation methods do not require complete data, the imputation has been used because the predicted ratings are more accurate when there are more ratings available in the rating matrix. However, the imputation process has been used as a pre-processing step in which missing data are imputed before the rating prediction process, then the ratings are predicted based on the original and imputed ratings. Prediction results often improve by using the imputed data with an extremely sparse rating matrix [65].

Even though the imputation alleviates the sparsity issue, we have to consider the error, which may be introduced from the imputed ratings. To get the benefit of the imputation and reduce the imputation error, we need to answer two important questions: which missing data should be imputed, and how to impute ratings [49]. For that, the most efficient imputation-based collaborative filtering methods do not impute all missing data, so they use strategies to select which missing data should be imputed. There are several methods to impute missing data, such as the ratings mean of either all known ratings or ratings of a particular item or user, predictive mean matching (PMM) [32], and linear regression. In addition, machine learning classification methods have been used for imputation, such as naive Bayes, neural networks, decision tree, decision table, lazy Bayesian rules, logistic regression, and others [76].

There are several imputation approaches that have been proposed with both collaborative filtering methods: memory-based and model-based collaborative filtering, which are sometimes called imputation-based collaborative filtering methods.

Because rating matrix sparsity affects the results of the recommendation system, we propose new approaches to reduce the sparsity using the trust user network and item auxiliary information. In our experimental datasets, users trust other users based on their ratings since they don't know any other information about each other except the ratings. We can expect that if a user did not provide a rating for an item, then his/her rating for that item will be similar to his/her trustees' rating.

1.1 Related Works

Nonnegative Matrix Factorization (NMF) [29] is a dimension reduction method which is vastly used in many applications, such as clustering [5, 25], text mining [45, 78], and image processing and analysis [57, 80]. NMF has been applied for collaborative filtering. Zhang et al. [80] used NMF to learn the missing values in the rating matrix, which is based on the collaborative filtering method. A nonnegativity constraint is enforced in the linear model to guarantee that all users' ratings can be represented as an additive linear combination of canonical coordinates. They introduced two methods on NMF to learn a constrained linear model from an incomplete rating matrix. The first one is based on the Expectation-Maximization (EM) procedure and the other is Weighted Nonnegative Matrix Factorization (WNMF), which has been applied in [38]. Ding et al. proposed [5] an unconstrained 3-factor NMF method that has an additional factor matrix to absorb the different scales in the two matrix factors in basic NMF.

Relying only on rating information is not sufficient because most datasets suffer from sparsity. In addition, cold-start users who did not rate many items have the most negative impact. To alleviate this issue, other sources of information have been used, such as user information (gender, occupation, location, interests, etc.)[13, 75], item categories [13, 75], rating reviews (helpfulness) [73], and social information (relationship between users or trust and distrust list) [13, 17, 18, 22, 23, 24, 35, 36, 37, 39, 40, 41].

Aux-NMF [75] is one of the studies that incorporate the user and item information into the NMF-based method. Their proposed method surpasses the SVD-based data update approach [74].

On the other hand, the social network has been utilized to alleviate the most serious problems of the recommendation system: rating matrix sparsity and cold-start users. The social network can be gathered from internal or external resources. Social media, such as Facebook, Twitter, and Instagram, are counted as external resources that can be used to recommend items to users [17]. Some review websites even allow users to create a list of users whose reviews they believe are trustworthy; that list of users is called a *trust list.* Social relationship information has been incorporated into both memory-based [18, 39, 40, 41] and model-based [13, 22, 23, 35, 36, 37] collaborative filtering methods.

In the memory-based approach, the neighborhood of users is defined based on the social network rather than the similarity measures. By analyzing the statistical information of the epinions.com dataset, Massa and Bhattacharjee [41] presented evidence that the total number of users who have a trust relationship between them is more than the total number of users who have a similarity between them because most users do not have a commonly rated item between them. The trust value can be calculated between more users than similarity by utilizing trust propagation. Massa and Bhattacharjee proposed [41], a new method that incorporates social network into memory-based collaborative filtering, [39, 40] which replaces the similarity measure with the trust metric to predict the missing ratings. Instead of computing the similarity between two users based on their commonly rated items, they computed trust weights between users based on the trust web network. The key differences in this method are in the neighbors' identification and weights. The results show that the new method using only trust metrics is more effective in terms of accuracy and coverage than either the purely collaborative filtering or the system that combines trust and similarity, especially with cold-start users.

Massa and Avesani [39, 40] used the MoleTrust algorithm, a local trust metric that is a depth-first graph walking algorithm with a tunable trust propagation horizon that sets the distance to which trust is propagated. However, other approaches have been proposed with similar ideas as [39, 40] but with a different walk algorithm that is used to propagate trust through the social network. Some examples of walk algorithms are random walk [27] and breadth-first walk: TidalTrust [10]. MoleTrust is similar to TidalTrust, but MoleTrust sets a maximum depth of the users regardless of any specific users or items.

On the other hand, the social network has been incorporated into the model-based collaborative filtering method. Hao Ma et. al [36, 37] integrated the social network structure and the user-item rating matrix based on probabilistic matrix factorization. In addition, they not only learned the user latent feature space and item latent feature space from a user-item matrix, but they also utilized user social network simultaneously and seamlessly. They connected two different data resources - the rating matrix and trust matrix - through the shared user latent feature space. They assumed that the user latent feature space in the rating matrix is the same as the user latent feature space in the trust matrix. This method has more accuracy than Maximum Margin Matrix Factorization (MMMF) [51], Probabilistic Matrix Factorization (PMF) [56], and Constrained Probabilistic Matrix Factorization (CPMF) [56] algorithms.

Ma et al. [35] introduced a method to fuse the users' tastes and their trusted friends' tastes together using the probabilistic matrix factorization framework. In addition, they balanced users' tastes with their trusted friends' tastes using a control parameter. Their proposed method - RSTE - achieved better accuracy than the SoRec [36].

He and Chu [18] proposed a model to make recommendations by taking into account the user's own preference, the item's general acceptance, and the influence of friends using probability distribution and expectation of the distribution (SNRN). The results show that SNRN surpasses traditional collaborative filtering method, especially with data sparsity and cold-start users.

Gu et al.[13] proposed a unified model for collaborative filtering using graph regularized and weighted nonnegative matrix factorization. They built user graph regularizations and item graph regularizations by utilizing internal and external information, such as the similarity between the users and items, users' demographics, social trust networks, and the items' genre. After that, they added the user and item graphs to weighted nonnegative matrix factorization to learn from the training dataset.

In "Trust prediction via aggregating heterogeneous social networks" [22] and "Social trust prediction using heterogeneous networks," [23] Huang et al. developed the joint manifold factorization (JMF) method to predict the trust and distrust values in the social network using the trust network and rating matrix, which is considered as auxiliary information. The two matrices, rating and trust, are different in the domain and scale - heterogeneous. The authors assumed that users tend to trust other users who have similar rating patterns, so the rating matrix and trust matrix may have similar row structures because the rows are represented users in both matrices. The results show that JMF surpasses classical trust prediction methods.

Moreover, the imputation process has been incorporated into collaborative filtering methods to alleviate rating matrix sparsity. Su et al. [67] proposed a new method - IBCF - in which a subset of missing data is imputed after dividing the rating matrix into subset matrices based on the number of ratings each item received. Two imputation techniques have been used: predictive mean matching (PMM) [32] and machine learning classifier algorithms [76], which include the decision tree (C4.5), decision table (dTable), Lazy Bayesian Rules (LBR), logistic regression (LR), naive Bayes (NB), neural networks (NN), one rule (OneR), decision list (PART), and support vector machine (SVM). To select which missing data should be imputed, an ensemble classifier has been used so that the missing data was imputed if and only if there were at least six votes from classifiers. Otherwise, the missing data is left as missing. In the end, the traditional Pearson correlation-based CF algorithm is used with each subset matrix to predict the ratings. The results show that using imputation in IBCF outperforms the content-boosted CF and traditional Pearson correlation-based CF, especially IBCF with naive Bayes. In addition, the IBCF approach has been improved to IBCF-NBM by using a different imputation approach based on the sparsity of the subset matrix [66], whereas naive Bayes is used for a relatively dense matrix and the mean imputation method for an extremely sparse matrix.

Also, imputed neighborhood-based collaborative filtering (INCF) has been proposed for the nearest and densest neighborhood approaches called INN-CF and IDN-CF, respectively [65]. The imputation techniques that are used with these methods include the baseline mean imputation (MEI) and an extension of the Bayesian multiple imputation (eBMI) [55, 65]. After that, the most similar users (nearest or densest) are found in the original rating matrix for each active user, then the traditional Pearson CF algorithm is applied to the imputed ratings of the most similar (or the densest) users in order to predict the ratings. The results of both IDN-CF and INN-CF with eBMI imputation method significantly outperform the commonly-used neighborhood-based CF. However, the baseline mean imputation method (MEI) did not improve the prediction performance when it was applied to IDN-CF and INN-CF, which demonstrated that the selection of the imputation method is important.

In addition, Ren et al.[49] proposed the Auto-Adaptive Imputation (AutAI) method for neighborhood-based collaborative filtering. The AutAI method can identify which missing data should be imputed automatically, which is called the *key set* of missing data. There are two methods of AutAI: user-based AutAI and item-based AutAI. AutAI achieves significant improvement with both similarity metrics, PCC and COS, compared to user-based PCC and user-based COS algorithms. Ren et al.[50] also proposed an improvement of AutAI method called Adaptive-Maximum imputation method (AdaM), which identifies an area to impute that will can maximize the imputation advantage and minimize the imputation error.

Furthermore, the imputation has been used with model-based collaborative filtering. Ranjbar et. al. [48] proposed a novel algorithm called IMULT that is based on the classic Multiplicative Update Rules (MULT). IMULT utilizes imputation to fill out the subset of unknown ratings. Several imputation methods are used, such as item-wise, user-wise, mean-wise and hybrid-wise. The IMULT method outperforms several MF approaches, specifically for rating matrices that are highly sparse. More details about AdaM and IMULT methods are introduced in Chapter 6.

The Hwang et. al. method [24] is the only method that we found using the trust network to impute missing ratings. Their method is based on the probabilistic matrix factorization (PMF) model. In it, two sets need to be defined for the imputation. The first one is the reliable neighbors set of an active user, which contains his/her trustees and trustors in the trust network. The second one is the candidate item set, which is the items that have been rated by a sufficient number of reliable neighbors. The sufficient number is set manually by a parameter. The imputation process is applied to candidate items only so that the imputation value is the aggregating of the corresponding ratings given by his/her reliable neighbors. Their method provided better recommendation accuracy than the original PMF model, especially for the cold-start users who rated fewer than five items.

1.2 Dissertation Organization

- Chapter 2 proposes a method to handle New-Items issues by incorporating the item auxiliary information into the NMF-based method through the imputation method without hurting the prediction accuracy of other items.
- Chapter 3 proposes an approach that handles the rating matrix sparsity specifically for the New-Users problem by utilizing the trust network information.
- Chapter 4 proposes a method to increase the accuracy results of Cold-Start-Users through imputation by utilizing the trust network information. In addition, the negative impact of the imputation is limited within the proposed method.
- Chapter 5 designs a selective imputation method that fuses the factored original rating matrix and the factored imputed rating matrix into one system.
- Chapter 6 compares imputation-based methods in terms of accuracy, and it analyzes the strength and weakness points for each method.
- Chapter 7 discusses the conclusions and suggestions for future research.

1.2.1 Technical Contributions

This dissertation aims to enhance the recommendation accuracy by incorporating auxiliary information, i.e., item auxiliary information and trust information, into the NMF-based methods through the imputation method. The prediction accuracy is analyzed for each user and item group to study the behavior with the proposed methods. The main focus of the proposed methods is New-Users, New-Items, and Cold-Start-Users.

In summary, we identify some factors that negatively impact the accuracy of NMF-based recommendation systems, thus, we propose imputation-NMF-based methods that are capable of tackling these negative factors.

1.2.2 Notational Conventions

In collaborative filtering, there are m users where $U = \{u_1, ..., u_m\}$ and n items where $E = \{e_1, ..., e_n\}$. Each user u_i can rate a set of items. Users represent the rating through an explicit numeric rating, such as a scale from one to five. In addition, the rating information is summarized in an $m \times n$ matrix, which is called a rating matrix $R \in \mathbb{R}^{m \times n}, 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n$. The rows in the rating matrix represent the users, and the columns represent items. If a particular user, u_i, rates a particular item, e_j, then the value of the intersection of the user's row and item's column in the rating matrix R_{ij} holds the rating value. If the rating is missing, that means the user did not rate that item.

The social information is summarized in an $m \times m$ matrix, which is called a trust matrix $T \in \mathbb{R}^{m \times m}, 1 \leqslant p \leqslant m, 1 \leqslant q \leqslant m$. The rows correspond to the users who create a trust relationship (trustor), and the columns correspond to the users who have been trusted by others (trustee). If user u_p trusts user u_q, the value of T_{pq} is equal to 1. On the other hand, a zero in the trust matrix means there is no trust relationship between the users.

The users information and items information are summarized in users feature matrix $F_U \in \mathbb{R}^{n \times K_U}$ and items feature matrix $F_I \in \mathbb{R}^{n \times K_I}$, respectively. Each user and item belongs to one or more features, k_U and k_I respectively.

1.2.3 Data Description

In this dissertation, specific data are required to evaluate the proposed approaches. The first is the rating matrix that represents users' ratings for items. The rating values in the experimental datasets are discrete values. The second needed data is the trust matrix, which describes the trust relationship between users. Several websites allow users to create a trust relationship between them, such as Epinions, FilmTrust, Ciao, and Douban. The last data is items' information, which will be utilized in Chapter 2.

Table 1.1: Statistics of the datasets.

Dataset	# Users	# Items	# Ratings	# Trust Relationships
Ciao	7,375	21,978	184,024	111,781
CiaoDVD	17,615	16,121	72,345	22,484
Epinions	22,166	15,000	180,889	355,727
FilmTrust	1,642	2,071	35,494	1,853

In the dissertation experiments, we adopt four datasets, Ciao[71], CiaoDVD[14], Epinions [71, 72], and FilmTrust[15], as experimental datasets. Table 1.1 shows the statistics information of the datasets.

Ciao and CiaoDVD

Ciao is a European website that displays items from different online shopping websites, such as Amazon, and compares the prices for the same item at different shopping websites. Users are allowed to rate the items using 5-scale integer ratings (from 1 to 5). To rate an item, a user must write a textual review with at least 120 words and provide the advantages and disadvantages of the item. In addition, users can trust each other so that when a user (trustor) agrees with another user's reviews (trustee), the trustor can add the trustee to his/her own trust list.

There are several datasets that have been extracted from the Ciao website. The first dataset from the Ciao website is the Ciao dataset. Tang et al. [71] crawled from Ciao.co.uk in May 2011. There are 7,375 users and 106,797 items. Each item belongs to one or more of the 28 different catalogs (DVDs, Books, Beauty, Music, Travel, Food & Drink, Sports & Outdoors, Entertainment, Health, Ciao Café, Shopping, Internet, Software, House & Garden, Education & Careers, Cars, Household Appliances, Telecommunications, Electronics, Musical Instruments & Equipment, Computers, Cameras, Family, Games, Fashion, Adult Products, Office Equipment, and Finance). Due to the MATLAB memory limitation, we only chose users who rated at least one item and items that received at least three ratings ending up with 7,375 users, 21,978 items, 184,024 ratings, and 111,781 trust relationships.

The second dataset is CiaoDVD. Guo et al. [14] crawled the CiaoDVD's dataset from ciao.co.uk, the DVD category in December 2013. The CiaoDVD dataset has 17,615 users, 16,121 items, 72,345 ratings, and 22,484 trust relationships as we see in Table 1.1. Each DVD item belongs to one of 17 genres (Action & Adventure, Comedy, Family, Drama, Horror, Science Fiction & Fantasy, Thriller & Mystery, Martial Arts, Musicals & Music Films, War, Westerns, Documentaries & Biographies, Special Interest, Sports, World Cinema, TV Series, and Anime).

Epinions

Epinions.com is a popular general consumer review website established in 1999. Users can rate an item using 5-scale integer ratings (from 1 to 5). However, users must write a review of at least 20 words for each rating [41]. Epinions is one of the most popular websites that allows users to build their web of trust. The web of trust is a list of trusted users and distrusted users. A user can be a trustor or trustee. The user can set the trust list to be public or private, but the distrust list is always private. This trust information is used to rank the reviews of products [9]. Due to the website's popularity in the research area, there are several datasets that have been extracted from Epinions.com. However, we select the Epinions dataset that was collected by Tang et al. in May 2011 [71, 72]. There are 22,166 users and 296,277 items. Each item belongs to one or more of 27 categories. The categories are: Online Stores & Services, Games, Movies, Books, Music, Personal Finance, Electronics, Home and Garden, Computer Hardware, Hotels & Travel, Restaurants & Gourmet, Magazines & Newspapers, Software, Media, Cars & Motorsports, Education, Sports & Outdoors, Wellness & Beauty, Kids & Family, Musical Instruments, Business & Technology, Pets, Computers & Internet, Web Sites & Internet Services, Gifts, Preview Categories, Photo & Optics. Due to the MATLAB memory limitation, we chose 15,000 out of 296,277 items, which are the first 5,000 items, the middle 5,000 items, and the last 5,000 items. We totaled 22,166 users, 15,000 items, 180,889 ratings and 355,727 trust relationships, as shown in Table 1.1.

FilmTrust

It was crawled from the entire FilmTrust website in June 2011 [15]. FilmTrust is a website that provides predictive recommendations about movies. However, FilmTrust does not recommend a list of movies to the users. Instead, FilmTrust suggests how much the user may like a chosen movie [11]. The FilmTrust dataset has 1,642 users, 2,071 items, 35,494 ratings, and 1,853 trust relationships as we see in Table 1.1. The rating is on a scale of a half star from half star to four stars. In addition, there is no information about items in this dataset.

1.2.4 Evaluation Strategy

In this dissertation, we evaluate the proposed approaches by measuring the accuracy of the predicted ratings, which is a common measure used to evaluate the performance of recommendation system methods [54]. The ratings in the rating matrix R are divided into a training set and test set. The training set is fed to recommendation system methods in order to predict the ratings of the test set. To measure the accuracy, the real ratings in the test set are compared with the predicted ratings. It is important to mention that in this dissertation, R refers to the rating matrix that holds only the ratings of the training set. On the other hand, the ratings of the test set are held in the R_{test} matrix.

The Mean Absolute Error (MAE) is used to evaluate the proposed approach; the MAE is the most often used measure for rating-based systems [1]. The MAE is defined as:

$$MAE = \frac{1}{|TestSet|} \sum_{r_{ij} \in TestSet} |r_{ij} - p_{ij}| \tag{1.1}$$

where r_{ij} is the actual value while p_{ij} is the predicted value.

The ratings are divided into two sets in which 80% of the ratings are used as a training set and 20% as a test set. The imputation process is applied after the data is split into training and test sets, and imputed ratings are calculated based only on the training ratings. We performed our experiments in a 5-fold cross-validation approach.

For evaluation purposes, the users are divided into three groups; the items are divided into three groups based on the total number of the ratings in training set. From the user perspective, the first group is New-Users who did not rate any items at all. The second group is Cold-Start-Users who rated at least one item and at most, four items. The last group is Heavy-Rater-Users who rated more than four items. On the other hand, from the item perspective, the first group is New-Items that did not receive any ratings at all. The second group is Cold-Start-Items that received at least one rating and at most four ratings. The last group is Heavy-Rated-Items that received more than four ratings. Only the original ratings by the users are considered. In the accuracy evaluation, the ratings in the test set are grouped based on the user or item group that the rating belongs to, then the MAE of each rating group is calculated. The names of the rating groups are corresponded to the user or item groups that the ratings belong to.

The machine we used is equipped with a 2.53Ghz quad-core +HT processor, 8GB RAM and is installed with UNIX operating system. The code was written and run in MATLAB. However, another machine has been used in Chapter 6.

2 Imputation with Item Auxiliary Information

The cold-start items, especially the New-Items, have negative impacts on NMF-based approaches, particularly the ones that utilize other information besides the rating matrix. We propose a different strategy that handles one of the most significant issues in the recommendation systems, the New-Items, by incorporating the item auxiliary information into Aux-NMF [75] by utilizing an imputation method without hurting other the prediction accuracy of other items. The proposed method imputes a limited number of ratings for each item in the New-Items group before NMF is applied to control the errors that may be introduced from the imputation. We study two factors that may affect the imputation: (1) the total number of the imputed ratings for each New-Item, and (2) the value and the average of the imputed ratings. Experiments on three different datasets were conducted to examine the proposed approach. The results show that our approach can handle the New-Items' negative impact and reduce the recommendation errors for the whole dataset.

2.1 Problem Description

Aux-NMF [75] is one of the studies that incorporates the users and items information into the NMF-based method. In Aux-NMF, the rating matrix $R_{m\times n}$ is factored into three matrices, $U_{m\times k}$, $V_{n\times l}$, and $S_{k\times l}$. The U matrix contains the latent factors for users, and the V contains the latent factors for items. In addition, the S matrix absorbs the different scales between U and V. More details about the matrix factorization will be introduced in Chapter 3. The objective function of Aux-NMF is defined as follows,

$$\begin{aligned} &min_{U\geq 0, S\geq 0, V\geq 0} f(R, W, U, S, V, C_U, C_I) = \\ &\alpha \cdot \|W \circ (R - USV^T)\|_F^2 + \beta \cdot \|U - C_U\|_F^2 + \gamma \cdot \|V - C_I\|_F^2 \end{aligned} \tag{2.1}$$

where α, β and γ are coefficients that control the weight of each part. C_U and C_I are the user cluster matrix and the item cluster matrix, which are obtained by running the K-Means clustering algorithm on the users feature matrix F_U and items feature matrix F_I.

Aux-NMF [75] can alleviate New-Users and New-Items impacts by adding the users and items cluster constraints so that in each iteration of updating the matrices U, S, and V, the β value is added to the U matrix and γ to the V matrix. When β is set to zero, the recommendation system cannot recommend items to New-Users. Similarly, when γ is set to zero, New-Items cannot be recommended to users. This is because the values in the row that represents this item in matrix V are zeros. Because our datasets have only item information, we study the impact of the items auxiliary information constraint, γ, in Aux-NMF on the rating prediction results.

In Table 2.1, we can see the α and γ values that result in the lowest MAE for each dataset. We set $\beta = 0$ because there is no users auxiliary information in our datasets. We observe that the CiaoDVD dataset mostly relies on items auxiliary information constraint, more than the rating matrix. Contrastingly, Ciao and Epinions datasets mostly rely on the rating matrix. Even though adding the items auxiliary information constraint can alleviate the New-Items issue, other items' MAE may become higher. Table 2.1 shows the lowest MAE for the whole dataset and for each item group: New-Items, Cold-Start-Items, and Heavy-Rated-Items.

We observe that each group of items has different α and γ values that result in the lowest MAE. With New-Items group, all the datasets prefer to set γ to the maximum value, 0.9, and α to the minimum, 0.1. This is because adding γ to the rows of New-Items in the V matrix allows the system to recommend New-Items to users. The best MAE of Cold-Start-Items is when $\alpha = 1$ and $\gamma = 0$ with all dataset. However, the best Heavy-Rated-Items MAE results with different α and γ settings for each dataset.

Table 2.2: New-Items ratings % in the test set.

Dataset	New-Items ratings%
Ciao	0.57%
CiaoDVD	13.22%
Epinions	5.34%

In addition, we observe that the percentage of the New-Items ratings in the test set affects the best settings of α and γ for the whole dataset. For example, CiaoDVD suffers from the highest New-Items ratings percentage in the test set

Table 2.1: MAE results of the whole dataset and each item group with all selected combinations of α and γ without imputing New-Item.

α	γ	All-Items	New-Items	Cold-Start-Items	Heavy-Rated-Items
Ciao					
0.1	0.9	0.8158	**3.0171**	0.9207	**0.7486**
0.2	0.8	0.8083	3.1542	0.8942	0.7489
0.3	0.7	0.8029	3.1828	0.8752	0.7495
0.4	0.6	0.7986	3.1849	0.8603	0.7501
0.5	0.5	0.7952	3.1849	0.8478	0.7508
0.6	0.4	0.7924	3.1849	0.8370	0.7518
0.7	0.3	0.7901	3.1849	0.8273	0.7529
0.8	0.2	0.7882	3.1849	0.8183	0.7544
0.9	0.1	**0.7867**	3.1849	0.8095	0.7562
1	0	0.7911	4.1654	**0.8007**	0.7586
CiaoDVD					
0.1	0.9	**2.0532**	**2.6477**	1.8222	2.0106
0.2	0.8	2.0698	2.8351	1.7997	2.0056
0.3	0.7	2.0750	2.9164	1.7832	2.0026
0.4	0.6	2.0762	2.9588	1.7695	2.0006
0.5	0.5	2.0760	2.9834	1.7576	1.9993
0.6	0.4	2.0750	2.9985	1.7467	1.9985
0.7	0.3	2.0738	3.0073	1.7364	**1.9982**
0.8	0.2	2.0726	3.0123	1.7271	1.9986
0.9	0.1	2.0720	3.0148	1.7189	2.0000
1	0	2.1810	3.8322	**1.7142**	2.0030
Epinions					
0.1	0.9	1.3005	**2.6663**	1.8002	1.1912
0.2	0.8	1.2991	2.8476	1.6772	1.1857
0.3	0.7	1.2957	2.9053	1.5988	1.1829
0.4	0.6	1.2927	2.9291	1.5426	1.1812
0.5	0.5	1.2900	2.9379	1.4986	1.1801
0.6	0.4	1.2876	2.9400	1.4628	1.1793
0.7	0.3	1.2857	2.9404	1.4323	1.1789
0.8	0.2	1.2841	2.9405	1.4056	**1.1786**
0.9	0.1	**1.2831**	2.9405	1.3831	1.1788
1	0	1.3349	3.9059	**1.3679**	1.1799

as shown in Table 2.2, and the lowest MAE for the whole dataset when the γ is set to the maximum value, as we see in Table 2.1. However, Cold-Start-Items and Heavy-Rated-Items get the lowest MAE with different α and γ values. If we set $\alpha = 0.1$ and $\gamma = 0.9$ for the whole CiaoDVD dataset, the Cold-Start-Items and Heavy-Rated-Items MAE are getting worse even if the whole dataset MAE is improved. On the other hand, the best Ciao and Epinions MAE are obtained when $\alpha = 0.9$ and $\gamma = 0.1$, which is almost similar to the best Cold-Start-Items and Heavy-Rated-Items parameters setting. However, the New-Items MAE, in this case, is much worse than the MAE of the best parameters setting of the New-Items group.

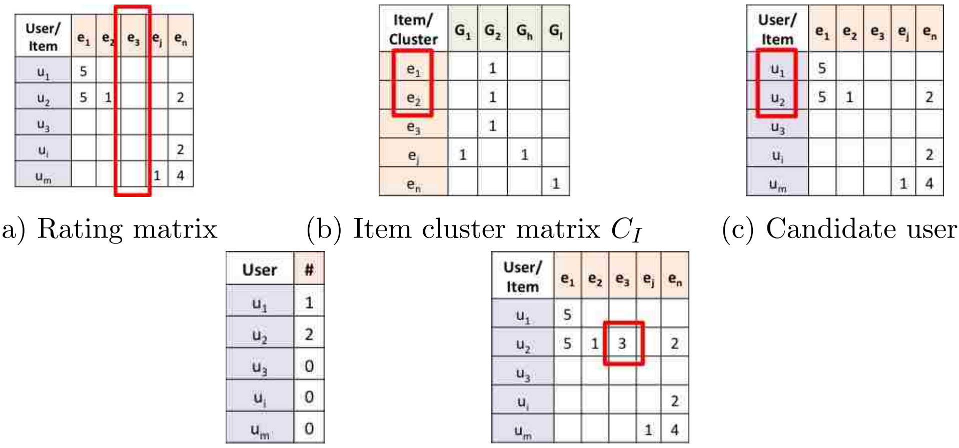
In this chapter, we propose a method to impute a subset of New-Items' ratings in the training set using the items auxiliary information to alleviate the impact of New-Items on items auxiliary information constraint and handle New-Items issue.

2.2 Proposed Method

We propose a different strategy that handles the New-Items issue by incorporating the item auxiliary information with Aux-NMF without hurting other items' prediction performance. In addition, the proposed method alleviates the impact of the New-Items on the best setting of the items auxiliary information constraint - γ -. Because imputed ratings introduce error to the system, our proposed method imputes limited ratings for each New-Items in which each dataset has a parameter of the maximum number of imputed ratings for each New-Item.

To perform the proposed imputation, we need to determine the subset of the real ratings that are used to calculate the imputed ratings, which are called source ratings, and the users who hold the imputed ratings. For each user, we count the total number of the ratings that the user did to all items that belong to the same New-Item cluster based on the item cluster matrix F_I. After ordering the users based on the total number of the ratings in descending order, the top-N users are selected to hold the imputed ratings. For each top-N user, only the user's real ratings are utilized to calculate the imputed ratings. Thereby, we ensure that the user rating pattern is maintained without involving other users' ratings that may have different rating patterns. On the other hand, the source ratings of the imputed rating for each top-N user are the ratings that the user did to all items that belong to the same New-Item cluster based on the item cluster matrix F_I.

Figure 2.1 is a simple example that illustrates the basic idea of the imputation. Figure 2.1 (a) is the rating matrix that presents the users, items, and the users' ratings to the items. As we see, item e_3 is a New-Item because there is no rating



User/ Item	e_1	e_2	e_3	e_j	e_n
u_1	5				
u_2	5	1			2
u_3					
u_i					2
u_m				1	4

(a) Rating matrix

Item/ Cluster	G_1	G_2	G_h	G_l
e_1		1		
e_2		1		
e_3		1		
e_j	1		1	
e_n				1

(b) Item cluster matrix C_I

User/ Item	e_1	e_2	e_3	e_j	e_n
u_1	5				
u_2	5	1			2
u_3					
u_i					2
u_m				1	4

(c) Candidate users

User	#
u_1	1
u_2	2
u_3	0
u_i	0
u_m	0

(d) Total ratings

User/ Item	e_1	e_2	e_3	e_j	e_n
u_1	5				
u_2	5	1	3		2
u_3					
u_i					2
u_m				1	4

(e) Imputed rating matrix

Figure 2.1: A simple example of the imputation process.

for it. To impute e_3, we need to find all items that belong to the same cluster as e_3. Figure 2.1 (b) displays the item cluster matrix C_I. Item e_3 belongs to clusters G_2 and items e_1 and e_2 belong to the same cluster as e_3. The candidate users that may hold the imputed rating are u_1 and u_2 because they did rate at least one of e_1 and e_2 items (Figure 2.1 (c)). User u_1 rated two items while user u_2 did one rating only that belongs to cluster G_2. If we decide to impute one rating for each New-Item, then u_2 will hold the imputed rating for e_3 because u_2 did the highest number of ratings, as we see in Figure 2.1 (d). The source ratings are the ratings that are used to calculate the imputed rating. In our example, the ratings 5 and 1 of u_2 are the source ratings. The average of the imputed source ratings is 3. The imputed rating of user u_2 to New-Item e_3 is equal to 3 as we see in Figure 2.1 (e).

In reality, introducing New-Items to the system is actually advertising items to the customers. For that, the prediction error of the users that have a high probability to like the New-Item should be less compared to the users that don't. There are two methods to calculate the imputed ratings. The first one is the average of the subset of the real ratings that are used to impute source ratings, and the second method is the most frequent rating appears in that subset.

2.2.1 Objective Function

To handle the New-Item issue, we replace the rating matrix R in Equation (2.1) with imputed rating matrix R' so that

$$r'_{ij} = \begin{cases} r_{ij} & \text{if } r_{ij} \neq 0 \\ \text{Imputed Rating} & \text{if total ratings of item } j = 0 \text{ and source ratings } \neq \varnothing \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where $r'_{ij} \in R'$, $r_{ij} \in R$, and Imputed Rating could be either the average of the source ratings or the most frequent rating value in a source ratings set. In addition, W in Equation (2.1) is redefined as a W' so that:

$$w'_{ij} = \begin{cases} 1 \text{ if } r'_{ij} \neq 0 \\ 0 \text{ if } r'_{ij} = 0 \end{cases} \quad (w'_{ij} \in W', r'_{ij} \in R') \tag{2.3}$$

We update Aux-NMF Equation (2.1) with Equations (2.2) and (2.3), and set β to zero due to the absent of users auxiliary information in our datasets, the objective function is:

$$min_{U \geq 0, S \geq 0, V \geq 0} f(R', W', U, S, V, C_I) = \alpha \cdot \|W' \circ (R' - USV^T)\|_F^2 + \gamma \cdot \|V - C_I\|_F^2 \tag{2.4}$$

We name this matrix factorization Aux-New-Items-NMF.

2.2.2 Update Formula

Let $L = f(R', W', U, S, V, C_I)$, which is the objective function of Aux-New-Items-NMF. The update formulae of L are as follows [75]

$$U_{ij} = U_{ij} \cdot \frac{[\alpha(W' \circ R')VS^T]_{ij}}{\{\alpha[W' \circ (USV^T)]VS^T\}_{ij}} \tag{2.5}$$

$$V_{ij} = V_{ij} \cdot \frac{[\alpha(W' \circ R')^T US + \gamma C_I]_{ij}}{\{\alpha[W' \circ (USV^T)]^T US + \gamma V\}_{ij}} \tag{2.6}$$

$$S_{ij} = S_{ij} \cdot \frac{[U^T(W' \circ R')V]_{ij}}{\{U^T[W' \circ (USV^T)]V\}_{ij}} \tag{2.7}$$

The derivation of the update formulas (2.5), (2.6), and (2.7) are similar to the update formulas derivation in [75].

2.2.3 Convergence Analysis

This section proves that the objective function (2.4) is nonincreasing under the update formulae (2.5), (2.6), and (2.7) by following [29].

Definition 2.2.1. *$H(u, u')$ is an auxiliary function for $F(u)$ if the conditions*

$$H(u, u') \geq F(u), H(u, u) = F(u) \tag{2.8}$$

are satisfied.

Lemma 2.2.1. *If H is an auxiliary function for F, then F is nonincreasing under the update*

$$u^{t+1} = \arg\min_{u} H(u, u^t) \tag{2.9}$$

Lemma 2.2.1 can be easily proven since we have $F(u^{t+1}) = H(u^{t+1}, u^{t+1}) \leq H(u^{t+1}, u^t) \leq H(u^t, u^t) = F(u^t)$.

The convergences of the update formulae (2.5), (2.6), and (2.7) will be proved by their equivalence to Equation (2.9), with proper auxiliary functions defined.

Let's rewrite the objective function L,

$$\begin{aligned} L = & tr[\alpha(W' \circ R')^T \cdot (W' \circ R')] + tr\{-2\alpha(W' \circ R')^T \cdot [W' \circ (USV^T)]\} \\ & + tr\{\alpha[W' \circ (USV^T)]^T \cdot [W' \circ (USV^T)]\} + tr(U^T U) + tr(\gamma V^T V) \\ & + tr(-2\gamma V^T C_I) + tr(\gamma C_I^T C_I) \end{aligned} \tag{2.10}$$

where $tr(*)$ is the trace of a matrix.

After eliminating the irrelevant terms, we can define the following functions that are related only to U, V, and S, respectively.

$$\begin{aligned} L(U) = & tr\{-2\alpha(W' \circ R')^T \cdot [W' \circ (USV^T)] + \alpha[W' \circ (USV^T)]^T \cdot [W' \circ (USV^T)] \\ & + U^T U\} \\ & = tr\{[-2\alpha(W' \circ R')VS^T]U^T + U^T[\alpha W' \circ (USV^T)VS^T] + U^T U\} \end{aligned} \tag{2.11}$$

$$\begin{aligned} L(V) = & tr\{-2\alpha(W' \circ R')^T \cdot [W' \circ (USV^T)] + \alpha[W' \circ (USV^T)]^T \cdot [W' \circ (USV^T)] \\ & + \gamma V^T V - 2\gamma V^T C_I\} \\ & = tr\{[-2\alpha(W' \circ R')^T US + \gamma C_I]V^T + V^T[\alpha(W' \circ (USV^T))^T US] + V^T(\gamma V)\} \end{aligned} \tag{2.12}$$

$$\begin{aligned} L(S) =& tr\{-2\alpha(W'\circ R')^T\cdot[W'\circ(USV^T)] + \alpha[W'\circ(USV^T)]^T\cdot[W'\circ(USV^T)]\} \\ &= tr\{[-2\alpha U^T(W'\circ R')V]S^T + [\alpha U^T(W'\circ(USV^T))V]S^T\} \end{aligned} \tag{2.13}$$

Lemma 2.2.2. *For any matrices* $X \in \mathbb{R}_+^{n\times n}, Y \in \mathbb{R}_+^{k\times k}, F \in \mathbb{R}_+^{n\times k}, F' \in \mathbb{R}_+^{n\times k}$, *and* X, Y *are symmetric, the following inequality holds*

$$\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{(XF'Y)_{ij}F_{ij}^2}{F'_{ij}} \geq tr(F^TXFY) \tag{2.14}$$

The Lemma 2.2.2 is proved in [5] and is used to build an auxiliary function for $L(U)$. The convergences of $L(V)$ and $L(S)$ are similar to $L(U)$.

Lemma 2.2.3.

$$\begin{aligned} H(U,U') =& -2\sum_{ij}\{[\alpha(W'\circ R')VS^T]U^T\}_{ij} \\ &+\sum_{ij}\frac{(\alpha W'\circ(U'SV^T)VS^T+U')_{ij}U_{ij}^2}{U'_{ij}} \end{aligned} \tag{2.15}$$

is an auxiliary function of $L(U)$ *and the global minimum of* $H(U,U')$ *can be achieved by*

$$U_{ij} = U'_{ij}\cdot\frac{[\alpha(W'\circ R')VS^T]_{ij}}{\{\alpha[W'\circ(U'SV^T)]VS^T+U'\}_{ij}} \tag{2.16}$$

The Lemma 2.2.3 is proved in [75]. The Lemma 2.2.3 can be used for (2.6) and (2.7), too.

2.2.4 Detailed Algorithm

In this section, the Aux-New-Items-NMF algorithm is presented. Algorithm 2.1 depicts the steps of performing Aux-New-Items-NMF on the imputed rating matrix R'. We perform this algorithm in two cases. The first case is when the imputed ratings are equal to the average of the source ratings, which is called the Average-Imputation case. The second case is when the imputed ratings are equal to the most frequent rating value in the source ratings, which is called Most-Imputation case. Figure 2.2 shows the flowchart of the New-Items imputation steps. However, the Aux-New-Items-NMF algorithm may take hundreds or thousands of iterations to converge to a local minimum. Thus, in the algorithm, we set an additional stop criterion - the maximum iteration count. In collaborative filtering, this value varies from 10 $\sim$ 100 and can produce good results [75].

Algorithm 2.1 Aux-New-Items-NMF

```
Require:
  User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
  Item feature matrix: $F_I \in \mathbb{R}^{n \times k_I}$;
  Column dimension of $U : k$;
  Column dimension of $V : l$;
  Coefficients in objective function: $\alpha$ and $\gamma$;
  Number of maximum iterations: $MaxIter$;
  Number of maximum imputed ratings for each New-Item: $MaxImputedRatings$;
Ensure:
  Item cluster membership indicator matrix: $C_I \in \mathbb{R}^{n \times l}$;
  Imputed rating matrix: $R' \in \mathbb{R}^{m \times n}$;
  Factor matrices: $U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times l}$, and $S \in \mathbb{R}^{k \times l}$;

function NEW-ITEMS IMPUTATION($R$, $C_{I_{Row}}$ , $j$ , Imputation Case )
    for each group $g_I$ in $C_{I_{Row}}$ do
        if $g_I == 1$ then
            $g_I Items = g_I Items$ + all items belong to $g_I$
        end if
    end for
    for each user $u_i$ do
        $candidateImputedUsers$ = count the total ratings of $u_i$ for all items in $g_I Items$
    end for
    $OrderedUsers$ = sort $candidateImputedUsers$ based on the total ratings in descending order
    for $u_{imputed} = 1 : MaxImputedRatings$ in $OrderUsers$ do
        if Imputation Case = Average then
            $r'_{u_{imputed}j}$= the average ratings of $u_{imputed}$ for all items in $g_i Items$
        else if Imputation Case = Most then
            $r'_{u_{imputed}j}$= the most frequent ratings value of $u_{imputed}$ for all items in $g_i Items$
        end if
    end for
    return $r'_{:j}$
end function

Cluster items into $l$ groups based on $F_I$ by K-Means algorithm → $C_I$;
Initialize $U, S$, and $V$ with random values;
for each item $e_j$ do
    if $e_j$ total ratings == 0 then
        $r'_{:j}$ = New-Items Imputation($R$, $C_{I_{e_j:}}$, $j$ , Imputation Case)
    else
        $r'_{:j} = r_{:j}$
    end if
end for
Build weight matrix $W'$ by Eq. (2.3);
```

```
Set $iteration = 1$ and $stop = false$;
**while** $(iteration < MaxIter)$ *and* $(stop == false)$ **do**
    $U_{ij} \leftarrow U_{ij} \cdot \frac{[\alpha(W'\circ R')VS^T]_{ij}}{\{\alpha[W'\circ(USV^T)]VS^T+U\}_{ij}}$
    $V_{ij} \leftarrow V_{ij} \cdot \frac{[\alpha(W'\circ R')^TUS+\gamma C_I]_{ij}}{\{\alpha[W'\circ(USV^T)]^TUS+\gamma V\}_{ij}}$
    $S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T(W'\circ R')V]_{ij}}{\{U^T[W'\circ(USV^T)]V\}_{ij}}$
    $L \leftarrow \alpha \cdot \|W' \circ (R' - USV^T)\|_F^2 + \gamma \cdot \|V - C_I\|_F^2$
    **if** $L$ increases in this iteration **then**
        $stop = true$;
        Restore $U, V$, and $S$ to their values in last iteration.
    **end if**
**end while**
**Return** $R', U, V, S,$, and $C_I$.
```

2.2.5 Complexity

The computational complexity of Aux-New-Items-NMF can be broken down into two phases: imputation and NMF phase (updating U, V, and S).

There are four basic steps to perform the imputation in Aux-New-Items-NMF which need to be considered in the computational complexity. Firstly, the time complexity of searching for New-Items is $O(mn)$. However, the time complexity for finding the items of the source ratings for each New-Item is $O(l+n)$. The time complexity to find the candidate users is $O(mn)$, and finally the time complexity of ordering the candidate users for each New-Item is $O(m^2)$. By combining the time complexity of all imputation steps, the total of time complexities is as follows,

$$TimeComplexity = O(mn) + n(l+n) + O(mn) + O(nm^2) \tag{2.17}$$

where m is the total number of the users in the rating matrix R, n is the total number of the items in the rating matrix R, and l is the total number of the item clusters. In addition, the time complexity of the imputation phase is considered in the worst case.

In general, the time complexity of the imputation in the Aux-New-Items-NMF method is quadratic. However, in reality, the time complexity is much less than the worst case. For instance, in the second term of Equation 2.17, we assume the total number of New-Items $NewItemsNum \ll n$. In addition, the items of the source rating items $sourceRatingItemsNum \ll n$ in the third term of Equation

2.17. Finally, we assume that the candidate users who rated at least one item of source ratings $candidateUsersNum \ll m$ in which the users who did not rate at least one item of source ratings could be eliminated before the sorting algorithm is applied. In addition, the sorting algorithm is applied for each New-Items such that $NewItemsNum \ll n$. For large scale datasets, the imputation process could be ran in parallel to reduce the computation time.

On the other hand, we suppose $k, l \ll \min(m, n)$, the time complexities of updating U, V, and S in each iteration are all $O(mn(k + l))$ [75].

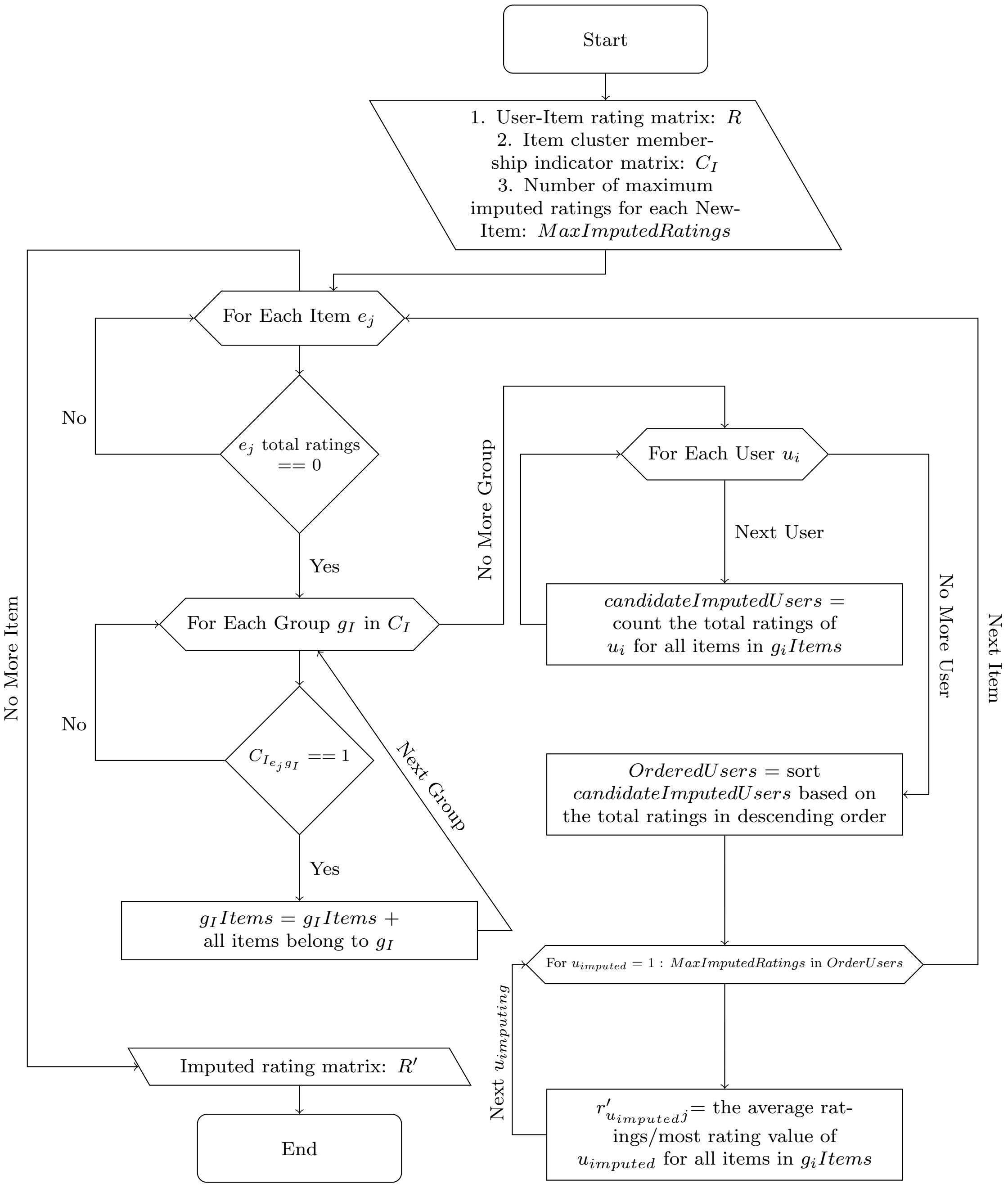

Start
1. User-Item rating matrix: R
2. Item cluster membership indicator matrix: C_I
3. Number of maximum imputed ratings for each New-Item: $MaxImputedRatings$
For Each Item e_j
No
e_j total ratings $== 0$
Yes
No More Item
For Each Group g_I in C_I
No
$C_{I_{e_j g_I}} == 1$
Yes
$g_I Items = g_I Items +$ all items belong to g_I
Next Group
No More Group
For Each User u_i
Next User
$candidateImputedUsers =$ count the total ratings of u_i for all items in $g_i Items$
No More User
Next Item
$OrderedUsers =$ sort $candidateImputedUsers$ based on the total ratings in descending order
For $u_{imputed} = 1 : MaxImputedRatings$ in $OrderUsers$
Next $u_{imputing}$
$r'_{u_{imputed}j} =$ the average ratings/most rating value of $u_{imputed}$ for all items in $g_i Items$
Imputed rating matrix: R'
End

Figure 2.2: New-Items imputation flowchart.

2.3 Experimental Study

In the experiment of this chapter, the FilmTrust dataset is excluded because of the absence of items' information. For the three other datasets, the parameters of the Aux-New-Items-NMF algorithm need to be determined in advance. They have been set based on the experiments. Table 2.3 gives the parameter setup in Aux-New-Items-NMF (see Algorithm 2.1).

Table 2.3: Parameter setup in Aux-New-Items-NMF.

Dataset	k	l	$MaxIter$	$MaxImputedRatings$
Ciao	10	20	10	15
CiaoDVD	2	15	10	3
Epinions	10	20	10	5

When the results before the imputation in Table 2.1 are compared to the results after the imputation in Table 2.4, it is notable that the imputation process improves the prediction results. Furthermore, the best α and γ settings are different in all the datasets. After the imputation, Ciao and Epinions datasets rely totally on the rating matrix with $\alpha = 1$ and $\gamma = 0$. In addition, CiaoDVD dataset relies almost on the rating matrix with $\alpha = 0.9$ and $\gamma = 0.1$.

Before the New-Items imputation, the best setting of the New-Items group is when α equals the minimum value, 0.1, and γ equals the maximum value, 0.9, with all datasets, as we see in Table 2.1. After imputing New-Items with the average of the source ratings, the New-Items prediction improves remarkably for all selected α and γ combinations in all datasets, as we see in Table 2.4. In addition, the best setting of CiaoDVD and Epinions New-Items group is $\alpha = 1$ and $\gamma = 0$. However, the Ciao dataset has the same α and γ best setting of New-Items group before and after the imputation. The best setting of α and γ for other items groups, Cold-Start-Items and Heavy-Rated-Items, remains the same for all datasets and the MAE is almost the same.

We observe that the best α and γ setting of New-Items group is the same as the item group that $MaxImputedRatings$ value is within its limits. For example, each New-Item in CiaoDVD and Epinions datasets is imputed with 3 and 5 imputed ratings, respectively, and the best α and γ setting of New-Items of both datasets

Table 2.4: MAE results of the whole dataset and each item group with all selected combinations of α and γ of Aux-New-Items-NMF.

α	γ	All-Items	New-Items	Cold-Start-Items	Heavy-Rated-Items
Ciao					
0.1	0.9	0.8036	**0.8332**	0.9212	**0.7487**
0.2	0.8	0.7954	0.8339	0.8945	0.7489
0.3	0.7	0.7897	0.8340	0.8754	0.7495
0.4	0.6	0.7855	0.8343	0.8604	0.7501
0.5	0.5	0.7820	0.8346	0.8479	0.7509
0.6	0.4	0.7792	0.8351	0.8370	0.7518
0.7	0.3	0.7769	0.8357	0.8273	0.7529
0.8	0.2	0.7750	0.8367	0.8182	0.7544
0.9	0.1	0.7735	0.8381	0.8095	0.7562
1	0	**0.7723**	0.8401	**0.8006**	0.7586
CiaoDVD					
0.1	0.9	1.9011	1.5036	1.8153	2.0118
0.2	0.8	1.8918	1.4921	1.7951	2.0066
0.3	0.7	1.8853	1.4839	1.7799	2.0034
0.4	0.6	1.8801	1.4771	1.7671	2.0012
0.5	0.5	1.8758	1.4708	1.7558	1.9998
0.6	0.4	1.8721	1.4647	1.7454	1.9988
0.7	0.3	1.8689	1.4588	1.7357	**1.9983**
0.8	0.2	1.8664	1.4532	1.7267	1.9986
0.9	0.1	**1.8649**	1.4486	1.7187	2.0001
1	0	1.8660	**1.4474**	**1.7140**	2.0036
Epinions					
0.1	0.9	1.2205	1.1633	1.7721	1.1930
0.2	0.8	1.2077	1.1302	1.6589	1.1871
0.3	0.7	1.1997	1.1018	1.5858	1.1839
0.4	0.6	1.1938	1.0762	1.5326	1.1819
0.5	0.5	1.1892	1.0539	1.4909	1.1805
0.6	0.4	1.1857	1.0350	1.4565	1.1795
0.7	0.3	1.1827	1.0174	1.4275	1.1788
0.8	0.2	1.1802	0.9986	1.4030	**1.1786**
0.9	0.1	1.1781	0.9752	1.3822	1.1788
1	0	**1.1780**	**0.9653**	**1.3674**	1.1801

are equal to the Cold-Start-Items group's best setting. However, the best α and γ setting of New-Items in the Ciao dataset is the same as Heavy-Rated-Items because each New-Item is imputed with 15 imputing ratings, which make them Heavy-Rated-Items. This explains why the best α and γ setting of Ciao New-Items dataset did not change after the imputation.

The difference between MAE of the item groups with the best α and γ setting of the whole dataset and of each item group is moot compared to the no imputation case. Before the New-Items imputation, the difference in the Epinions dataset between the lowest MAE of New-Items and MAE of the same group with the best α and γ setting of the whole dataset is the highest, which is 0.2742. However, after the imputation, the Ciao dataset has the most difference, which is between the lowest MAE of the Heavy-Rated-Items group and the MAE of them with the best α and γ setting of the whole dataset, which is 0.0099.

In conclusion, using item auxiliary information for imputation, not the NMF process, is a better strategy.

2.3.1 The Influence of Imputed Rating Value

In this section, we demonstrate how the value of the imputed ratings and the average of all the imputed ratings affect the results. There are two cases used to calculate the imputed rating value: Average-Imputation and Most-Imputation. The predicted rating is zero when the system cannot predict the rating, which is called unpredictable rating. This happens because of the impact of New-Users. After applying Aux-New-Items-NMF, some of the New-Item rows in matrix V are zeros even though all New-Items are imputed. For each rating value of New-Items in the test set, we consider its MAE as high when it is larger than the whole dataset MAE. On the other hand, we consider the MAE as a low when it is equal to or lower than the whole dataset MAE.

By applying the Average-Imputation case to the Ciao dataset, 96.12% of the rating value 4 of New-Items in the test set get low MAE (which is the highest percentage among all other rating values), as we see in Table 2.5. This is because of the average of the imputed ratings which is 4.10 as shown in Table 2.6. With the second imputation case, the average of the imputed ratings increases to 4.46,

Table 2.5: The percentage of the New-Items rating values in the test set and the percentage of their MAEs cases (high/low) after the New-Item imputation with both cases: Average and Most.

Rating Value	Rating %	Unpredictable Rating	High MAE		Low MAE	
			Average	Most	Average	Most
Ciao						
1	3.59%	2.22%	97.78%	97.78%	0.00%	0.00%
2	4.95%	3.75%	96.25%	96.25%	0.00%	0.00%
3	12.14%	1.41%	76.57%	89.71%	22.02%	8.88%
4	31.84%	1.90%	1.97%	17.33%	96.12%	80.77%
5	48.74%	1.77%	42.83%	12.83%	55.41%	85.40%
CiaoDVD						
1	4.85%	11.22%	71.42%	84.45%	17.36%	4.34%
2	8.88%	8.33%	21.25%	44.83%	70.43%	46.84%
3	18.80%	9.69%	0.80%	6.84%	89.52%	83.48%
4	33.15%	18.40%	0.06%	0.09%	81.53%	81.50%
5	34.33%	26.76%	2.22%	1.46%	71.03%	71.78%
Epinions						
1	4.68%	5.43%	91.82%	92.98%	2.75%	1.59%
2	7.20%	2.60%	90.70%	92.87%	6.70%	4.53%
3	17.64%	2.45%	17.66%	38.74%	79.89%	58.81%
4	33.82%	2.98%	1.61%	1.15%	95.41%	95.87%
5	36.66%	4.31%	25.55%	13.27%	70.14%	82.43%

as we see in Table 2.6. The low MAE percentage of rating value 5 for New-Items in the test set increases from 55.41% to 85.40%, which is the highest percentage among all other rating values, as we see in Table 2.5. On the other hand, the low MAE percentage of the rating value 4 declines to 80.77%. Because the imputed rating average of both imputation cases is above 4, none of the rating values 1 and 2 MAE of New-Items in the test set are low even though there are fewer 1 and 2 imputed ratings in the second imputation case, as we see in Table 2.7.

The CiaoDVD dataset has the lowest average of the imputed ratings in the first and second imputation cases among other datasets, as shown in Table 2.6. For the first imputation strategy, the average of the imputed ratings is 3.63. The rating value 3 of New-Items has the highest percentage of the low MAE, then rating values 4 and 5, respectively (Table 2.5). In addition, some 1 and 2 rating values of New-Items in the test set have low MAE. With the second imputation strategy, the imputed rating average increases to 4.04, as we see in Table 2.6.

Table 2.6: The average of the imputed ratings with both New-Items imputation cases: Average and Most.

Dataset	Average	Most
Ciao	4.10	4.46
CiaoDVD	3.63	4.04
Epinions	3.89	4.30

Table 2.7: The percentage and average for each imputed rating value range with both imputation cases: Average and Most.

rating value range		Ciao		CiaoDVD		Epinions	
>	<=	%	average	%	average	%	average
New-Item Average-Imputation Case							
0	1	0.00%	N/A	0.00%	N/A	0.00%	N/A
1	2	0.04%	1.52	0.00%	N/A	0.02%	2.00
2	3	1.82%	2.67	20.06%	2.74	2.71%	2.89
3	4	39.32%	3.72	52.55%	3.55	48.74%	3.56
4	5	58.82%	4.40	27.39%	4.42	48.52%	4.29
New-Item Most-Imputation Case							
0	1	0.18%	1	0.53%	1	0.19%	1
1	2	0.32%	2	0.34%	2	1.36%	2
2	3	4.29%	3	24.50%	3	10.69%	3
3	4	44.17%	4	44.13%	4	44.14%	4
4	5	51.04%	5	30.51%	5	43.63%	5

This leads to a decrease in the low MAE percentage of rating values 1, 2, and 3 (Table 2.5). However, there is almost no improvement in the rating prediction (low MAE percentage) of 4 and 5 values of the New-Items. This is probably because of several reasons. First, the total number of the ratings in the test set in CiaoDVD is much less than other datasets, as we see in Table 1.1. The second reason is that the unpredictable ratings are much more than other datasets, especially for the high rating values 4 and 5, as we see in Table 2.5. The third one is that the sum of the New-Items high rating values (4 and 5) percentage in the test set is the lowest compared to other datasets, as we see in Table 2.5. Due to these facts, the increase in the low MAE percentage of the high rating values (4 and 5) is not notable in this case, even though there is an increase in the average of imputed ratings. Although the percentages of imputed ratings with low values (1, 2, and 3) in the second imputation case are more than in the first imputation case, as we see in Table 2.7, the percentages of the high MAE of the low rating values (1, 2, and 3) increase because the average of the imputed ratings increased too.

The imputed ratings average of Epinions dataset is in between CiaoDVD and Ciao datasets, as shown in Table 2.6. With the first imputation case, the highest percentage of the low MAE is for rating value 4, then 3 and 5, respectively, as we see in Table 2.5 where the average of the imputed ratings is 3.89. However, the average of the imputed ratings in the second imputation case is 4.30, which raises the percentage of the low MAE of rating value 5 up to 82.43% and declines the percentage of the low MAE of rating value 3 to 58.81% as we see in Table 2.5. As we observe in other datasets, there are more imputed ratings of low value (1, 2, and 3) in the second imputation case than the first one, as we see in Table 2.7. However, the low MAE percentage of the low rating values (1, 2, and 3) decreases.

Table 2.8: The MAE of both New-Items imputation cases: Average and Most when $\alpha = 1$.

Imputation Case	All-Items MAE	New-Items MAE	Cold-Start Items MAE	Heavy-Rated Items MAE
Ciao				
Average	0.7723	0.8400	**0.8006**	0.7586
Most	**0.7720**	**0.7910**	**0.8006**	**0.7585**
CiaoDVD				
Average	**1.8660**	**1.4474**	**1.7140**	**2.0036**
Most	1.8700	1.4752	1.7152	2.0038
Epinions				
Average	**1.1780**	**0.9653**	**1.3674**	**1.1800**
Most	1.1796	0.9806	1.3711	1.1807

Table 2.8 shows the MAE results of both New-Items imputation cases: Average and Most when $\alpha = 1$ and $\gamma = 0$. We set $MaxImputedRatings$ of both New-Items imputation cases as is shown in Table 2.3. The results show MAE for the whole dataset and for each item group. Only MAE of Ciao dataset is slightly lower with the New-Items Most-Imputation case than the Average-Imputation case for the whole dataset and New-Items group. This is because Ciao dataset has the highest percentage of the rating value 5 in the test set among other datasets (Table 2.5). In addition, the most improvement in the prediction in the second imputation case is with rating value 5, as we see in Table 2.5. On the other hand, the best MAE

for other datasets is New-Items Average-Imputation case for the whole dataset and New-Items group.

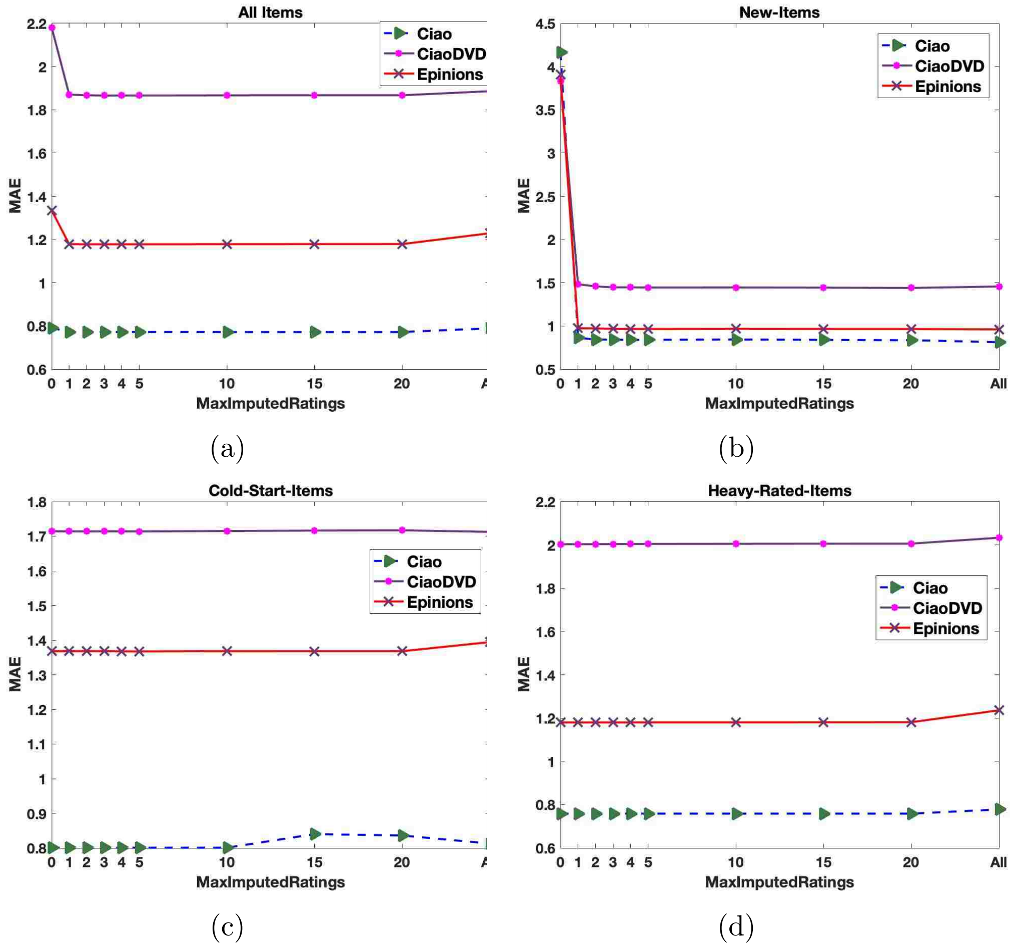
In conclusion, the prediction accuracy of the rating values that are close to the average of New-Items imputed ratings is better than other rating values. In addition, the influence of the imputed rating average is more effective than the value of the imputed ratings. Hence, the average of the imputed ratings determines which rating values will have high or low MAE compared to the whole dataset MAE. Because recommending New-Items to users considers an advertisement, we think that the users who have a high probability of liking the New-Item need to have more accurate prediction than the users who don't. Raising the average of the imputed ratings allows the system to predict the high rating values more accurately than the low rating values.

2.3.2 Parameter Study

In Aux-New-Items-NMF, the parameter $MaxImputedRatings$ needs to be set. We run the experiment with different total numbers of the imputed ratings for each New-Item. In this experiment, we set $\alpha = 1$ and $\gamma = 0$ with New-Item Average-Imputation case. In general, the MAEs of all three datasets are lower after New-Items imputation regardless of the total number of imputed ratings, $MaxImputedRatings$, as shown in Figure 2.3(a).

Adding more imputed ratings ($MaxImputedRatings$) improves the results of the New-Items group prediction results slightly. Nevertheless, adding only one imputed rating to each New-Item allows the system to recommend New-Items to users and reduces the New-Items MAE remarkably compared to none imputation case, i.e. Aux-NMF, as we see in Figure 2.3(b). When all available imputed ratings are imputed for each New-Item, CiaoDVD and Ciao MAE are worse. However, the result of Epinions dataset slightly improves but requires a long time to impute the rating matrix. This demonstrates that adding imputed ratings is not always advantageous because they introduce errors to the system at the same time, even for New-Items.

As we see in Figure 2.3(d), the results of Heavy-Rated-Items show that more imputed ratings lead to increasing the MAE of them. However, there is a dif-



(a) (b)

(c) (d)

Figure 2.3: The MAE of New-Item Average-Imputation case with different values of *MaxImputedRatings*.

ference in the increment ratio of MAE between the datasets. Ciao dataset has the lowest New-Items percentage in the training set among other datasets, as we see in Table 2.9. For that, the Heavy-Rated-Items MAE did not increase with the *MaxImputedRatings* increment but did increase when all possible imputed ratings of New-Items were imputed. On the other hand, the highest percentage of New-Items in the training set is in CiaoDVD and Epinions datasets; and, their Heavy-Rated-Items MAEs increase with almost every time the *MaxImputedRatings* is increased, as shown in Figure 2.3(d). Overall, the best of Heavy-Rated-Items MAE is without imputation process.

In general, to set the *MaxImputedRatings* parameter, we need to balance between the imputation advantage and the imputation error. Table 2.3 shows the best setting of *MaxImputedRatings* that improves the accuracy of ratings predic-

Table 2.9: The % of New-Items in the training set.

Dataset	Ciao	CiaoDVD	Epinions
%	0.27%	11.08%	12.45%

tion. There are two factors that may impact the *MaxImputedRatings* parameter setting. The first is the percentage of New-Items in the training set, and the second is the percentage of New-Items ratings in the test set. As we see in Tables 2.3 and 2.9, there is an inverse relationship between the best *MaxImputedRatings* parameter setting and the percentage of New-Items in the training set. In addition, there is an inverse relationship between the best *MaxImputedRatings* parameter setting and the percentage of New-Items ratings in the test set, as shown in Tables 2.2 and 2.3. Ciao has the lowest percentage of New-Items in the training set, the lowest percentage of New-Items ratings in the test set, and the highest *MaxImputedRatings*. On the other hand, the CiaoDVD dataset has the most percentage of New-Items ratings in the test set, the next highest percentage of New-Items in the training set, and the lowest *MaxImputedRatings*. Epinions dataset has the highest percentage of New-Items in the training set, the next highest percentage of New-Items ratings in the test set. However, the *MaxImputedRatings* is in middle between other datasets but it much closer to CiaoDVD.

In conclusion, the total number of the imputed ratings in the training set should be limited. The percentage of New-Items in the training set plays a critical factor in setting the value of *MaxImputedRatings*. If there is a high existence of New-Items in the training set, then the value of *MaxImputedRatings* should be small, especially if the system predicts plenty of ratings that belong to New-Items group and vice versa when the New-Items percentage in the training set is low.

2.4 Summary

In this chapter, we proposed a method to incorporate item auxiliary information into the Aux-NMF [75] using the imputation process. Our results show that the proposed method alleviates the impact of the New-Items on the items auxiliary information constraint - γ - in Aux-NMF [75]. In addition, Aux-New-Items-NMF allows the system to recommend New-Items to the users. Furthermore, using item auxiliary information for imputation, not for the NMF process, is a better strategy. Additionally, increasing the average of imputed ratings improves the prediction accuracy of users that have a high probability to like the New-Item. The total number of New-Items in the training set determines the total imputed ratings for each item in the New-Items group.

We propose an NMF-based approach to handle the New-Users issue by utilizing the trust network information. A subset of missing ratings in the rating matrix is imputed before NMF is applied to alleviate the sparsity issue and enhance the prediction accuracy. To survey each user group behavior with the imputation, we perform two cases of imputation: (1) when all users are imputed, and (2) when only New-Users are imputed. Experiments on four different datasets were conducted to examine the proposed approach. The results show that our approach can handle the New-Users issue and reduce the recommendation errors for the whole dataset, especially in the second imputation case.

3.1 Problem Description

The basic NMF is defined as follows [29]:

$$R_{m\times n} \approx U_{m\times k} \cdot V_{n\times k}^{T} \tag{3.1}$$

The goal is to find a pair of orthogonal nonnegative matrices U and V (such that, $U^TU = I$ and $V^TV = I$) that minimizes the Frobenius norm (Euclidean norm) $\|R - UV^T\|_F$. The objective function is:

$$f(R,U,V) = min_{U\geq 0,V\geq 0}\|R - UV^T\|_F^2 \tag{3.2}$$

In Ding et. al. [6], they proved that NMF is equivalent to K-Means clustering. Thus, by applying NMF on R, the users and the items are clustered into k groups. The two matrices U and V produced by NMF on R describe the clustering information of the objects such that each column vector of U, u_i, can be considered as a basis, and each data point r_i is approximated by a linear combination of these k bases, weighted by the components of V [77], where k is the rank of factor matrices.

In the collaborative filtering field, rating matrix R represents the relationships between users and items. We can obtain users and items clusters by performing NMF on rating matrix R. However, it is difficult to find two matrices U and V

that represent user clusters and item clusters respectively and that also have the same rank of the factor k, which is considered to be the substantial property of NMF. To solve this issue, NMTF (Nonnegative Matrix Tri-Factorization) [5] adds an extra factor matrix S to absorb the different scales of U and V. NMTF allows U and V to have a different number of the clusters, which are k and l, respectively. NMTF is defined as follows,

$$R_{m \times n} \approx U_{m \times k} \cdot S_{k \times l} \cdot V_{n \times l}^{T} \tag{3.3}$$

In NMTF, the rating matrix R is factored into three matrices, U, V, and S, where U is a matrix that contains the latent factors for users and V contains the latent factors for items. In this case, there is no requirement that U and V matrices have the same rank of factor k because S matrix absorbs the different scales between U and V.

The goal is to find a pair of orthogonal nonnegative matrices U and V that minimize the Frobenius norm (Euclidean norm) $\|R - USV^T\|_F$. The objective function is:

$$f(R, U, S, V) = min_{U \geq 0, S \geq 0, V \geq 0} \|R - USV^T\|_F^2 \tag{3.4}$$

However, one of the most significant issues with NMF (3.1) and NMTF (3.3) is that they require that the rating matrix not have missing ratings. As we mentioned before, the rating matrix in recommendation systems suffers from sparsity. Therefore, the rating matrix cannot be directly fed to NMF and NMTF. To handle that, all missing ratings should be imputed as a pre-processing step before NMF or NMTF is applied, which requires extra time to compute the missing ratings.

On the other hand, Weighted NMF (WNMF) [80] is one of the matrix factorization algorithms that can factorize a sparse rating matrix without the need to impute all missing ratings during the pre-processing step. The objective function of Weighted Nonnegative Matrix Factorization (WNMF) is as follows,

$$f(R, W, U, V) = min_{U \geq 0, V \geq 0} \|W \circ (R - UV^T)\|_F^2 \tag{3.5}$$

where $\circ$ is the element-wise multiplication. The weight matrix $W \in \mathbb{R}_{+}^{m \times n}$ indicates the value existence in the rating matrix R, such that

$$w_{ij} = \begin{cases} 1 \text{ if } r_{ij} \neq 0 \\ 0 \text{ if } r_{ij} = 0 \end{cases} \quad (w_{ij} \in W, r_{ij} \in R) \tag{3.6}$$

To get the advantages of both NMTF and WNMF, Equations (3.4) and (3.5) are combined to form Weighted Nonnegative Matrix Tri-Factorization (WNMTF). The objective function of WNMTF is as follows,

$$f(R, W, U, S, V) = min_{U \geq 0, S \geq 0, V \geq 0} \| W \circ (R - USV^T) \|_F^2 \tag{3.7}$$

Actually, WNMTF is equivalent to Aux-NMF [75] when α is set to 1 in Equation (2.1).

Generally, WNMTF cannot predict items to New-Users because the values in the row that represents this user in matrix U are zeros. Unpredictable ratings lead to high MAE, particularly in the case that the average value of the ratings in the test set is closer to the maximum rating value than the minimum. Aux-NMF [75] can alleviate this issue by adding the users' cluster constraint so that in each iteration of updating the matrices U, S, and V, the β value is added to the U matrix, as we see in Equation (2.1). However, there are some issues with Aux-NMF. First, we cannot guarantee that each dataset has users information to build the users and items features, which are used to cluster users. Users' information is really difficult to collect and is often untrustworthy. Most users do not provide their personal information for many reasons, e.g., they do not trust the reliability of the system to keep the privacy of their personal information. In addition, providing personal information is time-consuming. If the system forces users to provide their personal information at registration, the system may lose users.

In addition, it is difficult for the system to trust users' information for many reasons. First, most users' information changes over time, such as occupation, address, marital status, education, and life experiences. In addition, users' hobbies vary depending on age, occupation, marital status, living place, etc. Even if the system allows users to update their information, it is difficult to ensure that users do so regularly. In addition, some information is hard to be collected in a multichoice style, such as the address, because there are many counties and cities. However, if the system allows users to provide their address as a text, it is difficult to extract the address automatically for many reasons. For example, users may misspell the address or write an unreal address, such as sky, heaven, and so on. Furthermore, the system may get confused if the user enters the name of a city that is also the name of a country, and vice versa.

3.2 Proposed Method

We propose a new strategy that handles New-Users issues and reduces the rating matrix sparsity by incorporating the trust network into WNMTF Equation (3.7). To perform that strategy, we impute a subset of missing ratings using the trustees' ratings. In reality, users may trust each other based on their ratings. We assume if the user did not rate an item, then the rating of that item will be similar to his/her trustees' rating for that item. We impute each missing rating with the average rating of trustees in his/her trust list. If none of the trustees rated that item, we would keep the rating as a missing rating. By this method, we introduce ratings to New-Users so the system can recommend items to them and add more ratings to other users to reduce the sparsity; most trustee users are Heavy-Rater-Users. In addition, the imputation process adds more ratings to Cold-Start-Users who did not rate many items, which may help the recommendation system to recommend items more accurately. Our proposed method is different from [24] in the selection method of the missing ratings that are imputed, and the source ratings that are used to calculate the imputed ratings of the missing ratings. We name the proposed method Trust-WNMTF.

3.2.1 Objective Function

To recommend items to New-Users and alleviate the rating matrix R sparsity, we replace rating matrix R with imputed rating matrix R' such that,

$$r'_{ij} = \begin{cases} r_{ij} & \text{if } r_{ij} \neq 0 \\ \frac{\sum_{fr} r_{ij}}{|f_r|} & \text{if } r_{ij} = 0,\ \sum_{fr} r_{ij} > 0, \text{ and} |f_r| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where $r'_{ij} \in R'$, $r_{ij} \in R$, f_r is the set of trustees who rated item j, and $|f_r|$ is the total number of trustees who rated item j.

In addition, W is redefined with W' based on Equation (3.8), which is similar to Equation (2.3) in Chapter 2. By updating WNMTF(3.7) using Equations (3.8) and (2.3), the objective function of Trust-WNMTF is:

$$f(R', W', U, S, V) = min_{U \geq 0, S \geq 0, V \geq 0} \| W' \circ (R' - USV^T) \|_F^2 \tag{3.9}$$

3.2.2 Update Formula

The update formulae for Trust-WNMTF are as follows [75],

$$U_{ij} \leftarrow U_{ij} \cdot \frac{[(W' \circ R')VS^T]_{ij}}{\{[W' \circ (USV^T)]VS^T\}_{ij}} \tag{3.10}$$

$$V_{t_{ij}} \leftarrow V_{t_{ij}} \cdot \frac{[(W' \circ R')^T US]_{ij}}{\{[W' \circ (USV^T)]^T US\}_{ij}} \tag{3.11}$$

$$S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T (W' \circ R')V]_{ij}}{\{U^T [W' \circ (USV^T)]V\}_{ij}} \tag{3.12}$$

3.2.3 Convergence Analysis

The proof of convergence of the update formulae (3.10), (3.11), and (3.12) is similar to Section 2.2.3 when the items cluster constraint is omitted, i.e., γ is set to zero.

3.2.4 Detailed Algorithm

In this section, the Trust-WNMTF algorithm is presented in Algorithm 3.1, which describes the steps of performing Trust-WNMTF on the imputed rating matrix R'. We perform this algorithm with two cases. In the first case, the All-Users imputation case, all users are imputed. In the other case, only New-Users are imputed, therefore, this case is called the New-Users imputation case. However, it may take hundreds or thousands of iterations to converge to a local minimum. Thus, in the algorithm, we set an additional stop criterion - the maximum iteration count. In collaborative filtering, this value varies from 10 $\sim$ 100, which can produce good results [75].

Algorithm 3.1 Trust-WNMTF

```
Require:
  User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
  Trust matrix: $T \in \mathbb{R}^{m \times m}$;
  Column dimension of $U : k$;
  Column dimension of $V : l$;
  Number of maximum iterations: $MaxIter$;
  Imputation Case: $Case$;
Ensure:
  Imputed rating matrix: $R' \in \mathbb{R}^{m \times n}$;
  Factor matrices: $U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times l}$, and $S \in \mathbb{R}^{k \times l}$;

function IMPUTATION($R$, $T$, $i$, $j$)
    find the user's $i$ trustees from the trust matrix $T \rightarrow L_t$
    for each trustee $l_t$ in $L_t$ do
        if $r_{l_t j} > 0$ then
            $r_{ij}temp = r_{ij}temp + r_{l_t j}$
            $L_tCounter = L_tCounter + 1$
        end if
    end for
    if $L_tCounter > 0$ then
        $r'_{ij} = \frac{r_{ij}temp}{L_tCounter}$
    else
        $r'_{ij} = 0$
    end if
    return $r'_{ij}$
end function

Initialize $U, V$, and $S$ with random values;
if $Case ==$ All-Users Imputation then
    for each user $u_i$ do
        for each item $e_j$ do
            if $r_{ij} == 0$ then
                $r'_{ij} =$ Imputation($R$, $T$, $i$, $j$)
            else
                $r'_{ij} = r_{ij}$
            end if
        end for
    end for
else if $Case =$ New-Users Imputation then
    for each user $u_i$ do
        if total ratings for $u_i == 0$ then
            for each item $e_j$ do
                $r'_{ij} =$ Imputation($R$, $T$, $i$, $j$)
            end for
        else
            $r'_{i:} = r_{i:}$
        end if
    end for
end if
Build weight matrix $W'$ by Eq. (2.3);
Set $iteration = 1$ and $stop = false$;
while $(iteration < MaxIter)$ $and$ $(stop == false)$ do
    $U_{ij} \leftarrow U_{ij} \cdot \frac{[(W' \circ R')VS^T]_{ij}}{\{[W' \circ (USV^T)]VS^T\}_{ij}}$
    $V_{ij} \leftarrow V_{ij} \cdot \frac{[(W' \circ R')^T US]_{ij}}{\{[W' \circ (USV^T)]^T US\}_{ij}}$
    $S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T(W' \circ R')V]_{ij}}{\{U^T[W' \circ (USV^T)]V\}_{ij}}$
    $L \leftarrow \|W' \circ (R' - USV^T)\|_F^2$
    if $L$ increases in this iteration then
        $stop = true$;
        Restore $U, S$, and $V$ to their values in last iteration.
    end if
end while
Return $R', U, S$, and $V$.
```

3.2.5 Complexity

The computational complexity of Trust-WNMTF can be broken down into two phases: imputation and WNMTF phase (updating U, V, and S).

There are three basic steps to preform the imputation in Trust-WNMTF which need to be considered in the computational complexity. First, the time complexity of searching for missing ratings is $O(mn)$. On the other hand, the time complexity of finding the trustees of all users is $O(m^2)$. Finally, the time complexity to obtain the trustees ratings of the missing ratings is $O(m(mn))$. By combining the time complexity of all imputation steps, the total of the time complexities is as follows,

$$TimeComplexity = O(mn) + O(m^2) + O(m(mn)) \tag{3.13}$$

where m is the total number of the users in the rating matrix R and n is the total number of the items in the rating matrix R. In addition, the time complexity of the imputation phase is considered in the worst case.

The time complexity of the imputation in Trust-WNMTF is quadratic. As mentioned before, there are two imputation cases: All-Users and New-Users. The time complexity of the All-Users imputation case is closer to the worst case of the time complexity than the New-Users imputation case. With All-Users imputation case, we need to run the second term in Equation (3.13) with all users m; however, with New-Users imputation case, the second step of the imputation should be ran for New-Users only. We assume that $NewUserNum \ll m$. For the last step of the imputation of the Trust-WNMTF method, the total number of the trustees for each user $TrusteeNum \ll m$. However, as we mention with the second imputation step, finding the trustees' ratings is only for New-Users with the New-Users imputation case. For large scale datasets, the imputation process could be ran in parallel to reduce the computation time.

On the other hand, we suppose $k, l \ll \min(m, n)$, the time complexities of updating U, V, and S in each iteration are all $O(mn(k + l))$ [75].

3.3 Experimental Study

We run three experiments for all users together and each user group. The first is the the baseline method, i.e., the WNMTF Equation (3.7), which is considered a Non-Imputation case. The second experiment is called the All-Users imputation case where all users are imputed in the training set of Trust-WNMTF Equation (3.9). The last experiment is called the New-Users imputation case; here, only New-Users are imputed in the training set of Trust-WNMTF Equation (3.9). The setup of k, l, and $MaxIter$ parameters for the Trust-WNMTF algorithm is shown in Table 2.3. The setup of FilmTrust dataset parameters is $k = 3$, $l = 20$, and $MaxIter = 10$, in which they are set based on experiments.

With the All-Users Imputation case, the MAE of all users is almost equal to/higher than MAE of the Non-Imputation case except for the Epinions dataset, as we see in Table 3.1. To understand the reasons for these results, we need to analyze the results of each user group.

The New-Users group MAE improves in All-Users Imputation case with all datasets. The improvement ratio differs from one dataset to another. The Epinions and Ciao datasets get the most improvement. There are two factors that impact the results. The first is the difference in the percentage of the New-Users ratings between the All-Users imputation and Non-Imputation cases. The largest difference we observed is in the Epinions dataset at 7.5%, as we see in Figure 3.1. The other datasets have a difference in the percentage that is lower than 1.25%, as

Table 3.1: MAE results of all users and each user group (New-Users, Cold-Start-Users, and Heavy-Rater-Users) for each imputation case (Non-Imputation, All-Users Imputation, and New-User Imputation).

User Group/ Imputation Case	Non-Imp.	All-Users Imp.	New-Users Imp.
Ciao			
All-Users	0.7911	0.7907	**0.7530**
New-Users	4.2796	1.2860	**1.2649**
Cold-Start-Users	0.7976	**0.7893**	0.7986
Heavy-Rater-Users	**0.7431**	0.7839	0.7453
CiaoDVD			
All-Users	2.1810	2.1829	**2.1566**
New-Users	4.3050	4.1941	**4.1934**
Cold-Start-Users	**1.0425**	1.0861	1.0533
Heavy-Rater-Users	**1.2042**	1.2592	1.2206
Epinions			
All-Users	1.3349	1.1549	**1.1382**
New-Users	3.9187	1.9562	**1.9506**
Cold-Start-Users	**0.9472**	0.9742	0.9716
Heavy-Rater-Users	**1.0155**	1.0603	1.0411
FilmTrust			
All-Users	0.7289	0.7337	**0.7173**
New-Users	3.2304	2.5334	**2.5274**
Cold-Start-Users	0.7535	**0.7477**	0.7595
Heavy-Rater-Users	**0.6845**	0.7018	0.6849

shown in Figure 3.1. The second factor is the percentage of the New-Users ratings after the imputation process since we cannot guarantee that all New-Users get imputed ratings in this step. The lowest percentage is observed in the Ciao dataset, which is 0.18%, while the other datasets have a New-Users ratings percentage of more than 1%, as shown in Figure 3.1. Even though the CiaoDVD dataset does not have the smallest difference in the percentage of the New-Users between the Non-Imputation and All-Users imputation cases, the CiaoDVD dataset has the lowest improvement in MAE results of the New-Users group, as shown in Table 3.1, because it has the highest percentage of the New-Users ratings after the imputation process, which is 30.5%, as we see in Figure 3.1. The Cold-Start-Users group results improve with the Ciao and FilmTrust datasets only. However, all datasets get worse predictions with the Heavy-Rater-Users group after the All-Users imputations process.

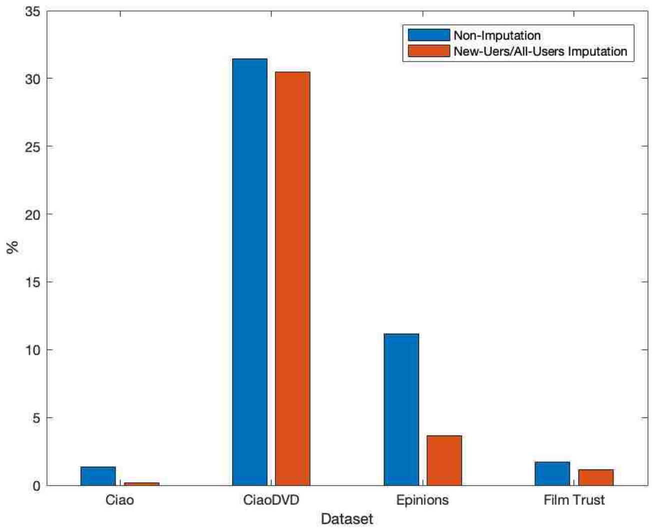


Figure 3.1: New-Users ratings % in the test set before and after the imputation.

After we analyze the results for each group, we find that the MAE after the All-Users Imputation process gets better for the New-Users group, worse for the Heavy-Rater-Users group, and is in between for the Cold-Start-Users group. A large difference in the percentages of the New-Users ratings between the Non-Imputation and All-Users imputation cases leads to an improvement in the MAE for the whole test set. This is observed in the Epinions dataset where 7.5% of the New-Users' ratings in the test set do not belong to the New-Users group after the imputation process, as we see in Figure 3.1. However, the other datasets have only a difference in the percentage of 1.5% or less, which results in a worse MAE in the All-User imputation than in the Non-Imputation case.

In the second method, the New-Users Imputation case, we impute only New-Users. All datasets get better results, as we see in Table 3.1. The results of the New-Users group improve with all datasets and are slightly better than the All-Users imputation case results. The improvement ratios of the New-Users group are almost the same as the All-Users imputation case, and the reasons for improvement are the same as well. However, the Cold-Start-Users group results are worse than the Non-Imputation method in all datasets. With the Heavy-Rater-Users group, the MAE increases even if only New-Users are imputed. Nevertheless, the increase in MAE is much lower than the All-Users imputation case.

In the New-Users Imputation case, the Epinions dataset gets the most improvement because 7.5% of the ratings of the New-Users group in test set do not belong to the New-Users group after the imputation process, as we see in Figure 3.1, which is the largest percentage among other datasets. On the other hand, the lowest ratio improvement is with the FilmTrust dataset because it has the lowest difference in the New-Users group ratings percentages between Non-Imputation and New-Users imputation cases, as we can see in Figure 3.1. The improvement ratio with CiaoDVD is in between the FilmTrust and Ciao datasets. The difference in the New-Users group ratings percentage is more in CiaoDVD than FilmTrust and slightly more than Ciao datasets. However, CiaoDVD still suffers from the highest percentage of the New-Users group ratings after the imputation process.

The Ciao dataset has a better result than the FilmTrust dataset because the Ciao New-Users group gets more improvement than the FilmTrust New-Users group. This happens because the Ciao dataset has a lower New-Users ratings percentage after the imputation process than the FilmTrust dataset, as we see in Figure 3.1.

In summary, the results of the Heavy-Rater-Users group is worse with both imputation cases in all datasets, especially with the All-Users imputation case. However, the results for the New-Users group improve with both imputation cases, especially with the New-Users imputation. In fact, by using the imputation process, the system can recommend items to New-Users. In all datasets, the Cold-Start-Users groups get a worse MAE in the New-Users imputation case than the Non-Imputation case. However, with the All-Users imputation case, the Cold-Start-Users group in the Ciao and FilmTrust datasets get better results while CiaoDVD and Epinions get worse results, which shows that imputing Cold-Start-Users is sometimes beneficial. In Chapter 4, we will explore the factors that impact Cold-Start-Users results.

3.3.1 Imputation Vs. WNMTF

In this section, we study the impact of the WNMTF on the imputed ratings. In particular, we analyze whether or not the WNMTF reduces the MAE of the imputed ratings. To test this, we calculate the MAE of a subset of test set ratings

Table 3.2: The % of imputed ratings in test set.

Dataset	% of ratings
Ciao	0.07 %
CiaoDVD	0.17 %
Epinions	0.80 %
FilmTrust	0.26 %

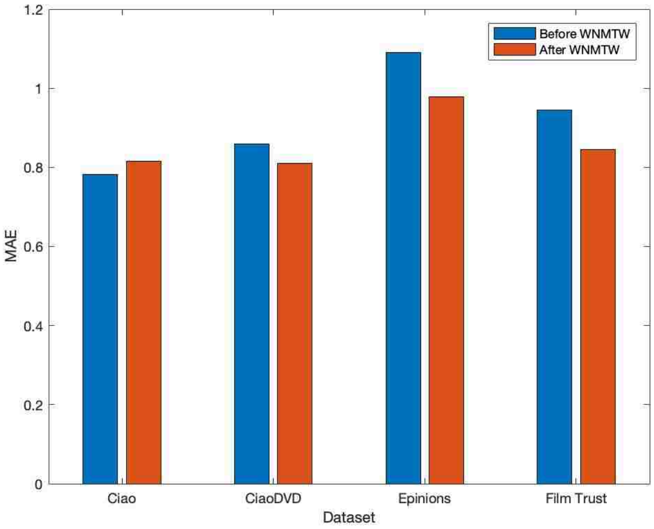


Figure 3.2: MAE results before and after WNMTF of the test set ratings that can be imputed.

that can be imputed using the training set in two cases. The first case is before WNMTF is applied. The MAE is calculated using the original ratings in the test set and the imputed ratings in the training set for the same ratings. In the second case, the MAE is calculated after applying WNMTF. We chose the New-Users imputation case because it has the lowest MAE.

As we can see from Table 3.2, the percentages of the ratings that are in the test set and are imputed are very low when compared with the total test set ratings. The MAE results after applying the WNMTF are improved with all datasets, except for Ciao as we can see in Figure 3.2, which has the lowest ratings percentages that are imputed, and in the test set results shown in Table 3.2. The Epinions dataset has the best improvement ratio among other datasets, as we see in Figure 3.2, and the largest ratings percentages that are imputed and in the test set results as shown in Table 3.2.

Based upon the conclusions of our results, we can say that the MAE result improves if the percentage of ratings that are imputed and in the test set exceeds a certain threshold, such as 10% in our experiment. In addition, more percentages of ratings are imputed and are in the test set results in a better improvement ratio after applying WNMTF.

3.4 Summary

In this chapter, we proposed the Trust-WNMTF method to incorporate trust network information into the WNMTF by utilizing the imputation. Our results show that the Trust-WNMTF New-Users Imputation case is better than Non-Imputation (WNMTF), especially when the dataset suffers from New-Users, but worse at some others. On the other hand, the results of the Trust-WNMTF All-User Imputation case indicate an impact of the error of the imputed ratings that may be introduced to the system. In addition, the WNMTF reduces the MAE of the subset from the test set that can be imputed when the percentage of that ratings is large.

4 Influential Factors of Imputation with Trust Network Information for Cold-Start Users

We propose an NMF-based approach to improve the Cold-Start-Users predictions since Cold-Start-Users suffer from a high error in the results. The proposed method utilizes the trust network information to impute a subset of the missing ratings before NMF is applied. We propose three strategies for selecting the subset of missing ratings that hold the imputed ratings in order to examine the influence of the imputation with both item groups: Cold-Start-Items and Heavy-Rated-Items; and we survey to find if the trustees' ratings could improve the results more than the other users. We analyze two factors that may affect the results of the imputation: (1) the total number of imputed ratings, and (2) the average of rating values in the training set before and after the imputation. Experiments on four different datasets are conducted to examine the proposed approach. The results show that our approach improves the predicted ratings of the Cold-Start-Users and alleviates the impact of the imputed ratings.

4.1 Problem Description

Generally, the Cold-Start-Users suffer from a high error in the prediction results compared to Heavy-Rater-Users, as we see in Table 3.1, either with WNMTF or Trust-WNMTF. In Chapter 3, all users are imputed with all available imputed ratings in the proposed method, Trust-WNMTF All-Users, in order to improve the accuracy of the rating prediction. Nevertheless, some datasets' prediction of Cold-Start-Users did not improve with the imputation even though some others improved. In this chapter, we intend to study the behavior of the Cold-Start-Users and Heavy-Rater-Users groups with the imputation process and we analyze the factors that affect the prediction accuracy when the imputation process is applied.

Table 4.1: The average of the rating values in the training set of the original rating matrix R for whole dataset and each user group.

Dataset	Whole Dataset	Cold-Start-Users	Heavy-Rater-Users	Rating Median Value
Ciao	4.1483	4.2164	4.1442	3
CiaoDVD	4.0711	4.2860	3.9369	3
Epinions	3.8742	3.9126	3.8640	3
FilmTrust	3.0028	3.1219	2.9954	2.75

As shown in Table 4.1, the average of the Cold-Start-Users rating values in the training set is higher than the whole dataset rating value average and the Heavy-Rater-Users rating value average in all datasets. In addition, the average of the training set ratings of all users is higher than the median value of the ratings.

This indicates that users tend to rate items that they like more than items that they do not like. This could be for several reasons. First, in the e-commerce era, it is easy for users to know all the information they need about the item before they make a decision whether to buy it. In addition, users tend to trust their choices. Further, users tend to buy what they know, such as a certain brand, instead of taking a risk and buying what they don't know. In reality, users did not try a lot of options to give a fair rating to items.

In general, Cold-Start-Users have higher MAE because of several reasons. The first reason is the lack of ratings in the training set. Even though the average of rating values of the Cold-Start-Users is the highest compared to other user groups, it does not have a significant influence on the whole dataset rating value average because of the lack of Cold-Start-Users ratings in the training set. In our proposed method, we have two goals: (1) improve the Cold-Start-Users predictions, and (2) limit the impact of the imputed ratings. This could be done by increasing the total number of the Cold-Start-Users ratings and simultaneously increasing the average of the training set rating values through the imputed ratings.

4.2 Proposed Method

We propose a new strategy to improve the Cold-Start-Users predictions by incorporating the trust information into MNMTF(3.7). Even though the proposed method is similar to Trust-WNMTF in Chapter 3, the proposed method in this chapter concentrates on restricting the negative impact of imputed ratings that appear clearly in Trust-WNMTF, especially with Heavy-Rater-Users. In addition, the proposed method aims to increase the accuracy of Cold-Start-Users predictions.

To perform the proposed method, we need to determine the source ratings of the imputed rating, which are defined as a subset of the real ratings that are used to calculate the value of the imputed ratings. Each user group is imputed with a limited number of imputed ratings. The items that have been rated by the user's trustees are considered as candidate items that may be imputed for that user (trustor). In addition, the subset of items from candidate items that are selected to hold the imputed ratings should be chosen carefully. To do that, we count the total number of the ratings for each candidate item from all users and the total number of the ratings for the item from the user's trustees only.

We propose three methods to select the imputed items from the candidate items set. The first method is called the Trustee case in which the candidate items are ordered based on the total number of the ratings for the items from the user's trustees descendingly, then by the total number of the ratings for the item from all users ascendingly in case of tie values. The second method is the CSI case in which the candidate items are ordered based on the total number of the ratings for the item from all users ascendingly, then the tie values are ordered based on the total number of the ratings for the items from the user's trustees descendingly. The last method is called HI because the candidate items are ordered based on the total number of the ratings for the item from all users descendingly, then by the total number of the ratings of the items from the user's trustee descendingly as well. Table 4.2 shows the summary of the three cases. The purpose of these methods is to examine the influence of the imputation with both item groups: Cold-Start-Items and Heavy-Rated-Items. In addition, we want to find if the trustees' ratings could improve the results more than the other users' ratings.

The source ratings for each imputed rating are all trustees' ratings for the selected item that will be imputed. However, the value of the imputed ratings equals the average of the rating values of the imputed user's trustees for that item (source ratings). In addition, we set a total number of the imputed ratings for

Table 4.2: The summary of the three proposed cases.

Rating Source	All Users		User's trustee only	
	Order priority	Order type	Order priority	Order type
Trustee	2	acs.	1	desc.
CSI	1	acs.	2	desc.
HI	1	desc.	2	desc.

each user group to limit the error that is introduced by the imputed rating. To do that, three parameters that define the total number of the imputed ratings for each user group need to be set in advance.

4.2.1 Objective Function

In the proposed method, we replace the rating matrix R in Equation (3.7) with the imputed rating matrix R', such that,

$$r'_{ij} = \begin{cases} r_{ij} & \text{if } r_{ij} \neq 0 \\ \frac{\sum_{fr} r_{ij}}{|fr|} & \text{if } r_{ij} = 0,\ \sum_{fr} r_{ij} > 0,\ |f_r| > 0, \text{ and meet the conditions} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $r'_{ij} \in R'$, $r_{ij} \in R$, and f_r is the set of trustees of users who rated at least one item, and $|f_r|$ is the total number of trustees who rated at least once. In the proposed method, the first condition is that each user's group has a limited number of imputed ratings. This condition must be satisfied so that the total number of the imputed ratings for each user does not exceed the parameter settings. The second condition is that the imputed item should belong to the corresponding case (Trustee, CSI, or HI) that is applied.

In addition, W' in Equation (2.3) is defined based on Equation (4.1). When we update Equation (3.7) by using Equations (4.1) and (2.3), the objective function is similar to Equation (3.9), Chapter 3, such that,

$$f(R', W', U, S, V) = min_{U\geq 0, S\geq 0, V\geq 0} ||W' \circ (R' - USV^T)||_F^2 \tag{4.2}$$

We name the proposed method, Trust-WNMTF++.

4.2.2 Update Formula

The update formulae are the same as Trust-WNMTF, Section 3.2.2 in Chapter 3.

4.2.3 Convergence Analysis

The convergence proof of the update formulae is the same as Section 3.2.3 in Chapter 3, as well.

4.2.4 Detailed Algorithm

Algorithm 4.1 depicts the steps of performing Trust-WNMTF++ on the imputed rating matrix R'. As mentioned previously, the algorithm is performed with three cases: Trustee, CSI, and HI. However, it may take hundreds or thousands of iterations to converge to a local minimum. Thus, in the algorithm, we set an additional stop criterion - the maximum iteration count. In collaborative filtering, this value varies from 10 $\sim$ 100, which can produce good results [75].

Algorithm 4.1 Trust-WNMTF++

```
Require:
  User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
  Trust matrix: $T \in \mathbb{R}^{m \times m}$;
  Column dimension of $U : k$;
  Column dimension of $V : l$;
  Total number of the imputed ratings for New-User group: $NUIR$ ;
  Total number of the imputed ratings for Cold-Start-User group: $CSUIR$;
  Total number of the imputed ratings for Heavy-Rater-User group: $HUIR$;
  Number of maximum iterations: $MaxIter$;
  Imputation Case: $ImpCase$;
Ensure:
  Imputed rating matrix: $R' \in \mathbb{R}^{m \times n}$;
  Factor matrices: $U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times l}$ and $S \in \mathbb{R}^{k \times l}$;

Initialize $U, V$, and $S$ with random values
Set $R' = R$
for each user $u_i$ do
    find the user's $i$ trustees from the trust matrix $T \rightarrow L_t$
    if $count(L_t) > 0$ then
        find all items that have been rated by $L_t \rightarrow candidateItems$
        if $count(candidateItems) > 0$ then
            for each $candidateItems$ $c_j$ do
                calculate the average of the rating values of $L_t$ users for item $c_j \rightarrow ImputedRatingValue$
                count the total number of ratings for $c_j$ from all users $\rightarrow totalRatingsAllUsers$
                count the total number of ratings for $c_j$ from $L_t$ $\rightarrow totalRatingsTrusteesUsers$
            **end for**
            **if** $ImpCase == Trustee$ **then**
                Order $candidateItems$ based on $totalRatingsTrusteesUsers$ descendingly, then for the tie values
                Order $candidateItems$ based on $totalRatingsAllUsers$ ascendingly
            **else if** $ImpCase == CSI$ **then**
                Order $candidateItems$ based on $totalRatingsAllUsers$ ascendingly, then for the tie values
                Order $candidateItems$ based on $totalRatingsTrusteesUsers$ descendingly
            **else if** $ImpCase == HI$ **then**
                Order $candidateItems$ based on $totalRatingsAllUsers$ descendingly, then for the tie values
                Order $candidateItems$ based on $totalRatingsTrusteesUsers$ descendingly
            **end if**
            **if** total ratings number of $u_i == 0$ **then**
                $topImpRatings = NUIR$
            **else if** total ratings number of $u_i > 0$ **and** total ratings number of $u_i < 5$ **then**
                $topImpRatings = CSUIR$
            **else if** total ratings number of $u_i > 4$ **then**
                $topImpRatings = HUIR$
            **end if**
            Set $ImputedRatingCounter = 0$
            Set $candidateItemsIndex = 1$
            **while** $ImputedRatingCounter < topImpRatings$ **do**
                $j$ = item index of $candidateItems(candidateItemsIndex)$
                **if** $r_{i,j} == 0$ **then**
                    $r'_{i,j} = ImputedRatingValue(candidateItemsIndex)$
                    $ImputedRatingCounter = ImputedRatingCounter + 1$
                **end if**
                $candidateItemsIndex = candidateItemsIndex + 1$
            **end while**
        **end if**
    **end if**
**end for**
Build weight matrix $W'$ by Eq. (2.3);
Set $iteration = 1$ and $stop = false$;
**while** $(iteration < MaxIter)$ $and$ $(stop == false)$ **do**
    $U_{ij} \leftarrow U_{ij} \cdot \frac{[(W' \circ R')VS^T]_{ij}}{\{[W' \circ (USV^T)]VS^T + U\}_{ij}}$
    $V_{ij} \leftarrow V_{ij} \cdot \frac{[(W' \circ R')^T US]}{\{[W' \circ (USV^T)]^T US + V\}_{ij}}$
    $S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T(W'\circ R')V]_{ij}}{\{U^T[W'\circ(USV^T)]V\}_{ij}}$
    $L \leftarrow \|W' \circ (R' - USV^T)\|_F^2$
    if $L$ increases in this iteration then
        $stop = true$;
        Restore $U, V$, and $S$ to their values in last iteration
    end if
end while
Return $R', U, V$, and $S$
```

4.2.5 Complexity

The computational complexity of Trust-WNMTF++ is similar to the computational complexity of Trust-WNMTF in Section 3.2.5. However, an extra step needs to be added to the time complexity which is the selection of the holder items. Two totals must be counted. The first one is the total number of the ratings from all users, $O(mn)$. This step could be done in searching of the missing ratings step. The second one is the total of number of the ratings from the user's trustees only, $O(m(mn))$. This step can be done in the process of obtaining the trustees ratings of the missing ratings step. The time complexity of sorting the items is $O(mn^2)$. By combining the time complexity of all imputation steps of Trust-WNMTF++, the time complexity is as follows,

$$TimeComplexity = O(mn) + O(m^2) + O(m(mn)) + O(mn^2) \tag{4.3}$$

where m is the total number of the users in the rating matrix R and n is the total number of the items in the rating matrix R. In addition, the time complexity of the imputation phase is considered in the worst case.

4.3 Experimental Study

Table 4.3: The best parameters setting of the proposed method with the best case of each dataset.

Dataset	$NUIR$	$CSUIR$	$HUIR$
Ciao	12	5	1
CiaoDVD	8	2	3
Epinions	3	4	2
FilmTrust	10	2	2

In this section, we present and discuss our experimental results. Our proposed method is compared with WNMTF 3.7 and with both proposed cases of the Trust-WNMTF approach in Chapter 3, All-Users and New-Users imputation, too. The parameters are set the same as in Chapter 3. In addition, Table 4.3 shows the additional parameters that must be set for the proposed method. Because the purpose of this chapter is to focus on Cold-Start-Users more than New-Users as in Chapter 3, the datasets are re-split to ensure sufficient availability of Cold-Start-Users.

Table 4.4: The MAE of WNMTF, Trust-WNMTF cases: All-Users and New-Users, and the proposed method Trust-WNMTF++ with the three cases (Trustee, CSI, and HI).

Method/Dataset	Ciao	CiaoDVD	Epinions	FilmTrust
Previous Methods				
WNMTF	0.8237	1.6503	1.0816	0.7288
Trust-WNMTF All-Users	0.8305	1.6721	1.0751	0.7439
Trust-WNMTF New-Users	0.8224	1.6462	1.0760	0.7269
Proposed Methods				
Trust-WNMTF++ Trustee	**0.8025**	**1.6348**	1.0382	0.7206
Trust-WNMTF++ CSI	0.8029	1.6368	**1.0372**	**0.7200**
Trust-WNMTF++ HI	0.8127	1.6411	1.0448	0.7226

Table 4.5: The percentage of each item group that is imputed in the training set with the three proposed cases: Trustee, CSI, and HI (CSI = Cold-Start-Items group; HI = Heavy-Rated-Items group).

Proposed Case	Trustee Case		CSI Case		HI Case	
Item Group	CSI	HI	CSI	HI	CSI	HI
Ciao	49.96%	50.04%	84.63%	15.37%	4.16%	95.84%
CiaoDVD	42.26%	57.74%	95.75%	4.25%	4.18%	95.82%
Epinions	16.47%	83.53%	57.88%	42.12%	0.86%	97.38%
FilmTrust	32.90%	67.10%	51.16%	48.84%	2.15%	97.85%

In general, the results of the three cases of our proposed method are better than WNMTF and Trust-WNMTF: All-Users and New-Users with all datasets are shown in Table 4.4. However, the results of the HI case are the worst compared to Trustee and CSI cases with all datasets. Furthermore, Ciao and CiaoDVD have better results with the Trustee case; and, Epinions and FilmTrust results are better with the CSI case. We noticed that the percentage of the Heavy-Rated-Items imputed in Epinions and FilmTrust with the Trustee case is very high compared to the other datasets, as we see in Table 4.5. This indicates that imputing Heavy-Rated-Items limits the advantages of the imputations and introduces large error. In addition, we noted that the lowest improvement of the proposed method is with CiaoDVD because of the existence of the New-Items where CiaoDVD has the most percentage of New-Items in the test set, especially with Cold-Start-Users group, as we see in Figure 4.1.

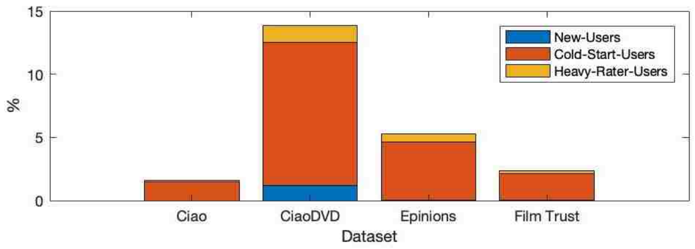


Figure 4.1: The percentage of New-Items ratings with each users group in the test set.

In Table 4.6, the results of each user group are shown with the previous methods and the best case of the proposed method, Trust-WNMTF++, for each dataset to examine each users group behavior with the imputation.

The New-Users group gets slightly better results than the Trust-WNMTF New-Users method, but it is worse in the Ciao dataset. This could be because the percentage of the imputed New-Users ratings in the test set in the Ciao dataset is the lowest among other datasets, as we see in Table 4.7.

The results of the Heavy-Rater-Users with the proposed method are the best compared to other methods even though the Non-Imputation method, i.e., WNMTF, of all datasets but not with FilmTrust, as shown in Table 4.6. This is because the averages of the rating values in the training set for the whole FilmTrust dataset and Heavy-Rater-Users are the lowest compared to other datasets. With FilmTrust, the best results of Heavy-Rater-Users is with WNMTF, and increasing the average of the training rating values after the imputation leads to more errors in the low ratings values. This issue will be explained in more detail in Chapter 5, Section 5.3.1. On the other hand, the worst result of Heavy-Rater-Users is obviously in the Trust-WNMTF All-Users method (as shown in Table 4.6) and markedly improved with the proposed method, Trust-WNMTF++, compared to

Table 4.6: The MAE for whole dataset and each user group of WNMTF, Trust-WNMTF: All-Users and New-Users, and the best case for each dataset of the proposed method Trust-WNMTF++.

Methods	All-Users	New-Users	Cold-Start-Users	Heavy-Rater-Users
Ciao				
WNMTF	0.8237	4.4118	0.8345	0.7452
Trust-WNMTF All-User	0.8305	1.4235	0.8399	0.7715
Trust-WNMTF New-User	0.8224	**1.3615**	0.8345	0.7453
Trust-WNMTF++ Trustee	**0.8025**	1.3999	**0.8118**	**0.7438**
CiaoDVD				
WNMTF	1.6503	4.3433	1.2397	1.0612
Trust-WNMTF All-User	1.6721	4.2832	1.2722	1.1122
Trust-WNMTF New-User	1.6462	4.2830	1.2442	1.0689
Trust-WNMTF++ Trustee	**1.6348**	**4.2824**	**1.2302**	**1.0606**
Epinions				
WNMTF	1.0816	3.9203	1.0770	0.9316
Trust-WNMTF All-User	1.0751	1.9541	1.0888	0.9769
Trust-WNMTF New-User	1.0760	1.9495	1.0964	0.9543
Trust-WNMTF++ CSI	**1.0372**	**1.9297**	**1.0529**	**0.9311**
FilmTrust				
WNMTF	0.7288	3.3677	0.7326	**0.6455**
Trust-WNMTF All-User	0.7439	2.7780	0.7487	0.6679
Trust-WNMTF New-User	0.7269	2.7735	0.7324	0.6463
Trust-WNMTF++ CSI	**0.7200**	**2.7639**	**0.7242**	0.6478

Trust-WNMTF All-Users. Even though the Trust-WNMTF New-Users method imputes only New-Users, Heavy-Rater-Users results are worse than the results of the Non-Imputation case, WNMTF. On the contrary, Heavy-Rater-Users results improve slightly with the proposed method, Trust-WNMTF++, compared to the Trust-WNMTF New-Users method, but the improvement of the accuracy is notable for the Epinions dataset. This is because the worst descent in the accuracy of the results is within Epinions among other datasets. We can conclude that the proposed method can handle the negative impact of the imputation on the Heavy-Rater-Users.

The Cold-Start-Users accuracy results improve compared to the WNMTF and Trust-WNMTF: All-Users and New-Users. When the proposed method, Trust-WNMTF++, is compared with the Non-Imputation method, WNMTF, we notice

Table 4.7: The percentage of the ratings for each users group in the test set before and after the imputation.

User Group	New-Users		Cold-Start-Users		Heavy-Rater-Users	
Imputation Case	Before	After	Before	After	Before	After
Ciao	0.05%	0.01%	86.43%	2.54%	13.52%	97.45%
CiaoDVD	13.92%	13.63%	73.95%	50.40%	12.12%	35.98%
Epinions	1.29%	0.43%	76.93%	23.77%	21.78%	75.80%
FilmTrust	0.30%	0.23%	86.19%	49.01%	13.50%	50.76%

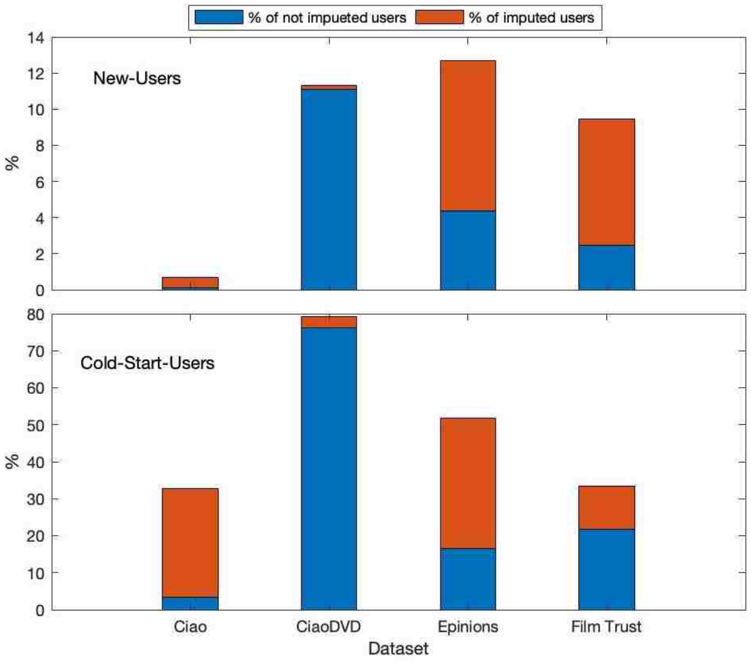


Figure 4.2: New-User and Cold-Start-User groups information in the training set with the best case of Trust-WNMTF++ for each dataset.

that there is a proportional relationship between the percentage of the total number of imputed Cold-Start-Users in the training set and the percentage of the accuracy increase of Trust-WNMTF++, as we see in Figure 4.2 and Table 4.6. For example, the Ciao dataset has the highest percentage of accuracy improvement and the highest percentage of the imputed Cold-Start-Users in the training set, as well. On the other hand, CiaoDVD has the lowest percentage of the improvement in the results and the lowest percentage of the imputed Cold-Start-Users in the training set, too. However, when the proposed method is compared to Trust-WNMTF New-Users, the datasets that have worse results with Trust-WNMTF New-Users and a high percentage of the imputed Cold-Start-Users in the training set with Trust-WNMTF++ get a better result than other datasets. For example, even though Ciao has the highest percentage of the imputed Cold-Start-Users in the training set, Epinions gets a better percentage of accuracy improvement than Ciao because Epinions gets worse MAE with the Trust-WNMTF New-Users method than Ciao. This is the same with CiaoDVD and FilmTrust datasets.

4.3.1 Influence of the Rating Value Average

Table 4.8: The average of the rating values in the training set for the whole dataset with WNMTF, Trust-WNMTF: All Users and New-Users, and the best case of the proposed method Trust-WNMTF++.

Dataset	WNMTF	Trust-WNMTF All-Users	Trust-WNMTF New-Users	Trust-WNMTF++
Ciao	4.1483	**4.1870**	4.1496	4.1569
CiaoDVD	4.0711	3.7887	4.0050	**4.0720**
Epinions	3.8742	3.8314	3.8382	**3.9129**
FilmTrust	3.0028	2.9376	2.9957	**3.0032**

In this section, we analyze the influence the rating value average of the training set has on the accuracy of the results, particularly with Cold-Start-Users since this chapter aims to improve the Cold-Start-Users results. The training set could refer either to only the original ratings as in the WNMTF method or to original and imputed ratings together as in the Trust-WNMTF and Trust-WNMTF++ methods.

As we see in Table 4.1, the average of the Cold-Start-Users original rating values in the training set is higher than the original rating value average of the whole dataset and Heavy-Rater-Users with all datasets. In addition, the rating value average of the training set with the Trust-WNMTF All-Users imputation case is the lowest among other methods in all datasets except Ciao, as shown in Table 4.8. Furthermore, we notice that Cold-Start-Users in the Ciao dataset has the lowest increase in MAE after the Trust-WNMTF All-Users imputation among other datasets, as we see in Table 4.6, which could be because the rating value average did not decrease as in other datasets compared to WNMTF.

With the Trust-WNMTF New-Users imputation case, all datasets have a higher rating value average of the training set than the Trust-WNMTF All-Users imputation case except Ciao, but it is higher than WNMTF (Table 4.8). However, the

Epinions dataset gets the lowest increase in the rating value average among other datasets, as we see in Table 4.8. In addition, the Cold-Start-Users result is worse with the Trust-WNMTF New-Users imputation than with the Trust-WNMTF All-Users imputation case only in Epinions dataset compared to other datasets (Table 4.6). This could be because of the impact of the average of the training set after the imputations.

The highest average of rating values is with the proposed method, Trust-WNMTF++, in all datasets except Ciao, which is the next highest, as we see in Table 4.8. In addition, the best prediction ratings results is with Trust-WNMTF++, as we see in Table 4.6. This indicates that the average of the rating values in the training set has an important influence on the accuracy of the rating prediction.

Even though the highest rating value average of Ciao datasets is with Trust-WNMTF All-Users, the accuracy result is not the best. This could be due to the huge gap between the average of original rating values, i.e., WNMTF method, and the highest average of rating values that may result in introducing error. This denotes the need to control the increase in the average rating values of the training set by utilizing imputed ratings.

We can conclude that increasing the average of the rating values in the training set through the imputed ratings has a strong influence on increasing the accuracy results of Cold-Start-Users. This is because the Cold-Start-Users has a higher average of the rating values in the training set than in the other users group. However, the increase ratio in the average of the rating values should be limited compared to the original ratings in the training set.

4.3.2 Parameter Settings

As mentioned earlier, each user is imputed with a limited number of imputed ratings based on the group that the user belongs to. In our experiment, we set the maximum imputed ratings for New-Users to 20, Cold-Start-Users to 5, and Heavy-Rater-Users to 3 imputed ratings. Table 4.3 shows the total number of the imputed ratings for each users group that results in the lowest MAE for the whole dataset.

For the New-Users group, there is an inverse relationship between the percentage of New-Users in the training set (Figure 4.2) and the best total of imputed ratings of New-Users $NUIR$ (Table 4.3). The Ciao dataset has the lowest New-Users percentage and the highest $NUIR$, and vice versa with the Epinions dataset. Despite the fact that the New-Users percentages in CiaoDVD and Epinions datasets are close, the values of $NUIR$ are not close to each other. This could be because the percentage of New-Users that cannot be imputed is huge in CiaoDVD compared to other datasets, as shown in Figure 4.2.

In addition, there is an inverse relationship between the percentage of ratings in the test set that belong to New-Users that have been imputed and the best total of imputed ratings of New-Users $NUIR$, as we see in Tables 4.3 and 4.7. Ciao dataset has the lowest percentage of the New-Users ratings in the test set that belong to imputed New-Users, 0.04%, and the highest $NUIR$ among other datasets, then FilmTrust comes after Ciao. On the other hand, Epinions has the highest percentage of the New-Users ratings in the test set that belongs to New-Users that have been imputed, 0.89%, and the lowest $NUIR$ among other datasets, then CiaoDVD as we see in Tables 4.3 and 4.7.

With the Cold-Start-Users group, there is a proportional relationship between the percentage of imputed Cold-Start-Users in the training set and the total imputed ratings for each Cold-Start-Users, $CSUIR$, as we see in Figure 4.2 and Table 4.3. For example, the highest imputed Cold-Start-Users in the training set is in the Ciao and Epinions datasets; and they have the highest $CSUIR$ values among other datasets. On the other hand, CiaoDVD and FilmTrust have the lowest percentage of imputed Cold-Start-Users in the training set, and they have the lowest $CSUIR$ value among other datasets. This could be because the rating prediction of the non-imputed Cold-Start-Users may hurt via imputed ratings of other imputed Cold-Start-Users. For that, we need to reduce $CSUIR$ if there is a high percentage of Cold-Start-Users that cannot be imputed.

Table 4.9: The average of rating values in the training set with/without imputing Heavy-Rater-Users.

Parameter Setting			Average of
$NUIR$	$CSUIR$	$HUIR$	rating value
Ciao			
12	5	**1**	4.1569
12	5	**0**	4.1548
CiaoDVD			
8	2	**3**	4.0720
8	2	**0**	4.0717
Epinions			
3	4	**2**	3.9129
3	4	**0**	3.9035
FilmTrust			
10	2	**2**	3.0032
10	2	**0**	3.0042

Even though the Cold-Start-Users group results improve with the proposed method and the Heavy-Rater-Users results do not improve, both Cold-Start-Users and Heavy-Rater-Users groups are imputed. This could be for several reasons. First, as mentioned before, imputing Cold-Start-Items improves the results more than imputing Heavy-Rated-Items. Because the candidate items are ordered based on the total ratings from all users ascendingly, imputing Heavy-Rater-Users allows us to impute more Cold-Start-Items. In addition, as we see in Table 4.9, the average of the rating values in the training set increases when Heavy-Rater-Users are imputed, which is one of the factors that results in a lower MAE. However, it decreases in the FilmTrust dataset when Heavy-Rater-Users are imputed even though it results in a lower MAE. This is because the average of the rating values for the whole dataset and Cold-Start-Users in the training set are the closest to the median value of the rating values among other datasets, as we see in Table 4.1.

There is an inverse relationship between the percentage of imputed Cold-Start-Users in the training set (Figure 4.2) and the best setting of the imputed ratings of Heavy-Rater-User, $HUIR$, as shown in Table 4.3. In addition, there is an inverse relationship between the best setting of the imputed ratings of Cold-Start-User, $CSUIR$, and $HUIR$. The Ciao dataset has the highest percentage of imputed

Cold-Start-Users in the training set, the highest $CSUIR$ value, and the lowest $HUIR$ value. On the other hand, the CiaoDVD dataset has the lowest percentage of imputed Cold-Start-Users in the training set, the lowest $CSUIR$ value, and the highest $HUIR$ value. The FilmTrust and Epinions datasets are in between. In general, the total of the best setting of the imputed ratings of Cold-Start-User and Heavy-Rater-Users together in our experiment with all datasets are in the same range, which are between four and six imputed ratings in total.

Generally, we conclude that the total number of the imputed ratings overall for all user groups should be limited. For New-Users group, if there is a large percentage of New-Users in the training set, then $NUIR$ should be a small value and vice versa. For Cold-Start-Users, less percentage of imputed Cold-Start-Users in the training set requires fewer of the total imputed ratings for each Cold-Start-Users, $CSUIR$, in order to control the imputation error that may be introduced via imputed ratings to the non-imputed Cold-Start-Users. Because increasing the average of the rating values in the training set plays an important role in improving accuracy, imputing Heavy-Rater-Users is one way to increase the average. However, if the Cold-Start-Users are imputed with a large total number of the imputed ratings, $CSUIR$, then the Heavy-Rater-Users should be imputed with a small total number of imputed ratings, $HUIR$, and vice versa.

4.3.3 Summary of Results

In conclusion, handling the lack of the Cold-Start-Users and Cold-Start-Items ratings by imputation could improve the rating prediction for each. One must consider that each imputed rating affects the average of the training rating values, which subsequently affects the prediction performance. In our experiment, the Cold-Start-Users ratings percentage in the test set is really high, which we believe represents the reality. On the other hand, the Cold-Start-Users rating value average in the training set does not have much effect on the whole training set rating average due to the lack of the ratings. Increasing the average of the rating values provides an opportunity to increase the accuracy of Cold-Start-Users. We suggest using the proposed method with the systems that predict ratings of Cold-Start-Users more than Heavy-Rater-Users.

4.4 Summary

In this chapter, we proposed a method to incorporate the trust network information into the WNMTF using the imputation process to improve the Cold-Start-Users and Heavy-Rater-Users prediction results compared to WNMTF and Trust-WNMTF. We proposed three strategies to select the subset of missing ratings to impute in order to examine the influence of the imputation with both item groups: Cold-Start-Items and Heavy-Rated-Items; and find if the trustees' ratings could improve the results more than the other users.

Our results show that imputing Cold-Start-Items improves the results of Cold-Start-Users with the Trust-WNMTF++ method, especially when the dataset suffers from Cold-Start-Users. However, the New-Users results are slightly better with most datasets. The error that was introduced from previously proposed methods is controlled in the proposed method. However, two factors must be taken into account: the total number of the imputed ratings, and the average of the ratings in the training set after the imputation.

5 Selective Imputation Strategies Based on Fused Factored Matrices

In this chapter, we propose a selective imputation NMF-based method that fuses the factored original rating matrix and the factored imputed rating matrix into one system. The outputs of the factored matrices provide four different ways to calculate the predicted ratings, which are called sub-predicted ratings. Our proposed method is capable of predicting the rating by utilizing either the imputed users, or imputed items, or both in order to limit the errors that may be introduced from the imputed ratings. We proposed five strategies to calculate the final predicted ratings from the sub-predicted ratings. The prediction results of the rating values that are not close to the average of the rating values could be enhanced by utilizing the proposed method. Experiments on four different datasets were conducted to examine the proposed approach. The results show that our approach improves the predicted rating, especially with Max Value strategy.

5.1 Problem Description

Even though the previously proposed methods in Chapters 3 and 4 showed that the imputation improves the predicted rating accuracy in general, there are some predicted ratings that have better results without the imputation method. This is because the imputation introduces errors to the system that negatively impact some predicted ratings results. In fact, the imputation process involves imputing the two parts of the recommendation system: users and items. Even though the users are imputed basically in the proposed methods in Chapters 3 and 4, the items are imputed as a consequence of the user imputation, and this is the same with Chapter 2 in which the users are imputed as a consequence of the New-Items imputation.

In addition, despite that the imputed ratings are limited in Trust-WNMTF++ in Chapter 4 to control the error, the parameter settings may be considered an issue because there is no specific strategy to set the parameters for each dataset without running experiments that may cost time and resources.

Imputing either the user or item could limit the error that is introduced from the imputed ratings. To predict a rating with an imputed user or imputed item only, we need to utilize the feature matrices of the imputed users or imputed items generated after the WNMTF is applied to the imputed rating matrix R'.

Furthermore, some experiments, for example in Chapter 2, show that the best prediction results are for rating values that are around the average of the imputed rating values. Therefore, the prediction results of rating values that are not close to the average of the imputed rating values are worse than others regardless of the percentage of the rating value in the test set; the percentage of the rating value in this test set does not have as significant of an effect as the average of the imputed rating values. In this chapter, we extend the experiment to study the effect of the the average of the rating values on the accuracy since the imputed ratings are considered as a subset of the all ratings in the training set.

We propose a method to conduct selective imputation that fuse the factored original rating matrix and the factored imputed rating matrix into one recommendation system. Our proposed method is capable of predicting the ratings by utilizing the feature matrix of the original users ratings and the original items ratings or substituting them with either the feature matrix of the imputed user or feature matrix of imputed items, or both in order to limit the error that may be introduced from the imputed ratings. The selected feature matrices in the proposed method could be identical for all predicted ratings or not: for each predicted rating, the U and V rows could be selected either from feature matrices of imputed rating matrix R or feature matrices of imputed rating matrix R'. The prediction results of rating values that are not close to the average of the rating values could be enhanced by utilizing the proposed method.

5.2 Proposed Method

In the proposed method, we aim to utilize the imputed ratings in a different way than the previously proposed methods in Chapters 2, 3, and 4. We propose a new method that calculates the predicted ratings in four different ways that utilize the feature matrices of the users and items either before or after the imputation to generate predicted ratings. Basically, the matrix factorization, i.e., WNMTF method, is performed twice to generate the feature matrices of the users and items. The first performance is with the original rating matrix R, and the second is with the imputed rating matrix R' in which all available imputed ratings for all users are imputed. The imputed ratings are obtained using the trust network as we proposed in Chapter 3 with the All-Users Imputation case. Equation (5.1) shows the outputs of the matrix factorization with the rating matrix R, and Equation (5.2) shows the outputs of the matrix factorization with the imputed rating matrix R'.

$$R_{m\times n} \approx U_{r_{m\times k}} \cdot S_{r_{k\times l}} \cdot V^T_{r_{n\times l}} \tag{5.1}$$

$$R'_{m\times n} \approx U_{t_{m\times k}} \cdot S_{t_{k\times l}} \cdot V^T_{t_{n\times l}} \tag{5.2}$$

The outputs of the matrix factorization are six matrices; four of them are feature matrices, two of them describe the users: U_r and U_t such that U_r holds the features of the users based on the users' original ratings and U_t holds the features of the users based on the original and all available imputed ratings of all users. In addition, there are two features matrices that describe the items: V_r and V_t in which V_r holds the features of the items using the items original ratings, and U_t holds the features of the items using the original and all available imputed ratings of the items. It is important to point out that even though the users are imputed basically in the proposed method, the items are imputed as a consequence of the user imputation.

By utilizing U_r, U_t, V_r, and V_t feature matrices from Equations (5.1) and (5.2), we can calculate the predicted ratings in four different ways. The first way is when no imputation method is involved either with users or items feature matrices generation (Equation (5.3)); or both users and items feature matrices result from the imputed rating matrix, R', as we see in Equation (5.6). The third and fourth ways are when one feature matrix (either of the users or the items) results from the imputed rating matrix R' and another feature matrix is not, as we see in Equations (5.4) and (5.5).

$$X_1 = U_r \cdot S_r \cdot V_r^T \tag{5.3}$$

$$X_2 = U_r \cdot (S_r + S_t)/2 \cdot V_t^T \tag{5.4}$$

$$X_3 = U_t \cdot (S_r + S_t)/2 \cdot V_r^T \tag{5.5}$$

$$X_4 = U_t \cdot S_t \cdot V_t^T \tag{5.6}$$

where each X holds the predicted ratings for the whole dataset. For each training rating and test rating, there are four predicted ratings, which are called sub-predicted ratings.

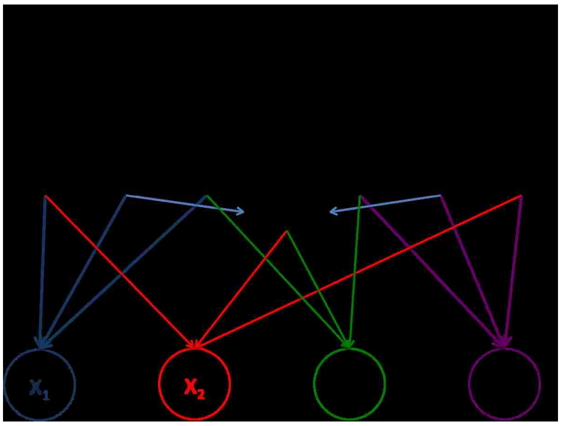


Figure 5.1: The classes of the sub-predicted ratings with the source category.

We classify the sub-predicted ratings into two different categories. The first category is based on the source of the sub-predicted ratings. There are four classes in this category. The first class is X_1 when the sub-predicted ratings result from U_r and V_r. The second is when the sub-predicted ratings result from U_r and V_t, which is called X_2 class. The third class is X_3 in which the sub-predicted ratings result from U_t and V_r, and the last class is X_4 when U_t and V_t are used to calculate the sub-predicted ratings. We call this category the *source category* of the sub-predicted ratings and Equations (5.3), (5.4), (5.5) and (5.6) represent this category. In addition, Figure 5.1 illustrates the four classes of the source category, as well.

The second category is based on the value of the sub-predicted ratings, $X1$, $X2$, $X3$,, and $X4$, compared to each other for each training and test rating. This category is named the *value category.* There are three classes under this category: the maximum value of sub-predicted ratings, the minimum value of non-zero sub-predicted ratings, and the "in-between" maximum and minimum value. Each sub-predicted rating of each training or test rating is assigned to one of these three classes. However, there are some cases that the values of all sub-predicted ratings are the same, e.g. when the rating is unpredictable, $X1 = X2 = X3 = X4 = 0$, which may belong to New-Users or New-Items. In this case, all the sub-predicted ratings for a specific training or test rating are classified as "same" class.

5.2.1 Objective Function

In our proposed method, we factor two rating matrices. The first matrix is the rating matrix R in which R represents the original ratings that are done by the users. The objective function of Weighted Nonnegative Matrix Tri-Factorization (WNMTF) with the original rating matrix R (Equation (5.1)) is as follows,

$$f(R, W, U_r, S_r, V_r) = min_{U_r \geq 0, S_r \geq 0, V_r \geq 0} \| W \circ (R - U_r S_r V_r^T) \|_F^2 \tag{5.7}$$

where $\circ$ is the element-wise multiplication. The weight matrix $W \in \mathbb{R}^{m \times n}$ is defined in Equation (3.6) based on the R.

The second matrix factored by WNMTF is the imputed rating matrix R'. As mentioned in Chapter 3, the rating matrix R is imputed with all available imputed ratings to form the imputed rating matrix R', which is defined in Chapter 3, Equation (3.8). The objective function of Weighted Nonnegative Matrix Tri-Factorization (WNMTF) with the imputed rating matrix R', Equation (5.2), is as follows,

$$f(R', W', U_t, S_t, V_t) = min_{U_t \geq 0, S_t \geq 0, V_t \geq 0} \| W' \circ (R' - U_t S_t V_t^T) \|_F^2 \tag{5.8}$$

where W' is defined in Chapter 3, Equation (2.3), based on R'. We name the proposed method Trust-Dual-WNMTF.

5.2.2 Update Formula

The update formulae for the objective function 5.7 are as follows [75],

$$U_{r_{ij}} \leftarrow U_{r_{ij}} \cdot \frac{[(W \circ R) V_r S_r^T]_{ij}}{\{[W \circ (U_r S_r V_r^T)] V_r S_r^T\}_{ij}} \tag{5.9}$$

$$V_{r_{ij}} \leftarrow V_{r_{ij}} \cdot \frac{[(W \circ R)^T U_r S_r]_{ij}}{\{[W \circ (U_r S_r V_r^T)]^T U_r S_r\}_{ij}} \tag{5.10}$$

$$S_{r_{ij}} \leftarrow S_{r_{ij}} \cdot \frac{[U_r^T (W \circ R) V_r]_{ij}}{\{U_r^T [W \circ (U_r S_r V_r^T)] V_r\}_{ij}} \tag{5.11}$$

On the other hand, the update formulae for the objective function 5.8 are as follows [75],

$$U_{t_{ij}} \leftarrow U_{t_{ij}} \cdot \frac{[(W' \circ R')V_t S_t^T]_{ij}}{\{[W' \circ (U_t S_t V_t^T)]V_t S_t^T\}_{ij}} \tag{5.12}$$

$$V_{t_{ij}} \leftarrow V_{t_{ij}} \cdot \frac{[(W' \circ R')^T U_t S_t]_{ij}}{\{[W' \circ (U_t S_t V_t^T)]^T U_t S_t\}_{ij}} \tag{5.13}$$

$$S_{t_{ij}} \leftarrow S_{t_{ij}} \cdot \frac{[U_t^T (W' \circ R') V_t]_{ij}}{\{U_t^T [W' \circ (U_t S_t V_t^T)] V_t\}_{ij}} \tag{5.14}$$

The time complexity of Trust-Dual-WNMTF is identical to the time complexity of Aux-New-Items-NMF in Chapter 2.

5.2.3 Convergence Analysis

The convergence proof of the derived update formulas is the same as Section 3.2.3 in Chapter 3.

5.2.4 Detailed Algorithm

In this section, we present the Trust-Dual-WNMTF algorithm. There are two phases in the proposed method. The first phase is the training phase in which the sub-predicted ratings are generated. Algorithm 5.1 depicts the steps of performing the training phase of the Trust-Dual-WNMTF method. We perform the matrix factorization twice, each in a separate performance. The first performance is with the original rating matrix R and the second is with the imputed rating matrix R'. The imputed rating matrix R' are generated using Algorithm 3.1 when the imputation case *Case* is All-Users Imputation. However, it may take hundreds or thousands of iterations to converge to a local minimum. Thus, in addition to the objective function criterion, an additional stop criterion (the maximum iteration count) is set in the algorithm. In collaborative filtering, this value varies from $10 \sim 100$, which can produce good results.

Algorithm 5.1 Trust-Dual-WNMTF - Training Phase

```
Require:
  User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
  Imputed User-Item rating matrix: $R' \in \mathbb{R}^{m \times n}$;
  Column dimension of $U : k$;
  Column dimension of $V : l$;
  Number of maximum iterations: $MaxIter$;
Ensure:
  $X_1, X_2, X_3$, and $X_4$;

Initialize $U, V$, and $S$ with random values
Set $U_r = U, V_r = V$, and $S_r = S$
Build weight matrix, $W$ by Eq. (3.6)
Set $iteration = 1$ and $stop = false$;
while $(iteration < MaxIter)$ $and$ $(stop == false)$ do
    $U_{r_{ij}} \leftarrow U_{r_{ij}} \cdot \frac{[(W \circ R) V_r S_r^T]_{ij}}{\{[W \circ (U_r S_r V_r^T)] V_r S_r^T\}_{ij}}$
    $V_{r_{ij}} \leftarrow V_{r_{ij}} \cdot \frac{[(W \circ R)^T U_r S_r]_{ij}}{\{[W \circ (U_r S_r V_r^T)]^T U_r S_r\}_{ij}}$
    $S_{r_{ij}} \leftarrow S_{r_{ij}} \cdot \frac{[U_r^T (W \circ R) V_r]_{ij}}{\{U_r^T [W \circ (U_r S_r V_r^T)] V_r\}_{ij}}$
    $L_r \leftarrow \|W \circ (R - U_r S_r V_r)^T)\|_F^2$
    if $L_r$ increases in this iteration then
        $stop = true$;
        Restore $U_r, V_r$, and $S_r$ to their values in last iteration.
    end if
end while
Set $U_t = U, V_t = V$, and $S_t = S$
Build weight matrix, $W'$ by Eq. (2.3)
Set $iteration = 1$ and $stop = false$;
while $(iteration < MaxIter)$ $and$ $(stop == false)$ do
    $U_{t_{ij}} \leftarrow U_{t_{ij}} \cdot \frac{[(W' \circ R') V_t S_t^T]_{ij}}{\{[W' \circ (U_t S_t V_t^T)] V_t S_t^T\}_{ij}}$
    $V_{t_{ij}} \leftarrow V_{t_{ij}} \cdot \frac{[(W' \circ R')^T U_t S_t]_{ij}}{\{[W' \circ (U_t S_t V_t^T)]^T U_t S_t\}_{ij}}$
    $S_{t_{ij}} \leftarrow S_{t_{ij}} \cdot \frac{[U_t^T (W' \circ R') V_t]_{ij}}{\{U_t^T [W' \circ (U_t S_t V_t^T)] V_t\}_{ij}}$
    $L_t \leftarrow \|W' \circ (R' - U_t S_t V_t)^T)\|_F^2$
    if $L_t$ increases in this iteration then
        $stop = true$
        Restore $U_t, V_t$, and $S_t$ to their values in the last iteration
    end if
end while
$X_1 = U_r \cdot S_r \cdot V_r^T$
$X_2 = U_r \cdot \frac{(S_r + S_t)}{2} \cdot V_t^T$
$X_3 = U_t \cdot \frac{(S_r + S_t)}{2} \cdot V_r^T$
$X_4 = U_t \cdot S_t \cdot V_t^T$
Return $X_1, X_2, X_3$, and $X_4$.
```

The second phase of the Trust-Dual-WNMTF algorithm is the test phase in which the final predicted ratings are calculated. We proposed several methods to calculate the final predicted ratings, R'', using the sub-predicted ratings. The first method is by simply calculating the final predicted ratings as the average of non-zero sub-predicted ratings, which is called Average Source, as we see in

Algorithm 5.2, such that,

$$r''_{ij} = \frac{\sum_{c=1}^{4} x_{c_{ij}}}{\sum_{c:x_c \neq 0} 1} \tag{5.15}$$

where $r''_{ij} \in R''$.

Algorithm 5.2 Trust-Dual-WNMTF - Test Phase - Average Source Method

Require:
X_1, X_2, X_3, and X_4;
User-Item test rating matrix: $R_{test} \in \mathbb{R}^{m \times n}$;
Ensure:
Final predicted ratings matrix: R'';

```
for each User $i$ do
    for each Item $j$ do
        if $r_{test_{ij}}$ in $R_{test} \neq 0$ then
            $nonZeroValueCount = \sum_{c=1}^{4} (x_{c_{ij}} \neq 0)$
            $r''_{ij} = \frac{\sum_{c=1}^{4} X_{c_{ij}}}{nonZeroValueCount}$
        end if
    end for
end for
Return $R''$.
```

Algorithm 5.3 presents the second method, which is based on the source category. After the matrix factorization is performed on R and R', we calculate the ratio of each class, δ_c, of the source category that holds the best predicted rating among sub-predicted ratings, which results in the lowest error of the original ratings in the training set. Then, these ratios are used to calculate the final predicted rating of each test rating,

$$r''_{ij} = \frac{\sum_{c=1}^{4} \delta_c \cdot x_{c_{ij}}}{\sum_{c:x_c \neq 0} 1} \tag{5.16}$$

where $r''_{ij} \in R''$.

However, when a New-User is imputed, the sub-predicted ratings X_1 and X_2 are zeros, which means unpredictable ratings, but not X_3 and X_4. To avoid the impact of the unpredictable ratings, we calculate the average of the non-zero sub-predicted ratings X_3 and X_4.

Algorithm 5.3 Trust-Dual-WNMTF - Test Phase - Ratio Source Method

```
Require:
  $X_1, X_2, X3$, and $X_4$;
  User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
  User-Item test rating matrix: $R_{test} \in \mathbb{R}^{m \times n}$;
Ensure:
  Final predicted ratings matrix: $R''$;

for each User $i$ do
    for each Item $j$ do
        if $r_{ij}$ in $R \neq 0$ then
            for each sub-predicted rating $c$ do
                $MAE\_x_c = abs(x_{c_{ij}} - r_{ij})$
            end for
            $minMAE = min(MAE\_x_1, MAE\_x_2, MAE\_x_3, MAE\_x_4)$
            if $minMAE == MAE\_x_1$ then
                $counter\_x_1 = counter\_x_1 + 1$
            else if $minMAE == MAE\_x_2$ then
                $counter\_x_2 = counter\_x_2 + 1$
            else if $minMAE == MAE\_x_3$ then
                $counter\_x_3 = counter\_x_3 + 1$
            else if $minMAE == MAE\_x_4$ then
                $counter\_x_4 = counter\_x_4 + 1$
            end if
        end if
    end for
end for
for $c = 1$ to $4$ do
    $Ratio\_x_c = counter\_x_c / |$ training set rating $|$
end for
for each User $i$ do
    for each Item $j$ do
        if $r_{test_{ij}}$ in $R_{test} \neq 0$ then
            $nonZeroValueCount = \sum_{c=1}^{4} (x_{c_{ij}} \neq 0)$
            if $nonZeroValueCount == 4$ then
                $r''_{ij} = \sum_{c=1}^{4} Ratio\_x_c * x_{c_{ij}}$
            else
                $r''_{ij} = \frac{(x3_{i,j} + x4_{i,j})}{2}$
            end if
        end if
    end for
end for
Return $R''$.
```

The third method is similar to the second, but we use the value category instead of the source category, as we see in Algorithm 5.4.

Algorithm 5.4 Trust-Dual-WNMTF - Test Phase - Ratio Value Method

```
Require:
  $X_1, X_2, X_3$, and $X_4$;
  User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
  User-Item test rating matrix: $R_{test} \in \mathbb{R}^{m \times n}$;
Ensure:
  Final predicted ratings matrix: $R''$;

for each User $i$ do
    for each Item $j$ do
        if  $r_{ij}$ in $R \neq 0$ then
            for each sub-predicted rating $c$ do
                $MAE\_x_c = abs(x_{c_{ij}} - r_{ij})$
            end for
            $[minMAE, minMAEIndex] = \min_{1 \leq c \leq 4} \{MAE\_x_c\}$
            $[MaxValue, MaxIndex] = \max_{1 \leq c \leq 4} \{x_{c_{i,j}}\}$
            $[MinValue, MinIndex] = \min_{1 \leq c \leq 4, x_c > 0} \{x_{c_{i,j}}\}$
            if $minMAEIndex == MaxIndex$ then
                $MaxValueCounter = MaxValueCounter + 1$
            else if $minMAEIndex == MinIndex$ then
                $MinValueCounter = MinValueCounter + 1$
            else
                $InBtnValueCounter = InBtnValueCounter + 1$
            end if
        end if
    end for
end for
$MaxValueRatio = MaxValueCounter/|$ training set rating $|$
$MinValueRatio = MinValueCounter/|$ training set rating $|$
$InBtnValueRatio = InBtnValueCounter/|$ training set rating $|$
for each User $i$ do
    for each Item $j$ do
        if $r_{test_{ij}}$ in $R_{test} \neq 0$ then
            $nonZeroValueCount = \sum_{c=1}^{4} (X_{c_{ij}} \neq 0)$
            $MaxValue = max(x_{1_{ij}}, x_{2_{ij}}, x_{3_{ij}}, x_{4_{ij}})$
            if nonZeroValueCount==4 then
                $MinValue = min(x_{1_{i,j}}, x_{2_{i,j}}, x_{3_{i,j}}, x_{4_{i,j}})$
            else if nonZeroValueCount==2 then
                $MinValue = min(x_{3_{i,j}}, x_{4_{i,j}})$
            end if
            if $nonZeroValueCount == 4$ then
                $InBtnValue = (X_{1_{i,j}} + X_{2_{i,j}} + X_{3_{i,j}} + X_{4_{i,j}} -$
                $MaxValue - MinValue)/2$
            else if $nonZeroValueCount == 2$ then
```

```
                $InBtnValue = (MaxValue + MinValue)/2$
            else if $nonZeroValueCount == 0$ then
                $InBtnValue = 0$
            end if
            if $nonZeroValueCount > 0$ then
                $r''_{ij} = MaxValueRatio * MaxValue +$
                $MinValueRatio * MinValue +$
                $InBtnValueRatio * InBtnValue$
            else
                $r''_{ij} = 0$
            end if
        end if
    end for
end for
Return $R''$.
```

In the last two methods, we set the final predicted ratings to only one value of sub-predicted ratings. We test this method with the value category to select either the maximum or minimum value of the sub-predicted ratings as final predicted ratings, as we see in Algorithm 5.5 and Algorithm 5.6, respectively, such that,

$$r''_{ij} = \max_{1 \le c \le 4} \{x_{c_{ij}}\} \tag{5.17}$$

$$r''_{ij} = \min_{1 \le c \le 4, x_c > 0} \{x_{c_{ij}}\} \tag{5.18}$$

where $r''_{ij} \in R''$.

Algorithm 5.5 Trust-Dual-WNMTF - Test Phase - Max Value Method

```
Require:
  $X_1, X_2, X_3$, and $X_4$;
  User-Item test rating matrix: $R_{test} \in \mathbb{R}^{m \times n}$;
Ensure:
  Final predicted ratings matrix: $R''$;

for each User $i$ do
    for each Item $j$ do
        if $r_{test_{ij}}$ in $R_{test} \neq 0$ then
            $r''_{ij} = max(x_{1_{i,j}}, x_{2_{i,j}}, x_{3_{i,j}}, x_{4_{i,j}})$
        end if
    end for
end for
Return $R''$.
```

Algorithm 5.6 Trust-Dual-WNMTF - Test Phase - Min Value Method

```
Require:
  $X_1, X_2, X_3$, and $X_4$;
  User-Item test rating matrix: $R_{test} \in \mathbb{R}^{m \times n}$;
Ensure:
  Final predicted ratings matrix: $R''$;

for each User $i$ do
    for each Item $j$ do
        if $r_{test_{ij}}$ in $R_{test} \neq 0$ then
            $nonZeroValueCount = \sum_{c=1}^{4} (x_{c_{ij}} \neq 0)$
            if $nonZeroValueCount == 4$ then
                $r''_{ij} = min(x_{1_{ij}}, x_{2_{ij}}, x_{3_{ij}}, x_{4_{ij}})$
            else if $nonZeroValueCount == 2$ then
                $r''_{ij} = min(x_{3_{ij}}, x_{4_{ij}})$
            else
                $r''_{ij} = 0$
            end if
        end if
    end for
end for
Return $R''$.
```

5.2.5 Complexity

The computational complexity of Trust-Dual-WNMTF (Training Phase) is similar to the computational complexity of Trust-WNMTF in Section 3.2.5.

5.3 Experimental Study

Table 5.1: The MAE for the whole dataset and each user group with WNMTF, Trust-WNMTF, Trust-WNMTF++, and the five strategies of the proposed method.

Methods		All-Users	New-Users	Cold-Start Users	Heavy-Rated Users
		Ciao			
Previous	WNMTF	0.8237	4.4118	0.8345	0.7452
	Trust-WNMTF All-User	0.8305	1.4235	0.8399	0.7715
	Trust-WNMTF New-User	0.8224	1.3615	0.8345	0.7453
	Trust-WNMTF++ Trustee	0.8025	1.3999	0.8118	0.7438
Proposed	Average Source	0.8118	1.3839	0.8210	0.7510
	Ratio Source	0.8122	1.3839	0.8218	0.7494
	Ratio Value	0.8023	1.3739	0.8115	0.7418
	Max Value	**0.7738**	**1.3451**	**0.7832**	**0.7118**
	Min Value	0.8980	1.4240	0.9101	0.8190
		CiaoDVD			
Previous	WNMTF	1.6503	4.3433	1.2397	1.0612
	Trust-WNMTF All-User	1.6721	4.2832	1.2722	1.1122
	Trust-WNMTF New-User	1.6462	4.2830	1.2442	1.0689
	Trust-WNMTF++ Trustee	1.6348	**4.2824**	1.2302	1.0606
Proposed	Average Source	1.6509	4.2841	1.2469	1.0925
	Ratio Source	1.6413	4.2841	1.2364	1.0775
	Ratio Value	1.6427	4.2839	1.2373	1.0834
	Max Value	**1.6237**	4.2827	**1.2159**	**1.0585**
	Min Value	1.7506	4.2859	1.3612	1.2152
		Epinions			
Previous	WNMTF	1.0816	3.9203	1.0770	0.9316
	Trust-WNMTF All-User	1.0751	1.9541	1.0888	0.9769
	Trust-WNMTF New-User	1.0760	1.9495	1.0964	0.9543
	Trust-WNMTF++ CSI	1.0372	1.9297	1.0529	0.9311
Proposed	Average Source	1.0498	1.9462	1.0647	0.9443
	Ratio Source	1.0479	1.9462	1.0639	0.9380
	Ratio Value	1.0424	1.9436	1.0570	0.9373
	Max Value	**1.0161**	**1.9262**	**1.0318**	**0.9068**
	Min Value	1.1411	1.9731	1.1596	1.0262
		FilmTrust			
Previous	WNMTF	0.7288	3.3677	0.7326	0.6455
	Trust-WNMTF All-User	0.7439	2.7780	0.7487	0.6679
	Trust-WNMTF New-User	0.7269	2.7735	0.7324	0.6463
	Trust-WNMTF++ CSI	**0.7200**	**2.7639**	**0.7242**	0.6478
Proposed	Average Source	0.7232	2.7687	0.7270	0.6524
	Ratio Source	0.7214	2.7687	0.7254	0.6498
	Ratio Value	0.7211	2.7677	0.7249	0.6505
	Max Value	0.7204	2.7662	0.7255	**0.6418**
	Min Value	0.7608	2.7726	0.7665	0.6793

In the experimental study of this chapter, the proposed method is compared with WNMTF 3.7, both proposed cases of the Trust-WNMTF approach in Chapter 3: All-Users and New-Users, and with the best case for each dataset that results in the lowest MAE of the Trust-WNMTF++ method from Chapter 4. The parameters are set the same as the parameter setting in Chapter 3.

Table 5.1 presents the results of the proposed method with five different strategies that calculate the final predicted ratings by using the four sub-predicted ratings. Firstly, the Min Value strategy is excluded from the comparison because the Min Value strategy results in the worst accuracy among all previous and proposed methods with all datasets.

On the other hand, the results with the other four proposed strategies are better than the WNMTF method. This indicates that the proposed method is able to utilize the imputation to enhance accuracy. Furthermore, the proposed method is better to utilize the imputed ratings than the previous method, Trust-WNMTF All-Users, to improve Cold-Start-Users and Heavy-Rater-Users groups accuracy results.

In addition, results in the four proposed strategies are better than the Trust-WNMTF New-User method except for one case, which is the Average Source with the CiaoDVD dataset. Among the five proposed strategies, the Min Value results in the worst rating prediction; then, the Average Source strategy seems to be the next worst results, except for the Ciao dataset.

However, only the Max Value strategy surpasses the Trust-WNMTF++ method with all datasets except FilmTrust. Figure 5.2 illustrates the percentage of each class of the value category that results in the best predicted ratings for all test ratings. With all datasets, more than half of the best sub-predicted ratings are the maximum/same value. However, the FilmTrust dataset has the highest percentage of the best sub-predicted ratings that are the minimum value, which leads to slightly worse results compared to the Trust-WNMTF++ results. In addition, the proposed method improves the results of the Heavy-Rater-Users in addition to New-Users and Cold-Start-Users, as we see in Table 5.1.

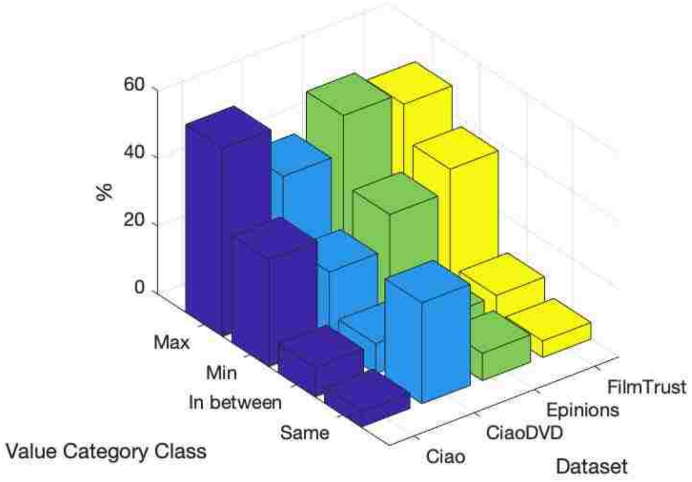


Figure 5.2: The percentage of each value category class that results in the best predicted ratings for all test ratings.

5.3.1 Rating Value Vs. Rating Value Average

Because we perform our experiment in a 5-fold cross-validation approach, the percentage of the rating values in the training set and test set are identical where the training and test set together form the rating matrix R. From Tables 4.1, 5.2, and 5.3, we observe the relationship between the average of the rating values in the training set and the rating value that has the best accuracy results in the Non-Imputation method, i.e., WNMTF. Unexpectedly, the percentage of the rating values in the training set does not have a significant effect on the rating value that has the best accuracy results. For example, despite 50% of the rating value is 5 in Ciao, the rating value 4 has the lowest MAE among other rating values, which is the closest rating value to the rating value average (Table 4.1). Consequently, the rating value that is the largest percentage in the test set does not have the lowest MAE, which leads to high MAE of the whole dataset.

Our proposed approach - Max Value - handles this issue where the rating value that is the largest percentage in the test set has the most improvement percentage in MAE with all datasets. We suppose that the prediction accuracy of rating value 5 is more significant than other rating values due to the fact that the fundamental objective of the recommendation system is to recommend the items most relevant to the users' taste.

Table 5.2: The percentage of each rating value in the training/test set and their MAEs with several methods (1).

Rating Value	1	2	3	4	5
			Ciao		
% in R	4.48%	5.12%	11.1%	29.66%	**49.63%**
WNMTF	2.1353	1.6052	0.9744	**0.6314**	**0.7058**
Max Value	2.4366	1.9230	1.2151	0.6632	**0.4723**
Min Value	1.9126	1.3925	**0.8326**	0.5981	0.9491
			CiaoDVD		
% in R	3.66%	6.43%	13.79%	31.04%	**45.08%**
WNMTF	1.8140	1.5839	**1.3995**	**1.4773**	1.8432
Max Value	2.0943	1.8279	1.5979	1.5209	**1.6350**
Min Value	1.5657	**1.3608**	1.2635	1.5123	2.1342
			Epinions		
% in R	7.91%	9.09%	12.62%	28.44%	**41.94%**
WNMTF	1.9271	1.3449	**0.9518**	**0.8747**	1.0443
Max Value	2.2401	1.6288	1.0988	0.8250	**0.7573**
Min Value	1.6740	**1.1166**	0.8222	0.8953	1.3085

Table 5.3: The percentage of each rating value in the training/test set and their MAEs with several methods (2).

Rating Value	0.5	1	1.5	2	2.5	3	3.5	4
				FilmTrust				
% in R	2.99%	3.21%	4.52%	8.76%	12.38%	22.18%	20.11%	**25.84%**
WNMTF	1.6942	1.5312	1.1334	0.7983	**0.5793**	**0.4988**	**0.5835**	0.8053
Max Value	1.8791	1.7363	1.3135	0.9515	0.6573	0.4930	0.4966	**0.6777**
Min Value	1.5857	1.4230	1.0348	**0.7133**	0.5098	0.4803	0.6803	0.9752

On the other hand, with Min Value method, the results of the rating values that are lower than the average of the rating values have the most improved percentage in the accuracy. Our proposed method enhances the results of the rating values that are not close to the average of the rating values either higher or lower. Because our datasets have more rating values that are higher than the average of the rating values, the Max Value method outperforms the other methods. However, we expect that the Min Value method benefits the systems that focus on predicting the ratings of the low rating values or, in other words, the items that the users do not like.

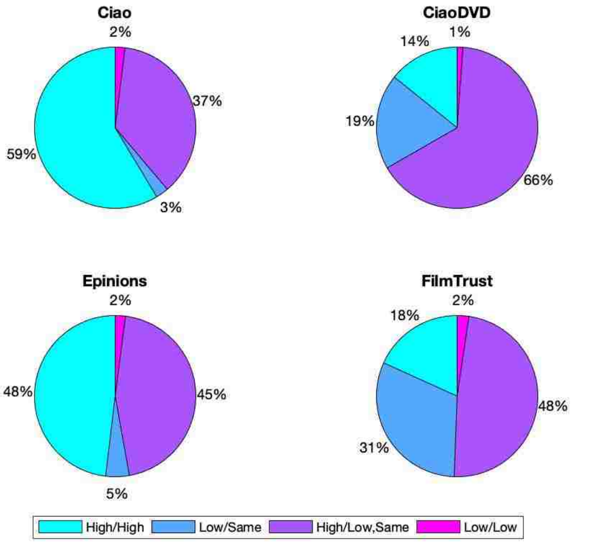


Figure 5.3: The percentage of predicted ratings that belong to one of the four combinations of rating value average categories (High, Low, or Same) for the user/item with Max Value strategy.

5.3.2 Influence of Increase/Decrease the Users/Items Rating Average

The value rating averages of the users and items may increase or decrease after the imputation. We conclude in Section 5.3.1 that the average of the rating values in the training set has a strong influence on the accuracy. In our experimental study, the rating value average for the user and item could be classified into High, Low, or Same based on the comparison of the rating value average before and after the imputation. In some cases, e.g., there is no imputation at all, the rating value average remains the same.

For each predicted rating of the test set p_{ij}, the average of rating values in the training set for user u_i and item e_j that hold p_{ij} is surveyed. The predicted rating p_{ij} equals the maximum value among the four sub-predicted ratings, i.e., Max Value strategy of the proposed method. As shown in Figure 5.3, there are four different combinations of the rating value average classes. The first combination is High/High category in which both rating value averages of the user and the item are the highest either before or after the imputation. On the contrary, in the Low/Low category, the rating value averages of the user and item are the lowest either before or after the imputation. There are other cases in between the High/High and Low/Low categories. If the rating value average of either the user or the item is the highest, but the other either the lowest or same rating value average, then the predicted test rating belongs to the ”High/Low, Same” category. The last category is Low/Same where the rating value average of either user or item is the lowest and the same for the other.

As we see in Figure 5.3, we observe that approximately half of the predicted ratings in Ciao and Epinions belong to the users and items that have the highest rating value averages, High/High category. In addition, these two datasets record the highest percentage of the improvement compared to the previous methods, as shown in Table 5.1. However, with other datasets, CiaoDVD and FilmTrust, around half of the predicted ratings belong to the High/Low, Same category. Furthermore, the percentages of accuracy improvement for CiaoDVD and FilmTrust are not as good as the others.

5.3.3 Summary of Results

To conclude, the Max Value strategy results in the lowest MAE when compared to other proposed strategies, which corroborates the concept that increasing the average of the rating values in the training set either for the users or the items leads to improving the prediction accuracy. In addition, the improvement in the prediction is obvious when the maximum value of the sub-predicted ratings belongs to the highest rating value averages for the users and items either before or after the imputation.

5.4 Summary

In this chapter, we proposed a method to conduct the selective imputation method that fuses the factored original rating matrix and the factored imputed rating matrix into one system. Our proposed method is capable of predicting the rating by utilizing either the imputed users or imputed items, or both in order to limit the error that may be introduced from the imputed ratings. The results show that our proposed method surpasses Trust-WNMTF++ in Chapter 4. Furthermore, the Max Value strategy surpasses other proposed strategies. The prediction results of rating values that are not close to the average of the rating values could be enhanced by utilizing the proposed method either through the Max Value or the Min Value strategy.

6 Comparison Between Selected Methods of Imputation-Based Recommendation Systems

Even though the proposed method in Chapter 5 improves the accuracy results of the New-Users and Cold-Start-Users, the New-Items cannot be introduced to the users yet. In this chapter, the New-Item imputation method in Chapter 2 is integrated with the proposed methods in Chapter 5. In addition, we compare the proposed method with two popular imputation-based CF methods, AdaM [50] and IMULT [48].

6.1 Proposed Method

We propose an approach that integrates the New-Item imputation method proposed in Chapter 2 into the Trust-Dual-WNMTF method in Chapter 5. The imputed ratings for New-items are inserted in the imputed rating matrix R', which is utilized with Trust-Dual-WNMTF in Chapter 5. The source of the New-Items imputed ratings is the same as in Chapter 2. However, in this chapter, all available imputed ratings for New-items are utilized instead of a limited number of them, which we performed in Chapter 2. In addition, the Average case is used to calculate the imputed ratings from the source ratings. After that, the Max Value method, as shown in Algorithm 5.5, is applied to calculate the final predicted ratings. This method is named Trust-New-Items-Dual-WNMTF.

6.2 Background

Two popular imputation-based collaborative filtering methods are compared to our proposed method, Trust-New-Items-Dual-WNMTF, in terms of the accuracy: AdaM and IMULT. Our proposed method, Adam, and IMULT are similar in utilizing the imputation to handle the rating matrix sparsity and improve the recommendation accuracy. On the contrary, AdaM and IMULT rely only on the rating information to perform the imputation whereas the trust information does not.

6.2.1 AdaM

The Adaptive-Maximum imputation method (AdaM) [50] is a neighborhood-based and imputation-based collaborative filtering method. Its basic idea is to identify an area to impute that can maximize the imputation advantage and minimize the imputation error. The imputation area is determined from both the user and the item perspectives in order to accomplish the maximum imputation. On the other hand, there is at least one real rating preserved for each item in the identified imputation area in order to reduce the imputation error.

From the user perspective, to predict the rating r_{as}, in which the active user is u_a and the active item is t_s, the imputation area is determined by two sides: the maximum set of possible neighbors related to u_a, which is called U_a, and the maximum set of possible items related to the active item t_s, which is called T_s. U_a and T_s are defined as follows

$$U_a = \{u_{a'} | r_{a's} \neq \emptyset\} \cup \{u_a\} \tag{6.1}$$

$$T_s = \{t_j | t_j \subset [S_a \cup S_{a'_1} \cup ... \cup S_{a'_j} \cup ... \cup S_{a'_l}]\} \tag{6.2}$$

where $S_{a'_j}$ is all the items that have been rated by the user $u_{a'_j} \in U_a$, and $l = |U_a|$.

Based on the subset of users in U_a and the subset of the items in T_s, the *max neighbourhood* for the active user u_a to predict the rating of item t_s is defined as:

$$N_{as} = \{r_{a'j} | u_{a'} \in U_a, t_j \in T_s\} \tag{6.3}$$

The max neighbourhood N_{as} considers a subset matrix from the rating matrix R. The rating value, i.e., $r_{a'j}$, in N_{as} can be either a missing rating or not. All the missing ratings in the matrix N_{as} are defined as the *key set* for the prediction p^u_{as}. Each missing rating $r_{a'j} \in N_{as}$, i.e., the *key set* ratings, is imputed with $r'_{a'j}$, such that:

$$r'_{a'j} = \bar{u}_{a'} + \frac{\sum_{u_x \in N_k(u_{a'})} sim(u_{a'}, u_x) \times (r_{xj} - \bar{u}_x)}{\sum_{u_x \in N_k(u_{a'})} sim(u_{a'}, u_x)} \tag{6.4}$$

where $sim(u_{a'}, u_x)$ is the PCC similarity [52] between $u_{a'}$ and u_x, $\bar{u}_{a'}$ is the average of the rating values of user $u_{a'}$, and k is the total number of the neighbors.

After all missing ratings in max neighbourhood N_{as} are imputed, AdaM predicts the rating p^u_{as} of r_{as} from the user perspective by utilizing both the imputed ratings and the observing ratings in N_{as} so that

$$p^u_{as} = \bar{u}_a + \frac{\sum_{u_x \in N_k(u_a)} sim'(u_a, u_x) \times (r_{xs} - \bar{u}_x)}{\sum_{u_x \in N_k(u_a)} sim'(u_a, u_x)} \tag{6.5}$$

where sim' is defined as follows,

$$sim'(u_a, u_x) = \frac{\sum_{t_j \in T_s} (r_{aj} - \bar{u}_a)(r_{xj} - \bar{u}_x)}{\sqrt{\sum_{t_j \in T_s} (r_{aj} - \bar{u}_a)^2 \sum_{t_j \in T_s} (r_{xj} - \bar{u}_x)^2}} \tag{6.6}$$

All steps should be performed from the item perspective to calculate the predicted rating p^e_{as} of r_{as}. More details can be found in [50]. The final predicted rating of r_{as} can be obtained as follows,

$$p_{as} = \lambda p^u_{as} + (1 - \lambda) p^e_{as} \tag{6.7}$$

where λ is a predefined parameter that determines the involved ratio of predictions from the user perspective and the item perspective.

Based on AdaM experiment results, AdaM significantly outperforms in terms of accuracy other related imputation-based methods, which include the default voting (Default Voting) method [3], the EMDP method [34], the SCBPCC method [79], AutAI-Fusion method [49], and two traditional collaborative filtering algorithms, the user-based CF (UPCC) and the item-based CF (IPCC), and one model-based algorithm, the Slope One algorithm [30].

6.2.2 Imputed MULT (IMULT)

Imputed MULT [48] is a model-based and imputation-based method that is based on the Multiplicative update rules (MULT) method. MULT [29] is one of the techniques that is used to solve the NMF problem, Equation (3.1), which considers a gradient descent-based approach with a special choice of learning stepsizes. On the other hand, the IMULT method in [48] utilizes the imputation as a pre-processing step before MULT is applied in order to alleviate the lack of ratings. The objective function of IMULT is as follows,

$$f(U,V) = \min \frac{1}{2}||P_{\Omega}(R - UV^T)||_F^2 + \frac{\delta}{2}||P_{\Psi}(R' - UV^T)||_F^2 \tag{6.8}$$

where $R' \in \mathbb{R}^{m\times n}$ is the imputed rating matrix that holds only the imputed ratings, δ is the learning rating of the imputed ratings where $0 < \delta \leq 1$, Ω is the set of ratings in R such that $r_{ij} > 0$, and $P_{\Omega}(.) : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^{m\times n}$ is defined as:

$$P_{\Omega}(x_{ij}) = \begin{cases} x_{ij} & \text{(ij)} \in \Omega \\ 0 & \text{otherwise} \end{cases}, \tag{6.9}$$

and Ψ is the set of the missing ratings in R, and $P_{\Psi}(.) : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^{m\times n}$ is defined as:

$$P_{\Psi}(x_{ij}) = \begin{cases} x_{ij} & \text{(ij)} \in \Psi \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

In addition, Equation (6.8) can be re-written as follows,

$$f(U,V) = \min \sum_{(ij)\in\Omega\cup\Psi} \frac{\varphi_{ij}}{2}(r^*_{ij} - \sum_{l=1}^{k} u_{il}v^T_{jl})^2 \tag{6.11}$$

where $\varphi_{ij} = \begin{cases} \delta & (ij) \in \Psi \\ 1 & (ij) \in \Omega \end{cases}$ and $r^*_{ij} = \begin{cases} r'_{ij} & (ij) \in \Psi \\ r_{ij} & (ij) \in \Omega \end{cases}$.

We can calculate the imputed ratings for the missing ratings in four different ways. The first is the mean-wise in which all the missing ratings in R are imputed with the average of all ratings in R, such that, $r' = \frac{\sum_{(ij)\in\Omega} r_{ij}}{|\Omega|}$. The second method is when the missing ratings for each item j are imputed with the average of the rating values of the item j (item-wise) where $r'_j = \frac{\sum_{(:j)\in\Omega} r_{ij}}{|(:j)\in\Omega|}$, in which $|(:j) \in \Omega|$ is the total number of the ratings for the item j. The next way is user-wise in which the missing ratings for each user i are imputed with the average of the rating values of the user i, such that $r'_i = \frac{\sum_{(i:)\in\Omega} r_{ij}}{|(i:)\in\Omega|}$, in which $|(i:) \in \Omega|$ is the total number of the ratings for the user i. The last way is hyper-wise in which the missing ratings are imputed with the linear combination of the user-wise and item-wise such that, $r'_{ij} = \alpha r'_i + (1-\alpha)r'_j$ where $0 \leq \alpha \leq 1$.

The IMULT method outperforms Multiplicative update rules (MULT) [29], Alternating Least Squares (ALS) [44], Stochastic Gradient Descent (SGD) [31], Regularization Stochastic Gradient Descent (RSGD) [69], and SVD++ [28] in terms of accuracy.

6.3 Experimental Study

In this section, we calculate MAE for the whole dataset, for each user group, and for each item group. FilmTrust is excluded from this experiment because the New-Items imputation method is not applicable due to absent item information.

The machine we use for the AdaM and IMULT methods is 95 Teraflops Dell C6220 Server, which consists of 16 cores in which 4 nodes per 2U chassis Dual Intel E5-2670 8 Core (Sandy Bridge) @ 2.6 GHz 2 sockets/node x 8 cores/socket; and 64 GB/node of 1600 MHz RAM 500 GB local (internal) SATA disk Linux OS (RHEL).

6.3.1 Parameter settings

Table 6.1: Parameter setup in AdaM and IMULT.

Dataset	λ	IMULT k
Ciao	1	3
CiaoDVD	0.8	2
Epinions	1	3

The parameters setting for our proposed method, Trust-New-Items-Dual-WNMTF, is shown in Table 2.3. However, $MaxImputedRatings$ is excluded in this chapter because all available imputed ratings for New-Items are utilized.

With the AdaM method, two parameters need to be set in advance. The first parameter is the total number of neighbors, k, which is set to 30 based on the AdaM experiment [50]. On the other hand, we run AdaM with several values of λ. Table 6.1 shows the best value of λ for each dataset that results in the best accuracy.

In the experiments for the IMULT method, the learning rate of imputed ratings, δ, is set at 0.1 based on the experiment in [48]. In addition, the initialized U and V are fixed for the proposed methods and IMULT as well. The maximum of iterations is set to 10, which is the same as our proposed methods. The rank k for U and V in IMULT is shown in Table 6.1 for each dataset. Furthermore, we run IMULT with all four imputation methods: mean-wise, user-wise, item-wise, and hyper-wise where α is set to 0.5 with hyper-wise IMULT method.

6.3.2 Results and Discussion

Table 6.2: The % of New-Users and New-Items ratings in the test set.

Dataset	New-Users	New-Items
Ciao	0.05%	1.58%
CiaoDVD	13.92%	13.87%
Epinions	1.29%	5.28%

The WNMTF method considers a baseline in which the MAE of New-Users and New-Items are in the worst case because their ratings cannot be predicted at all. On the other hand, integrating New-Items imputation into the Trust-Dual-WNMTF method (Chapter 5) does not increase the errors with all users group, as we see in Tables 5.1 and 6.3.

AdaM

The AdaM accuracy results, in general, are the worst compared to IMULT and Trust-New-Items-Dual-WNMTF with all datasets, as we see in Table 6.3. Furthermore, the AdaM MAE results are worse than the baseline method, WNMTF, which supports the assumption that model-based methods surpass neighborhood-based methods in terms of accuracy. It is important to mention that the imputation process in the AdaM method does not predict the ratings of New-Users and New-Items. However, based on the AdaM method, the New-Items ratings are predicted as the average of the user ratings with the user perspective case of AdaM. On the other hand, AdaM predicts the New-Users ratings based on the average of the item ratings with the item perspective case of AdaM. Consequently, the percentages of New-Users and New-Items ratings in the test set have a strong influence on the λ value that results in the lowest MAE for the whole dataset. Table 6.1 shows the best λ value that results in the lowest MAE.

For the Ciao and Epinions datasets, the percentage of New-Items ratings in the test set is higher than the percentage of New-Users, as shown in Table 6.2. In addition, the lowest MAE with AdaM is when the value of λ is set to 1. In

Table 6.3: The MAE results of the whole dataset and for each user and item group with the best setting of Trust-New-Items-Dual-WNMTF, AdaM, and IMULT methods.

Method	Pers.	All	New-	Cold-Start-	Heavy-Rate(r/d)
Ciao					
WNMTF	Users	0.8237	4.4118	0.8345	0.7452
	Items		4.1248	0.7865	0.7607
Trust-New-Items-Dual-WNMTF	Users	**0.7220**	**1.2636**	**0.7269**	**0.6890**
	Items		0.8916	**0.7326**	**0.7127**
AdaM	Users	1.5065	4.4118	1.4108	2.1009
	Items		**0.8444**	1.4666	1.5492
IMULT (Hyper-wise)	Users	0.7930	2.3414	0.7979	0.7556
	Items		2.1147	0.7511	0.7810
CiaoDVD					
WNMTF	Users	1.6503	4.3433	1.2397	1.0612
	Items		3.8668	1.2602	1.3075
Trust-New-Items-Dual-WNMTF	Users	1.2541	4.2659	**0.7773**	**0.7051**
	Items		1.1840	1.2365	1.2759
AdaM	Users	1.6505	3.5792	1.2828	1.6695
	Items		1.3760	1.4803	1.7684
IMULT (Mean-wise)	Users	**0.8388**	**0.8098**	0.8493	0.8144
	Items		**0.8832**	**0.8550**	**0.8217**
Epinions					
WNMTF	Users	1.0816	3.9203	1.0770	0.9316
	Items		3.9385	1.0287	0.9188
Trust-New-Items-Dual-WNMTF	Users	**0.8578**	**1.9073**	**0.8509**	**0.8202**
	Items		**0.8905**	0.8904	**0.8540**
AdaM	Users	1.2944	3.9203	1.2409	1.3274
	Items		0.9039	1.6637	1.2984
IMULT (Hyper-wise)	Users	0.9832	2.0771	0.9797	0.9316
	Items		2.0527	**0.8836**	0.9287

this case, all predicted ratings of New-Items are not zero, i.e., predictable, which reduces the MAE, but all New-Users ratings are unpredictable. In addition, the MAE of all user groups and all item groups with the AdaM method are the worst among all three methods except New-Items, where the best results of New-Items in Ciao is with AdaM, and the next best of New-Items in Epinions is with AdaM, as we see in Table 6.3. However, our proposed method cannot guarantee that all New-Items are imputed, which leads to unpredictable ratings of New-Items. In contrast, all New-Items ratings are predictable in AdaM when $\lambda > 0$.

On the other hand, the percentage of New-Users ratings in the CiaoDVD test set is slightly larger than the percentage of New-Items, as we see in Table 6.2, thus the best MAE result is when the λ is 0.8. When $0 < \lambda < 1$, both New-Users and New-Items ratings are predictable. Due to the fact that most New-Users cannot be imputed by our proposed method, Trust-New-Items-Dual-WNMTF, with the CiaoDVD dataset as we see in Table 4.7, the AdaM results of the New-Users is better, but our proposed method results in better results with New-Items. Otherwise, the results of other user and item groups are the worst with AdaM out of the three methods.

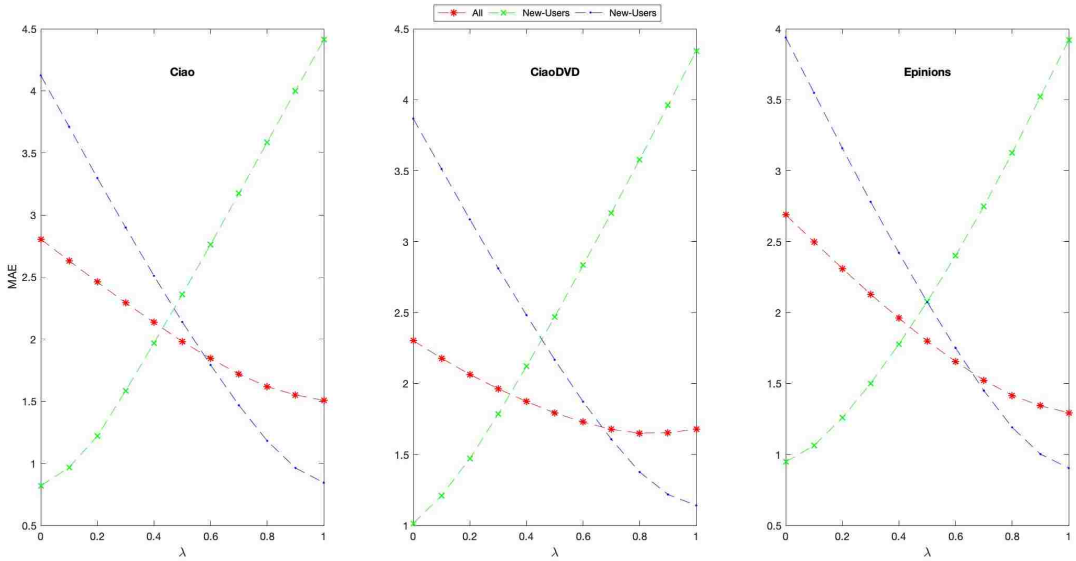


Figure 6.1: The MAE results for the whole dataset, New-Users, and New-Items with different values of λ of AdaM method.

Figure 6.1 shows the relationship between the MAE of New-Users, New-Items, the whole dataset in general, and λ value. By increasing λ value, the New-Users results get better, but results are worse for New-Items. This indicates that AdaM cannot simultaneously predict New-Users and New-Items ratings with the best accuracy.

In our opinion, one of the biggest downsides of AdaM is that the imputation is applied for each predicted rating in which each predicted rating has its own imputed ratings. In addition, the similarity between the users and items needs to be calculated twice. The first calculation is with the original ratings for the user and item to calculate the imputed ratings. The second is with the original and imputed ratings for the user and item to predict the rating. In general, the AdaM method is not suitable for large-scale datasets due to the intensive calculations for each predicted rating.

IMULT

As we see in the IMULT Algorithm in [48], for each rating $r_{ij} \in R$ and $r'_{ij} \in R'$, only the U row that corresponds to the user i - i.e. u_i - and the V row that corresponds to the item j - i.e. v_j - are updated. This is unlike the updating rules method where all rows of U and V matrices are updated simultaneously, which is what we used in our proposed methods.

In case of relying only on the rating matrix R, i.e., MULT, the total number of the user rating i and item j determine the total number of the updating times of rows u_i and v_j, respectively. For that, if the user or item suffers from the lack of ratings, cold-start issue, the MULT update rules do not converge to the optimum value for that user or item. Furthermore, the U and V rows that correspond to New-Users and New-Items, respectively, will not be updated. Accordingly, the predicted ratings of either New-Users or New-Items are based on the initial value of the corresponding U and V rows, respectively. In our experiment, we set the predicted ratings p_{ij} to zero in case the rows u_i or v_j have not been updated at all.

Table 6.3 shows the best IMULT case that results in the lowest MAE of the whole dataset. IMULT results are better than the baseline method, WNMTF, which indicates that the imputation is beneficial. However, our proposed method surpasses IMULT with Ciao and Epinions but not with CiaoDVD. This is because most New-Users cannot be imputed in CiaoDVD by our proposed method, which results in a high percentage of unpredictable ratings for New-Users, as shown in Table 4.7. In addition, some of the New-Items ratings in the test belong to New-Users, as well, as we see in Figure 4.1.

It is notable that neither user-wise IMULT nor item-wise IMULT results in the best overall MAE for all datasets. This is due to the fact that with user-wise IMULT, New-Users ratings are unpredictable since the New-Users cannot be imputed; contrastingly, all New-Items ratings are predictable based on the

Table 6.4: The MAE results of the whole dataset and for each user and item group with user-wise IMULT and item-wise IMULT.

Method	Pers.	All	New-	Cold-Start-	Heavy-Rate(r/d)
Ciao					
IMULT (User-wise)	Users	0.8695	4.4118	0.8757	0.8152
	Items		**0.8510**	0.8416	0.8839
IMULT (Item-wise)	Users	0.8239	**0.8165**	0.8324	0.7726
	Items		4.1248	0.7868	0.7606
CiaoDVD					
IMULT (User-wise)	Users	1.3384	4.3433	0.8642	0.7856
	Items		**1.1453**	1.2728	1.4014
IMULT (Item-wise)	Users	1.1730	**1.0218**	1.2170	1.0777
	Items		3.8668	0.8094	0.7176
Epinions					
IMULT (User-wise)	Users	1.0867	3.9203	1.0609	1.0096
	Items		**0.9095**	1.0800	1.0991
IMULT (Item-wise)	Users	1.0236	**0.9547**	1.0453	0.9531
	Items		3.9385	0.8611	0.8633

average of the user ratings. This is the same with the item-wise IMULT method in which the New-Items ratings cannot be predicted and all New-Users ratings are predictable based on the average of the item ratings, as well. As we see in Table 6.4, the best result of New-Users is with the item-wise IMULT method and with user-wise IMULT for New-Items ratings compared to IMULT best case that results in the lowest overall MAE in Table 6.3.

Even though the hyper-wise IMULT method is capable of predicting New-Users and New-Items ratings, their MAEs are high compared to New-Items MAE with user-wise IMULT and New-Users MAE with item-wise IMULT. As shown in Tables 6.3 and 6.4, the MAE results of New-Users and New-Items with hyper-wise IMULT in Ciao and Epinions datasets are in between the user-wise IMULT and item-wise IMULT methods. This is because the U rows that correspond to New-Users are updated depending on half of the average value of the item ratings based on λ value, which results in poor predicted ratings of New-Users. This is the same with the rows of New-Items in V matrix. Due to the high percentage of New-Users and New-Items ratings in the CiaoDVD test set shown in Table 6.2, the mean-wise IMULT method results in the best accuracy instead of hyper-wise.

In general, the imputation in IMULT improves the accuracy results in comparison to WNMTF. Based on our experiment, the IMULT method is not suitable for large scale datasets that hold enormous ratings, especially given that IMULT cannot be run in parallel. However, different parameters setting may result in better results than our experiment.

6.3.3 Comparison Summary

We conclude that if trust information is available, as we see in Ciao and Epinions datasets, then our proposed method, Trust-New-Items-Dual-WNMTF, is capable of predicting the ratings of New-Users and New-Items simultaneously with high accuracy compared to other imputation-based methods. In addition, the imputation error is limited; thus, the imputation enhances the accuracy of other user groups. On the contrary, with the AdaM and IMULT methods, the accuracy of New-Users and New-Items predicted ratings could be good for either New-Users or New-Items depending upon the utilized imputation method, but not both simultaneously.

6.3.4 Trust Imputation Influence

In this section, we study the influence of the imputation process based on the trust information against the imputation based on the rating information only. To perform this, we select the ratings in the test set that belong to imputed New-Users; then, their MAEs are calculated based on the best settings of the three methods: AdaM, IMULT, and Trust-New-Items-Dual-WNMTF. The percentage of the ratings in the test set that belong to imputed New-Users based on our proposed method is shown in Table 4.7.

As we see in Table 6.5, when the imputation is based on the trust information, i.e., Trust-New-Items-Dual-WNMTF, the results are better in terms of accuracy than when the imputation is based on the ratings, i.e., AdaM and IMULT, with all datasets. However, the lack of social information in the recommendation systems may still be considered an issue.

Table 6.5: The MAE of the ratings in the test set that belong to imputed New-Users based on our proposed method with AdaM, IMULT, and Trust-New-Items-Dual-WNMTF.

Dataset	AdaM	IMULT	Trust-New-Items-Dual-WNMTF
Ciao	4.3589	2.2901	**0.7022**
CiaoDVD	3.7675	0.7735	**0.7200**
Epinions	3.9451	2.0840	**0.9292**

6.4 Summary

In this chapter, we integrated the New-Items imputation method into Trust-Dual-WNMTF method to build a recommendation system that is capable of predicting the New-Users and New-Items ratings with the ability not only to limit the imputation error on other users and item groups but also to enhance overall accuracy. We compared our proposed method, Trust-New-Items-Dual-WNMTF, with two popular imputation-based methods: AdaM and IMULT. The results show that an imputation-based method that utilizes trust information is more accurate than others that don't utilize trust information.

7 Conclusions and Future Work

This dissertation presents research for incorporating trust information into NMF-based collaborative filtering recommender systems through the imputation methods to enhance the accuracy of the recommendation. This work involves the study of the factors that impact utilizing imputation with the NMF-based method either positively or negatively with different users and item groups. This chapter summarizes the dissertation work and proposes some future research topics.

7.1 Research Accomplishments

In the last ten years, we believe that the most significant Internet applications have been shopping, entertainment, and socializing. It cannot be denied that the data on shopping and entertainment websites is too large. Customers cannot possibly surf the entire websites' products, which manifests the urgent necessity for a recommendation system that can facilitate filtering the products based on specific information. On the other hand, economists and marketers realize the great potential of the recommendation systems, which includes, but is not limited to, promoting the products and increasing profits. Essentially, the accuracy of the recommendation systems relies mostly on the available customer information, which indeed mostly is absent, especially for new customers. However, users tend to express themselves through social websites that may appear through the social interactions between users. The information on social websites seems to be a great and accurate source of customers' preferences to accomplish the recommendation system's goals.

This dissertation discusses several topics in NMF-based collaborative filtering based recommender systems. Essentially, this dissertation can be divided into three parts: (1) Imputation-based methods that enhance the accuracy; (2) Incorporating social information into NMF-based methods; (3) Investigation and comparison of popular imputation-based methods.

7.1.1 Imputation with Item Auxiliary Information

The New-Items negatively impact the accuracy of the recommendation system due to the fact that New-Items cannot be introduced to the users. Chapter 2 proposes the Aux-New-Items-NMF method that incorporates the item auxiliary information into Aux-NMF through utilizing the imputation process. Our results show that using the item auxiliary information for imputation, not the NMF process, is a better strategy to introduce New-Items to the users without hurting the prediction accuracy of other item groups. In order to control the errors that may be introduced from the imputation, a limited number of ratings are imputed for each item in the New-Items group before NMF is applied. However, the total number of New-Items in the training set determines the total imputed ratings for each New-Item. We demonstrated the influence of the value and average of the imputed ratings in which the prediction accuracy of the rating values that are close to the average of imputed ratings is better than other rating values. Users that have a high probability to like the New-Item need to have more accurate prediction than the users that don't like the item because recommending New-Items to the users is considered an advertisement. By increasing the average of the imputed ratings, the prediction of the high rating values is more accurate than the prediction of the low rating values.

7.1.2 Imputation with Trust Network Information for New Users

The New-Users ratings cannot be predicted with the WNMTF method, which leads to an increase in the recommendation error. Chapter 3 proposes the Trust-WNMTF method that incorporates trust network information into WNMTF through the imputation approach to handle the New-Users issue and to alleviate the rating matrix sparsity. Two cases of imputation are proposed in Chapter 3. The first case is when the available imputed ratings of all users are imputed, i.e., All-Users Imputation case; another case is when only New-Users are imputed, i.e., New-Users Imputation case. Our results show that the accuracy of the New-Users group improves with both imputation cases, especially with the New-Users imputation. In fact, by using the imputation process, the system can recommend items to New-Users. Moreover, the Cold-Start-Users group gets a worse MAE in the New-Users imputation case than the Non-Imputation case; but, some datasets get better accuracy with the All-Users imputation case, which shows that imputing Cold-Start-Users sometimes is beneficial. However, the accuracy of the Heavy-Rater-Users group is worse with both imputation cases in all datasets, especially with the All-Users imputation case. There are two factors that impact the accuracy results. The first factor is the difference in the percentage of the New-Users ratings between the proposed method and Non-Imputation cases. The second factor is the percentage of the New-Users ratings in the test set after the imputation process since we cannot guarantee that all New-Users can be imputed in this step. In addition, the WNMTF reduces the MAE of the subset from the test set that can be imputed when the percentage of ratings in that subset is large.

7.1.3 Influential Factors on Imputation with Trust Network Information for Cold-Start Users

As shown in Chapter 3, the Cold-Start-Users suffer from high error in the prediction results compared to Heavy-Rater-Users. Even though the prediction accuracy of Cold-Start-Users with some datasets was improved with the Trust-WNMTF All-Users method, some others did not improve because of the errors that are introduced from the imputed ratings. Chapter 4 proposes a method that utilizes the trust network information to impute a subset of the missing ratings before WNMTF is applied to improve Cold-Start-Users accuracy. Three strategies are proposed to select the subset of missing ratings that hold the imputed ratings: Trustee, CSI, and HI. Performance analysis shows that imputing the items that have been rated by the user's trustees, Trustee case, improves the accuracy and limits the imputation error. But, it is important to take into consideration that imputing Heavy-Rated-Items introduces more errors from the imputed ratings than imputing Cold-Start-Items even if the Heavy-Rated-Items have been rated by the user's trustees. By selecting the imputed items carefully, the accuracy of Cold-Start-Users improves, especially when the total number of imputed Cold-Start-Users in the training set is large. In addition, the New-Users group gets slightly better results, and the negative impact of the imputation on the Heavy-

Rater-Users is limited. The average of the rating values in the training set is a critical factor for the accuracy of the predicted ratings in which increasing the average of the rating values in the training set by the imputed ratings leads to increasing the accuracy results of Cold-Start-Users. Nonetheless, the increase ratio in the average of the rating values should be limited when compared to the average of the original ratings in the training set.

7.1.4 Selective Imputation Strategies Based on Fused Factored Matrices

Performance analysis shows that although the imputation improves the accuracy of the predicted ratings in general, there are some predicted ratings that have better results without the imputation method. Chapter 5 proposes a selective imputation NMF-based method that fuses the factored original rating matrix and the factored imputed rating matrix to build one system: Trust-Dual-WNMTF. The proposed method is capable of predicting the ratings by utilizing either the imputed users, or imputed items, or both to limit the errors that may be introduced from the imputation. Five strategies are proposed with Trust-Dual-WNMTF to calculate the final predicted ratings. The results show that Trust-Dual-WNMTF is able to utilize the imputation to improve the accuracy of Heavy-Rater-Users in addition to New-Users and Cold-Start-Users, especially with Max Value strategy. The improvement in the prediction is obvious when the maximum value of the sub-predicted ratings belongs to the highest rating value averages of the users and items either before or after the imputation. The strength of the Trust-Dual-WNMTF method is that the prediction results of rating values that are not close to the average of the rating values could be enhanced by utilizing the proposed method with either Max Value or Min Value strategy.

7.1.5 Comparison Between Selected Methods of Imputation-Based Recommendation Systems

Chapter 6 integrates the New-Items imputation method into the Trust-Dual-WNMTF method to build a recommendation system that is capable of predicting the New-User and New-Items ratings. In addition, Chapter 6 conducts a comparison between our proposed method and two popular imputation-based methods, AdaM and IMULT. The accuracy of an imputation-based method that utilizes trust information is higher than other imputation-based methods that rely on the user ratings only.

7.2 Extension Techniques of the Dissertation

In this dissertation, we utilized one type of social information for the imputation method, which is the trust relationship. The trust relationship is considered as a special case of a one direction relationship, i.e., follow-ship relationship. In our datasets, the trust information is within the recommendation system, i.e. internal source. We assume that this work can be extended to more different types of social information in different aspects. The source of social information could be external. In addition, the relationship could be either more general of follow-ship or friendship.

On the other hand, the abundance of auxiliary information is definitely a crucial factor to enhance the recommendation accuracy of the proposed methods in this dissertation. Finally, we assume the proposed methods fit properly the systems that strive to enhance the prediction accuracy of the items that have a high probability to be liked by users, which we believe is one of the fundamental objectives of the recommendation system.

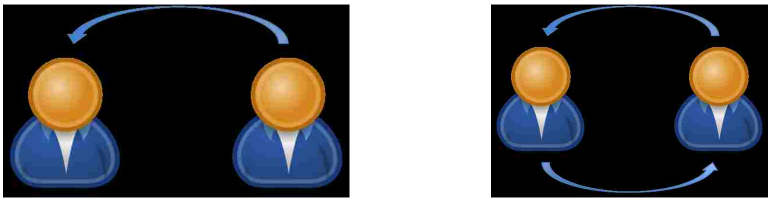
7.3 Suggestions for Future Work

In the future, it would be interesting to integrate external social information into state-of-the-art NMF-based collaborative filtering methods. In addition, deep learning has gained a great interest in many research fields, including recommendation systems. In general, the top topics that should be studied are: (1) Developing Trust-Dual-WNMTF; (2) Influence of social information types on the imputation; (3) Deep learning recommendation systems.

Developing Trust-Dual-WNMTF

In Chapter 5, our proposed strategies are able to choose either the maximum or the minimum value among the four sub-predicted rating values. However, we want to design a model that is capable of choosing the best out of these sub-predicted values, maximum value, minimum value, or ”in-between” value, by utilizing machine learning classification methods, such as K-nearest neighbors. The critical point is the feature selection process of the sub-predicted ratings.

Influence of Social Information Types on The Imputation



(a) Follow-ship

(b) Friendship

Figure 7.1: The interaction types in the online relationship.

In our proposed methods, we utilize internal social information, specifically trust relationship. However, there are different types of relationships that can be classified based on several aspects. For example, the relationship source can be either external or internal from the recommendation system perspective. Some questions are raised, such as, does the internal relationship between the users have a stronger influence on the recommendation than an external relationship? On the other hand, there are two types of interactions between the users. The first is when one user follows another, i.e., followship, when the relationship is from one direction, as we see in Figure 7.1a. In our dissertation, the trust relationship is considered a special type of followship relationship because the relationship is based on product reviewing. The second type is the friendship in which the relationship is from both directions between the users, as shown in Figure 7.1b. The last classification of the relationship is the realism. Currently, in addition to the real-life relationship, there is a virtual relationship. The common features between the users in a virtual relationship may totally differ from the real-life relationship. It would be interesting to study the influence of these types of relationships on the imputation to enhance the recommendation systems.

Deep Learning Recommendation Systems

One of the new hot research topics in recommendation systems is deep learning. It brings further opportunities to improve the recommendation system performance because of its ability to solve many complex tasks. In addition, deep learning has the capacity to utilize various sources and heterogeneous content information, such as texts, images, audios, and even videos. Deep learning is a machine learning method that is based on neural networks. In addition, matrix factorization can be considered as neural networks [19, 20]. From the recommendation system perspective, matrix factorization is able to recognize the low-order interactions between the users and items, and it models the interaction by linearly combining users and items latent factors [20]. On the contrary, deep neural is capable of observing high-order feature interactions and then modeling the interactions between the users and items nonlinearity with nonlinear activations including, but are not limited to, relu, sigmoid, tanh, and others. The nonlinearity modeling allows the system to catch the complex and intricate patterns of the interactions between users and items [81]. Guo et al. [16] demonstrate that combining the power of deep learning and matrix factorization into one system results in better performance.

In the future, we may study the behavior of each user and item group with the deep learning recommendation system. The lack of ratings is considered an issue with the deep learning recommender system. Thus, the imputation process could be utilized to alleviate this issue. However, it would be interesting to analyze the accuracy of the results when the imputation is utilized and to study the influential factors of the imputation. A comparison between the imputation with the NMF-based method and deep learning method can be conducted.

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6):734–749, 2005.

[2] Punam Bedi, Harmeet Kaur, and Sudeep Marwaha. Trust based recommender system for semantic web. In *IJCAI'07 Proceedings of the 20th International Joint Conference on Artifical Intelligence*, volume 7, pages 2677–2682, 2007.

[3] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[4] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Ionference on Knowledge Discovery and Data Mining*, pages 160–168. ACM, 2008.

[5] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD*, pages 126–135. ACM, 2006.

[6] Chris HQ Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005.

[7] Alireza Farhangfar, Lukasz A Kurgan, and Witold Pedrycz. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(5):692–709, 2007.

[8] Rana Forsati, Hanieh Mohammadi Doustdar, Mehrnoush Shamsfard, Andisheh Keikha, and Mohammad Reza Meybodi. A fuzzy co-clustering approach for hybrid recommender systems. *International Journal of Hybrid Intelligent Systems*, 10(2):71–81, 2013.

[9] Rana Forsati, Mehrdad Mahdavi, Mehrnoush Shamsfard, and Mohamed Sarwat. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Transactions on Information Systems*, 32(4):17:1–17:38, October 2014.

[10] Jennifer Golbeck. Generating predictive movie recommendations from trust in social networks. In *iTrust'06 Proceedings of the 4th International Conference on Trust Management*, pages 93–104. Springer, 2006.

[11] Jennifer Golbeck, James Hendler, et al. FilmTrust: movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference*, volume 96, pages 282–286, 2006.

[12] Nathaniel Good, J Ben Schafer, Joseph A Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, John Riedl, et al. Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*, pages 439–446, 1999.

[13] Quanquan Gu, Jie Zhou, and Chris HQ Ding. Collaborative filtering: weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 199–210. SIAM, 2010.

[14] Guibing Guo, Jie Zhang, Daniel Thalmann, and Neil Yorke-Smith. ETAF: An extended trust antecedents framework for trust prediction. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 540–547. IEEE, 2014.

[15] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *IJCAI Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2619–2625, 2013.

[16] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 1725–1731. AAAI Press, 2017.

[17] Ankit Gupta, Rohan Jain, and Shiwei Song. Movie recommendations using social networks. Technical report, Stanford University Stanford, 2008.

[18] Jianming He and Wesley W Chu. A social network-based recommender system (SNRS). In *Data Mining for Social Network Data*, pages 47–74. Springer, Boston, MA, 2010.

[19] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364. ACM, 2017.

[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[21] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[22] Jin Huang, Feiping Nie, Heng Huang, and Yi-Cheng Tu. Trust prediction via aggregating heterogeneous social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1774–1778. ACM, 2012.

[23] Jin Huang, Feiping Nie, Heng Huang, Yi-Cheng Tu, and Yu Lei. Social trust prediction using heterogeneous networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(4):17, 2013.

[24] Won-Seok Hwang, Shaoyu Li, Sang-Wook Kim, and Kichun Lee. Data imputation using a trust network for recommendation. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 299–300. ACM, 2014.

[25] Jingu Kim and Haesun Park. Sparse nonnegative matrix factorization for clustering. Technical report, Georgia Institute of Technology, 2008.

[26] Arnd Kohrs and Bernard Merialdo. Clustering for collaborative filtering applications. In *Computational Intelligence for Modelling, Control & Automation.* IOS, 1999.

[27] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 195–202. ACM, 2009.

[28] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[29] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562. MIT Press, 2001.

[30] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM, 2005.

[31] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[32] Roderick JA Little. Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, 6(3):287–296, 1988.

[33] Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data.* John Wiley & Sons, Hoboken, New Jersey, 2014.

[34] Hao Ma, Irwin King, and Michael R Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46. ACM, 2007.

[35] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 203–210. ACM, 2009.

[36] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. SoRec: social recommendation using probabilistic matrix factorizationec. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 931–940. ACM, 2008.

[37] Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems*, 29(2):9:1–9:23, April 2011.

[38] Yun Mao and Lawrence K Saul. Modeling distances in large-scale networks by matrix factorization. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 278–287. ACM, 2004.

[39] Paolo Massa and Paolo Avesani. Trust-aware collaborative filtering for recommender systems. In *Proceedings of the International Conference on Cooperative Information Systems (CoopIS)*, pages 492–508, 2004.

[40] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 17–24. ACM, 2007.

[41] Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. In *International Conference on Trust Management*, pages 221–235. Springer, 2004.

[42] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[43] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*, pages 187–192, 2002.

[44] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[45] V Paul Pauca, Farial Shahnaz, Michael W Berry, and Robert J Plemmons. Text mining using non-negative matrix factorizations. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, volume 4, pages 452–456. SIAM, 2004.

[46] Dmitry Y Pavlov and David M Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, pages 1465–1472, 2002.

[47] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[48] Manizheh Ranjbar, Parham Moradi, Mostafa Azami, and Mahdi Jalili. An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems. *Engineering Applications of Artificial Intelligence*, 46:58–66, 2015.

[49] Yongli Ren, Gang Li, Jun Zhang, and Wanlei Zhou. The efficient imputation method for neighborhood-based collaborative filtering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 684–693. ACM, 2012.

[50] Yongli Ren, Gang Li, Jun Zhang, and Wanlei Zhou. Adam: adaptive-maximum imputation for neighborhood-based collaborative filtering. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 628–635. ACM, IEEE, 2013.

[51] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 713–719. ACM, 2005.

[52] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[53] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[54] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor. *Recommender Systems Handbook*. Springer, New York City, New York, 2015.

[55] Donald B Rubin. *Multiple Imputation for Nonresponse in Surveys*, volume 81. John Wiley & Sons, Hoboken, New Jersey, 2004.

[56] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS'07 Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 1257–1264, 2007.

[57] Roman Sandler and Michael Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1590–1602, 2011.

[58] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system - a case study. Technical report, Minnesota University - Dept. of Computer Science, Minneapolis MN, 2000.

[59] Joseph L Schafer. *Analysis of Incomplete Multivariate Data*. CRC press, Boca Raton, FL, 1997.

[60] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 704–711, 2003.

[61] Parag Singla and Matthew Richardson. Yes, there is a correlation - from social networks to personal behavior on the web. In *Proceedings of the 17th International Conference on World Wide Web*, pages 655–664. ACM, 2008.

[62] Rashmi R Sinha and Kirsten Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, volume 106, 2001.

[63] Ian Soboroff and Charles Nicholas. Combining content and collaboration in text filtering. In *Proceedings of the IJCAI'99 Workshop on Machine Learning for Information Filtering*, volume 99, pages 86–91, 1999.

[64] Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

[65] Xiaoyuan Su, Taghi M Khoshgoftaar, and Russell Greiner. Imputed neighborhood based collaborative filtering. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 633–639. IEEE Computer Society, 2008.

[66] Xiaoyuan Su, Taghi M Khoshgoftaar, and Russell Greiner. A mixture imputation-boosted collaborative filter. In *FLAIRS Conference*, pages 312–316, 2008.

[67] Xiaoyuan Su, Taghi M Khoshgoftaar, Xingquan Zhu, and Russell Greiner. Imputation-boosted collaborative filtering using machine learning classifiers. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 949–950. ACM, 2008.

[68] K Suresh Joseph and T Ravichandran. A imputed neighborhood based collaborative filtering system for web personalization. *International Journal of Computer Applications*, 19(8), 2011.

[69] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of machine learning research*, 10(Mar):623–656, 2009.

[70] Mojdeh Talabeigi, Rana Forsati, and Mohammad Reza Meybodi. A hybrid web recommender system based on cellular learning automata. In *Granular Computing (GrC), 2010 IEEE International Conference on*, pages 453–458. IEEE, 2010.

[71] Jiliang Tang, Huiji Gao, and Huan Liu. mTrust: discerning multi-faceted trust in a connected world. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pages 93–102. ACM, 2012.

[72] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. eTrust: Understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 253–261. ACM, 2012.

[73] Suhang Wang, Jiliang Tang, and Huan Liu. Toward dual roles of users in recommender systems. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1651–1660. ACM, 2015.

[74] Xiwei Wang and Jun Zhang. SVD-based privacy preserving data updating in collaborative filtering. In *Proceedings of the World Congress on Engineering*, volume 1, pages 377–384, 2012.

[75] Xiwei Wang, Jun Zhang, Pengpeng Lin, Nirmal Thapa, Yin Wang, and Jie Wang. Incorporating auxiliary information in collaborative filtering data update with privacy preservation. *International Journal of Advanced Computer Science and Applications*, 5(4):224–235, 2014.

[76] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, Burlington, MA, 2016.

[77] H Wu and Z Liu. Non-negative matrix factorization with constraints. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 506–511, 2010.

[78] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 267–273. ACM, 2003.

[79] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM, 2005.

[80] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 549–553. SIAM, 2006.

[81] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):5, 2019.

[82] Yi Zhang and Jonathan Koren. Efficient Bayesian hierarchical user modeling for recommendation system. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 47–54. ACM, 2007.

[83] Cai-Nicolas Ziegler and Georg Lausen. Analyzing correlation between trust and user similarity in online communities. In *International Conference on Trust Management*, pages 251–265. Springer, 2004.

Vita

Personal Data:

Name: Fatemah Alghamedy

Place of Birth: Boulder, CO, USA

Educational Background:

- Master of Science Degree in Computer Science, Arkansas State University, USA , 2013 (4 GPA)

  Thesis Title: CFPh-Growth Tree: A Data Structure for Mining Association Rules with Skewed Support Distribution

- Bachelor of Science Degree in Computer Science, Science College for Girls, Dammam, Saudi Arabia, 2006 (94.13% GPA). First honors degree, Third highest GPA in Computer Science Department.

Professional Experience:

- Research Assistant. Markey Cancer Center, University of Kentucky, Lexington, KY, USA. 9/2018 - .

- Grader. Department of Computer Science, University of Kentucky, Lexington, KY, USA. 1/2018-5/2018.

- Research Assistant. Markey Cancer Center, University of Kentucky, Lexington, KY, USA. 1/2017 - 5/2017.

- Grader. Department of Computer Science, University of Kentucky, Lexington, KY, USA. 1/2016 - 5/2016.

- Teacher Assistant. Department of Computer Science, Arkansas State University, Jonesboro, AR, USA. 8/2013 - 12/2013.

- Instructor. Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. 12/2007 - 2/2010.

- Teacher Assistant. King Faisal University, Dammam, Saudi Arabia. 2/2007 - 6/2007.

Awards:

- Student travel support from Saudi Arabian Cultural Mission (SACM), April 2019.

- Student travel support from Saudi Arabian Cultural Mission (SACM), December 2018.

- Student travel support from Graduate School Fellowship, University of Kentucky, November 2018.

- Student travel support from Graduate School Fellowship, University of Kentucky, March 2018.

- Full scholarship for Ph.D. degree. Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

- Outstanding graduating student for 2013-14. Arkansas State University, Jonesboro, AR, USA.

- Student travel support from Saudi Arabian Cultural Mission (SACM), October 2013.

- CRA-W Grad Cohort 2011 Participation Grant, April 2012.

- Full scholarship for Master degree. Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

Posters:

- Imputing Item Auxiliary Information in NMF-based Collaborative Filtering, 2nd Commonwealth Computational Summit, University of Kentucky, Lexington, KY, USA, 2018.

- Incorporating Protein Dynamics Through Ensemble Docking in Machine Learning Models to Predict Drug Binding, Markey Cancer Research Day, University of Kentucky, Lexington, KY, USA, 2018.

- Imputing Trust Network Information in NMF-based Recommendation Systems, 1st Commonwealth Computational Summit, University of Kentucky, Lexington, KY, USA, 2017.

Publications:

- **Fatemah Alghamedy** and Jun Zhang. Imputation strategies for cold-start users in NMF-based recommendation systems. In Proceedings of the *3rd International Conference on Information System and Data Mining (ICISDM2019).* Houston, Texas, USA, To appear.

- **Fatemah Alghamedy** and Jun Zhang. Enhance NMF-based recommendation systems with social information imputation.*Computer Science & Information Technology*, Vol. 8, No. 15, pp. 37-54, November, 2018.

- **Fatemah Alghamedy**, Maryam Al-Ghamdi, and Jun Zhang. Imputing item auxiliary information in NMF-based collaborative filtering. *Computer Science & Information Technology*, Vol. 8, No. 15, pp. 21-36, November, 2018.

- **Fatemah Alghamedy**, Jeevith Bopaiah, Derek Jones, Xiaofei Zhang, Heidi Weiss, Sally Ellingson. Incorporating protein dynamics through ensemble docking in machine learning models to predict drug binding. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, Vol. 2017, pp. 26–34, May, 2018.

- Derek Jones, Jeevith Bopaiah, **Fatemah Alghamedy**, Nathan Jacobs, Heidi Weiss, W.A. de Jong, Sally Ellingson. Polypharmacology within the full kinome: a machine learning approach. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, Vol. 2017, pp. 98–107, May, 2018.

- **Fatemah Alghamedy**, Xiwei Wang, and Jun Zhang. Imputing trust network information in NMF-based collaborative filtering. In Proceedings of the *Proceedings of the ACMSE 2018 Conference*, pp. 2:1–2:8. Richmond, Kentucky, USA, March 29 - 31, 2018.