IMPROVED PROBABILISTIC MATRIX FACTORIZATION MODEL FOR SPARSE DATASETS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

YILMAZ AR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INFORMATION SYSTEMS

OCTOBER 2014

IMPROVED PROBABILISTIC MATRIX FACTORIZATION MODEL FOR SPARSE DATASETS

submitted by **YILMAZ AR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Nazife Baykal
Director, Graduate School of **Informatics Institute** ________________

Prof. Dr. Yasemin Yardımcı Çetin
Head of Department, **Information Systems** ________________

Assist. Prof. Dr. Tuğba Taşkaya Temizel
Supervisor, **Information Systems Department** ________________

Examining Committee Members:

Assoc. Prof. Dr. Aysu Betin Can
Information Systems Department, METU ________________

Assist. Prof. Dr. Tuğba Taşkaya Temizel
Information Systems Department, METU ________________

Assoc. Prof. Dr. Şahin Emrah
Computer Engineering Department, Ankara University ________________

Assist. Prof. Dr. Erhan Eren
Information Systems Department, METU ________________

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU ________________

Date: **01.10.2014**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: YILMAZ AR

Signature :

ABSTRACT

IMPROVED PROBABILISTIC MATRIX FACTORIZATION MODEL FOR SPARSE DATASETS

AR, YILMAZ
Ph.D., Department of Information Systems
Supervisor : Assist. Prof. Dr. Tuğba Taşkaya Temizel

October 2014, 83 pages

The amount of information on the World Wide Web has increased significantly owing to advancing web and information technologies. This has made it difficult for users to obtain relevant and useful information thus there is a need for information filtering. Recommender Systems (RS) have emerged as a technique to overcome the problem. Collaborative Filtering (CF) that is one of the widely used RS approaches aims to predict users' preference concerning an item. The main idea behind CF is the users who agreed in the past will agree in the future. The Probabilistic Matrix Factorization (PMF) is the preferred CF technique in the literature due to its high accuracy and scalability. This thesis demonstrates the importance of the initialization techniques for the user and the item latent vectors in the PMF algorithm with real and synthetic datasets and proposes five different initialization techniques. The suggested approaches produce better results in comparison with the state-of-the-art techniques in particularly very sparse datasets.

Keywords: Probabilistic Matrix Factorization, Latent Vectors, Recommender Systems

ÖZ

SEYREK VERİ KÜMELERİ İÇİN İYİLEŞTİRİLMİŞ OLASILIKSAL MATRİS ÇARPANLARINA AYRIŞIMI MODELİ

AR, YILMAZ
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Tuğba Taşkaya Temizel

Ekim 2014 , 83 sayfa

Dünya çapındaki ağ üzerindeki bilgi miktarı, ağ ve bilgi teknolojilerindeki ilerlemeler nedeniyle önemli ölçüde artmıştır. Bu durum kullanıcılar için ilgili ve yararlı bilgiler elde etmeyi zor hale getirmiştir ve bu nedenle bilgi filtreleme ihtiyacı oluşmuştur. Öneri Sistemleri (ÖS) bu probleme bir çözüm olarak ortaya çıkmıştır. Yaygın olarak kullanılan ÖS yaklaşımlarından biri olan Ortak Filtreleme (OF), kullanıcıların bir ürün üzerindeki tercihini tahmin etmeyi amaçlamaktadır. OF ardındaki ana fikir, geçmişte aynı fikirde olan kullanıcıların, gelecekte de aynı fikirde olacaklarıdır. Bir OF tekniği olarak Olasılıksal Matris Çarpanlarına Ayrışımı (OMÇA) genellikle yüksek doğruluk ve ölçeklenebilirlik nedeniyle literatürde tercih edilmektedir. Bu tezde, OMÇA metodunda yer alan kullanıcı ve ürün gizli vektörlerin başlatma tekniklerinin önemi gerçek ve sentetik veri kümeleri ile gösterilerek yeni beş başlatma tekniği önerilmektedir. Önerilen yaklaşımlar literatürdeki diğer başlatma teknikleri ile karşılaştırıldığında çok seyrek veri setleri için daha iyi sonuçlar üretmektedir.

Anahtar Kelimeler: Olasılıksal Matris Çarpanlarına Ayrışımı, Gizli Yöneyler, Öneri Sistemleri

Dedicated to my wife, daughter, son and my parents

ACKNOWLEDGMENTS

I am grateful to my supervisor Assist. Prof. Dr. Tuğba Taşkaya Temizel for her immense patience and continuing support she gave me throughout my research. I am certain that this thesis would not have been possible without her guidance. I am indebted to my thesis committee for their insightful comments and valuable contributions. Last but not least I wish to thank my family for their endless patience during my study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

PMF	Probabilistic Matrix Factorization
RperU	Number of Ratings per User
RperI	Number of Ratings per Item
CF	Collaborative Filtering
RS	Recommender Systems
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
WWW	World Wide Web
BPMF	Bayesian Probabilistic Matrix Factorization
BPMFSR	Bayesian Probabilistic Matrix Factorization with Social Relations
BPMFSRIC	Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents
PCC	Pearson Correlation Coefficient
MCMC	Markov Chain Monte Carlo

CHAPTER 1

INTRODUCTION

The number of studies concerning recommender systems (RS) has grown significantly in the recent years owing to the popularity of e-platforms together with their increasing number of items and users. In the literature there are noteworthy survey studies that aim to summarize this extensive field [7, 9]. The domain in the RS has always presented a challenge with high-dimensional, complex, noisy, multi-modal (text, discrete ratings and video), sparse user and item information. Due to the complexity of RS, the studies [48, 8, 19, 51, 9, 2, 42] in the literature were only able to solve a specific part of the problem. For example, many studies disregarded the problem of cold start users or others only considered the user-item rating information while ignoring the item features or user review content in their proposals. For some solutions, a thorough parameter tuning should be undertaken, as in standard matrix factorization algorithms. RS can be mainly analyzed in two key groups based on the information they use and how they approach the problem.

Content based RS extract the features of the items each user liked in the past to generate user preference (liking or disliking) information [41, 43]. New items that are highly correlated with these preferences are recommended to the users. The over-specialization problem, limited-content problem and new-user problem are important limitations of content based methods.

Collaborative Filtering (CF) methods make use of mutual user preferences on items including user-item ratings or the items with which they have interacted (purchased/watched/followed) in order to generate new user-item associations. Memory-based CF methods principally compute similarities between each pair of users or items using user rating data to determine the target item for recommendation. Two typical examples of memory-based methods are item-based CF and user-based CF which compute similarities between items and users [51]. Latent factor models, such as matrix factorization models, characterize the information about both items and the users to the same latent factor space where a high-dimensional feature vector comprising the observation data is squashed into a lower dimensional latent vector space. CF methods are extensively discussed and explained in [26]. Probabilistic Matrix Factorization (PMF) is the widely used approach in model-based methods. Pioneered by [20], PMF aims to create dyadic data modeling where a probabilistic low-rank approximation to the observation matrix is carried out to reveal structure in the data [1]. The main idea behind PMF is that user preferences are determined by some unseen factors. Despite being a powerful modeling method, the basic approach exhibits the following deficiencies [1, 44, 55]:

- Sensitivity to parameter settings: Regularization parameters and other optimization parameters such as epoch size, batch size and the learning parameters should be carefully decided to circumvent over-fitting to the training dataset.

- Cold start user problem: For new users or items with no or few ratings, it cannot produce reliable recommendation since the matrices are formed based on rating observations.
- The use of single type of information (no side information): Disregarding explicit information such as temporal (review dates) data, user profiles or demographics data (such as gender, age and education) and item content (item features) often results in limited success in recommendation performance as the basic algorithm depends on the mere incorporation of user-item rating information.
- Extreme sparseness of the rating data: The limited number of co-ratings causes problems for learning with the PMF as it cannot converge successfully.

Numerous advancements in PMF were proposed to resolve the issues given above. Methods that incorporated side information into latent vectors as priors aimed to ensure richer prior information in the latent vectors [1, 44, 58]. A neighborhood-aware adaption of the algorithm [55] was proposed in which the tagging data was used to select the neighbors of each user and each item to which unique Gaussian distributions were added to ensure similar users (items) would have similar latent features. In addition to using user-item ratings data, some studies showed a significant improvement in the accuracy when additional information was included in their model. In the case of absence or scarcity in the amount of explicit rating data, implicit feedback information that comprise the indirect reflections of user opinion or behavior such as purchase data or search patterns are valuable in filling the gap. The positive impact of the incorporation of temporal dynamics on user interests and preferences, and the attachment of confidence scores to each observation such as the frequency of certain user actions such as how many times a certain type of item is preferred by the user are also shown to provide better accuracy results in a Netflix dataset [27]. A trust network and user-item rating matrix based ensemble model connected through the shared user latent feature space was developed and demonstrated superior performance compared to single models using basic PMF method in [33].

To solve the over-fitting problem caused by improper tuning of regularization parameters, a Bayesian treatment of PMF model where model capacity was controlled by integrating over all model parameters and hyperparameters was proposed in [50]. Two other Bayesian based models, *Bayesian Probabilistic Matrix Factorization with Social Relations (BPMFSR)* and *BPMFSR and Item Contents (BPMFSRIC)* were developed, where the user hyperparameters and item hyperparameters are assumed to be different for each user and the item vector, are sampled according to social relations and item contents [30].

1.1 Problem Definition

PMF models require a thorough tuning of the parameters for learning. Presented in this thesis is another important problem of PMF models which is the initialization of latent vectors. To remedy this five different initialization methods are introduced and I show that:

- People usually tend to give high ratings to items. This behavior can be explained by the observation that people generally prefer to rate when they are fond of a specific item [12]. Hence, it is common to see many items with high item ratings in e-platforms.

- Many e-platforms have very sparse datasets where it is common to see a small number of item ratings.
- Only using rating data when constructing the latent vector matrices can give significantly better results than those that are initialized using different methods such as employing random numbers from Gaussian and Uniform distributions. In particular, people's bias in rating can be used in the initialization.

1.2 Contribution of the Thesis

In this thesis, it is argued that:

- The initialization of latent factor matrices significantly affects the performance of the PMF algorithm. This argument is different from those provided in [1, 44] where they incorporated side information as Gaussian priors. Only using the rating data in the initialization of matrices may alone significantly improve performance.
- People usually rate an item and give comments when they have a positive attitude toward the item. So the average rating score of most items is usually high in e-platforms[12]. The incorporation of this biased information into latent vectors may improve the accuracy. The performance improvement is expected to be notable particularly in datasets with few ratings.
- The characteristics of dataset used in the initialization of latent vectors affect the performance of PMF algorithm therefore, the rating matrix density and skewness of ratings can be used in the initialization process. Additionally the number of ratings per user and the number of ratings per item is very important on PMF training.

The main contributions of this thesis are: (1) To remedy the problems mentioned above, I propose five initialization methods that utilize only the rating data after demonstrating the problem. The main idea of the proposed methods is to initialize the user and item latent vectors using the distribution statistics of the rating information. (2) A systematic large-scale simulation is undertaken based on different initialization settings and the results are compared. These approaches were applied to several real-world datasets and simulated datasets each having different distribution characteristics in order to demonstrate the performance of the methods in a comparative study. (3) A flexible dataset simulation tool is developed in which the characteristics of datasets namely rating matrix density and skewness are utilized. In addition to rating matrix density, the concept of user density and item density computed as the ratings per user and the ratings per item respectively are used in the proposed methods.

1.3 Organization of the Thesis

Chapter 2 presents an overview of RS and different approaches are discussed. The drawbacks of these methods are provided and the metrics used to evaluate RS are given.

Chapter 3 gives a detailed explanation of PMF and discusses the previous improvement efforts. It also investigates the drawbacks of existing PMF approaches.

Chapter 4 presents the proposed methods. The motivation of the proposed methods, the detailed description, and flowcharts are given.

Chapter 5 provides the characteristics of the datasets, experimental settings, experiments, and the results. It provides a comparison of proposed methods with the existing studies.

Chapter 6 gives the conclusion and future directions of the research.

CHAPTER 2

RECOMMENDER SYSTEMS

2.1 Introduction

The amount of information on the World Wide Web (WWW) has increased significantly owing to advancing web and information technologies. Therefore, it has become difficult for users to obtain relevant and useful information thus there is a need for information filtering. Recommender Systems (RS) have emerged as a technique to overcome the problem of information overload by filtering millions of items and providing personalized suggestions to users. The term "item" covers many different elements including; products, news, music, entertainment and points of interest such as restaurants and nightclubs. There is a widespread use of RS by internet companies for example Netflix.com for movie recommendation, Amazon.com for item recommendation and Last.fm for music recommendation. These internet companies aim to increase sales or to expand the use of their sites by providing recommendations that will be appreciated by the user. To date there have been several RS proposed that use various recommendation approaches [48, 25, 8, 19, 51, 9, 2, 42].

2.2 Recommender System Approaches

2.2.1 Content-Based Recommender Systems

Content based RS claim that in future users will prefer the items that they preferred in the past. Content based methods aim to extract the features of items the target user liked in the past and in future will recommend similar items to the same target user. In other words, the items that are highly correlated with the preferences of the target user are added to his or her recommendation list. Over-specialization, limited-content, and new-user are three important problems of content-based methods. In terms of over-specialization since a user can only be recommended items that are similar to those previously preferred, content-based methods usually cannot recommend *novel* items. The limited-content problem is defined as the specified features not being sufficient to describe the different characteristics of the items in the system [31]. The new-user problem results from the new user not having rated enough items to obtain accurate recommendations. The content based systems need to collect several ratings from a user to learn her/his preferences [31].

Content-based RS create profiles for each user and the item contents are compared with these profiles. Then, a score for the preference of the user in relation to an item is computed. To compute user's preference for an item, other users' information is not required. Only the users' own profile and item descriptions are necessary [41, 43].

Content-based RS consist of three recommendation processes; content analyzer, profile learner and filtering component [31]. The main aim of the content analyzer is to represent the features of different items in a suitable form for the next process. Various feature extraction methods are used to convert feature representations into structured forms. The profile learner collects user preferences and aims to find a generalization of this data to create user profiles [31]. One of the advantages of content-based RS is that a user profile is constructed based on the owner's information but not that of other users. The filtering component compares user profiles and item representations to produce recommendations. Content-based RS are able to recommend items that have not been previously rated [31].

2.2.2 Collaborative Filtering Recommender Systems

CF RS make use of the similarity between users in order to recommend an item. According to this approach users who agreed in the past will agree in the future. First similar users are found and the traditional CF RS only utilize the users' rating information on items for similarity calculation. Second, similar users' ratings on the given item are fused to predict unknown preferences. Many similarity computation methods exist in the literature some of which only use a user-item rating matrix such as Pearson correlation. Other methods use 'trust' between the users which can be either explicit or implicit. Prediction approaches exist in which the preference of the given user is obtained using aggregation of similar users' weighted preferences. There are two types of CF methods: memory-based and model-based.

The memory-based methods are also called neighborhood methods. For each pair of users, a similarity weight is computed using the items they rated. Then, the item that will be recommended to the user is chosen as that which has the maximum weighted average of the ratings of similar users. In addition to these user-based approaches, item-based methods have been developed [51]. In this approach the main task is to find similar items. Item-item similarities are computed and then used to predict ratings in the same manner on user-based methods. Model-based methods, also known as latent factor models, characterize both the items and users from a specified number of features.

In a very recent study, Bobadilla et al. proposed a generalized version of RS, which provide recommendation to a group of users on a restricted group of items [5]. For example a group of four friends could ask for joint recommendations of films similar to 'Avatar' or 'Titanic'. This constraint was implemented as a constraint to CF.

2.2.3 Trust-Based Recommender Systems

Many trust-based RS have been proposed in the literature [21, 36, 39, 24, 42, 17, 40]. Some use explicit trust values [36, 39, 17, 33], others compute trust from ratings [42]. The propagation of trust is preferred in some studies [39, 57] while some studies do not diffuse trust values in a social network [42, 35, 33]. While some exploit trust values to find similar users and incorporate them into traditional CF methods [21], the others develop independent trust-based recommender models in which the trust values are not used just as the user's similarity values. Some of the trust values of a user do not depend on the requesting user and is a global reputation in the system. Moreover, these values are not dependent on the asked item and constants in all the items for all the advice seekers [32, 35, 33]. On the other hand, in some approaches trust values are not constant and dependent on user-item pairs. They may use different trust values for each user-item pair for different advice seekers [42].

Golbeck used trust values provided by member of the social network [15]. These trust values were not public; no one other than the member who expressed the trust knew these values. The reason behind hiding trust information from the users was to learn about the actual opinions of the users, which they might withhold for fear of offending or upsetting others. The study carried out by Golbeck et al. was performed on the web environment Film Trust. 500 users created about 11,250 ratings (from half to four stars) and reviews of about 1250 movies. Then for each user-movie pair in the testing dataset, three different calculations are performed in order to predict the movie's rating by the user. The first calculates the simple average of ratings given to that item. The second weights the ratings with trust values. These values can be obtained in two ways; given by the user who search for advice or inferred by TidalTrust [14] algorithm which calculates the trust value for each rater at a given depth. The third calculation is standard CF algorithm. The first experiments do not show any statistical difference between the trust-based ratings and simple average, the reason being that the majority of the users' actual ratings are very close to the average rating. A random sampling of movies in their dataset showed that about 50% of all ratings are within the range of the mean ± a half star [15]. Therefore, the trust-based average of ratings is not better than the simple average of ratings. In the study another experiment was performed to test three approaches using only the users who do not give the average rating. Different levels of disagreement with the average (from half to two stars) were tested. When the gap was equal or greater than 1 star, the trust-based recommendation outperforms CF and the simple average.

O'Donovan and Smyth used two trust models in their study [42]. They made use of rating information to compute the profile-level and item-level trust. Profile level trust was the percentage of accurate recommendations that a user has contributed. Normally when a user was connected with the recommendation process, there are other users who participate in the same recommendation. The study assumes that it is the user alone that is responsible for the recommendation when calculating the percentage of correct recommendations. If a user has been nominated 100 times as a followee recommendation and for 40 times he has actually been followed, the profile level trust value for this user will be computed as 0.40. The users' profile may be more reliable than others when it comes to predicting the ratings for certain items. Item-level trust measures the percentage of the correct recommendations of a user on an item i. The assumption on computing profile-level trust is also made when calculating item-level trust. This work also describes the number of ways in which computed trust values can be incorporated in the standard CF algorithm. The experiments show that using trust values has a positive impact on the overall prediction error rates. The limitation of this study is that the authors use a global trust value and provide no personalization or trust propagation.

Massa and Avesani proposed trust-aware RS to solve the problems on standard CF algorithms [36, 38, 39]. One of the big challenges of the CF approach is the sparseness of the user-item rating matrix that means the number of items is generally very high while the number of rated items is low. Therefore, it is very unlikely to find two random users who have rated common items. This makes the process of finding similar users very difficult; in some situations this is even impossible. In their work, the authors propose propagating trust over the trust network and estimating a trust weight that can be used in place of the similarity weight. Their experiment undertaken on an Epinions dataset showed that the proposed approach is most effective in terms of accuracy and preserves good coverage. The limitation of this study is the trust value representation in which the web of trust is built on binary relationships among users and the trust value propagation is only performed based on the distances between users.

Hwang and Chen proposed a trust-based RS that only uses an user-item rating matrix as the input [21]. Their system undertakes two computations; one for the trust matrix and the other for the user similarity matrix. Both computations use the ratings matrix. Then, the trust matrix is incorporated into the standard CF method instead of the user similarity matrix.

Golbeck created 28 profiles in order to investigate the features of profile similarity that are based on movies rated by the subject. She also explored the relation between the features of profile similarity and the trust values [16]. Profiles are generated to represent the preferences of hypothetical users. The study showed that there is a correlation between trust and the largest single difference in ratings. As the single largest difference increases, trust decreases. The largest single difference is the maximum difference between the user's rating and profile's rating on a given movie and it affects the user trust in the profile. Another finding of the study is that there is a correlation between trust and the agreement on movies that have been assigned extreme ratings. In this study, the range of the ratings was between 1 and 10, and the extreme ratings are defined as 1, 2, 9, and 10.

Ray and Mahanti reconstructed a trust network by removing the trust links between users in which their correlation is below a specified threshold value [47]. The Pearson Correlation Coefficient is used to compute the correlation between the users. After this reconstruction, three different weighting schemes; trust, trust multiplied by correlation, and trust multiplied by top-*n* correlation were used to produce the rating predictions. The experiments were performed using different values of the correlation threshold on the Epinions dataset. The results showed that the best predictions were obtained when the correlation threshold value was 0.5 and the weighting schema was trust multiplied by top-5 correlation. However, they only used the Epinions dataset in their experiments thus, their results cannot be generalized to other datasets. [47].

2.2.4 Hybrid Recommender Systems

Hybrid recommender systems combine two or more methods in order to overcome the limitations of each method. Traditional hybrid recommender systems usually collaborate with other approaches. Burke has given details of possible combination techniques [9].

Bobadilla et al. used the significance of users and/or items in CF approach [3]. The proposed method weights the rating of an item according to its importance. The authors assumed that none of the items or the users have the same importance when making recommendations to other users. For example; a recent and much-advertised Apple item can be regarded as more significant in comparison with an outdated MP3 device that is still available.

Bobadilla et al. assumed that the value of the similarity must be modulated by the value of the singularity, in a way that a very singular similarity should be awarded a higher value than a very normal similarity [4]. For example, if 95% of the users rated positively for the item, the similarity derived (for this item) between two users who belong to the remaining 5% (very singular) must be greater than the similarity derived between two users who belong to the

95%(not very singular). Using the singularity measure suggested that a great improvement is achieved comparing to the results obtained using traditional similarity measures, both in terms of prediction quality and recommendation quality [4].

Pradel et al. presented a case study on real-life purchase dataset [45]. They compared different type of RS that used on purchase data rather than rating data. Their work discusses that the rating data is declarative and can be unreliable or unavailable. It also compares three different methods namely; item-based collaborative filtering, matrix factorization, and association rules. Surprisingly the most accurate results come from the simplest form of bi-gram association rules. Their results also show that factors such as how recent was the purchase and context-awareness may be at least as important as the choice or the design of a well-performing algorithm. One important issue was also indicated in that the purchase data only shows the action of buying; it does not indicate customers' opinion of the item.

2.2.5 Social Network-Based Recommender Systems

Liu and Lee proposed a way of incorporating social network information into collaborative filtering [29]. They collected data about users' preferences and their social network information from Cyworld, which was one of the most popular web sites in South Korea. The members of this site can share photos and post blogs. They can connect to their immediate circle of friends by adding them as "ilchon" (close friends). The authors conducted a web survey in which 30 home page skin items are presented to survey participants who were then asked to rate their preferences using a 5-point scale where 5 indicates "best". A distributor group was formed from those members who have at least 10 Cyworld friends. These 27 distributors had a total of 313 friends. The distributors had two tasks; to participate in the survey and to distribute the survey to their friends. The subjects also include 119 members who have no friend relationship with any distributor. In this study, four experiments were performed. First, the recommendations are generated through collaborative filtering using nearest neighbors. Then, in the second experiment the friends are utilized instead of the nearest neighbors to make predictions. The third experiment combined the nearest neighbors and friends into a new neighbor group. In the fourth experiment, the influence of friends among the neighbors is emphasized. The results indicate that combining the nearest neighbors and social network information (experiment 3) greatly improves the prediction accuracy. The most important limitation of the study is using all the friends' information without filtering it because the size of experiment set is very small. The average number of friends is 12 which is very small compared to a real dataset.

Walter et al. proposed a model for a trust-based recommendation system using a social network information [53]. They investigated the conditions and to what extent the existence of a trust system enhances the performance of a recommendation system on a social network. In this model when a user decides to purchase an item, s/he asks her/his neighborhood for recommendation. If neighbors are unable to provide a recommendation they pass this query to their neighbors. Therefore, the network replies to a user's query with a set of recommendations. The system decision about recommending an item depends on these recommendations. Basically the most frequently recommended item would be recommended. However, the situation is not so simple; the users' preferences are heterogeneous, so the basic approach may not be a good solution. Therefore, the trustworthiness of the users should be incorporated to the system.

Carmagnola et al. developed a Social Networks-Based Algorithm for Social Recommender Systems (SONARS) [10]. SONARS recommends content to the members of a social network based on the trend of the network and the influence relationships among the members. This approach uses three complementary theories of social influence namely social conformity, social comparison, and social facilitation. Social conformity claims that people belonging to a group usually experience a "pressure to conform" (i.e. they tend to change their attitudes and behaviors to match the expectations of the others). Social comparison states that people search for information about the opinions of other members to evaluate how they compare and create their own attitudes and behaviors. Social comparison mostly takes place when people are without an objective means of evaluation. Usually the effects of social comparison are obvious if people compare themselves to similar individuals. Social facilitation takes place when people are encouraged to perform a target behavior as a consequence of the physical or virtual company of others. Based on these theories, the base idea of SONARS is that taking part in social relationships can cause individuals to change their attitudes and behaviors and they are more likely to be interested in what people belonging to their social network like, independently of their real preferences [10].

Chen and Fong proposed a framework of collaborative filtering on social network and a novel approach in measuring trust factors by data mining over a survey dataset provided by a Facebook Project [13]. In this study, the similarity between the profiles of a pair of users and the trust between them were fused into the CF algorithm. The trust values are computed considering the activities on Facebook and users do not need to state the trust values explicitly. The proposed framework, collaborative filtering trust network (CFTN), collects many sources of static and dynamic data to generate recommendations. The static attributes are classified into two groups; "demographic" and "interest". There are three groups of dynamic attributes; "tags", "activities", and "applications". Demographic, interest, activity, and application attributes are used to compute similarity while tags are used to calculate trust value. The dataset was collected from 124 students. The dataset contained the answers to the questions pertaining to perceptions on trust and privacy, messaging, pictures, and groups on Facebook. The authors stated that although the sample size of the survey was relatively small their work demonstrates the concept of estimating trust factors from Facebook data [13].

2.3 Challenges in Recommender Systems

Many RS utilize only user ratings on items to predict the unknown preference of a user on an item. The rating information has limited explanation of the preference of users on items. To find the similar users who are crucial for recommender system approaches such as CF, the known rating information may not be sufficient. Using only rating information can also cause other problems such as those created by a cold start user, cold start item, and sparseness.

When a new user registers to the system, s/he has no rating history. Therefore s/he cannot receive recommendations. The same problem is also valid for items. New items have no ratings, and without ratings it is very difficult for the item to be recommended. Even when it is possible to compute a similarity weight, due to data sparseness, this is often derived from few overlapping ratings and it is hence a noisy and unreliable value [38]. Thus any predictions presented might be inaccurate.

Overspecialization, a limitation for content-based RS, prevents users from encountering new items that they may enjoy. A similar problem is only making mainstream popular recommendations however, some of the users might not like that type of item and look for more diverse items. Being accurate is not always good for RS. If users only receive mainstream or overspecialized item recommendations this may make users stop receiving recommendations.

In general, RS do not use blogs or discussion forums as an information source. They prefer structured data such as item ratings, explicit trust values and a number of review ratings. However, blogs and forums can contain quite valuable real experiences that can meet users' needs. These hidden recommendations need to be extracted and used by RS. A related problem is writing many reviews or being active in a social network does not necessarily make that user an expert. A real expert or high reputation members may not be active and might write less or high-quality reviews. If the recommendation algorithm only looks for quantity, that not-active-but-expert users could be overlooked.

One of the important challenges is that different domains have different properties. The approaches that might be used in subjective domains of taste such as book or movies may not be used in factual and objective domains. A user might be interested in other user's opinions that are similar to her/him in subjective domains, but s/he might prefer the opinions of an expert in factual domains such as fitness equipment. Usually the same recommender algorithms are applied in all domains. The new approaches that exploit the domain or category characteristics can help to improve the accuracy of RS.

The other challenge is the trade-off between the accuracy and coverage of the predictions. The higher the coverage, the lower the accuracy of the recommendation system will become. For example, a RS predicts what rating a user might give to a certain item. Then, the system needs to filter the items that have predicted ratings above a user provided threshold. If the threshold is high, the number of recommended items will be low but they will meet the user's expectations. If the threshold is low, the number of recommended items will be high and there is a high probability that not all of the recommended items will meet users' expectations.

2.4 Evaluation Metrics for Recommender Systems

Root Mean Square Error (RMSE) is widely used as the evaluation metric in the literature [50, 27, 49].

RMSE is defined as in Equation 2.1

$$RMSE = \sqrt{\frac{1}{|R|}\sum_{(i,j)\epsilon R}(\hat{r}_{ij} - r_{ij})^2} \tag{2.1}$$

where R is the set of tested rating values defined for user-item (i, j) pairs and $|R|$ is the number of elements in R. $\hat{r}_{ij}$ is the predicted rating value for user i to item j and r_{ij} is the actual rating value.

Mean Absolute Error (MAE) is another widely used evaluation metric on RS. MAE is defined as in Equation 2.2.

$$MAE = \frac{1}{|R|} \sum_{(i,j) \epsilon R} |\hat{r}_{ij} - r_{ij}| \tag{2.2}$$

where R is the set of tested rating values defined for user-item (i, j) pairs and $|R|$ is the number of elements in R. $\hat{r}_{ij}$ is the predicted rating value for user i to item j and r_{ij} is the actual rating value.

CHAPTER 3

PROBABILISTIC MATRIX FACTORIZATION

3.1 Formal Definition

PMF characterizes both the users and items on certain number of factors inferred from the user-item rating data. This method simultaneously maps user and item characteristics into a feature space. I created an $N \times M$ rating matrix where N and M indicate the number of users and items respectively. N users having D features are modeled by a matrix U with the size $N \times D$ and M items having D features is modeled by matrix V with the size $M \times D$. The rating value R take values between $[1, K]$. A low-rank matrix factorization approach aims to approximate $\widehat{R}$ by a multiplication of low-rank factors [56].

$$\widehat{R} = UV^T \tag{3.1}$$

where U is an $N \times D$ user latent matrix and V is an $M \times D$ item latent matrix, with $D < min(M, N)$.

PMF is a probabilistic linear model with Gaussian observation noise. The conditional distribution over the observed ratings is defined as shown in Equation 3.2 [49].

$$p(R|U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left(\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)\right)^{I_{ij}} \tag{3.2}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the Gaussian distribution probability density function with mean μ and variance σ^2. I_{ij} is equal to 1 if user i rated item j, and equal to 0 otherwise. R_{ij} represent the rating of user i for item j.

Zero-mean spherical Gaussian priors are placed on the user and item feature vectors as can be seen in Equations 3.3 and 3.4.

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} \left(\mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I})\right) \tag{3.3}$$

$$p(V|\sigma_V^2) = \prod_{j=1}^{M} \left(\mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I})\right) \tag{3.4}$$

The following equation (3.5) gives the log of the posterior distribution over the user and item features.

$$
\begin{aligned}
\ln p(U, V|R, \sigma^2, \sigma_V^2, \sigma_U^2) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 \\
&- \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_j \\
&- \frac{1}{2}\Big(\Big(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\Big)\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2\Big) + C
\end{aligned}
\tag{3.5}
$$

where C is a constant that does not depend on the parameters. Maximizing the log-posterior over item and user features is equivalent to minimizing the sum of squared errors objective function with quadratic regularization terms (Equation 3.6).

$$
E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N} \parallel U_i \parallel_F^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \parallel V_j \parallel_F^2 \tag{3.6}
$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$ and $\parallel . \parallel_F$ denotes the Frobenius norm. A local minimum of the objective function can be obtained using the gradient descent in U and V in Equation 3.6.

A simple linear-Gaussian model can generate predictions outside the range of the valid rating values. Therefore, the dot item between the user and item specific feature vectors passes through the logistic function.

$$
g(x) = 1/(1 + exp(-x)) \tag{3.7}
$$

which bounds the range of predictions to the interval $[0, 1]$.

$$
p(R|U, V, \sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\Big[\mathcal{N}\Big(R_{ij}|g(U_i^T V_j), \sigma^2\Big)\Big]^{I_{ij}} \tag{3.8}
$$

The ratings between $[1, K]$ are mapped to the interval $[0, 1]$ using the normalization function.

$$
t(x) = (x - 1)/(K - 1) \tag{3.9}
$$

so that the range of valid rating values matches the range of predictions made by the given model.

The PMF graphical model is given in Figure 3.1. V_j is item j's latent vector and U_i is user i's latent vector. σ^2 is the variance of Gaussian distribution.

The aim of PMF training is to obtain the matrix UV^T that minimizes the sum-squared distance to the target matrix R.

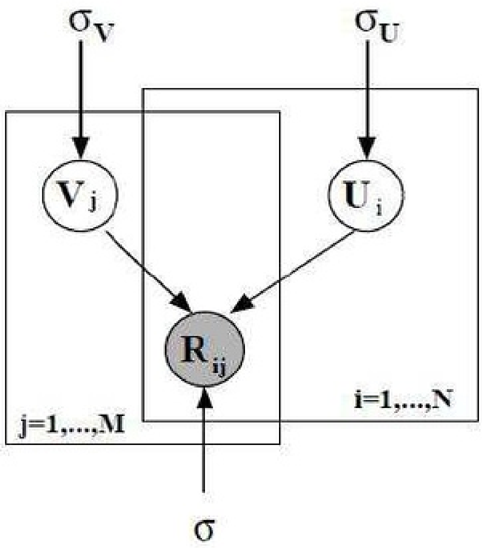

Figure 3.1: Probabilistic Matrix Factorization Graphical Model [50]

3.2 Related Work on PMF

A PMF model is presented to solve existing RS problems. PMF scales linearly according to the number of ratings and performs well on large and sparse datasets [50]. The main idea behind PMF is that user preferences are determined by some unobserved factors. The details of PMF algorithm are given in Section 3.1.

Salakhutdinov and Mnih aimed to improve PMF using a Bayesian approach [49]. In this approach, the model capacity is monitored automatically by integrating over all model parameters and hyperparameters. They claim that their approach has significantly better prediction accuracy than traditional PMF models when applied to the Netflix dataset. The graphical model of the Bayesian PMF is given in Figure 3.2. In this approach, the likelihood of observed ratings is modeled as PMF in Equation 3.10.

$$p(R|U,V,\alpha) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left(\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1})\right)^{I_{ij}} \tag{3.10}$$

Salakhutdinov and Mnih assumed prior distributions over the user and item latent vectors as Gaussian.

$$p(U|\mu_U, \Lambda_U) = \prod_{i=1}^{N} \mathcal{N}(U_i|\mu_U, \Lambda_U^{-1}) \tag{3.11}$$

$$p(V|\mu_V, \Lambda_V) = \prod_{j=1}^{M} \mathcal{N}(V_i|\mu_V, \Lambda_V^{-1}) \tag{3.12}$$

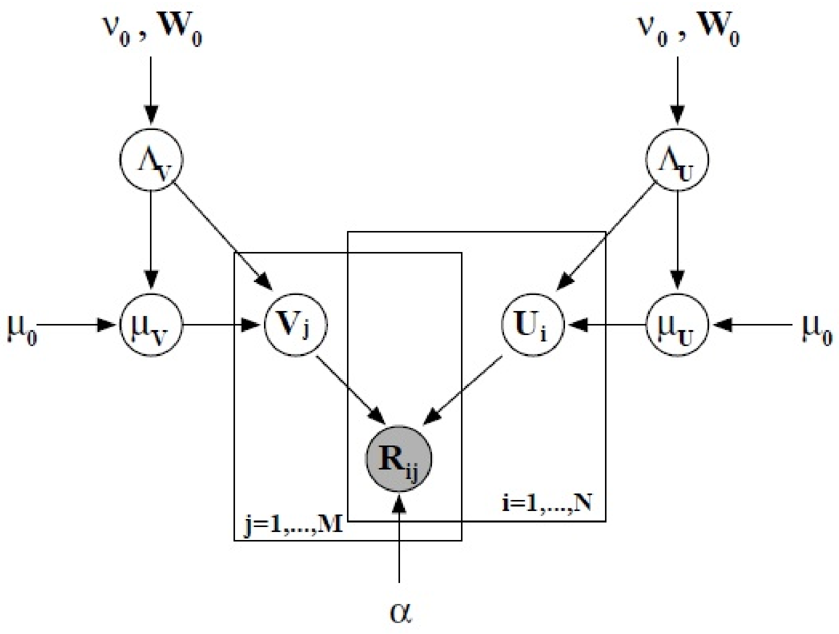

Figure 3.2: Bayesian Probabilistic Matrix Factorization Graphical Model [49]

The authors also placed Gaussian-Wishart priors on the user and item hyperparameters $\Theta_U = \{\mu_U, \Lambda_U\}$ and $\Theta_V = \{\mu_V, \Lambda_V\}$ as shown in Equation 3.13 and 3.14 respectively.

$$\begin{aligned} p(\Theta_U|\Theta_0) &= p(\mu_U|\Lambda_U)p(\Lambda_U) \\ &= \mathcal{N}(\mu_U|\mu_0, (\beta_0\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0, \nu_0) \end{aligned} \tag{3.13}$$

$$\begin{aligned} p(\Theta_V|\Theta_0) &= p(\mu_V|\Lambda_V)p(\Lambda_V) \\ &= \mathcal{N}(\mu_V|\mu_0, (\beta_0\Lambda_V)^{-1})\mathcal{W}(\Lambda_V|W_0, \nu_0) \end{aligned} \tag{3.14}$$

$\mathcal{W}$ is the Wishart distribution with ν_0 degrees of freedom and W_0 is a $D \times D$ scale matrix (Equation 3.15).

$$\mathcal{W}(\Lambda|W_0, \nu_0) = \frac{1}{C}|\Lambda|^{(\nu_0-D-1)/2}exp(-\frac{1}{2}Tr(W_0^{-1}\Lambda)) \tag{3.15}$$

where C is the normalizing constant and $\Theta_0 = \{\mu_0, \nu_0, W_0\}$.

Normally, in PMF only the user-item rating matrix is used as an input. Ma et al. fused user's social network graph with the user-item rating matrix in order to make more accurate predictions [33]. They proposed a method (SoRec) that integrates the social network structure and the user-item rating matrix, based on probabilistic factor analysis. The graphical model for SoRec is given in Figure 3.3.

These two different data resources are connected through the shared user latent feature space. The user latent feature space in the social network structure is the same as in the user-item

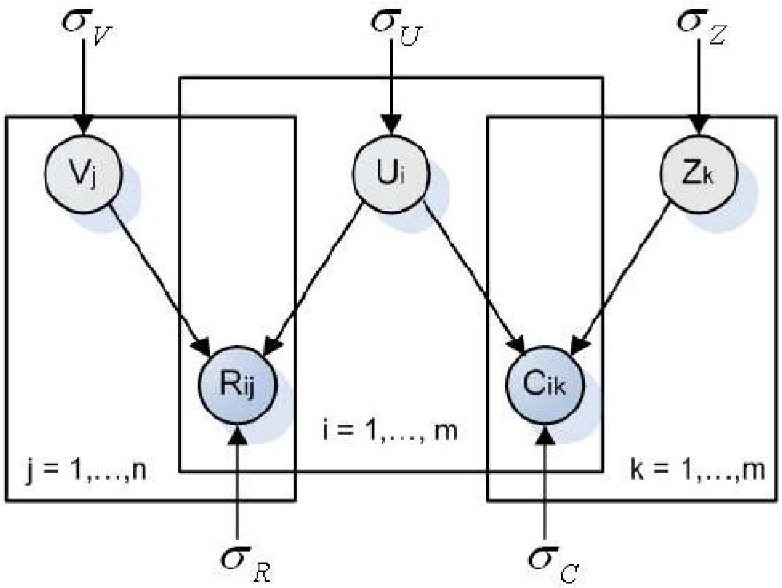

Figure 3.3: Graphical Model for Social Recommendation [33]

rating matrix. By performing factor analysis based on probabilistic matrix factorization, the user and item latent vectors are learned.

Koren et al. who were the members of the Netflix Prize competition winner group, improve matrix factorization techniques using additional information such as implicit feedback, temporal effects, and confidence levels [27].

Adams et al. proposed a study that incorporated side information into the PMF [1]. They proposed a dependent probabilistic matrix factorization (DPMF) that replaces scalar latent values with functions. These functions vary over the space of the additional information. The comparison of DPMF with PMF is given in Figure 3.4.

Ma et al. proposed a matrix factorization method with social regularization [34]. They incorporated social network information in to the matrix factorization model as a regularization term. There are two approaches developed in this study; average-based regularization and individual-based regularization. The first approach is based on the idea that people consult their friends for suggestions. They added a parameter that minimizes the tastes between the user and their friends into the objective function. The friends' tastes are weighted by the similarity function. PCC is used as a similarity function. Average-based recommendation model objective function is given in Equation 3.16.

$$\begin{aligned} \min_{U,V} \mathcal{L}(R, U, V) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\alpha}{2} \sum_{i=1}^{N} \|U_i - \frac{\sum_{f \epsilon \mathcal{F}^+_{(i)}} Sim(i,f) \times U_f}{\sum_{f \epsilon \mathcal{F}^+_{(i)}} Sim(i,f)}\|_F^2 \\ + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|V\|_F^2 \end{aligned} \tag{3.16}$$

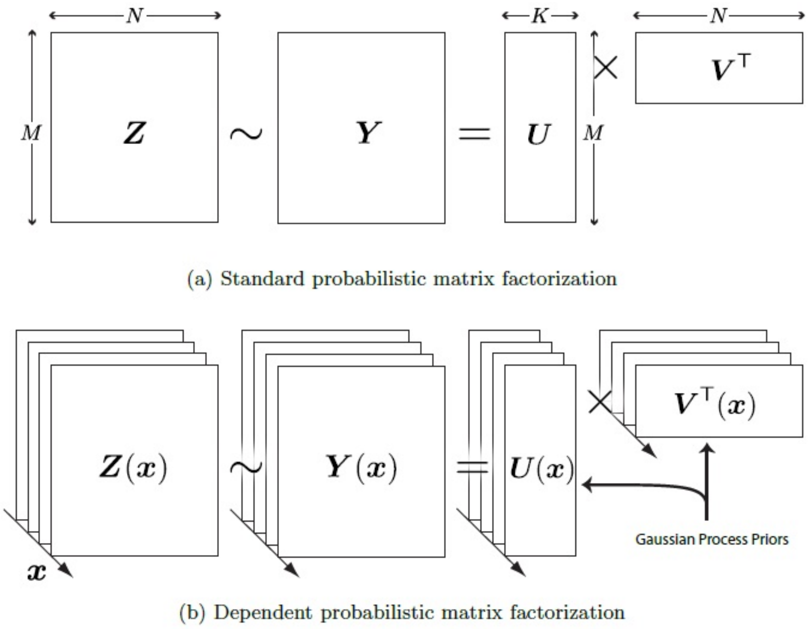

Figure 3.4: Dependent Probabilistic Matrix Factorization Graphical Model [1]

where $Sim(i, f) \ \epsilon \ [0, 1]$ shows the similarity between user u_i and user u_f. $\mathcal{F}^+_{(i)}$ is the set of friends of user u_i. λ_1 and λ_2 are the regularization parameters and $||.||_F$ denotes the Frobenius norm.

The authors claim that average-based regularization is insensitive to users who have friends with diverse preferences so the individual-based regularization proposes another regularization term that specifies constraints between the user and their friends individually. The same similarity function is also used in this regularization. The individual-based recommendation model objective function is given in Equation 3.17.

$$\begin{aligned} \min_{U,V} \mathcal{L}(R, U, V) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\beta}{2} \sum_{i=1}^{N} \sum_{f \epsilon \mathcal{F}^+_{(i)}} Sim(i, f) ||U_i - U_f||_F^2 \\ + \lambda_1 ||U||_F^2 + \lambda_2 ||V||_F^2 \end{aligned} \tag{3.17}$$

Zhou et al. proposed the Kernelized Probabilistic Matrix Factorization (KPMF) that incorporates additional side information [58]. Normally, the user (and item) latent vectors are independent with Gaussian priors in PMF. Each row (in U) and each column (in V^T) are updated independently from other rows or columns. The row and column independencies do not respect the covariance structure among the rows and columns. KPMF uses latent vectors spanning all the rows (and all the columns) with Gaussian Process priors. They claim that their approach successfully exploits the underlying covariances among the rows and columns of the data matrix simultaneously and enables integration of social network structure of users. The prior distribution of each column of the latent matrices, $U_{:,d}$ and $V_{:,d}$, is a zero-mean Gaussian process [46] in KPMF. Gaussian processes are a generalization of the multivariate Gaussian distribution, which is determined by a mean vector and covariance matrix. The

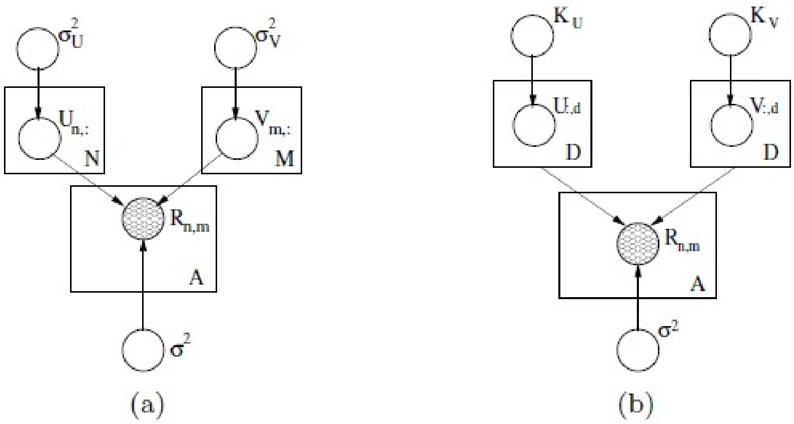

Figure 3.5: The Generative Processes of KPMF and Comparison with PMF's[58]

Gaussian Process $GP(m(x), k(x, x^{ß}))$ is specifed by a mean function $m(x)$ and a covariance function $k(x, x^{ß})$. Assuming K_U and K_V denote the full covariance matrix for rows of R (Ratings Matrix) and columns of R respectively the authors claim that using K_U and K_V in the priors forces the latent factorization to simultaneously capture the covariances among rows and columns. The generative model for KPMF is given as:

- 1. Generate $U_{:,d} \sim GP(0, K_U)$,
- 2. Generate $V_{:,d} \sim GP(0, K_V)$,
- 3. For each non-missing entry $R_{n,m}$ generate $R_{n,m} \sim \mathcal{N}(U_{n,:}V_{m,:}^T, \sigma^2)$, where σ is constant.

The generative processes of PMF and KPMF are given in Figure 3.5.

The likelihood of observed ratings is given in Equation 3.18.

$$p(R|U, V, \sigma^2) = \prod_{n=1}^{N}\prod_{m=1}^{M}\left(\mathcal{N}(R_{nm}|U_{n,:}V_{m,:}^T, \sigma^2)\right)^{\delta_{nm}} \tag{3.18}$$

with the priors over U and V given in Equations 3.19 and 3.20 respectively.

$$p(U|K_U) = \prod_{d=1}^{D} GP(U_{:,d}|0, K_U) \tag{3.19}$$

$$p(U|K_V) = \prod_{d=1}^{D} GP(V_{:,d}|0, K_V) \tag{3.20}$$

Liu et al. claimed that the user hyperparameters and item hyperparameters are different for each user and item vector [30] and proposed two approaches that improve Bayesian PMF proposed in [49]. These are the Bayesian Probabilistic Matrix Factorization with Social Relations

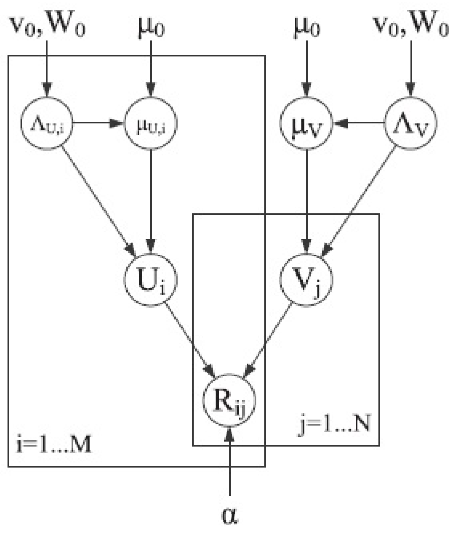

Figure 3.6: BPMFSR Graphical Model [30]

(BPMFSR) and the BPMFSR and Item Contents (BPMFSRIC) for recommendation. In these methods, the user and item hyperparameters are sampled according to the social relations and item contents. The BPMFSR method is based on the idea that it is not reasonable that hyperparameters Θ_U are the same for different users in BPMF. To fix this problem, BPMFSR assumes that every user has its own hyperparameters. As a result, the prior distribution of latent vector U is given as:

$$p(U) = \prod_{i=1}^{N} N(U_i|\mu_{U,i}, \Lambda_{U,i}^{-1}) \tag{3.21}$$

where $\Theta_{U,i} = \{\mu_{U,i}, \Lambda_{U,i}\}$ are the hyperparameters for user i. The user's preference is influenced by his/her friends. So in this approach, the conditional distribution over user hyperparameters is conditioned on the feature vectors of the user's friends.

BPMFSRIC method is based on the idea that it is not reasonable that hyperparameters Θ_V are the same for different items in BPMF. To fix this problem, BPMFSRIC assumes that every item has its own hyperparameters. As a result, the prior distribution of latent vector V is given as:

$$p(V) = \prod_{j=1}^{M} N(V_j|\mu_{V,j}, \Lambda_{V,j}^{-1}) \tag{3.22}$$

where $\Theta_{V,j} = \{\mu_{V,j}, \Lambda_{V,j}\}$ are the hyperparameters for item j. C_j denotes the item set in which every item links to item j. The item sets can be constructed using item tags and categories. To integrate item contents to PMF, the conditional distribution over item hyperparameters $\Theta_{V,j} = \{\mu_{V,j}, \Lambda_{V,j}\}$ is only conditioned on the feature vectors of items in C_j. The graphical models of BPMFSR and BPMFSRIC are given in Figures 3.6 and 3.7 respectively.

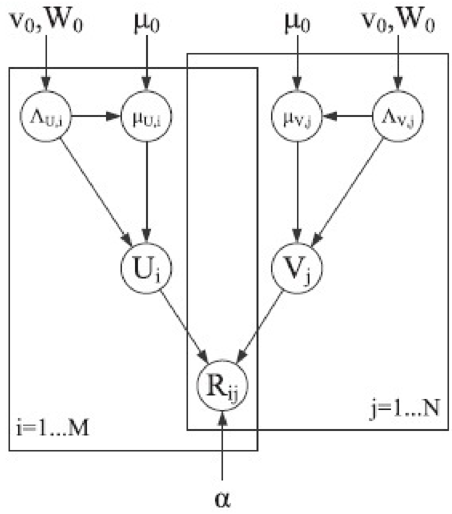

Figure 3.7: BPMFSRIC Graphical Model [30]

3.3 Improvements on PMF

The improvement efforts on PMF can be divided into 3 groups. The first group consists of the improvement efforts on the latent vector initialization process. The user and item latent vectors are initialized using different approaches. Some studies use side information, which is any information other than rating. This can be the date of rating, user's residential address or the trust network between the users. BPMFSR use the trust network and user-item rating matrix to create user and item latent vectors [30]. DPMF replaces scalar values in latent vectors with functions [1]. Other approaches use only user-item rating matrix. BPMF controls the model capacity automatically and only uses the rating matrix [50]. The study in this thesis can be located in this group. Since only information obtained from user-item rating matrix is used in the proposed approaches. We used the trimmed mean of the ratings in the base method. The rating matrix density and skewness of the ratings in addition to the trimmed mean are used in the improved methods in this thesis.

The second group comprises the efforts performed on the learning phase of PMF. SoRec [33] uses a trust network on the learning phase of PMF algorithm in addition to the user-item rating matrix. KPMF uses the covariances among the rows and columns of data matrix [58]. This method also exploits the social network structure. The matrix factorization with social regularization is another method that incorporates social network information as a regularization term into the objective function [34].

The last group contains the improvement efforts that go to post-processing. In these efforts, the obtained prediction values are updated based on the side information related to the user or item and again this side information can be in a broad range.

CHAPTER 4

THE PROPOSED METHODS

The first method proposed in this thesis is an initialization method that only uses the user-item rating matrix. This base method uses the trimmed-mean of the ratings in the training set to initialize user and item latent vectors. Then, four additional approaches are presented, involving the use of dataset characteristics, such as the rating matrix density and skewness.

4.1 An Initialization Method for the Latent Vectors in PMF based on the Mean Rating Information (INMED)

As mentioned in Chapter 3, users and items are characterized by latent factors where each user u and item v are mapped onto a latent feature space. Each rating r is computed as a dot product of user and item latent vectors (see Equation 3.1). Then, the squared error between the real rating value and predicted rating value is computed and utilized during the training phase.

The average rating score of most products is usually high in many e-platforms [12]. This is due to the fact that people tend to give higher ratings in general. As a result, this biased information can be positioned on U and V matrices by filling each element with a constant value c', which is calculated by decomposing the overall average rating in the dataset into U and V in a uniform way. The performance improvement is expected to be notable particularly in datasets with few ratings. In addition, PMF is known to have a local minima problem, particularly in very sparse datasets due to the small number of the training samples. As a consequence, the initialization of latent vectors with a constant value that produces the trimmed mean of ratings can result in the algorithm converging to a better local minima in very sparse datasets compared to other initialization techniques, such as random initialization.

Assuming that U' is $N \times D$ and V' is $M \times D$ on initial user and item latent matrices, respectively, u'_{ij} and v'_{kj} are the matrix entries in U' and V' matrices, respectively where $i \in [1, 2, ..., N]$, $j \in [1, 2, ..., D]$ and $k \in [1, 2, ..., M]$.

Let $\tilde{r}$ is taken as the mean value (trimmed mean) between 5% and 95% percentiles of all the rating values in the dataset. Due to the non-normality of the dataset, we cannot calculate the mean directly. In such cases, the median is a candidate metric that has a robustness of validity but it lacks the robustness of efficiency. The trimmed mean is a measure of location that tries to balance efficiency and validity.

Assuming that each entry on the matrices' entries is equal to a certain constant value c'.

$$u'_{ij} = c', for\ \forall i, j \tag{4.1}$$

$$v'_{kj} = \begin{cases} c', & \tilde{r} > (K+1)/2 \\ -c', & \tilde{r} \leq (K+1)/2 \end{cases}, for\ \forall k, j \tag{4.2}$$

Thus, using the Equation 3.1, an $N \times M$ matrix, R', is obtained where x_{ik} denotes the ith row and $k_t h$ column matrix element.

$$\forall i, k. \exists x_{ik} \in \mathbb{R}.\ x_{ik} = \pm {c'}^2 D \tag{4.3}$$

The objective here is to find the c'. So, using the Equations 3.7 and 3.9,

$$\forall i, k. \exists x_{ik} \in \mathbb{R}.\ g(x_{ik}) = t(\tilde{r}) \tag{4.4}$$

the following result is obtained;

$$c' = \begin{cases} sqrt\left(\frac{-ln\left(\frac{K-\tilde{r}}{\tilde{r}-1}\right)}{D}\right), & \tilde{r} > (K+1)/2 \\ sqrt\left(\frac{ln\left(\frac{K-\tilde{r}}{\tilde{r}-1}\right)}{D}\right), & \tilde{r} \leq (K+1)/2 \end{cases} \tag{4.5}$$

4.2 Improved INMED I: Based on INMED and the Rating Matrix Density (Constant Interval Center)

The method 4.1 was proposed to facilitate successful PMF convergence in very sparse datasets. However, there are other issues to be considered in the PMF initialization for a better convergence depending on the characteristics of the datasets.

The rating matrix density of datasets and the number of ratings per user and item in these datasets are quite variable. Even in very sparse datasets, a dataset can consist of users and items with a high number of ratings. Filling the matrices with constant values can work in cases where the number of ratings per user and item is low since there is a limited number of training samples. However, it may not be successful when the number of ratings per user and/or item is higher. As a remedy to the latter case, rather than assigning a constant value, a random initialization can be undertaken within a range to help PMF converge successfully and avoid the same local minima caused by constant initialization.

Definition 1. The sparseness value of a dataset is computed with the following formula:

$$sv_D = 1 - \left(\frac{R_{Count}}{N * M}\right) \tag{4.6}$$

where R_{Count} is the number of ratings in the dataset. N and M denote the number of users and the number of items, respectively.

Definition 2. The rating matrix density of a dataset is computed with the following formula:

$$rmd_D = \frac{R_{Count}}{N * M} \tag{4.7}$$

Based on these arguments, the Improved INMED I method is proposed.

Here, *RperU* is defined as the average number of ratings per user,

$$RperU = \frac{R_{Count}}{N} \tag{4.8}$$

and *RperI* as the average number of ratings per item:

$$RperI = \frac{R_{Count}}{M} \tag{4.9}$$

RperU and RperI values are significant for PMF initialization when the datasets are very sparse. Even though these values are related to the rating matrix density value, two datasets with the same rating density can have different RperU and RperI values. For example, item-dense datasets have many users and a limited number of items, therefore their RperI values are greater than their RperU values. On the other hand, in user-dense datasets, RperU is higher than RperI.

Assuming that *cMatrix* and *sMatrix* are matrices that will be initialized with a constant value and random initialization, respectively:

$$\begin{cases} cMatrix = U \;\; and \;\; sMatrix = V \;\; Ar = RperI \;\; if \;\; RperU < RperI \\ cMatrix = V \;\; and \;\; sMatrix = U \;\; Ar = RperU \;\; if \;\; RperU \geq RperI \end{cases} \tag{4.10}$$

where Ar is the threshold which that be used in Equation 4.13. cMatrix is initialized as follows:

$$cm'_{kj} = \Big\{ c' \quad , for \; \forall k, j \tag{4.11}$$

where *cm'* is the matrix entries in *cMatrix*.

The computation of c' is given in Equation 4.5 in Section 4.1.

sMatrix is initialized randomly within a range using a beta distribution that is constructed using the rating distribution characteristics of the given dataset.

Definition 3. Rating Fuzziness

In the majority of portals, users are forced to rate a given item with a discrete value within a specific range, usually between 1 and 5. However, this rating is fuzzy since the user is forced to round the number to the nearest order of magnitude. For example, although a user would like to rate an item as 2.7 or 3.2, s/he is obliged to give a rating of 3. Therefore, it is not possible to know the original number that was rounded by the user. The concept of rating fuzziness also help us generate the beta distribution more accurately since it provides more data points to fit.

Based on this rating fuzziness assumption, a beta distribution was generated from the rating distribution in the dataset as follows:

The new lower bound of the rating distribution was defined as [lower-bound-0.5] and the new upper bound as [upper-bound+0.5]. Assuming that r indicates different rating values that a user is allowed to give in a portal, *count*(r_i) indicates the number of ratings for the rating value r_i.

For example, in a system where the rating range is defined between 1 and 5, the new lower-bound and upperbound will be 0.5 and 5.5, respectively. r comprised values such as [1,2,3,4,5]. *count*(r_i) of each rating value r_i can be computed by counting the rating value r_i in the given dataset.

In the current study, the *count*(r_i) number of points were generated in $[r_i - 0.5; r_i + 0.5]$ using the equal-width discretization, therefore the mean of the dataset did not change. The newly created ratings between 0.5 and 5.5 were mapped between 0 and 1. After this preprocessing, the ratings were fit to the beta distribution and the parameters α and β were obtained. This preprocessing ensured that the distribution characteristics of the original dataset were maintained; i.e. it did not change the mean value of the ratings. sMatrix was initialized by generating random numbers, G, from the fitted beta distribution, BetaDist, as shown in Equation 4.12. Then the generated random numbers, G, were scaled to the interval $[c'_1, c'_2]$ as shown in Equation 4.13.

A constant value s' was produced based on the standard deviation of the distribution of the rating values and rating matrix density values. Then, a new range was defined using s' around the trimmed mean value. The INMED value of the upper and lower boundaries of this new range was calculated. For the datasets that were very sparse, these two points were very close to the INMED value. On the other hand, if the dataset was dense, these two values were relatively far from the INMED value.

$$G = BetaDist(\alpha, \beta) \tag{4.12}$$

$$sm'_{ij} = \begin{cases} h, & for\ \forall i,j\ \ h \in G'\ [c'_1, c'_2],\ \ Z_i > Ar \\ c', & for\ \forall i,j\ \ N_i \leq Ar \end{cases} \tag{4.13}$$

where Z_i is the number of ratings that i_{th} element has in the sMatrix. *sm'* is the matrix entries in *sMatrix*. G' $[c'_1, c'_2]$ denotes the random numbers generated from the fitted beta distribution, G, that was scaled within the interval between c'_1 and c'_2. The computation of c' is given in Equation 4.5 in Section 4.1.

$$c_1' = \begin{cases} sqrt\Big(\frac{-ln\Big(\frac{K-(\tilde{r}-s')}{(\tilde{r}-s')-1}\Big)}{D}\Big), & (\tilde{r}-s') > (K+1)/2 \\ -1 * sqrt\Big(\frac{ln\Big(\frac{K-(\tilde{r}-s')}{(\tilde{r}-s')-1}\Big)}{D}\Big), & (\tilde{r}-s') \leq (K+1)/2 \end{cases} \tag{4.14}$$

$$c_2' = \begin{cases} sqrt\Big(\frac{-ln\Big(\frac{K-(\tilde{r}+s')}{(\tilde{r}+s')-1}\Big)}{D}\Big), & (\tilde{r}+s') > (K+1)/2 \\ -1 * sqrt\Big(\frac{ln\Big(\frac{K-(\tilde{r}+s')}{(\tilde{r}+s')-1}\Big)}{D}\Big), & (\tilde{r}+s') \leq (K+1)/2 \end{cases} \tag{4.15}$$

$$s' = \sigma_D * (LA + \frac{UA - LA}{(1 + Q * exp(-1 * B * (-log10(rmd_D) - MG)))^{(1/\nu)}}) \tag{4.16}$$

where LA is the lower asymptote, UA is upper asymptote and B is the growth rate. σ_D is standard deviation of ratings in dataset D. rmd_D is the rating matrix density of the dataset D. MG is the time of maximum growth if $Q = \nu$. The flow chart of the Improved INMED I is given in Figure 4.1.

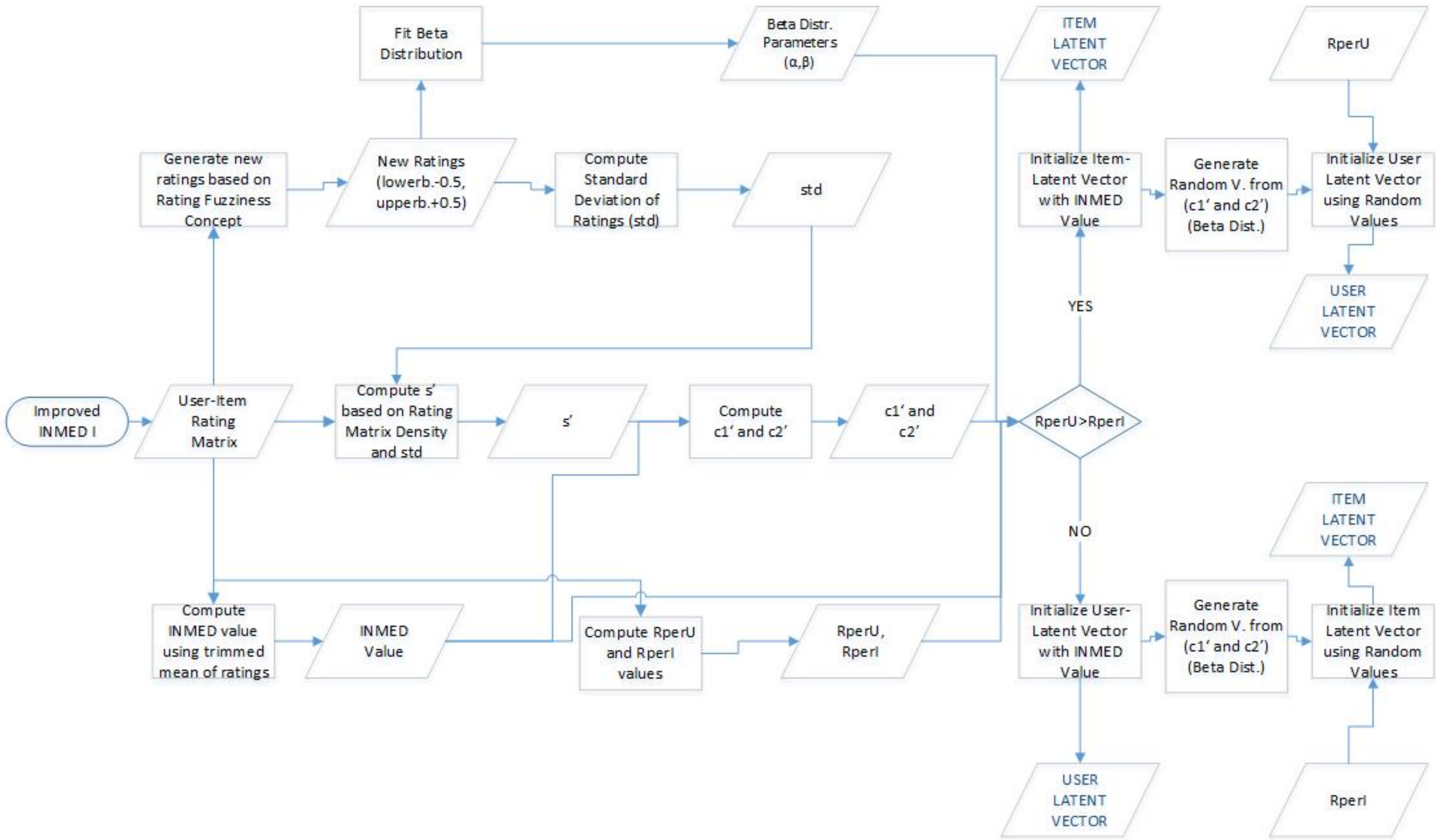

Figure 4.1: The Flow Chart of Improved INMED I Method

4.3 Improved INMED II: Based on INMED, the Rating Matrix Density, and the Skewness of the Rating Distribution (Constant Interval Center)

In the proposed method 4.2 a constant INMED value was used to initialize one of the latent vectors depending on the number of ratings per user and item. The INMED value was computed using the trimmed mean of the ratings as previously discussed in section 4.1. For certain rating distributions, the trimmed mean of the ratings was very close to 3.0, which resulted in an INMED value that was very close to zero. Therefore, the initialization values of one of the latent vectors were very close to zero. This made the training of PMF very difficult to converge. To correct this, a parameter d' was defined between 0 and 1.0. This parameter was computed using the skewness of ratings since the skewness was close to zero where the mean of distribution was around 3.0.

$$cm'_{kj} = \begin{cases} c' & , for \; \forall k, j \end{cases} \tag{4.17}$$

$$c' = \begin{cases} \left(\frac{-ln\left(\frac{K-(\tilde{r}+d')}{(\tilde{r}+d')-1}\right)}{D}\right), & (\tilde{r}+d') > (K+1)/2 \\ \left(\frac{ln\left(\frac{K-(\tilde{r}-d')}{(\tilde{r}-d')-1}\right)}{D}\right), & (\tilde{r}-d') \leq (K+1)/2 \end{cases} \tag{4.18}$$

$$d' = LA + \frac{UA - LA}{(1 + Q * exp(-1 * B * (-log10(skew_D) - MG)))^{(1/\nu)}} \tag{4.19}$$

where LA is the lower asymptote, UA is upper asymptote and B is the growth rate. $skew_D$ is the skewness of ratings on dataset D. MG is the time of maximum growth if $Q = \nu$.

sMatrix initialization is given in Equation 4.13. The flow chart of the Improved INMED II is given in Figure 4.2.

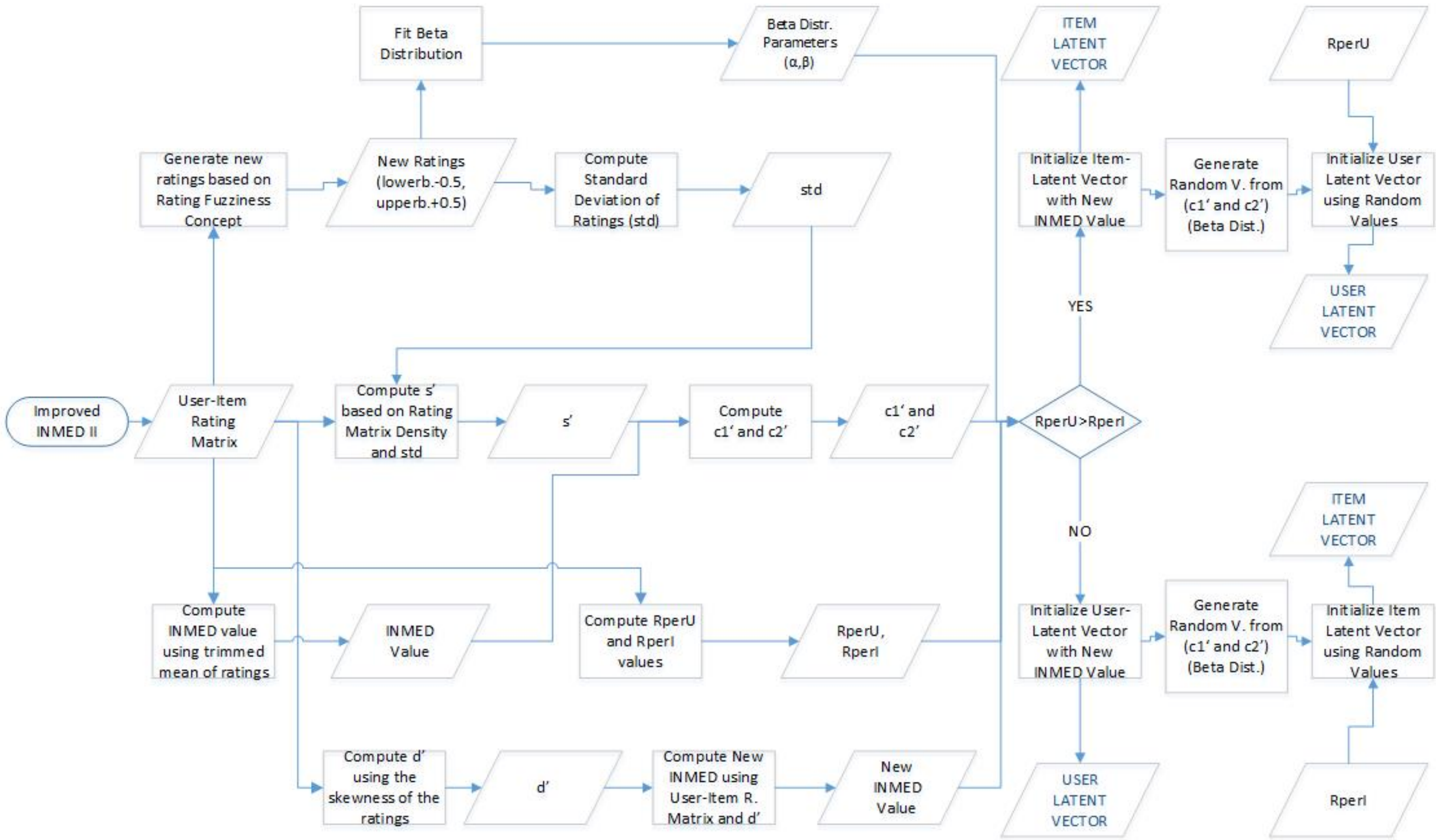

Figure 4.2: The Flow Chart of Improved INMED II Method

4.4 Improved INMED III: Based on INMED and the Rating Matrix Density (Variable Interval Center Based on the Rating Matrix Density)

In method 4.2, one of the latent vectors was initialized with the values that were randomly selected from an interval that was specified using the beta distribution. The interval center was assigned to the computed INMED value. The interval width was based on the rating matrix density.

In this method, the width of the interval and the interval center were determined based on the rating matrix density. The width of the interval was determined using the method 4.2. This method is different in terms of selecting the center of the interval. The new interval center was determined by moving the INMED value toward zero. The assumption here is that if the interval center is moved toward zero, the initialization values can be selected from both negative and positive real values. Depending on the interval width, the interval center that is close to zero ensures that the initial values are selected from both sides of zero. The generalized logistic function was used to compute the shift amount using the sparseness of the dataset. When the training data was dense, the interval center was moved toward zero. On the other hand, when the training data was sparse, the interval center was moved closer to the INMED value. In this method, both the interval center and the width of the interval moved back and forth based on the characteristics of the training data.

$$sm'_{ij} = \begin{cases} h,\ for\ \forall i,j \quad h \in G'\ [c'_1, c'_2], & \tilde{r} > (K+1)/2\ \ and \quad Z_i > Ar \\ c',\ \ for\ \ \forall i,j & \tilde{r} > (K+1)/2\ \ and\ \ Z_i \leq Ar \\ h,\ for\ \forall i,j \quad h \in G'\ [c'_3, c'_4], & \tilde{r} \leq (K+1)/2\ \ and \quad Z_i > Ar \\ -c',\ \ for\ \ \forall i,j & \tilde{r} \leq (K+1)/2\ \ and\ \ Z_i \leq Ar \end{cases} \tag{4.20}$$

The computation of c' is given in Equation 4.5 in Section 4.1.

$$c'_1 = \begin{cases} sqrt\Big(\frac{-ln\left(\frac{K-(\tilde{r}-l'-s')}{(\tilde{r}-l'-s')-1}\right)}{D}\Big), & (\tilde{r}-l'-s') > (K+1)/2 \\ -1 * sqrt\Big(\frac{ln\left(\frac{K-(\tilde{r}-l'-s')}{(\tilde{r}-l'-s')-1}\right)}{D}\Big), & (\tilde{r}-l'-s') \leq (K+1)/2 \end{cases} \tag{4.21}$$

$$c'_2 = \begin{cases} sqrt\Big(\frac{-ln\left(\frac{K-(\tilde{r}-l'+s')}{(\tilde{r}-l'+s')-1}\right)}{D}\Big), & (\tilde{r}-l'+s') > (K+1)/2 \\ -1 * sqrt\Big(\frac{ln\left(\frac{K-(\tilde{r}-l'+s')}{(\tilde{r}-l'+s')-1}\right)}{D}\Big), & (\tilde{r}-l'+s') \leq (K+1)/2 \end{cases} \tag{4.22}$$

$$c'_3 = \begin{cases} sqrt\Big(\frac{-ln\left(\frac{K-(\tilde{r}+l'-s')}{(\tilde{r}+l'-s')-1}\right)}{D}\Big), & (\tilde{r}+l'-s') > (K+1)/2 \\ -1 * sqrt\Big(\frac{ln\left(\frac{K-(\tilde{r}+l'-s')}{(\tilde{r}+l'-s')-1}\right)}{D}\Big), & (\tilde{r}+l'-s') \leq (K+1)/2 \end{cases} \tag{4.23}$$

$$c'_4 = \begin{cases} sqrt\Big(\frac{-ln\left(\frac{K-(\tilde{r}+l'+s')}{(\tilde{r}+l'+s')-1}\right)}{D}\Big), & (\tilde{r}+l'+s') > (K+1)/2 \\ -1 * sqrt\Big(\frac{ln\left(\frac{K-(\tilde{r}+l'+s')}{(\tilde{r}+l'+s')-1}\right)}{D}\Big), & (\tilde{r}+l'+s') \leq (K+1)/2 \end{cases} \tag{4.24}$$

$$s' = \sigma_D * (LA + \frac{UA - LA}{(1 + Q * exp(-1 * B * (-log10(rmd_D) - MG)))^{(1/\nu)}}) \tag{4.25}$$

$$l' = \sigma_D * (LA + \frac{UA - LA}{(1 + Q * exp(-1 * B * (-log10(rmd_D) - MG)))^{(1/\nu)}}) \tag{4.26}$$

where LA is the lower asymptote, UA is upper asymptote and B is the growth rate. σ_D is standard deviation of ratings in the dataset D. rmd_D is the rating matrix density of the dataset D. cMatrix initialization is given in Equation 4.13. MG is the time of maximum growth when $Q = \nu$. The flow chart of the Improved INMED III is given in Figure 4.3.

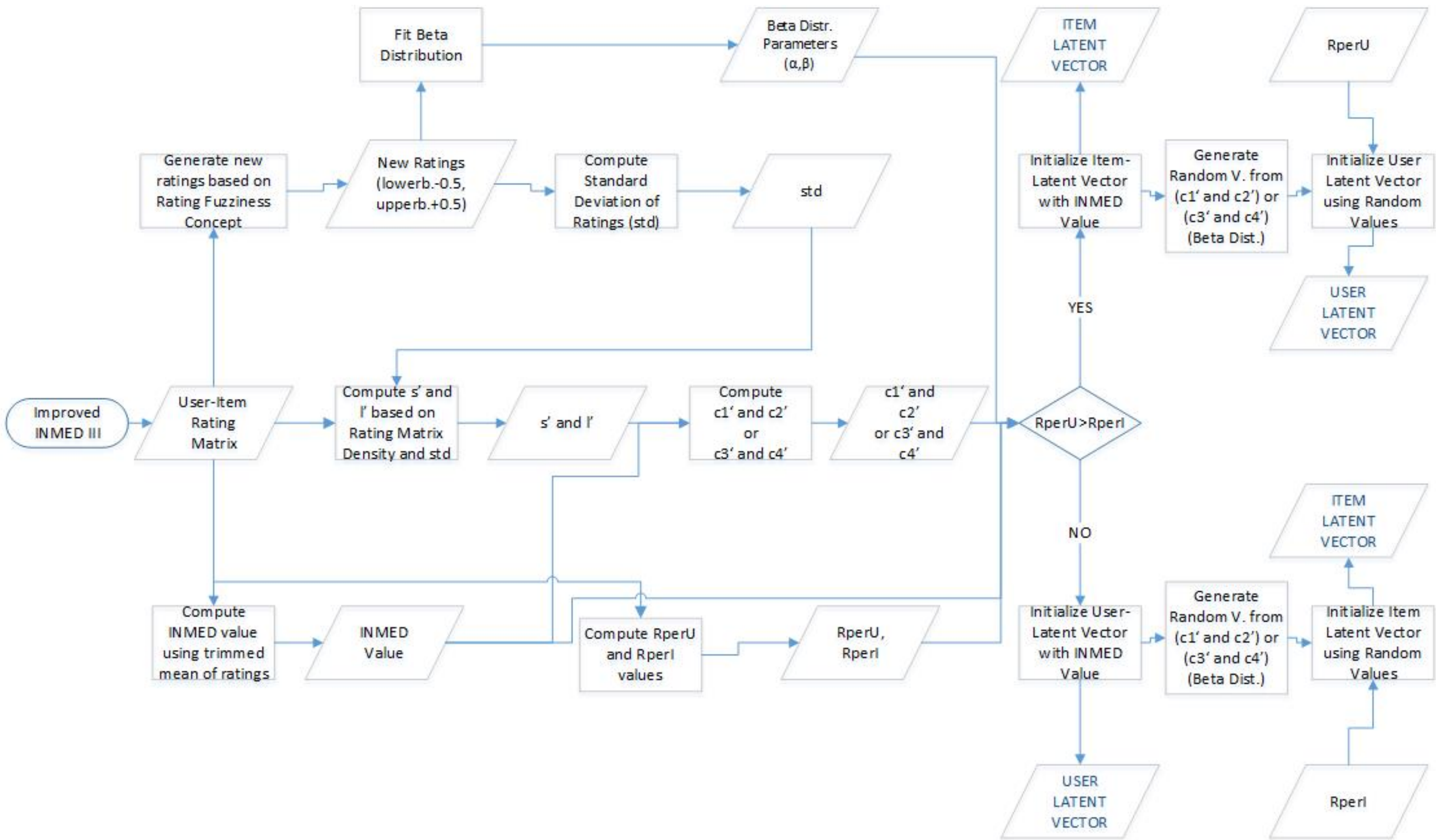

Figure 4.3: The Flow Chart of Improved INMED III Method

4.5 Improved INMED IV: Based on INMED, the Rating Matrix Density, and the Skewness of the Rating Distribution (Variable Interval Center Based on the Rating Matrix Density)

This method combines the approaches proposed in the previous sections. In this method, one of the latent vectors was initialized as described in method 4.4 with the numbers selected randomly from the intervals that were moved toward zero. The other latent vector was initialized as described in method 4.3 using the dislocated INMED constant based on skewness. The decision regarding which latent vector would be initialized with constant values, and which would be initialized with values from the specified intervals was made by considering the number of unique users and the number of unique items in the training data. The user latent vectors were initialized with constant values where there were more users than items. On the other hand, when the number of ratings per user was high, the interval that was calculated using method 4.4 was used for the initialization. The flow chart of the Improved INMED IV is given in Figure 4.4.

The characteristics of the datasets that were used in the proposed INMED-based methods are given in Table 4.1.

Table 4.1: The characteristics of the datasets that were used in INMED-based methods.

Characteristics	INMED	I.INMED I	I.INMED II	I.INMED III	I.INMED IV
Trimmed Mean	Yes	Yes	Yes	Yes	Yes
Rating Matrix Density	No	Yes	Yes	Yes	Yes
Ratings Skewness	No	No	Yes	No	Yes
Ratings per User/Item	No	Yes	Yes	Yes	Yes

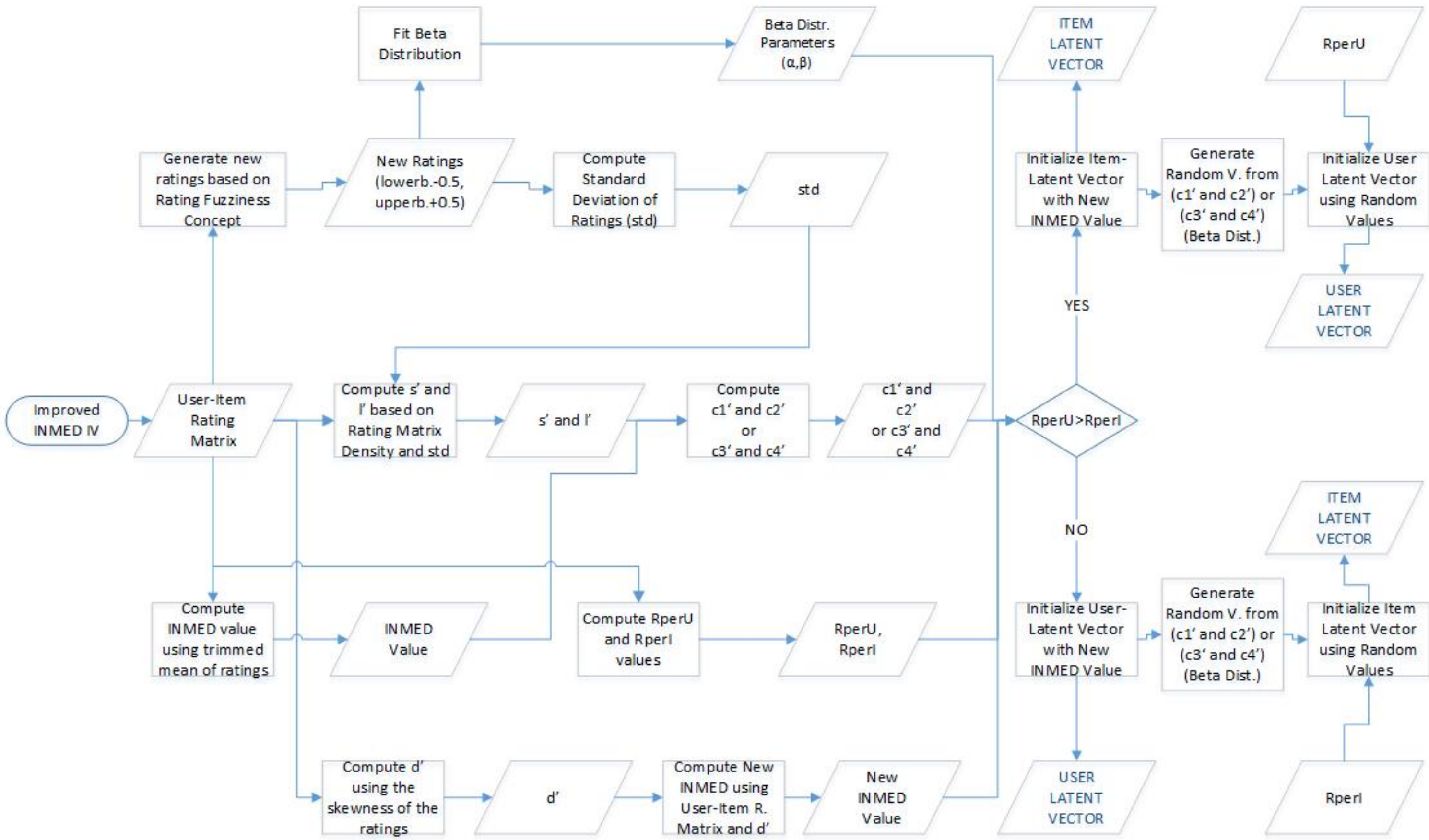

Figure 4.4: The Flow Chart of Improved INMED IV Method

CHAPTER 5

EVALUATION

5.1 Dataset Characteristics

Seven datasets were used to perform the experiments. Three were crawled by the author from Epinions.com in March 2012. Epinions is a social e-platform with a very large user community, in which users rate and write reviews about products. The rating scale ranges between 1 and 5. The crawler was based on Massa's idea [42]. At first, an initial user with sufficient amount of trust information was chosen. Starting from this initial user, all the users who was on the web of trust of the initial user and those who trusted the initial user were visited. All the items that had been reviewed by the visited users were also extracted. At the end, the crawler listed the users who had at least one of the following: ratings, issued or received trust statements or review ratings. Three datasets were obtained (Movies (*M_ALL*), Home and Garden (*HG_ALL*) and Wellness and Beauty (*WB_ALL*)). The remaining four datasets (MovieLens, LastFM, Douban, Epinions) were downloaded from [18, 22, 23, 52], respectively.

Additional datasets were created using two techniques; subsampling the datasets and eliminating specific items. Subsampling three datasets, different datasets were created with different distribution characteristics. Six different distributions were created with different ratings between 1 and 5. The percentage of each rating in each distribution is given in Table 5.1. The first distribution exhibits a positive linear trend where the majority of the ratings are gathered around 4 and 5. The second distribution has a linear trend. The third distribution shows the Gaussian characteristics where the mean of the ratings is three. Each rating gets the 20% of the total ratings in the fourth distribution. The fifth and sixth distributions are the mirrors of the second and first distributions, respectively.

Since the median rating (3 in the subsamples) was very important for the proposed INMED formula, subsamples were created using the means that could be either less or greater than the middle point. The new datasets were named as D1,...,D6, E1,...,E6 and M1,...,M6 subsampled from the Douban, Epinions and MovieLens datasets respectively, giving a total of 18 datasets. The rating distributions of the datasets Di, Ei, and Mi were the same where i indicates the distribution ID in Table 5.1.

The second technique used to create new datasets was to eliminate certain items from the original datasets. A significant number of the items in 3 datasets (M-ALL, HG-ALL, and WB-ALL) had been rated only once. Therefore, in order to effectively measure the performance of the proposed methods, some variations of the datasets were created where the items having

Table 5.1: The Rates of Ratings 1-5 in 6 Subsampled Datasets

Distribution ID	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
1	0.05	0.10	0.15	0.30	0.40
2	0.07	0.13	0.20	0.27	0.33
3	0.10	0.20	0.40	0.20	0.10
4	0.20	0.20	0.20	0.20	0.20
5	0.33	0.27	0.20	0.13	0.07
6	0.40	0.30	0.15	0.10	0.05

less than or equal to 1, 3 and 5 ratings were discarded. These datasets were named with a prefix denoting the category type (*M* for Movie, *HG* for Home and Garden, *WB* for Wellness and Beauty), followed by a number that indicates the upper bound frequency of the discarded items. For example, the *HG*_3 dataset comprised the Home and Garden dataset from which the products having less than and equal to 3 ratings were removed. These abbreviations will be used in the following sections of this thesis.

As a result, 34 datasets were obtained 27 of which were different subsamples of the original seven datasets.

Table 5.2 displays the maximum number of ratings given by each user and assigned to each item. It also shows the average number of ratings for all users and items. For example, in the *M_ALL* dataset, the average number of ratings given by each user and assigned to each item are 8.14 and 5.49, respectively. The maximum number of ratings given by a user is 1662 and the item with the highest number of ratings received 843 ratings.

According to the characteristics of all the datasets given in Table 5.3, the user-item rating matrices are sparse in all the datasets except MovieLens.

The motivation behind the approach in this thesis was to incorporate people's biases regarding the ratings into the latent vectors. As stated before, people are usually more inclined to rate an item when they are satisfied with it. Table 5.4 illustrates the distribution of ratings for all the datasets. Each column shows the number of ratings for each category. For example, in the M_ALL dataset, 12516 people rated an item as 1 and 18026 people rated an item as 2. The percentage of people who rated 4 or 5 was considerably high with 65% in the M-ALL dataset and 80% in the HG-ALL dataset.

5.2 Experimental Settings

In this thesis, 70% of the data was used for training, 10% for validation and the remaining 10% for testing. The percentages were determined in line with the literature [33, 35, 32].

The experiments were conducted using MATLAB. In each experiment, a 10 fold cross validation was used. The fold divisions were carried out after randomizing the dataset and repeated three times.In the literature, the PMF experiments were carried out with latent vector dimen-

Table 5.2: The Basic Statistics of All Datasets

Statistics	Max. # of Ratings		Avg. # of Ratings	
	User	Product	User	Product
D1	27	207	1.52	5.73
D2	23	182	1.54	5.63
D3	34	165	1.64	5.53
D4	46	222	1.63	5.49
D5	80	394	1.76	5.51
D6	89	490	1.83	5.52
E1	83	150	2.77	1.88
E2	79	202	2.81	1.90
E3	112	284	2.90	2.00
E4	97	309	2.84	2.02
E5	91	414	2.87	2.21
E6	89	462	2.88	2.29
M1	114	248	8.82	16.21
M2	123	191	8.83	16.14
M3	116	126	9.04	15.78
M4	119	139	8.91	15.69
M5	146	130	9.18	15.55
M6	193	167	9.33	15.41
M_ALL	1662	843	8.14	5.49
M_1	1142	843	7.45	10.84
M_3	975	843	6.81	19.29
M_5	865	843	6.43	26.13
HG_ALL	854	260	4.59	2.22
HG_1	268	260	3.38	5.07
HG_3	132	260	2.72	10.1
HG_5	100	260	2.42	14.55
WB_ALL	1162	168	6.69	2.39
WB_1	572	168	5.22	5.24
WB_3	311	168	4.23	9.94
WB_5	197	168	3.73	14.14
Epinions	1023	2026	16.55	4.76
LastFM	50	611	49.07	5.27
MovieLens	2314	3428	165.60	269.89

Table 5.3: The Characteristics of All Datasets

Datasets	# of Users	# of Items	# of Ratings	#ofUsers/#ofItems	Rating Matrix Density
D1	32857	8730	50000	3.76	1.74e-4
D2	32387	8880	50000	3.65	1.74e-4
D3	30501	9048	50000	3.37	1.81e-4
D4	30163	9105	50000	3.36	1.79e-4
D5	28349	9077	50000	3.12	1.94e-4
D6	27279	9061	50000	3.01	2.02e-4
E1	18020	26548	50000	0.68	1.05e-4
E2	17802	26253	50000	0.68	1.07e-4
E3	17216	25053	50000	0.69	1.16e-4
E4	17629	24722	50000	0.71	1.15e-4
E5	17426	22626	50000	0.77	1.27e-4
E6	17356	21833	50000	0.79	1.32e-4
M1	5669	3084	50000	1.84	2.86e-3
M2	5663	3097	50000	1.83	2.85e-3
M3	5532	3168	50000	1.75	2.85e-3
M4	5614	3186	50000	1.76	2.80e-3
M5	5447	3215	50000	1.69	2.86e-3
M6	5358	3244	50000	1.65	2.88e-3
M_ALL	21346	31663	173818	0.67	2.57e-4
M_1	21015	14440	156595	1.46	5.16e-4
M_3	20537	7247	139820	2.83	9.40e-4
M_5	20201	4969	129818	4.07	1.29e-3
HG_ALL	10140	20898	46503	0.49	2.20e-4
HG_1	9427	6288	31893	1.50	5.38e-4
HG_3	8308	2237	22594	3.71	1.22e-3
HG_5	7551	1256	18278	6.01	1.92e-3
WB_ALL	8209	22984	54945	0.36	2.91e-4
WB_1	7568	7544	39505	1.00	6.92e-4
WB_3	6809	2896	28787	2.35	1.46e-3
WB_5	6265	1652	23360	3.79	2.26e-3
Epinions	40163	139738	664824	0.29	1.19e-4
LastFM	1892	17632	92834	0.11	2.78e-3
MovieLens	6040	3706	1000209	1.63	4.47e-2

Table 5.4: Ratings Distributions of All Datasets

Datasets	1	2	3	4	5	Total
D1	2500	5000	7500	15000	20000	50000
D1 (%)	0.05	0.10	0.15	0.30	0.40	1.00
D2	3500	6500	10000	13500	16500	50000
D2 (%)	0.07	0.13	0.20	0.27	0.33	1.00
D3	5000	10000	20000	10000	5000	50000
D3 (%)	0.10	0.20	0.40	0.20	0.10	1.00
D4	10000	10000	10000	10000	10000	50000
D4 (%)	0.20	0.20	0.20	0.20	0.20	1.00
D5	16500	13500	10000	6500	3500	50000
D5 (%)	0.33	0.27	0.20	0.13	0.07	1.00
D6	20000	15000	7500	5000	2500	50000
D6 (%)	0.40	0.30	0.15	0.10	0.05	1.00
E1	2500	5000	7500	15000	20000	50000
E1 (%)	0.05	0.10	0.15	0.30	0.40	1.00
E2	3500	6500	10000	13500	16500	50000
E2 (%)	0.07	0.13	0.20	0.27	0.33	1.00
E3	5000	10000	20000	10000	5000	50000
E3 (%)	0.10	0.20	0.40	0.20	0.10	1.00
E4	10000	10000	10000	10000	10000	50000
E4 (%)	0.20	0.20	0.20	0.20	0.20	1.00
E5	16500	13500	10000	6500	3500	50000
E5 (%)	0.33	0.27	0.20	0.13	0.07	1.00
E6	20000	15000	7500	5000	2500	50000
E6 (%)	0.40	0.30	0.15	0.10	0.05	1.00
M1	2500	5000	7500	15000	20000	50000
M1 (%)	0.05	0.10	0.15	0.30	0.40	1.00
M2	3500	6500	10000	13500	16500	50000
M2 (%)	0.07	0.13	0.20	0.27	0.33	1.00
M3	5000	10000	20000	10000	5000	50000
M3 (%)	0.10	0.20	0.40	0.20	0.10	1.00
M4	10000	10000	10000	10000	10000	50000
M4 (%)	0.20	0.20	0.20	0.20	0.20	1.00
M5	16500	13500	10000	6500	3500	50000
M5 (%)	0.33	0.27	0.20	0.13	0.07	1.00
M6	20000	15000	7500	5000	2500	50000
M6 (%)	0.40	0.30	0.15	0.10	0.05	1.00
M_ALL	12516	18026	30815	54870	57591	173818
M_ALL (%)	0.07	0.10	0.18	0.32	0.33	1.00
M_1	11275	16244	27029	48993	53054	156595
M_1 (%)	0.07	0.10	0.17	0.31	0.34	1.00
M_3	10039	14585	23686	43128	48382	139820
M_3 (%)	0.07	0.10	0.17	0.31	0.35	1.00
M_5	9362	13560	21779	39650	45467	129818
M_5 (%)	0.07	0.10	0.17	0.31	0.35	1.00
HG_ALL	2853	2711	3936	13558	23445	46503
HG_ALL (%)	0.06	0.06	0.08	0.29	0.50	1.00
HG_1	2182	2053	2622	8666	16370	31893
HG_1 (%)	0.07	0.06	0.08	0.27	0.51	1.00
HG_3	1604	1486	1866	5985	11653	22594
HG_3 (%)	0.07	0.07	0.08	0.26	0.52	1.00
HG_5	1336	1239	1488	4783	9432	18278
HG_5 (%)	0.07	0.07	0.08	0.26	0.52	1.00
WB_ALL	3594	4497	6570	16705	23579	54945
WB_ALL (%)	0.07	0.08	0.12	0.30	0.43	1.00
WB_1	2735	3363	4532	11436	17439	39505
WB_1 (%)	0.07	0.09	0.11	0.29	0.44	1.00
WB_3	2086	2472	3160	8038	13031	28787
WB_3 (%)	0.07	0.09	0.11	0.28	0.45	1.00
WB_5	1748	2027	2483	6417	10685	23360
WB_5 (%)	0.07	0.09	0.11	0.27	0.46	1.00
Epinions	43228	50678	75525	194340	301053	664824
Epinions (%)	0.07	0.08	0.11	0.29	0.45	1.00
LastFM	3205	18550	57004	13444	631	92834
LastFM (%)	0.03	0.20	0.61	0.14	0.01	1.00
MovieLens	56174	107557	261197	348971	226310	1000209
MovieLens (%)	0.06	0.11	0.26	0.35	0.23	1.00

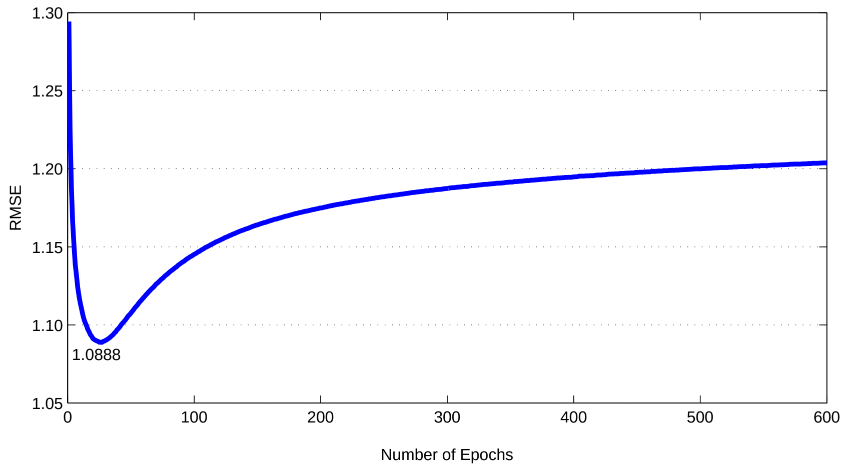

Figure 5.1: The validation dataset RMSE on 600 epochs (Movies)

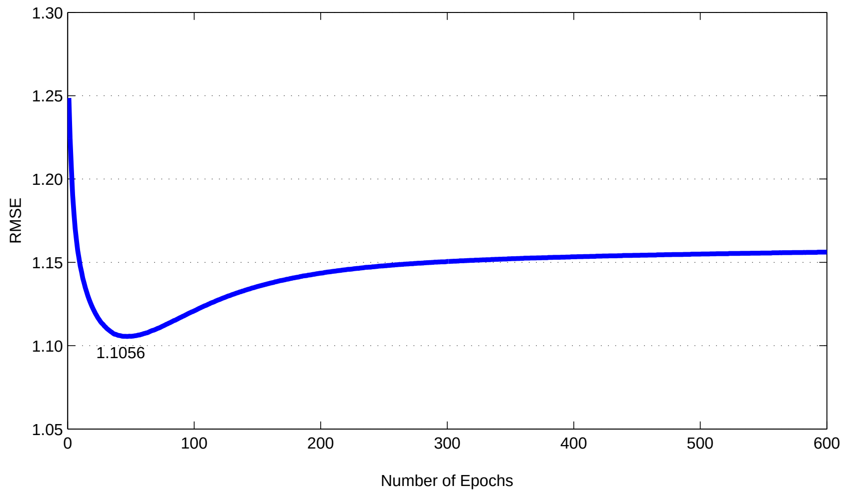

Figure 5.2: The validation dataset RMSE on 600 epochs (Home and Garden)

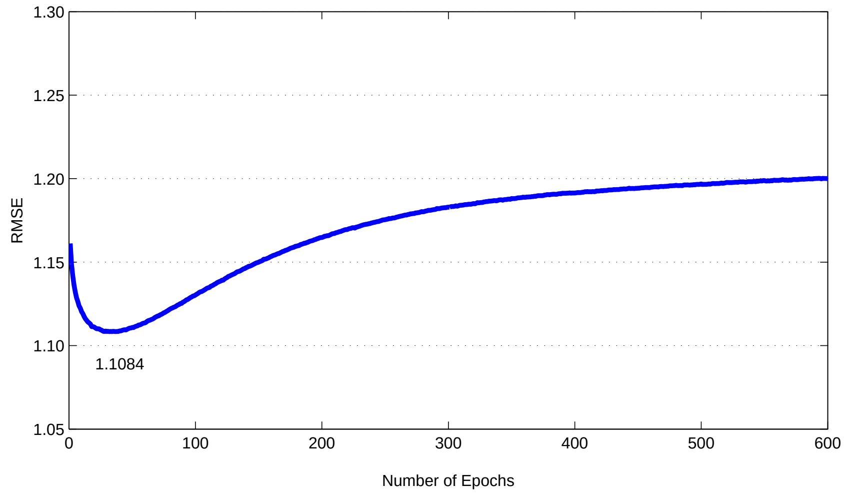

Figure 5.3: The validation dataset RMSE on 600 epochs (Wellness and Beauty)

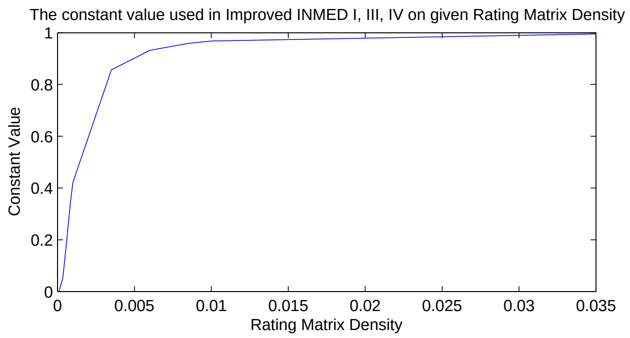

Figure 5.4: The produced value using generalized logistic function on given rating matrix density. This value is multiplied with standard deviation of ratings to compute s′ and l′.

sions of a size equivalent to 5 or mostly 10 [33, 35, 32]. As a result, in all the experiments, the number of dimensions of feature vectors was set to 10.

The generalized logistic function was used to map the rating matrix density and skewness of ratings in the interval range [0,1]. The objective here was to map the lower rating density values around 0 and the higher ones close to 1. Similarly, high skewness values were mapped close to 0 and low ones were mapped to higher values. The parameters of the generalized logistic function in Equations 4.16, 4.25, and 4.26 were specified as follows: $UA = 1$, $LA = 0$, $Q = 1/3$, $B = -3$, $MG = 3$ and $\nu = 1/-log10(rmd_D)$. The produced constant values on given rating matrix densities are shown on Figure 5.4. The same parameters of the generalized logistic function in Equation 4.19 were determined as follows: $UA = 1$, $LA = 0$, $Q = 1/10$, $B = -10$, $MG = 0.1$ and $\nu = 1/abs(skew_D)$.

The parameters required for the PMF training were carefully selected. Various values of learning rate and momentum values were analyzed and those that produced more successful results for all the initialization methods (the proposed method and other compared initialization methods) were chosen. The dataset was subdivided into mini-batches and the latent vectors were updated following each mini-batch. The mini-batch size was chosen as 1000. The number of batches is the amount that covers the related training data. The maximum number of epochs was chosen as 600. The regularization parameter was set to 0.001. The momentum and the learning rate were set to 0.9 and 0.005, respectively. During the training of datasets, when the RMSE on the validation set increased at least three times, the first of these epochs was selected as a candidate point. After selecting five candidate points, the epoch number with the minimum validation RMSE was determined as the termination point. If during the training phase, the algorithm could not find five candidate points before reaching the maximum epoch number of 600, the minimum of the selected candidate points was specified as the termination point. An example for M-ALL dataset is given in Figure 5.1. In M-ALL dataset, the minimum validation RMSE was obtained around epoch 27 with an RMSE of 1.0888. In HG-ALL dataset, the RMSE was converged around epoch 50 when the RMSE was 1.1056 (Figure 5.2). In WB-ALL dataset, the training was terminated around epoch 37 when the RMSE was 1.1084 (Figure 5.3). In all three runs illustrated in Figures 5.1, 5.2, and 5.3, the latent vectors were initialized randomly from a Gaussian distribution with $\mu = 0.5$ and $\sigma = 0.1$.

5.3 Performance Validation

In each fold, the values of the initialization matrices were calculated applying the proposed algorithms to the training data. Then, different initialization methods and values were compared. The details of the compared initialization methods are given in Table 5.5.

INMED computes the mean value (trimmed mean) between 5% and 95% percentiles of all the rating values for each fold in the training of all datasets. For the purpose of comparison, INMED was also computed using the trimmed mean value between 15% and 85% percentiles and between 25% and 75% percentiles of the training data. Using these trimmed means for each fold of all datasets, the initialization values of user and product latent vectors were computed.

The PMF RMSE results for D1,...,D6, E1,...,E6, and M1,...,M6 are given in Tables 5.6, 5.7 and 5.8, respectively. The RMSE average and standard deviation of all the runs in the experiments are presented. Tables 5.9, 5.10, 5.11, and 5.12 shows the PMF RMSE results of the datasets M_ALL,...,M_5, HG_ALL,...,HG_5, WB_ALL,...,WB_5 and original datasets (Epinions, MovieLens, and LastFM), respectively.

The results show that in dataset $D1$ all the improved INMED methods (I, II, III, IV) produced better results along with C-0.3, INMED(5-95), INMED(Mean), INMED(Median), Gaussian R1, and Gaussian R2. The improved INMED methods gave the best results in dataset $D2$ along with INMED(5-95), INMED(Mean), Gaussian R1, and Gaussian R2. In datasets $D3$ and $D4$, the Improved INMED II and IV produced the better results. On the other hand, none of the improved INMED methods gave the best result in dataset $D6$. These results indicate that when the mean of the ratings is larger than 3.0, all improved INMEDs produce the best results. If the mean is very close to 3.0, both Improved INMED II and IV make the best predictions. When the mean of the ratings is considerably smaller than 3.0, the proposed methods do not produce better results than other initialization approaches.

Table 5.5: The Description of Initialization Methods

Initialization Method	Description
C-v	The values of all the user and item latent vectors are initialized between 0 and 0.8 with an increment of 0.1. We reported the result of each initialization method's performance where each experiment was labeled as C-v where v indicates the constant initialization value. For example, the experiment C-0.1 and the experiment C-0.2 refer to the experiments where the matrices are initialized with 0.1 and 0.2 values, respectively.
INMED(5-95)	INMED initialization where the trimmed mean is calculated as the mean of all rating values between 5% and 95% percentiles.
INMED(15-85)	INMED initialization,where the trimmed mean is calculated as the mean of all rating values between 15% and 85% percentiles.
INMED(25-95)	INMED initialization,where the trimmed mean is calculated as the mean of all rating values between 25% and 75% percentiles.
Uniform R.	The initialization values are generated randomly from a uniform distribution in the interval [0, 1].
SND	The initialization values are generated randomly from a Standard Normal Distribution
0.1*SND	0.1 times The initialization values generated randomly from a Standard Normal Distribution
0.5 + 0.1*SND	0.5 plus 0.1 times The initialization values generated randomly from a Standard Normal Distribution
INMED(Mean)	INMED initialization,using the mean of the ratings
INMED(Median)	INMED initialization,using the median of the ratings
Gaussian R1	Gaussian distribution with μ=INMED(5,95) and σ=0.01
Gaussian R2	Gaussian distribution with μ=INMED(5,95) and σ=0.05
IINMED_I	Improved INMED I
IINMED_II	Improved INMED II
IINMED_III	Improved INMED III
IINMED_IV	Improved INMED IV

Although the distribution of the datasets $Ei(i = 1, ...6)$ were the same as the distribution of datasets Di, the performance of the Improved INMED methods was quite different. Only in datasets $E1$ and $E2$, the improved INMED methods gave better results along with some other initialization methods. The Improved INMEDs did not give the best results in $E3$, $E4$, $E5$, and $E6$. Although the rating distribution and total number of ratings of $Ei(i = 1, ...6)$ and $Di(i = 1, ...6)$ datasets were the same, the number of unique users and items was different. These values directly affect the rating matrix density, ratings per user and ratings per item. The rating matrix density in Di and Ei datasets was close to each other; $1.95 * 10^{-4}$ and $1.30 * 10^{-4}$, respectively. The actual difference was in the values of rating per users and rating per items that were around 1.45 and 4.55 for $Di(i = 1, ...6)$ and 2.40 and 1.85 for $Ei(i = 1, ...6)$. These results show that when these two values and the rating matrix density value are small, the constant initialization approaches perform better.

The rating matrix density of datasets $Mi(i = 1, ...6)$ wass greater than that of datasets $Di(i = 1, ...6)$ and $Ei(i = 1, ...6)$ (about 2.2×10^{-4}). The number of ratings per user and the number of per item were both greater; around 6.5 and 11.5, respectively. In dataset $M1$, the Improved INMED III and IV had better results. The Improved INMED II produced the best result in $M3$ and $M4$ datasets. In datasets $M2$, $M5$ and $M6$, Improved INMED I and II made the best predictions along with some other initialization methods.

Table 5.6: The PMF RMSE ($\mu \pm \sigma$) results for D1,...,D6 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**
C-0.0	1.4832±0.0049	1.4142±0.0055	1.0954±0.0045	1.4142±0.0054	1.4142±0.0057	1.4832±0.0045
C-0.1	1.1406±0.0061	1.1549±0.0058	1.0093±0.0051	1.2540±0.0084	1.1314±0.0070	1.1026±0.0042
C-0.2	1.0352±0.0064	1.0766±0.0056	1.0045±0.0044	1.2218±0.0071	**1.0831±0.0072**	1.0379±0.0050
C-0.3	**1.0038±0.0068**	1.0646±0.0058	1.0394±0.0057	1.2399±0.0087	1.0904±0.0080	**1.0346±0.0061**
C-0.4	1.0214±0.0077	1.0937±0.0063	1.0970±0.0075	1.2862±0.0110	1.1321±0.0088	1.0716±0.0074
C-0.5	1.0606±0.0088	1.1378±0.0071	1.1573±0.0087	1.3396±0.0122	1.1828±0.0094	1.1208±0.0083
C-0.6	1.0982±0.0095	1.1781±0.0080	1.2050±0.0094	1.3852±0.0125	1.2259±0.0099	1.1632±0.0088
C-0.7	1.1306±0.0095	1.2141±0.0083	1.2501±0.0104	1.4250±0.0128	1.2629±0.0102	1.1984±0.0090
C-0.8	1.2013±0.0112	1.2974±0.0087	1.3780±0.0129	1.5196±0.0224	1.3395±0.0095	1.2630±0.0098
INMED(5-95)	**1.0051±0.0071**	**1.0629±0.0057**	1.0654±0.0079	1.3600±0.0227	**1.0850±0.0079**	1.0431±0.0066
INMED(15-85)	1.0101±0.0073	1.0642±0.0058	1.0656±0.0079	1.3564±0.0242	1.0896±0.0081	1.0537±0.0072
INMED(25-75)	1.0130±0.0075	1.0650±0.0058	1.0622±0.0086	1.3537±0.0164	1.0913±0.0081	1.0588±0.0074
Uniform R.	1.1102±0.0092	1.1982±0.0086	1.2374±0.0096	1.4057±0.0114	1.4952±0.0101	1.5268±0.0101
SND	2.0388±0.0082	1.9807±0.0122	1.6749±0.0101	1.9806±0.0108	1.9706±0.0102	2.0205±0.0094
0.1*SND	1.4836±0.0049	1.4146±0.0054	1.0960±0.0045	1.4146±0.0053	1.4147±0.0059	1.4837±0.0046
0.5+0.1*SND	1.0630±0.0083	1.1420±0.0072	1.1598±0.0092	1.3371±0.0087	1.4141±0.0088	1.4369±0.0110
INMED(Mean)	**1.0037±0.0069**	**1.0631±0.0056**	1.0664±0.0077	1.3606±0.0199	**1.0824±0.0077**	**1.0373±0.0063**
INMED(Median)	**1.0051±0.0070**	1.0705±0.0059	1.0954±0.0045	1.4142±0.0054	1.1010±0.0083	1.0430±0.0065
Gaussian R1	**1.0052±0.0071**	**1.0629±0.0057**	1.0654±0.0081	1.3594±0.0227	**1.0850±0.0079**	1.0430±0.0066
Gaussian R2	**1.0059±0.0071**	**1.0635±0.0058**	1.0744±0.0094	1.3750±0.0258	**1.0857±0.0078**	1.0435±0.0067
IINMED_I	**1.0051±0.0071**	**1.0629±0.0057**	1.0647±0.0079	1.3573±0.0234	**1.0850±0.0079**	1.0431±0.0067
IINMED_II	**1.0051±0.0070**	**1.0614±0.0056**	**0.9751±0.0047**	**1.1824±0.0079**	1.0879±0.0079	1.0432±0.0066
IINMED_III	**1.0049±0.0071**	**1.0627±0.0057**	1.0659±0.0094	1.3561±0.0230	**1.0846±0.0079**	1.0426±0.0067
IINMED_IV	**1.0049±0.0071**	**1.0613±0.0057**	**0.9749±0.0046**	**1.1824±0.0078**	1.0875±0.0079	1.0428±0.0066

Table 5.7: The PMF RMSE ($\mu \pm \sigma$) results for E1,...,E6 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
C-0.0	1.4832±0.0037	1.4142±0.0056	1.0954±0.0070	1.4142±0.0053	1.4142±0.0050	1.4832±0.0054
C-0.1	1.2729±0.0067	1.2776±0.0058	**1.0735±0.0090**	**1.3624±0.0100**	1.2231±0.0042	1.1954±0.0054
C-0.2	1.1715±0.0062	1.2044±0.0062	1.0971±0.0225	**1.3721±0.0293**	**1.1851±0.0038**	1.1296±0.0056
C-0.3	1.1128±0.0065	**1.1759±0.0071**	1.1663±0.0380	1.4291±0.0521	1.2076±0.0047	**1.1210±0.0064**
C-0.4	1.1204±0.0084	1.2082±0.0082	1.2642±0.0490	1.5177±0.0691	1.2837±0.0064	1.1762±0.0071
C-0.5	1.1721±0.0097	1.2699±0.0092	1.3610±0.0555	1.6092±0.0791	1.3690±0.0079	1.2523±0.0081
C-0.6	1.2253±0.0101	1.3276±0.0097	1.4337±0.0581	1.6813±0.0830	1.4373±0.0091	1.3164±0.0090
C-0.7	1.2682±0.0105	1.3741±0.0096	1.5016±0.0601	1.7371±0.0846	1.4897±0.0094	1.3643±0.0096
C-0.8	1.3286±0.0111	1.4514±0.0106	1.6586±0.0684	1.8639±0.1032	1.5771±0.0082	1.4309±0.0068
INMED(5-95)	**1.1075±0.0070**	**1.1771±0.0068**	1.0870±0.0076	1.3932±0.0098	1.1965±0.0045	1.1326±0.0067
INMED(15-85)	**1.1092±0.0076**	**1.1760±0.0071**	1.0871±0.0083	1.3916±0.0094	1.2059±0.0048	1.1487±0.0069
INMED(25-75)	1.1117±0.0079	**1.1761±0.0071**	1.0860±0.0081	1.3920±0.0102	1.2095±0.0050	1.1565±0.0070
Uniform R.	1.2113±0.0086	1.3183±0.0094	1.3775±0.0094	1.5909±0.0109	1.7182±0.0118	1.7766±0.0079
SND	2.0380±0.0148	1.9699±0.0137	1.6997±0.0087	1.9785±0.0156	1.9861±0.0096	2.0355±0.0118
0.1*SND	1.4835±0.0037	1.4148±0.0055	1.0960±0.0071	1.4147±0.0053	1.4148±0.0052	1.4838±0.0054
0.5+0.1*SND	1.1752±0.0096	1.2749±0.0096	1.3120±0.0101	1.5330±0.0101	1.6434±0.0122	1.6991±0.0104
INMED(Mean)	**1.1100±0.0067**	1.1798±0.0066	1.0875±0.0074	1.3933±0.0088	1.1909±0.0043	1.1244±0.0066
INMED(Median)	**1.1075±0.0070**	1.1802±0.0075	1.0954±0.0070	1.4142±0.0053	1.2278±0.0053	1.1326±0.0067
Gaussian R1	**1.1076±0.0070**	**1.1772±0.0068**	1.0870±0.0076	1.3928±0.0099	1.1965±0.0045	1.1326±0.0068
Gaussian R2	**1.1082±0.0071**	**1.1779±0.0066**	1.0904±0.0075	1.4014±0.0113	1.1974±0.0045	1.1333±0.0068
IINMED_I	**1.1075±0.0070**	**1.1771±0.0068**	1.0874±0.0075	1.3956±0.0118	1.1965±0.0045	1.1326±0.0067
IINMED_II	**1.1076±0.0070**	1.1801±0.0070	1.0904±0.0067	1.4014±0.0051	1.2088±0.0046	1.1329±0.0067
IINMED_III	**1.1075±0.0070**	**1.1771±0.0068**	1.0887±0.0080	1.3973±0.0099	1.1965±0.0045	1.1325±0.0067
IINMED_IV	**1.1076±0.0070**	1.1801±0.0070	1.0901±0.0068	1.4012±0.0052	1.2087±0.0046	1.1329±0.0067

Table 5.8: The PMF RMSE ($\mu \pm \sigma$) results for M1,...,M6 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	M1	M2	M3	M4	M5	M6
C-0.0	1.4832±0.0047	1.4142±0.0043	1.0954±0.0049	1.4142±0.0055	1.4142±0.0060	1.4832±0.0060
C-0.1	1.0453±0.0051	1.1037±0.0056	1.0299±0.0117	1.2651±0.0218	1.1045±0.0072	1.0505±0.0067
C-0.2	1.0275±0.0052	**1.0896±0.0058**	1.0264±0.0150	1.2620±0.0322	**1.0881±0.0070**	**1.0309±0.0061**
C-0.3	1.0269±0.0057	1.0923±0.0062	1.0352±0.0190	1.2654±0.0379	**1.0883±0.0075**	**1.0280±0.0062**
C-0.4	1.0385±0.0066	1.1047±0.0067	1.0500±0.0222	1.2713±0.0384	1.0986±0.0080	1.0375±0.0065
C-0.5	1.0536±0.0074	1.1195±0.0072	1.0674±0.0243	1.2785±0.0367	1.1129±0.0081	1.0515±0.0065
C-0.6	1.0688±0.0079	1.1345±0.0075	1.0852±0.0258	1.2914±0.0355	1.1277±0.0084	1.0662±0.0065
C-0.7	1.0834±0.0085	1.1495±0.0079	1.1026±0.0270	1.3194±0.0483	1.1425±0.0086	1.0804±0.0063
C-0.8	1.1020±0.0089	1.1694±0.0086	1.1214±0.0272	1.3460±0.0565	1.1575±0.0090	1.0947±0.0067
INMED(5-95)	1.0297±0.0060	**1.0905±0.0060**	1.0495±0.0140	1.2711±0.0085	**1.0871±0.0074**	**1.0302±0.0062**
INMED(15-85)	1.0330±0.0062	1.0921±0.0062	1.0506±0.0122	1.2710±0.0089	**1.0879±0.0073**	1.0328±0.0062
INMED(25-75)	1.0346±0.0063	1.0926±0.0063	1.0498±0.0124	1.2754±0.0280	**1.0885±0.0075**	1.0341±0.0063
Uniform R.	1.1060±0.0079	1.1816±0.0087	1.1591±0.0082	1.4014±0.0104	1.4184±0.0306	1.3687±0.0234
SND	1.9119±0.0592	1.8877±0.0193	1.5514±0.0099	1.8962±0.0133	1.7529±0.0253	1.6968±0.0246
0.1*SND	1.3858±0.1159	1.3976±0.0329	1.0959±0.0050	1.4143±0.0063	1.3247±0.0871	1.3407±0.1557
0.5+0.1*SND	1.0628±0.0073	1.1320±0.0075	1.0989±0.0069	1.3375±0.0099	1.2632±0.0122	1.2143±0.0149
INMED(Mean)	1.0279±0.0058	**1.0897±0.0060**	1.0487±0.0099	1.2713±0.0085	**1.0865±0.0072**	**1.0286±0.0061**
INMED(Median)	1.0298±0.0059	1.0956±0.0064	1.0954±0.0049	1.4142±0.0055	1.0907±0.0076	**1.0301±0.0061**
Gaussian R1	1.0297±0.0060	**1.0906±0.0061**	1.0486±0.0139	1.2584±0.0121	**1.0869±0.0072**	**1.0301±0.0061**
Gaussian R2	1.0301±0.0059	**1.0908±0.0061**	1.0607±0.0151	1.2906±0.0187	**1.0872±0.0073**	**1.0302±0.0062**
IINMED_I	1.0313±0.0059	**1.0912±0.0060**	1.0751±0.0099	1.3475±0.0255	**1.0874±0.0072**	**1.0307±0.0061**
IINMED_II	1.0314±0.0060	**1.0888±0.0060**	**0.9990±0.0043**	**1.2087±0.0058**	**1.0883±0.0072**	**1.0307±0.0062**
IINMED_III	**1.0166±0.0056**	1.0955±0.0060	1.0476±0.0057	1.3129±0.0071	1.1343±0.0061	1.0384±0.0066
IINMED_IV	**1.0165±0.0057**	1.0921±0.0058	1.0297±0.0187	1.2782±0.0398	1.1311±0.0061	1.0383±0.0066

Movies dataset was denser than Epinions dataset and sparser than MovieLense dataset. Although the number of examples in Movies dataset was not as many as in Epinions and MovieLens, it was found higher than the other datasets (Home and Garden, Wellness and Beauty, and the subsampled datasets). In the sparse datasets and small dense datasets, constant initialization and random initialization methods produced similar results. The density of Movies dataset was higher than the density of Epinions dataset but lower than that of MovieLens dataset. As a result, the RMSE results of constant-based methods and random-based methods were not the same as the results obtained from Epinions dataset. The performance of both constant and random initialization techniques in Movies dataset was lower than their performance in MovieLens dataset.

Home and Garden, and Wellness and Beauty datasets had a smaller number of examples and there was no significant difference between constant-based initialization and random-based initialization. The INMED based methods made the best predictions.

In Epinions dataset, the INMED-based methods along with C-0.4 and Gaussian R1 performed better than other methods. Since Epinions dataset was sparse, the difference between the INMED based methods was not significant. Both the INMED methods that used only the constant INMED value and the Improved INMEDs that had random initialization values produced similar results. On the other hand, in MovieLens dataset, which was very dense, the difference in performance between the constant initialization methods (constant INMEDs and other constants) and variable initialization methods were quite significant (0.94 vs 0.86). The mean of LastFM dataset was close to 3.0. The RMSE results show that top two initialization methods are Improved INMED II and IV.

Table 5.9: The PMF RMSE ($\mu \pm \sigma$) results for M_ALL,...,M_5 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**M_ALL**	**M_1**	**M_3**	**M_5**
C-0.0	1.4257±0.0023	1.4352±0.0031	1.4436±0.0031	1.4489±0.0032
C-0.1	1.1129±0.0030	1.1037±0.0026	1.0999±0.0043	1.0993±0.0045
C-0.2	1.0768±0.0033	1.0697±0.0028	1.0662±0.0043	1.0658±0.0042
C-0.3	1.0641±0.0035	1.0566±0.0033	1.0515±0.0043	1.0502±0.0038
C-0.4	1.0674±0.0037	1.0583±0.0038	1.0506±0.0044	1.0477±0.0037
C-0.5	1.0779±0.0040	1.0673±0.0042	1.0562±0.0045	1.0512±0.0035
C-0.6	1.0903±0.0042	1.0782±0.0044	1.0644±0.0048	1.0572±0.0035
C-0.7	1.1028±0.0045	1.0899±0.0045	1.0738±0.0049	1.0647±0.0036
C-0.8	1.1162±0.0046	1.1033±0.0045	1.0848±0.0041	1.0730±0.0038
INMED(5-95)	1.0643±0.0035	1.0568±0.0033	1.0516±0.0043	1.0502±0.0039
INMED(15-85)	1.0638±0.0036	1.0560±0.0034	1.0504±0.0043	1.0487±0.0039
INMED(25-75)	1.0638±0.0036	1.0560±0.0034	1.0501±0.0044	1.0482±0.0038
Uniform R.	1.1348±0.0037	1.1302±0.0049	1.1210±0.0045	1.1122±0.0050
SND	1.6952±0.0136	1.6734±0.0403	1.6289±0.0212	1.6282±0.0226
0.1*SND	1.3220±0.0642	1.3190±0.0779	1.3125±0.0878	1.3017±0.0841
0.5+0.1*SND	1.0987±0.0040	1.0908±0.0047	1.0801±0.0046	1.0715±0.0045
INMED(Mean)	1.0653±0.0035	1.0578±0.0032	1.0530±0.0043	1.0518±0.0039
INMED(Median)	1.0639±0.0035	1.0560±0.0035	1.0501±0.0043	1.0484±0.0038
Gaussian R1	**1.0585±0.0034**	1.0501±0.0032	**1.0433±0.0045**	**1.0427±0.0039**
Gaussian R2	1.0650±0.0036	1.0575±0.0037	1.0504±0.0046	1.0482±0.0044
IINMED_I	**1.0584±0.0038**	**1.0490±0.0036**	1.0460±0.0049	1.0458±0.0041
IINMED_II	**1.0584±0.0035**	**1.0488±0.0037**	**1.0452±0.0044**	1.0457±0.0041
IINMED_III	**1.0589±0.0034**	**1.0483±0.0036**	1.0465±0.0047	1.0509±0.0040
IINMED_IV	**1.0591±0.0038**	**1.0481±0.0038**	**1.0450±0.0041**	1.0500±0.0041

Table 5.10: The PMF RMSE ($\mu \pm \sigma$) results for HG_ALL,...,HG_5 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**HG_ALL**	**HG_1**	**HG_3**	**HG_5**
C-0.0	1.6161±0.0048	1.6318±0.0048	1.6363±0.0059	1.6389±0.0082
C-0.1	1.2176±0.0075	1.2313±0.0088	1.2244±0.0120	1.2264±0.0143
C-0.2	1.1241±0.0086	1.1490±0.0082	1.1433±0.0136	1.1418±0.0143
C-0.3	**1.0828±0.0089**	1.1056±0.0085	1.1012±0.0146	1.1016±0.0153
C-0.4	**1.0831±0.0096**	1.1033±0.0095	**1.0926±0.0159**	**1.0916±0.0171**
C-0.5	1.1072±0.0104	1.1259±0.0103	1.1061±0.0168	1.1013±0.0177
C-0.6	1.1351±0.0111	1.1545±0.0113	1.1278±0.0177	1.1173±0.0177
C-0.7	1.1606±0.0117	1.1826±0.0115	1.1526±0.0186	1.1356±0.0180
C-0.8	1.1889±0.0119	1.2153±0.0111	1.1869±0.0236	1.1657±0.0235
INMED(5-95)	**1.0806±0.0095**	**1.1007±0.0092**	**1.0920±0.0157**	**1.0923±0.0165**
INMED(15-85)	1.0875±0.0100	1.1070±0.0098	**1.0942±0.0163**	**1.0928±0.0172**
INMED(25-75)	1.0923±0.0101	1.1125±0.0099	**1.0976±0.0168**	**1.0948±0.0175**
Uniform R.	1.1476±0.0099	1.1700±0.0124	1.1510±0.0171	1.1429±0.0203
SND	2.0420±0.0107	2.1105±0.0166	2.1265±0.0187	2.1243±0.0242
0.1*SND	1.6168±0.0048	1.6285±0.0113	1.6309±0.0119	1.6364±0.0090
0.5+0.1*SND	1.1143±0.0109	1.1339±0.0098	1.1109±0.0165	1.1046±0.0175
INMED(Mean)	**1.0787±0.0094**	**1.0998±0.0090**	**1.0931±0.0150**	**1.0941±0.0160**
INMED(Median)	-	-	-	-
Gaussian R1	**1.0805±0.0096**	**1.1007±0.0092**	**1.0920±0.0155**	**1.0920±0.0166**
Gaussian R2	**1.0812±0.0092**	**1.1012±0.0091**	**1.0926±0.0155**	**1.0928±0.0165**
IINMED_I	**1.0806±0.0095**	**1.1008±0.0095**	**1.0923±0.0158**	**1.0966±0.0172**
IINMED_II	**1.0804±0.0095**	**1.1007±0.0093**	**1.0922±0.0157**	**1.0966±0.0173**
IINMED_III	**1.0805±0.0095**	**1.0986±0.0090**	**1.0953±0.0135**	1.1000±0.0138
IINMED_IV	**1.0804±0.0095**	**1.0983±0.0091**	**1.0953±0.0138**	**1.0997±0.0146**

Table 5.11: The PMF RMSE ($\mu \pm \sigma$) results for WB_ALL,...,WB_5 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**WB_ALL**	**WB_1**	**WB_3**	**WB_5**
C-0.0	1.5376±0.0034	1.5548±0.0055	1.5702±0.0043	1.5781±0.0059
C-0.1	1.2369±0.0088	1.2398±0.0097	1.2367±0.0078	1.2400±0.0155
C-0.2	1.1640±0.0087	1.1724±0.0101	1.1738±0.0085	1.1785±0.0153
C-0.3	**1.1324±0.0093**	**1.1412±0.0105**	**1.1431±0.0103**	**1.1497±0.0157**
C-0.4	**1.1323±0.0100**	**1.1405±0.0106**	**1.1403±0.0125**	**1.1468±0.0163**
C-0.5	1.1482±0.0102	1.1567±0.0109	1.1540±0.0132	1.1585±0.0166
C-0.6	1.1667±0.0108	1.1771±0.0111	1.1730±0.0141	1.1756±0.0166
C-0.7	1.1856±0.0113	1.1976±0.0113	1.1933±0.0139	1.1945±0.0175
C-0.8	1.2066±0.0119	1.2196±0.0109	1.2143±0.0142	1.2131±0.0175
INMED(5-95)	**1.1293±0.0094**	**1.1377±0.0104**	**1.1389±0.0113**	**1.1457±0.0160**
INMED(15-85)	**1.1305±0.0098**	**1.1388±0.0105**	**1.1393±0.0121**	**1.1457±0.0162**
INMED(25-75)	**1.1321±0.0100**	**1.1409±0.0105**	**1.1412±0.0124**	**1.1474±0.0165**
Uniform R.	1.1963±0.0114	1.2095±0.0100	1.2025±0.0149	1.2041±0.0174
SND	1.9261±0.0095	1.9861±0.0114	2.0249±0.0157	2.0577±0.0159
0.1*SND	1.5380±0.0034	1.5533±0.0080	1.5653±0.0123	1.5683±0.0173
0.5+0.1*SND	1.1620±0.0101	1.1688±0.0094	1.1618±0.0145	1.1642±0.0159
INMED(Mean)	**1.1302±0.0094**	**1.1388±0.0103**	**1.1403±0.0109**	**1.1471±0.0157**
INMED(Median)	**1.1297±0.0096**	**1.1380±0.0103**	**1.1396±0.0109**	**1.1463±0.0159**
Gaussian R1	**1.1292±0.0094**	**1.1379±0.0105**	**1.1391±0.0112**	**1.1457±0.0161**
Gaussian R2	**1.1300±0.0094**	**1.1380±0.0098**	**1.1393±0.0113**	**1.1461±0.0158**
IINMED_I	**1.1292±0.0094**	**1.1375±0.0104**	**1.1397±0.0111**	**1.1483±0.0160**
IINMED_II	**1.1293±0.0096**	**1.1376±0.0104**	**1.1398±0.0113**	**1.1483±0.0161**
IINMED_III	**1.1298±0.0096**	**1.1418±0.0102**	1.1462±0.0090	1.1586±0.0147
IINMED_IV	**1.1298±0.0097**	**1.1421±0.0103**	1.1461±0.0088	1.1587±0.0146

Table 5.12: The PMF RMSE ($\mu \pm \sigma$) results for original datasets on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	Epinions	LastFM	MovieLens
C-0.0	1.5620±0.0014	0.7141±0.0036	1.2594±0.0012
C-0.1	1.1086±0.0024	0.4886±0.0030	0.9435±0.0015
C-0.2	1.0720±0.0025	0.4643±0.0031	0.9435±0.0015
C-0.3	1.0543±0.0026	0.4586±0.0031	0.9436±0.0015
C-0.4	**1.0521±0.0027**	0.4587±0.0031	0.9438±0.0015
C-0.5	1.0588±0.0028	0.4602±0.0031	0.9439±0.0015
C-0.6	1.0683±0.0029	0.4624±0.0031	0.9441±0.0015
C-0.7	1.0789±0.0030	0.4648±0.0031	0.9443±0.0015
C-0.8	1.0902±0.0030	0.4675±0.0032	0.9446±0.0016
INMED(5-95)	**1.0515±0.0026**	0.4904±0.0029	0.9436±0.0015
INMED(15-85)	**1.0518±0.0026**	0.4851±0.0029	0.9435±0.0015
INMED(25-75)	1.0527±0.0027	0.7141±0.0036	0.9435±0.0015
Uniform R.	1.1053±0.0027	0.5038±0.0045	0.8758±0.0019
SND	1.7639±0.0082	0.8338±0.0054	0.8848±0.0020
0.1*SND	1.4204±0.1099	0.6437±0.0448	0.8673±0.0017
0.5+0.1*SND	1.0749±0.0028	0.4655±0.0037	0.8699±0.0023
INMED(Mean)	1.0522±0.0026	0.4865±0.0029	0.9435±0.0015
INMED(Median)	1.0522±0.0026	0.7141±0.0036	0.9436±0.0015
Gaussian R1	**1.0508±0.0027**	0.4831±0.0031	**0.8629±0.0020**
Gaussian R2	1.0541±0.0026	0.4860±0.0031	0.8673±0.0021
IINMED_I	**1.0516±0.0026**	0.4849±0.0032	0.8691±0.0017
IINMED_II	**1.0515±0.0026**	**0.4505±0.0032**	0.8692±0.0022
IINMED_III	**1.0515±0.0026**	0.4913±0.0030	0.8688±0.0023
IINMED_IV	**1.0515±0.0026**	0.4522±0.0032	0.8690±0.0018

Table 5.13: The PMF MAE ($\mu \pm \sigma$) results for D1,...,D6 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	D1	D2	D3	D4	D5	D6
C-0.0	1.3000±0.0065	1.2000±0.0070	0.8000±0.0058	1.2000±0.0073	1.2000±0.0072	1.3000±0.0057
C-0.1	0.9595±0.0062	0.9681±0.0065	0.7860±0.0048	1.0531±0.0089	0.9355±0.0066	0.9085±0.0035
C-0.2	0.8469±0.0055	0.8853±0.0057	0.7994±0.0039	1.0088±0.0073	0.8854±0.0065	0.8396±0.0041
C-0.3	0.7968±0.0053	0.8521±0.0052	0.8247±0.0048	0.9994±0.0068	**0.8699±0.0063**	**0.8143±0.0046**
C-0.4	0.7926±0.0059	0.8586±0.0051	0.8635±0.0059	1.0183±0.0070	0.8877±0.0061	0.8272±0.0055
C-0.5	0.8043±0.0067	0.8772±0.0053	0.9030±0.0068	1.0444±0.0074	0.9128±0.0064	0.8491±0.0067
C-0.6	0.8152±0.0072	0.8933±0.0058	0.9328±0.0074	1.0666±0.0077	0.9332±0.0067	0.8663±0.0075
C-0.7	0.8257±0.0074	0.9094±0.0061	0.9624±0.0082	1.0870±0.0082	0.9516±0.0072	0.8810±0.0079
C-0.8	0.8585±0.0097	0.9569±0.0074	1.0559±0.0106	1.1480±0.0150	0.9969±0.0067	0.9139±0.0085
INMED(5-95)	**0.7895±0.0055**	0.8569±0.0053	0.7926±0.0060	1.1471±0.0215	**0.8713±0.0063**	**0.8137±0.0048**
INMED(15-85)	**0.7897±0.0057**	0.8525±0.0052	0.7927±0.0061	1.1445±0.0226	**0.8700±0.0063**	0.8186±0.0052
INMED(25-75)	**0.7903±0.0057**	0.8516±0.0053	0.7921±0.0061	1.1416±0.0147	**0.8699±0.0062**	0.8210±0.0054
Uniform R.	0.8406±0.0079	0.9258±0.0072	0.9781±0.0083	1.1058±0.0094	1.1823±0.0085	1.2006±0.0088
SND	1.6614±0.0074	1.6129±0.0125	1.3697±0.0105	1.6152±0.0113	1.6118±0.0096	1.6538±0.0098
0.1*SND	1.3038±0.0065	1.2051±0.0068	0.8103±0.0059	1.2051±0.0071	1.2052±0.0072	1.3039±0.0058
0.5+0.1*SND	0.8070±0.0065	0.8822±0.0050	0.9074±0.0080	1.0495±0.0069	1.1101±0.0076	1.1194±0.0092
INMED(Mean)	0.7939±0.0054	0.8611±0.0051	0.7928±0.0060	1.1473±0.0181	0.8730±0.0064	**0.8138±0.0044**
INMED(Median)	**0.7894±0.0056**	**0.8488±0.0053**	0.8000±0.0058	1.2000±0.0073	**0.8711±0.0061**	**0.8137±0.0047**
Gaussian R1	**0.7896±0.0055**	0.8568±0.0052	0.7930±0.0061	1.1464±0.0216	**0.8712±0.0064**	**0.8138±0.0047**
Gaussian R2	**0.7908±0.0055**	0.8574±0.0051	0.8029±0.0054	1.1669±0.0256	0.8719±0.0062	**0.8147±0.0047**
IINMED_I	**0.7895±0.0055**	0.8568±0.0053	0.7927±0.0059	1.1438±0.0223	**0.8711±0.0063**	**0.8137±0.0047**
IINMED_II	**0.7893±0.0055**	**0.8493±0.0051**	**0.7656±0.0048**	**0.9653±0.0069**	**0.8688±0.0062**	**0.8138±0.0047**
IINMED_III	**0.7895±0.0055**	0.8568±0.0052	0.7933±0.0058	1.1431±0.0227	**0.8711±0.0063**	**0.8136±0.0048**
IINMED_IV	**0.7893±0.0055**	**0.8493±0.0051**	**0.7657±0.0042**	**0.9662±0.0070**	**0.8686±0.0061**	**0.8136±0.0046**

The PMF MAE results for D1,...,D6, E1,...,E6, and M1,...,M6 are given in Tables 5.13, 5.14, and 5.15, respectively. The MAE average and standard deviation of all the runs in the experiments are also presented. Tables 5.16, 5.17, 5.18, and 5.19 show the PMF MAE results of datasets M_ALL,...,M_5, HG_ALL,...,HG_5, WB_ALL,...,WB_5 and the original datasets (Epinions, MovieLens, and LastFM) respectively.

Tables 5.20 and 5.21 show the first, second and third performance ranks of initialization methods using the two sample t-test on RMSE and MAE results of all the datasets, respectively. Each entry in the tables presents the rank number of the given initialization method in the selected dataset. Only the first, second and third ranks are presented to facilitate reading. The remaining cells have been left empty.

Table 5.14: The PMF MAE ($\mu \pm \sigma$) results for E1,...,E6 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
C-0.0	1.3000±0.0050	1.2000±0.0072	**0.8000±0.0069**	1.2000±0.0068	1.2000±0.0070	1.3000±0.0067
C-0.1	1.0873±0.0065	1.0869±0.0064	0.8274±0.0082	**1.1658±0.0087**	1.0271±0.0052	1.0004±0.0062
C-0.2	0.9857±0.0058	1.0197±0.0059	0.8760±0.0181	**1.1691±0.0253**	0.9932±0.0046	0.9349±0.0059
C-0.3	0.9001±0.0056	0.9601±0.0069	0.9259±0.0308	1.1850±0.0476	**0.9756±0.0048**	0.8899±0.0054
C-0.4	**0.8774±0.0065**	0.9589±0.0073	0.9944±0.0385	1.2326±0.0634	1.0098±0.0054	0.9039±0.0051
C-0.5	0.8884±0.0075	0.9830±0.0080	1.0626±0.0425	1.2859±0.0721	1.0571±0.0064	0.9398±0.0055
C-0.6	0.9011±0.0079	1.0050±0.0088	1.1118±0.0446	1.3265±0.0766	1.0934±0.0075	0.9686±0.0060
C-0.7	0.9132±0.0082	1.0242±0.0088	1.1621±0.0464	1.3604±0.0794	1.1218±0.0079	0.9903±0.0065
C-0.8	0.9308±0.0092	1.0592±0.0090	1.2946±0.0570	1.4518±0.0994	1.1710±0.0081	1.0164±0.0049
INMED(5-95)	0.8801±0.0056	0.9730±0.0063	**0.8038±0.0076**	1.1835±0.0092	**0.9779±0.0049**	**0.8836±0.0052**
INMED(15-85)	**0.8765±0.0059**	0.9617±0.0066	**0.8036±0.0073**	1.1824±0.0085	**0.9758±0.0047**	0.8902±0.0052
INMED(25-75)	**0.8762±0.0061**	0.9586±0.0069	0.8041±0.0073	1.1829±0.0097	**0.9755±0.0047**	0.8940±0.0052
Uniform R.	0.9144±0.0069	1.0187±0.0079	1.0957±0.0084	1.2653±0.0104	1.3893±0.0113	1.4379±0.0085
SND	1.6517±0.0145	1.5971±0.0124	1.3903±0.0090	1.6073±0.0148	1.6166±0.0106	1.6568±0.0120
0.1*SND	1.3037±0.0049	1.2055±0.0070	0.8101±0.0069	1.2053±0.0068	1.2054±0.0070	1.3040±0.0066
0.5+0.1*SND	0.8915±0.0076	0.9881±0.0083	1.0323±0.0085	1.2186±0.0093	1.3212±0.0106	1.3654±0.0103
INMED(Mean)	0.8923±0.0055	0.9820±0.0063	**0.8035±0.0077**	1.1835±0.0084	0.9806±0.0047	0.8865±0.0053
INMED(Median)	0.8800±0.0056	**0.9485±0.0072**	**0.8000±0.0069**	1.2000±0.0068	**0.9779±0.0050**	**0.8835±0.0052**
Gaussian R1	0.8804±0.0056	0.9727±0.0064	0.8040±0.0076	1.1833±0.0094	**0.9778±0.0047**	**0.8837±0.0052**
Gaussian R2	0.8825±0.0056	0.9734±0.0062	0.8077±0.0075	1.1941±0.0109	0.9785±0.0047	**0.8854±0.0053**
IINMED_I	0.8801±0.0056	0.9727±0.0062	**0.8036±0.0078**	1.1858±0.0107	**0.9779±0.0046**	**0.8835±0.0052**
IINMED_II	0.8800±0.0056	0.9656±0.0066	0.8308±0.0075	1.2043±0.0058	0.9793±0.0047	**0.8836±0.0052**
IINMED_III	0.8801±0.0056	0.9729±0.0063	**0.8036±0.0077**	1.1874±0.0096	0.9780±0.0048	**0.8836±0.0052**
IINMED_IV	0.8800±0.0056	0.9657±0.0067	0.8293±0.0074	1.2038±0.0057	0.9793±0.0047	**0.8837±0.0052**

Table 5.15: The PMF MAE ($\mu \pm \sigma$) results for M1,...,M6 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**
C-0.0	1.3000±0.0061	1.2000±0.0058	0.8000±0.0051	1.2000±0.0071	1.2000±0.0076	1.3000±0.0077
C-0.1	0.8348±0.0051	0.8954±0.0054	0.8021±0.0075	1.0364±0.0187	0.8941±0.0063	0.8353±0.0058
C-0.2	0.8208±0.0049	0.8842±0.0051	0.8086±0.0107	1.0467±0.0370	0.8823±0.0062	0.8212±0.0056
C-0.3	**0.8123±0.0049**	**0.8785±0.0049**	0.8151±0.0130	1.0401±0.0400	**0.8748±0.0064**	**0.8112±0.0054**
C-0.4	**0.8129±0.0056**	0.8804±0.0047	0.8233±0.0145	1.0315±0.0364	**0.8748±0.0068**	**0.8092±0.0053**
C-0.5	0.8164±0.0060	0.8849±0.0051	0.8328±0.0158	1.0284±0.0323	0.8790±0.0069	0.8120±0.0054
C-0.6	0.8206±0.0064	0.8905±0.0053	0.8421±0.0168	1.0339±0.0299	0.8837±0.0069	0.8160±0.0052
C-0.7	0.8254±0.0068	0.8964±0.0056	0.8512±0.0176	1.0560±0.0447	0.8890±0.0070	0.8205±0.0049
C-0.8	0.8341±0.0069	0.9073±0.0061	0.8615±0.0175	1.0757±0.0544	0.8959±0.0075	0.8261±0.0051
INMED(5-95)	**0.8116±0.0052**	**0.8790±0.0050**	0.8058±0.0075	1.0364±0.0071	**0.8761±0.0062**	**0.8097±0.0053**
INMED(15-85)	**0.8117±0.0053**	**0.8783±0.0049**	0.8064±0.0085	1.0362±0.0070	**0.8751±0.0063**	**0.8092±0.0052**
INMED(25-75)	**0.8120±0.0054**	**0.8783±0.0048**	0.8058±0.0082	1.0415±0.0310	**0.8746±0.0061**	**0.8091±0.0054**
Uniform R.	0.8558±0.0064	0.9357±0.0063	0.9169±0.0066	1.1463±0.0088	1.1448±0.0289	1.0859±0.0185
SND	1.5476±0.0679	1.5350±0.0222	1.2453±0.0090	1.5427±0.0146	1.3930±0.0231	1.3245±0.0238
0.1*SND	1.1904±0.1335	1.1851±0.0371	0.8102±0.0052	1.2065±0.0078	1.1123±0.0895	1.1459±0.1719
0.5+0.1*SND	0.8275±0.0058	0.9005±0.0051	0.8622±0.0052	1.0957±0.0099	1.0200±0.0111	0.9665±0.0136
INMED(Mean)	**0.8120±0.0050**	**0.8796±0.0050**	0.8065±0.0085	1.0364±0.0069	**0.8769±0.0063**	**0.8106±0.0053**
INMED(Median)	**0.8116±0.0051**	**0.8784±0.0048**	0.8000±0.0051	1.2000±0.0071	**0.8741±0.0063**	**0.8098±0.0053**
Gaussian R1	**0.8116±0.0051**	**0.8788±0.0048**	0.8068±0.0072	1.0309±0.0127	**0.8759±0.0064**	**0.8098±0.0051**
Gaussian R2	**0.8122±0.0049**	**0.8794±0.0048**	0.8127±0.0085	1.0772±0.0198	**0.8762±0.0062**	**0.8102±0.0053**
IINMED_I	**0.8132±0.0049**	0.8814±0.0048	0.8187±0.0054	1.1385±0.0268	0.8780±0.0060	**0.8101±0.0053**
IINMED_II	0.8133±0.0050	**0.8774±0.0046**	**0.7950±0.0032**	**0.9974±0.0040**	**0.8771±0.0060**	**0.8102±0.0053**
IINMED_III	**0.8111±0.0049**	0.8885±0.0052	0.8197±0.0063	1.1078±0.0060	0.9171±0.0048	0.8250±0.0057
IINMED_IV	**0.8107±0.0048**	0.8844±0.0052	0.8124±0.0119	1.0561±0.0387	0.9130±0.0051	0.8248±0.0057

Table 5.16: The PMF MAE ($\mu \pm \sigma$) results for M_ALL,...,M_5 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**M_ALL**	**M_1**	**M_3**	**M_5**
C-0.0	1.2261±0.0030	1.2382±0.0038	1.2484±0.0038	1.2546±0.0044
C-0.1	0.8919±0.0031	0.8804±0.0025	0.8777±0.0038	0.8782±0.0047
C-0.2	0.8564±0.0028	0.8495±0.0026	0.8486±0.0036	0.8492±0.0042
C-0.3	0.8358±0.0028	0.8295±0.0028	0.8277±0.0036	0.8285±0.0038
C-0.4	0.8300±0.0029	**0.8219±0.0030**	0.8184±0.0038	0.8183±0.0035
C-0.5	0.8301±0.0029	**0.8207±0.0034**	**0.8153±0.0037**	0.8138±0.0035
C-0.6	0.8322±0.0031	**0.8220±0.0033**	**0.8149±0.0040**	**0.8120±0.0033**
C-0.7	0.8350±0.0032	0.8244±0.0034	**0.8158±0.0038**	**0.8119±0.0034**
C-0.8	0.8393±0.0032	0.8291±0.0035	0.8189±0.0033	**0.8129±0.0034**
INMED(5-95)	0.8366±0.0029	0.8299±0.0025	0.8280±0.0037	0.8287±0.0036
INMED(15-85)	0.8338±0.0029	0.8268±0.0027	0.8245±0.0038	0.8251±0.0038
INMED(25-75)	0.8335±0.0028	0.8261±0.0028	0.8238±0.0038	0.8241±0.0036
Uniform R.	0.8775±0.0027	0.8723±0.0043	0.8654±0.0040	0.8591±0.0039
SND	1.3605±0.0335	1.3261±0.0641	1.2616±0.0297	1.2602±0.0312
0.1*SND	1.1074±0.0755	1.1048±0.0920	1.0985±0.1029	1.0849±0.0991
0.5+0.1*SND	0.8515±0.0029	0.8441±0.0041	0.8362±0.0038	0.8308±0.0043
INMED(Mean)	0.8392±0.0027	0.8325±0.0027	0.8305±0.0037	0.8313±0.0038
INMED(Median)	0.8326±0.0030	0.8258±0.0029	0.8237±0.0036	0.8246±0.0039
Gaussian R1	0.8309±0.0025	0.8237±0.0029	0.8189±0.0038	0.8189±0.0035
Gaussian R2	0.8415±0.0031	0.8343±0.0033	0.8288±0.0038	0.8280±0.0044
IINMED_I	**0.8267±0.0028**	**0.8222±0.0030**	0.8229±0.0043	0.8247±0.0039
IINMED_II	**0.8264±0.0024**	**0.8217±0.0029**	0.8220±0.0038	0.8242±0.0039
IINMED_III	**0.8273±0.0026**	0.8224±0.0030	0.8275±0.0040	0.8363±0.0046
IINMED_IV	**0.8271±0.0028**	**0.8222±0.0031**	0.8257±0.0034	0.8350±0.0044

Table 5.17: The PMF MAE ($\mu \pm \sigma$) results for HG_ALL,...,HG_5 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	**HG_ALL**	**HG_1**	**HG_3**	**HG_5**
C-0.0	1.4809±0.0069	1.4995±0.0061	1.5042±0.0082	1.5077±0.0110
C-0.1	0.9963±0.0070	0.9968±0.0065	0.9921±0.0107	0.9998±0.0129
C-0.2	0.8884±0.0057	0.9187±0.0063	0.9159±0.0097	0.9141±0.0120
C-0.3	0.8309±0.0055	0.8592±0.0062	0.8619±0.0086	0.8635±0.0107
C-0.4	0.8072±0.0054	0.8303±0.0071	0.8318±0.0090	0.8355±0.0114
C-0.5	0.7994±0.0062	**0.8194±0.0079**	0.8166±0.0098	0.8192±0.0120
C-0.6	**0.7963±0.0066**	**0.8159±0.0088**	**0.8101±0.0103**	**0.8111±0.0129**
C-0.7	**0.7946±0.0071**	**0.8155±0.0089**	**0.8077±0.0110**	**0.8069±0.0139**
C-0.8	**0.7947±0.0068**	**0.8148±0.0085**	**0.8070±0.0125**	**0.8053±0.0151**
INMED(5-95)	0.8099±0.0054	0.8349±0.0067	0.8379±0.0085	0.8420±0.0112
INMED(15-85)	0.8046±0.0056	0.8268±0.0072	0.8275±0.0091	0.8311±0.0115
INMED(25-75)	0.8028±0.0058	0.8240±0.0074	0.8229±0.0094	0.8260±0.0115
Uniform R.	0.8283±0.0065	0.8489±0.0091	0.8437±0.0107	0.8439±0.0144
SND	1.6729±0.0107	1.7174±0.0163	1.7271±0.0199	1.7228±0.0242
0.1*SND	1.4828±0.0072	1.4915±0.0205	1.4941±0.0216	1.5031±0.0132
0.5+0.1*SND	0.8068±0.0062	0.8273±0.0072	0.8216±0.0102	0.8236±0.0121
INMED(Mean)	0.8142±0.0056	0.8407±0.0065	0.8447±0.0084	0.8486±0.0111
INMED(Median)	-	-	-	-
Gaussian R1	0.8099±0.0054	0.8351±0.0068	0.8379±0.0085	0.8419±0.0113
Gaussian R2	0.8113±0.0053	0.8361±0.0069	0.8388±0.0083	0.8430±0.0112
IINMED_I	0.8098±0.0054	0.8345±0.0068	0.8364±0.0086	0.8371±0.0115
IINMED_II	0.8098±0.0054	0.8344±0.0068	0.8364±0.0085	0.8373±0.0116
IINMED_III	0.8101±0.0055	0.8377±0.0067	0.8544±0.0079	0.8606±0.0110
IINMED_IV	0.8101±0.0055	0.8376±0.0067	0.8541±0.0071	0.8602±0.0112

Table 5.18: The PMF MAE ($\mu \pm \sigma$) results for WB_ALL,...,WB_5 on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	WB_ALL	WB_1	WB_3	WB_5
C-0.0	1.3750±0.0045	1.3960±0.0067	1.4154±0.0057	1.4259±0.0070
C-0.1	1.0107±0.0081	1.0104±0.0098	1.0079±0.0088	1.0134±0.0137
C-0.2	0.9350±0.0075	0.9475±0.0090	0.9513±0.0077	0.9562±0.0137
C-0.3	0.8900±0.0070	0.9026±0.0083	0.9090±0.0079	0.9156±0.0127
C-0.4	0.8739±0.0066	0.8843±0.0080	0.8884±0.0087	0.8947±0.0124
C-0.5	**0.8690±0.0070**	**0.8784±0.0081**	**0.8812±0.0092**	**0.8872±0.0127**
C-0.6	**0.8671±0.0072**	**0.8769±0.0083**	**0.8797±0.0097**	**0.8862±0.0127**
C-0.7	**0.8664±0.0076**	**0.8768±0.0086**	**0.8804±0.0096**	**0.8872±0.0132**
C-0.8	**0.8682±0.0074**	**0.8781±0.0082**	**0.8809±0.0109**	**0.8876±0.0130**
INMED(5-95)	0.8800±0.0070	0.8919±0.0081	0.8975±0.0081	0.9042±0.0123
INMED(15-85)	0.8759±0.0067	0.8864±0.0078	0.8906±0.0085	0.8970±0.0124
INMED(25-75)	0.8740±0.0069	0.8838±0.0078	0.8873±0.0087	0.8933±0.0125
Uniform R.	0.9046±0.0082	0.9147±0.0069	0.9118±0.0106	0.9138±0.0135
SND	1.5705±0.0097	1.6163±0.0107	1.6465±0.0143	1.6739±0.0182
0.1*SND	1.3782±0.0045	1.3908±0.0138	1.4062±0.0236	1.4075±0.0316
0.5+0.1*SND	0.8838±0.0068	0.8904±0.0071	0.8885±0.0102	**0.8924±0.0124**
INMED(Mean)	0.8847±0.0069	0.8964±0.0079	0.9025±0.0079	0.9094±0.0124
INMED(Median)	0.8822±0.0069	0.8940±0.0080	0.9001±0.0081	0.9070±0.0125
Gaussian R1	0.8802±0.0066	0.8920±0.0080	0.8976±0.0081	0.9042±0.0123
Gaussian R2	0.8839±0.0066	0.8941±0.0075	0.8987±0.0082	0.9051±0.0122
IINMED_I	0.8804±0.0067	0.8914±0.0080	0.8970±0.0079	0.9032±0.0119
IINMED_II	0.8802±0.0067	0.8917±0.0078	0.8971±0.0081	0.9032±0.0119
IINMED_III	0.8815±0.0068	0.8999±0.0078	0.9193±0.0074	0.9295±0.0119
IINMED_IV	0.8814±0.0065	0.8998±0.0081	0.9197±0.0070	0.9300±0.0119

Table 5.19: The PMF MAE ($\mu \pm \sigma$) results for original datasets on different initialization techniques. The superior results on two sample t-test (at %5 significance level) are shown bold.

Initialization Method	Epinions	LastFM	MovieLens
C-0.0	1.4043±0.0017	0.4273±0.0034	1.0213±0.0015
C-0.1	0.8740±0.0023	0.3521±0.0021	0.7490±0.0013
C-0.2	0.8390±0.0022	0.3362±0.0020	0.7489±0.0013
C-0.3	0.8184±0.0018	0.3304±0.0019	0.7488±0.0013
C-0.4	0.8088±0.0019	0.3298±0.0020	0.7487±0.0013
C-0.5	0.8042±0.0020	0.3300±0.0020	0.7487±0.0013
C-0.6	**0.8025±0.0021**	0.3305±0.0021	0.7488±0.0013
C-0.7	**0.8024±0.0021**	0.3312±0.0021	0.7488±0.0013
C-0.8	0.8037±0.0022	0.3321±0.0023	0.7490±0.0013
INMED(5-95)	0.8124±0.0018	0.3530±0.0021	0.7488±0.0013
INMED(15-85)	0.8093±0.0020	0.3502±0.0021	0.7488±0.0013
INMED(25-75)	0.8078±0.0021	0.4273±0.0034	0.7488±0.0013
Uniform R.	0.8362±0.0019	0.3729±0.0026	0.6867±0.0016
SND	1.3647±0.0080	0.5583±0.0038	0.6909±0.0017
0.1*SND	1.2168±0.1472	0.4235±0.0128	0.6827±0.0013
0.5+0.1*SND	0.8174±0.0020	0.3363±0.0022	0.6837±0.0016
INMED(Mean)	0.8148±0.0020	0.3510±0.0021	0.7488±0.0013
INMED(Median)	0.8143±0.0020	0.4273±0.0034	0.7487±0.0013
Gaussian R1	0.8125±0.0020	0.3466±0.0022	**0.6802±0.0017**
Gaussian R2	0.8183±0.0019	0.3473±0.0022	0.6827±0.0015
IINMED_I	0.8123±0.0020	0.3479±0.0022	0.6836±0.0012
IINMED_II	0.8123±0.0018	**0.3258±0.0020**	0.6837±0.0017
IINMED_III	0.8124±0.0019	0.3550±0.0020	0.6835±0.0018
IINMED_IV	0.8123±0.0018	0.3269±0.0019	0.6837±0.0014

Table 5.20: The first, second, and third ranks of initialization methods on the two sample t-test results on RMSE on all datasets

	C-0.0	C-0.1	C-0.2	C-0.3	C-0.4	C-0.5	C-0.6	C-0.7	C-0.8	INMED(5-95)	INMED(15-85)	INMED(25-75)	Uniform R.	SND	0.1*SND	0.5+0.1*SND	INMED(Mean)	INMED(Median)	Gaussian R1	Gaussian R2	IINMED_I	IINMED_II	IINMED_III	IINMED_IV
D1				1	3					1	2	2					1	1	1	1	1	1	1	1
D2				2						1	2	2					1	3	1	1	1	1	1	1
D3		3	2																			1		1
D4			2	3																		1		1
D5			1	2						1	2	2					1	3	1	1	1	2	1	2
D6			2	1						3							1	3	3	3	3	3	3	3
E1				2	3					1	1	2					1	1	1	1	1	1	1	1
E2			3	1						1	1	1					2	2	1	1	1	2	1	2
E3	3	1	2							2	2	2			3		2	3	2	2	2	2	2	2
E4		1	1							2	2	2					2		2	3	2		3	3
E5			1							3							2		3	3	3		3	
E6			3	1						3							2	3	3		3		3	
M1			2	2						2	3						2	2	2	3	3	3	1	1
M2			1	2						1	2	2					1	3	1	1	1	1	3	2
M3		2	2	2	3					3	3	3					3		3			1	3	2
M4		2	2	2	2	3				3	3	3					3		2			1		3
M5			1	1	3					1	1	1					1	2	1	1	1	1		
M6			1	1	3					1	2	2					1	1	1	1	1	1	3	3
M_ALL				2	3					2	2	2					2	2	1	2	1	1	1	1
M_1				3							3	3						3	2	3	1	1	1	1
M_3				3	3					3	3	3						3	1	3	2	1	2	1
M_5					2						3	3						3	1	3	2	2		3
HG_ALL				1	1	3				1	2	2					1		1	1	1	1	1	1
HG_1				2	2					1	2	3					1		1	1	1	1	1	1
HG_3				2	1	2				1	1	1				3	1		1	1	1	1	1	1
HG_5				2	1	2	3			1	1	1				2	1	1	1	1	1	1	2	1
WB_ALL			3	1	1	2	3			1	1	1				3	1	1	1	1	1	1	1	1
WB_1			3	1	1	2	3			1	1	1				3	1	1	1	1	1	1	1	1
WB_3				1	1	3				1	1	1					1	1	1	1	1	1	2	2
WB_5			3	1	1	2	3			1	1	1				2	1	1	1	1	1	1	2	2
Epinions				3	1					1	1	2					2	2	1	3	1	1	1	1
MovieLens															2	3			1	2	3	3	3	3
LastFM				3	3																	1		2

Table 5.21: The first, second, and third ranks of initialization methods on the two sample t-test results on MAE on all datasets

	C-0.0	C-0.1	C-0.2	C-0.3	C-0.4	C-0.5	C-0.6	C-0.7	C-0.8	INMED(5-95)	INMED(15-85)	INMED(25-75)	Uniform R.	SND	0.1*SND	0.5+0.1*SND	INMED(Mean)	INMED(Median)	Gaussian R1	Gaussian R2	IINMED_I	IINMED_II	IINMED_III	IINMED_IV
D1				3	2					1	1	1					2	1	1	1	1	1	1	1
D2				2	3					3	2	2						1	3	3	3	1	3	1
D3		2								3	3	3					3		3		3	1	3	1
D4			3	2																		1		1
D5			3	1	3					1	1	1					2	1	1	2	1	1	1	1
D6				1	3					1	2	2					1	1	1	1	1	1	1	1
E1					1	3				2	1	1				3		2	2	2	2	2	2	2
E2				2	2						2	2						1				3		3
E3	1									1	1	2					1	1	3	3	1		1	
E4	3	1	1	3						2	2	2					2	3	2	2	2		2	
E5				1						1	1	1					3	1	1	2	1	2	2	2
E6				3						1	3						2	1	1	1	1	1	1	1
M1			3	1	1	2	3			1	1	1					1	1	1	1	1	1	1	1
M2			3	1	2	3				1	1	1					1	1	1	1	2	1		3
M3	2	2	3							3	3	3					3	2	3			1		
M4		2	3	2	2	2	2			3	3	2					3		2			1		
M5			3	1	1	2	3			1	1	1					1	1	1	1	2	1		
M6			3	1	1	1	2	3		1	1	1					1	1	1	1	1	1		
M_ALL					2	2	3					3						3	2		1	1	1	1
M_1					1	1	1												3		1	1	2	1
M_3					2	1	1	1	2			3						3	2		3	3		
M_5					3	2	1	1	1										3					
HG_ALL						2	1	1	1		3	3												
HG_1						2	1	1	1		3	3				3								
HG_3						2	1	1	1		3	3				2								
HG_5						2	1	1	1		3	3				2								
WB_ALL					2	1	1	1	1	3	2	2						3	3		3	3	3	3
WB_1					2	1	1	1	1	3	2	2				3		3	3	3	3	3		
WB_3					2	1	1	1	1	3	2	2				2		3	3	3	3	3		
WB_5					2	1	1	1	1	3	2	2				1	3	3	3	3	3	3		
Epinions					3	2	1	1	2			3												
MovieLens															2	3			1	2	3	3	2	3
LastFM				3	3	3	3															1		2

Table 5.22: Comparisons of INMED with the other studies in the literature depicted in [30]

Dimension	Training	Metrics	Hao Ma et al.'s method[34]	BPMF[49]	INMED	BPMFSR[30]
10	40%	MAE	0.9261±0.0034	0.8535±0.0018	**0.8346±0.0006**	0.8411±0.0006
		RMSE	1.2046±0.0050	1.0858±0.0026	1.0766±0.0008	**1.0695±0.0008**
	60%	MAE	0.9313±0.0023	0.8383±0.0018	**0.8190±0.0015**	0.8359±0.0006
		RMSE	1.2003±0.0047	1.0704±0.0027	**1.0613±0.0018**	1.0655±0.0009
	80%	MAE	0.9018±0.0026	0.8144±0.0020	**0.8093±0.0023**	0.8114±0.0022
		RMSE	1.1630±0.0027	1.0498±0.0023	1.0516±0.0028	**1.0435±0.0025**
	90%	MAE	0.8915±0.0036	0.8081±0.0020	0.8059±0.0028	0.8056±0.0020
		RMSE	1.1510±0.0055	1.0435±0.0035	1.0480±0.0036	**1.0334±0.0040**
30	40%	MAE	0.9341±0.0040	0.8446±0.0073	**0.8346±0.0006**	0.8381±0.0010
		RMSE	1.2043±0.0021	1.0785±0.0069	1.0766±0.0008	**1.0666±0.0012**
	60%	MAE	0.9332±0.0031	0.8423±0.0057	**0.8190±0.0015**	0.8348±0.0017
		RMSE	1.2015±0.0039	1.0741±0.0056	**1.0612±0.0018**	1.0640±0.0025
	80%	MAE	0.9135±0.0019	0.8156±0.0018	**0.8093±0.0023**	0.8124±0.0016
		RMSE	1.1736±0.0025	1.0496±0.0026	1.0515±0.0028	**1.0403±0.0024**
	90%	MAE	0.9078±0.0039	0.8087±0.0025	**0.8060±0.0028**	0.8078±0.0024
		RMSE	1.1661±0.0051	1.0434±0.0034	1.0480±0.0036	**1.0352±0.0030**

5.4 A Comparison of the Current Study with Other Studies

5.4.1 A Comparison of the Proposed Method with Other PMF-Based Methods

The primary datasets used in this study were crawled by the author. As a result, it is not possible to make a direct comparison with other methods in the literature. Therefore, the same Epinions.com dataset was used in the majority of other studies in literature [30, 37]. Table 5.22 shows the results of INMED and other PMF approaches in the literature. For a fair comparison, the same main parameter settings were used; the latent vector dimension was set to 10 and 30, and the training dataset ratio was selected as 40%, 60%, 80%, and 90%. The learning rate, momentum and regularization parameters were selected as 0.005, 0.9, and 0.001, respectively. These values were chosen in line with the literature [50, 49]. A 10-fold cross-validation was used and 5 runs were carried out for each fold, giving a total of 50 runs.

In Table 5.22, the results reported in [30] that combined the social information and the item contents with user ratings by modifying the model in BPMF[49] were used. Hao Ma et al. proposed a regularization based method in which a social regularization term was added to the loss function[34].

According to the results, the proposed method produced the best result in 9 out of 16 cases, was second best in 3 out of 16 cases and was third best in 4 out of 16 cases with respect to MAE and RMSE.

Z-Test was used to compare the performance of the proposed method with the performance of other methods at the significance level of 5%. The results showed that in 9 out of 16 cases the method proposed in this thesis produced the best results. In 6 out of 16 cases, BPMFSR was superior. In one case, both BPMFSR and INMED produced similar results. Compared to BPMF, the results showed that in 12 out of 16 cases, the proposed method produced significantly better results than BPMF. In 4 out of 16 cases, BPMF outperformed INMED.

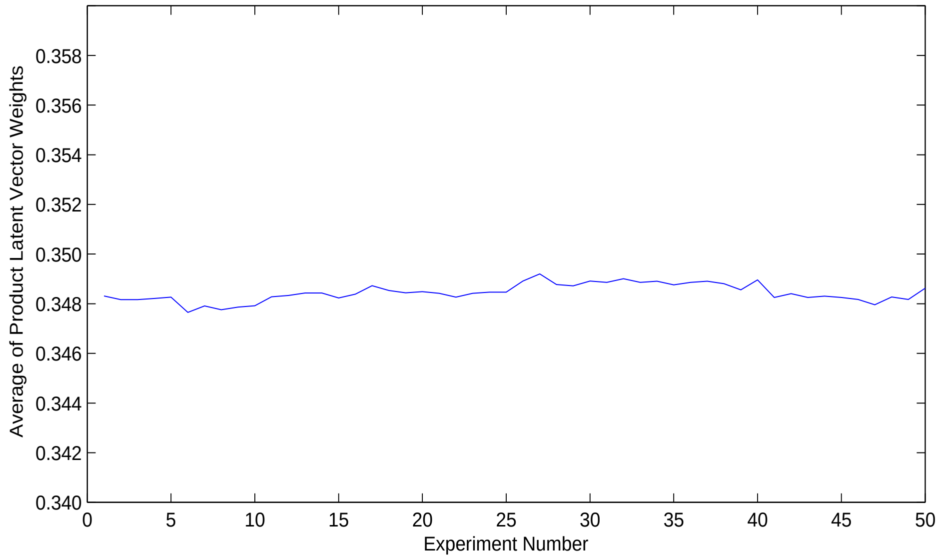

Figure 5.5: The mean of the product latent vector weights with INMED method for 50 runs (Dimension:10) after the training phase has been completed. The average of the initial weights was 0.3519. The experiment number shows the fold and the run number i.e. 1 implies the first fold in the first run, 2 is the first fold in the second run and etc.

The experiments showed that INMED produced similar results for 10 and 30 dimensions where the training dataset percentage was the same. This can be attributed to all the matrices in the latent vectors having been initialized with the same constant value. Therefore, particularly for sparse datasets, the weights cannot deviate significantly from their initial values (see Section 5.5 for details). In addition, the number of batches and the batch size defined in the experiments resulted in using all the training dataset. So, the error propagation at the end of each epoch ends up similarly. The random selection of batches (initializing random seeds based on the system time to ensure disparity) did not affect the results. To clarify this issue, the average of the product latent vector weights is given in Figures 5.5 and 5.6, respectively. These results show that the weights did not deviate much from the initial weights over 50 runs.

In terms of the complexity of the methods, INMED is less complex compared to the other approaches. Since INMED only uses the user-product rating information to compute the trimmed mean of the training data for the initialization of the matrices, the computational complexity is $O(1)$. It should be noted that here the complexity of the training approach (PMF) is not taken into consideration.

BPMF also only uses the user-product rating information and assumes that the prior distributions over the user and movie feature vectors are Gaussian [49]. Gaussian-Wishart priors are also placed on the user and product hyperparameters. To evaluate the predictive distribution of the rating value, Markov Chain Monte Carlo (MCMC) methods are used. Gibbs sampling algorithm is an MCMC algorithm that cycles through all the latent variables. The most expensive part of training BPMF models is the inversion of $D \times D$ matrix per feature vector performed in Gibbs sampling. The computational complexity is $O(D^3)$.

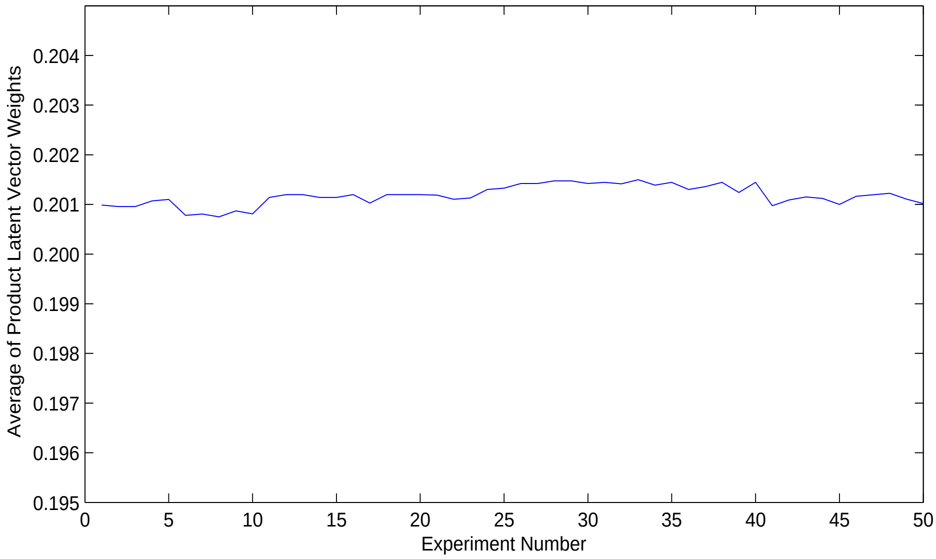

Figure 5.6: The mean of the product latent vector weights with INMED method for 50 runs (Dimension:30) after the training phase has been completed. The average of the initial weights was 0.2032.

BPMFSR uses both user-product rating and trust network information. Gibbs sampling algorithm is used to sample both user and product latent vectors [30]. Initially, BPMFSR generates hyperparameters for each user and, using these hyperparameters, produces latent vectors for each user. If a user has very few trusted friends (less than a specified threshold), the hyperparameters are sampled using the latent vectors of all users. If a user has sufficient number of friends, the hyperparameters are sampled using the latent vectors of the same user's friends (The product hyperparameters and product latent vectors are generated in the same way as in BPMF). BPMFSR uses different hyperparameters for different users instead of uniform user hyperparameters as in BPMF. It is suggested that using social relations (trust network) can improve the accuracy of the prediction [30]. The computational complexity of sampling all user feature vectors and sampling all product latent vectors is $O(K)$ where K is the number of non-zero entries in the rating matrix. The computational complexity is slightly higher in one iteration compared to BPMF due to the additional operations. However, it has also been reported that the computational complexity of sampling all user and product hyperparameters is much less than that of sampling user and product feature vectors [30].

Hao Ma et al. proposed a method that added social regularization term in the sum-of-squared-errors objective function to minimize the tastes between a user and his/her friends [34]. A user may have different similarities with their friends. Therefore, the regularization term is changed to reflect the different similarities between the user and user's friends. Then, a similarity function is added to the social regularization. The similarity value is computed for only between the user and user's friends based on the trust network. Therefore, the similarity computation is not as costly as expected. This method uses the trust network in addition to the user-product rating matrix. However, the prediction accuracy of this method is not as good as other approaches.

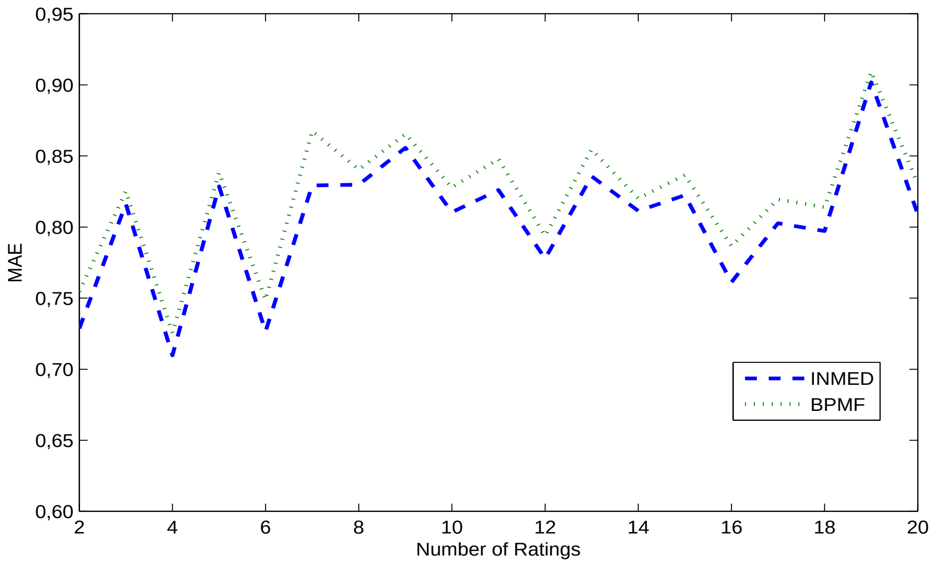

Figure 5.7: MAE results of predictions using INMED and BPMF on Movielens cold start users.

INMED was also compared with BPMF discussed in Section 5.4.1 using the cold start users created in MovieLens dataset. For both methods, the same parameters were used; learning rate:0.005, momentum:0.9, number of batches:880, batch size: 1000, max epoch:50, dimension:10 and lambda parameter:0.001. A total of 8449 ratings that belonged to cold start users were used as the testing dataset and the remaining ratings as the training data. A total of 50 runs were undertaken. Figures 5.7 and 5.8 show the comparative results of INMED and BPMF for cold start users of MovieLens dataset with respect to MAE and RMSE.

The results show that INMED performed better than BPMF in terms of cold start users in MovieLens dataset in all cases.

The RMSE and MAE results from Improved INMED methods were compared with those obtained from the BPMF algorithm. All the parameters of the PMF algorithm were the same both in the Improved INMEDs and BPMF except the termination of the training phase of the methods. The number of epochs was determined as 50 in BPMF. Therefore, BPMF did not use any validation dataset; instead all the training data was used for training (80% in the experiments described in this section). However, the validation data was used for the proposed INMED and Improved INMED methods to terminate the training phase (70% for training and 10% for validation). The termination of the training procedure is given in Section 5.2. A comparison of INMEDs and BPMF in terms of RMSE and MAE is given in Tables 5.23 and 5.24, respectively. The details of the BPMF method are given in Section 5.4.1.

The results show that the performance of the Improved INMEDs depends on the rating matrix density. On datasets that were sparser, the Improved INMEDs performed better than BPMF. On the other hand, BPMF made better predictions on denser datasets.

Table 5.23: The RMSE Comparison of Improved INMED results with BPMF on 32 Datasets. The superior results on two sample t-test (at %5 significance level) are shown bold.

Datasets	BPMF	IINMED_I	IINMED_II	IINMED_III	IINMED_IV
D1	1.0203±0.0064	**1.0051±0.0071**	**1.0051±0.0070**	**1.0049±0.0071**	**1.0049±0.0071**
D2	1.0718±0.0059	**1.0629±0.0057**	**1.0614±0.0056**	**1.0627±0.0057**	**1.0613±0.0057**
D3	0.9788±0.0052	1.0647±0.0079	**0.9751±0.0047**	1.0659±0.0094	**0.9749±0.0046**
D4	**1.1628±0.0074**	1.3573±0.0234	1.1824±0.0079	1.3561±0.0230	1.1824±0.0078
D5	**1.0610±0.0064**	1.0850±0.0079	1.0879±0.0079	1.0846±0.0079	1.0875±0.0079
D6	**1.0208±0.0049**	1.0431±0.0067	1.0432±0.0066	1.0426±0.0067	1.0428±0.0066
M1	**0.9889±0.0070**	1.0313±0.0059	1.0314±0.0060	1.0166±0.0056	1.0165±0.0057
M2	**1.0424±0.0062**	1.0912±0.0060	1.0888±0.0060	1.0955±0.0060	1.0921±0.0058
M3	**0.9585±0.0043**	1.0751±0.0099	0.9990±0.0043	1.0476±0.0057	1.0297±0.0187
M4	**1.1387±0.0056**	1.3475±0.0255	1.2087±0.0058	1.3129±0.0071	1.2782±0.0398
M5	**1.0432±0.0058**	1.0874±0.0072	1.0883±0.0072	1.1343±0.0061	1.1311±0.0061
M6	**0.9974±0.0078**	1.0307±0.0061	1.0307±0.0062	1.0384±0.0066	1.0383±0.0066
E1	1.1277±0.0059	**1.1075±0.0070**	**1.1076±0.0070**	**1.1075±0.0070**	**1.1076±0.0070**
E2	1.1914±0.0073	**1.1771±0.0068**	**1.1801±0.0070**	**1.1771±0.0068**	**1.1801±0.0070**
E3	**1.0594±0.0072**	1.0874±0.0075	1.0904±0.0067	1.0887±0.0080	1.0901±0.0068
E4	**1.3133±0.0062**	1.3956±0.0118	1.4014±0.0051	1.3973±0.0099	1.4012±0.0052
E5	**1.1826±0.0045**	1.1965±0.0045	1.2088±0.0046	1.1965±0.0045	1.2087±0.0046
E6	**1.1225±0.0066**	1.1326±0.0067	1.1329±0.0067	1.1325±0.0067	1.1329±0.0067
M_ALL	**1.0352±0.0032**	1.0584±0.0038	1.0584±0.0035	1.0589±0.0034	1.0591±0.0038
M_1	**1.0246±0.0029**	1.0490±0.0036	1.0488±0.0037	1.0483±0.0036	1.0481±0.0038
M_3	**1.0244±0.0046**	1.0460±0.0049	1.0452±0.0044	1.0465±0.0047	1.0450±0.0041
M_5	**1.0260±0.0043**	1.0458±0.0041	1.0457±0.0041	1.0509±0.0040	1.0500±0.0041
HG_ALL	1.0942±0.0087	**1.0806±0.0095**	**1.0804±0.0095**	**1.0805±0.0095**	**1.0804±0.0095**
HG_1	1.1129±0.0082	**1.1008±0.0095**	**1.1007±0.0093**	**1.0986±0.0090**	**1.0983±0.0091**
HG_3	1.1107±0.0144	**1.0923±0.0158**	**1.0922±0.0157**	**1.0953±0.0135**	**1.0953±0.0138**
HG_5	1.1103±0.0176	**1.0966±0.0172**	**1.0966±0.0173**	**1.1000±0.0138**	**1.0997±0.0146**
WB_ALL	1.1368±0.0084	**1.1292±0.0094**	**1.1293±0.0096**	**1.1298±0.0096**	**1.1298±0.0097**
WB_1	**1.1412±0.0084**	**1.1375±0.0104**	**1.1376±0.0104**	**1.1418±0.0102**	**1.1421±0.0103**
WB_3	1.1462±0.0112	**1.1397±0.0111**	**1.1398±0.0113**	1.1462±0.0090	1.1461±0.0088
WB_5	1.1556±0.0141	**1.1483±0.0160**	**1.1483±0.0161**	1.1586±0.0147	1.1587±0.0146
MovieLens	**0.8447±0.0014**	0.8691±0.0017	0.8692±0.0022	0.8688±0.0023	0.8690±0.0018
LastFM	**0.4471±0.0031**	0.4849±0.0032	0.4505±0.0032	0.4913±0.0030	0.4522±0.0032

Table 5.24: The MAE Comparison of Improved INMED results with BPMF on 32 Datasets. The superior results on two sample t-test (at %5 significance level) are shown bold.

Datasets	**BPMF**	**IINMED_I**	**IINMED_II**	**IINMED_III**	**IINMED_IV**
D1	0.8275±0.0055	**0.7895±0.0055**	**0.7893±0.0055**	**0.7895±0.0055**	**0.7893±0.0055**
D2	0.8841±0.0056	0.8568±0.0053	**0.8493±0.0051**	0.8568±0.0052	**0.8493±0.0051**
D3	0.7689±0.0046	0.7927±0.0059	**0.7656±0.0048**	0.7933±0.0058	**0.7657±0.0042**
D4	0.9743±0.0081	1.1438±0.0223	**0.9653±0.0069**	1.1431±0.0227	**0.9662±0.0070**
D5	0.8720±0.0068	**0.8711±0.0063**	**0.8688±0.0062**	**0.8711±0.0063**	**0.8686±0.0061**
D6	0.8240±0.0043	**0.8137±0.0047**	**0.8138±0.0047**	**0.8136±0.0048**	**0.8136±0.0046**
M1	**0.7891±0.0057**	0.8132±0.0049	0.8133±0.0050	0.8111±0.0049	0.8107±0.0048
M2	**0.8453±0.0045**	0.8814±0.0048	0.8774±0.0046	0.8885±0.0052	0.8844±0.0052
M3	**0.7616±0.0033**	0.8187±0.0054	0.7950±0.0032	0.8197±0.0063	0.8124±0.0119
M4	**0.9331±0.0043**	1.1385±0.0268	0.9974±0.0040	1.1078±0.0060	1.0561±0.0387
M5	**0.8418±0.0046**	0.8780±0.0060	0.8771±0.0060	0.9171±0.0048	0.9130±0.0051
M6	**0.7902±0.0062**	0.8101±0.0053	0.8102±0.0053	0.8250±0.0057	0.8248±0.0057
E1	0.9134±0.0048	**0.8801±0.0056**	**0.8800±0.0056**	**0.8801±0.0056**	**0.8800±0.0056**
E2	1.0060±0.0078	0.9727±0.0062	**0.9656±0.0066**	0.9729±0.0063	**0.9657±0.0067**
E3	0.8154±0.0065	**0.8036±0.0078**	0.8308±0.0075	**0.8036±0.0077**	0.8293±0.0074
E4	**1.1225±0.0072**	1.1858±0.0107	1.2043±0.0058	1.1874±0.0096	1.2038±0.0057
E5	0.9922±0.0050	**0.9779±0.0046**	**0.9793±0.0047**	**0.9780±0.0048**	**0.9793±0.0047**
E6	0.9103±0.0052	**0.8835±0.0052**	**0.8836±0.0052**	**0.8836±0.0052**	**0.8837±0.0052**
M_ALL	**0.8180±0.0022**	0.8267±0.0028	0.8264±0.0024	0.8273±0.0026	0.8271±0.0028
M_1	**0.8077±0.0025**	0.8222±0.0030	0.8217±0.0029	0.8224±0.0030	0.8222±0.0031
M_3	**0.8065±0.0034**	0.8229±0.0043	0.8220±0.0038	0.8275±0.0040	0.8257±0.0034
M_5	**0.8077±0.0037**	0.8247±0.0039	0.8242±0.0039	0.8363±0.0046	0.8350±0.0044
HG_ALL	0.8458±0.0059	**0.8098±0.0054**	**0.8098±0.0054**	**0.8101±0.0055**	**0.8101±0.0055**
HG_1	0.8717±0.0065	**0.8345±0.0068**	**0.8344±0.0068**	**0.8377±0.0067**	**0.8376±0.0067**
HG_3	0.8746±0.0083	**0.8364±0.0086**	**0.8364±0.0085**	0.8544±0.0079	0.8541±0.0071
HG_5	0.8763±0.0113	**0.8371±0.0115**	**0.8373±0.0116**	0.8606±0.0110	0.8602±0.0112
WB_ALL	0.9030±0.0062	**0.8804±0.0067**	**0.8802±0.0067**	**0.8815±0.0068**	**0.8814±0.0065**
WB_1	0.9104±0.0068	**0.8914±0.0080**	**0.8917±0.0078**	0.8999±0.0078	0.8998±0.0081
WB_3	0.9182±0.0079	**0.8970±0.0079**	**0.8971±0.0081**	0.9193±0.0074	0.9197±0.0070
WB_5	0.9267±0.0113	**0.9032±0.0119**	**0.9032±0.0119**	0.9295±0.0119	0.9300±0.0119
MovieLens	**0.6613±0.0010**	0.6836±0.0012	0.6837±0.0017	0.6835±0.0018	0.6837±0.0014
LastFM	**0.3250±0.0021**	0.3479±0.0022	**0.3258±0.0020**	0.3550±0.0020	0.3269±0.0019

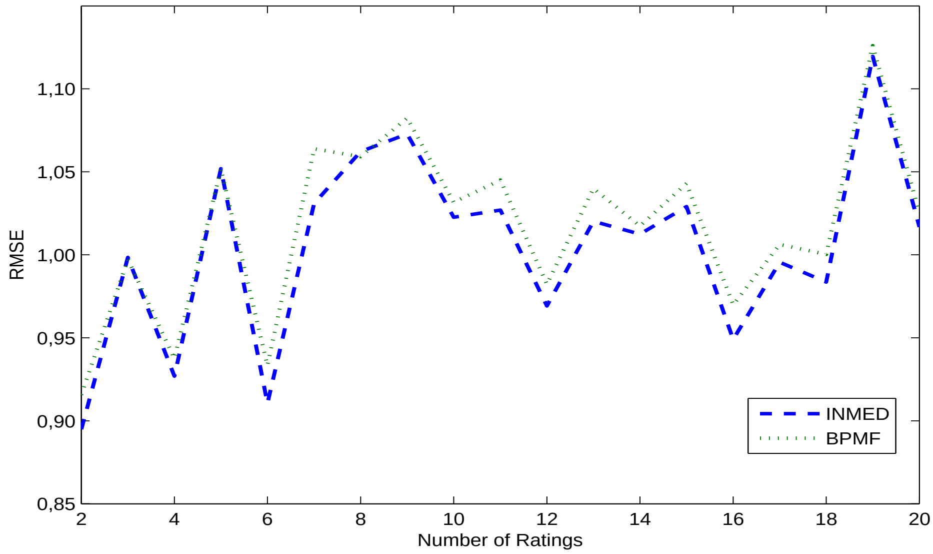

Figure 5.8: RMSE results of predictions using INMED and BPMF on Movielens cold start users.

5.4.2 Comparison with Memory-Based Methods

This study was also compared with the memory-based collaborative filtering approach developed as a remedy for the cold start problem [6]. Although the scope of that study is different from that of the current study, both are similar in terms of proposing solutions for sparse datasets where the dataset comprises a significant number of cold start users. Bobadilla et al. proposed a new similarity metric called the Mean-Jaccard-Differences metric (MJD), which combines six similarity measures to obtain a global similarity metric between the pairs of users using an optimization based on neural learning. A weight w was assigned to each individual measure to specify its relative degree of importance and all weights were obtained in the neural learning process. The MJD similarity between the users x and y (see Equation 5.1) was calculated as follows:

$$MJD_{x,y} = \frac{1}{6}(w_0 v^0_{x,y} + w_1 v^1_{x,y} + w_2 v^2_{x,y} + w_3 v^3_{x,y} + w_4 \mu_{x,y} + w_5 Jaccard_{x,y}) \tag{5.1}$$

v^0 shows the number of cases where both users (x and y) gave the same rating. v^1 and v^2 show the number of cases in which users voted with a difference of one and three scores respectively. v^3 represents the number of cases where both users (x and y) voted different from each other. $\mu_{x,y}$ indicates the mean squared difference and provides the most simple and intuitive measures of similarity. $Jaccard_{x,y}$ shows the proportion of total number of items rated by both user x and user y. MJD was found to be better than all similarity methods including COR, (which performed the worst among all similarity measures) in [6].

Although the proposed initialization method is designed for PMF (a model based approach), MovieLens dataset was used with the same experimental settings described in [6]. MAE was used for comparison but coverage, precision and recall were not included since the scope of

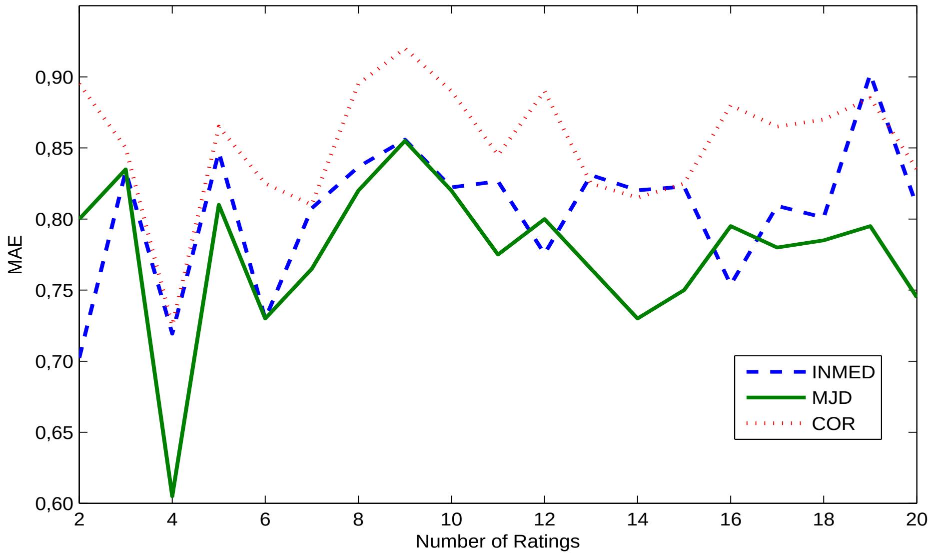

Figure 5.9: MAE results of predictions using INMED, MJD and COR on MovieLens cold start users.

this study did not extend to finding the k-neighbors of each cold-start user requesting recommendations. The experiment was carried out using Movielens dataset, which had 6040 users, 3706 items, and 1000209 ratings. The cold start users were generated inline with the study of Bobadilla et al. [6]. For each user with a number of 20 to 30 ratings, 5 to 20 ratings were randomly discarded to obtain a total of 990472 ratings in the dataset (8449 belonging to cold start users). The leave-one-out cross validation was utilized and then the performance results of the predicted ratings belonging to the cold start users were reported, and found to be in agreement with the results of the compared study. Figure 5.9 shows the MAE results of the predictions made using INMED and collaborative filtering approaches using the MJD and Pearson correlation (COR).

The results show that INMED produced the best result in 3 cases, performed similar to MJD in 4 cases and performed worse than to MJD in 11 cases. Compared to COR, the results from INMED were better in 12 out of 18 cases, similar in 4 cases and worse in 2 cases.

Although MJD produced better results in more cases, this similarity computation approach inherently has the disadvantages of memory-based CF methods. This method need to compute similarity between each pair of users and it is a costly procedure. Additionally, in very sparse datasets the required information to compute similarity might not be sufficient.

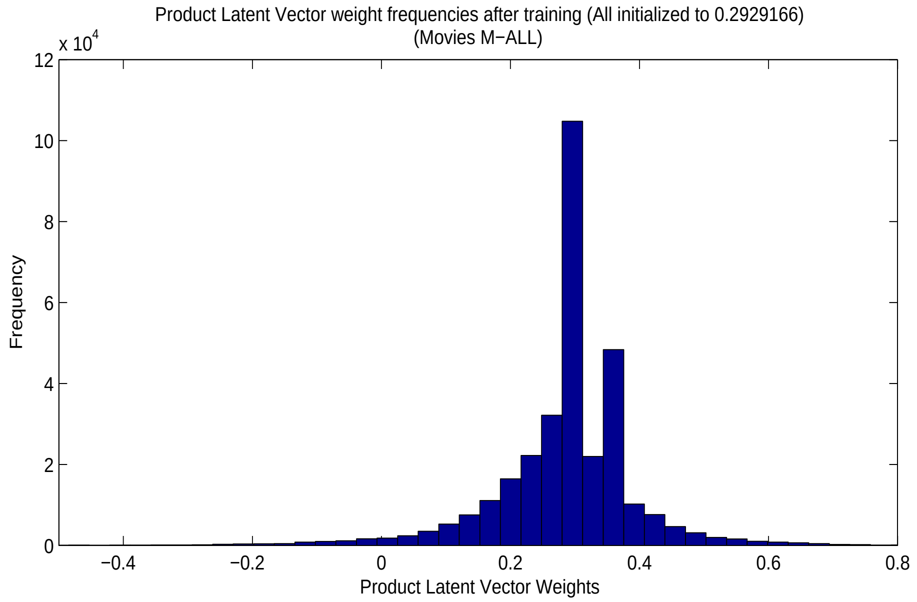

Figure 5.10: The final item (product) latent vector weights after PMF training with INMED method (Movies M-ALL)

Table 5.25: The characteristics of Movies Dense dataset

Datasets	# of Users	# of Products	# of Ratings	Rating Matrix Density
Movies Dense	749	923	32877	4.76e-2

5.5 Findings

5.5.1 Findings on the Effect of Rating Matrix Density on the Initial Values of Latent Vectors

In order to investigate the reason why the initial values in the latent vectors played a significant role in the performance, the latent vector weight distributions were plotted.

Figures 5.10 and 5.11 show that at the end of the training period, the weights of both product and user latent vectors mostly gathered around the initial points. The distribution did not spread much. There is a sharp peak around the initialization point. The results indicated two possibilities: (1) either the training was stuck at the local minima (2) or the majority of users or products had very few ratings (for example, most products may have had one or two ratings); so, the training data was not sufficient to successfully update all the latent vectors. Therefore, a denser dataset was created to understand the real cause.

In order to create the dense dataset, all the users who had given less than 20 ratings in total and all the products that received less than 20 ratings were removed from the original dataset. The dataset characteristics of the Movies Dense dataset are given in Table 5.25.

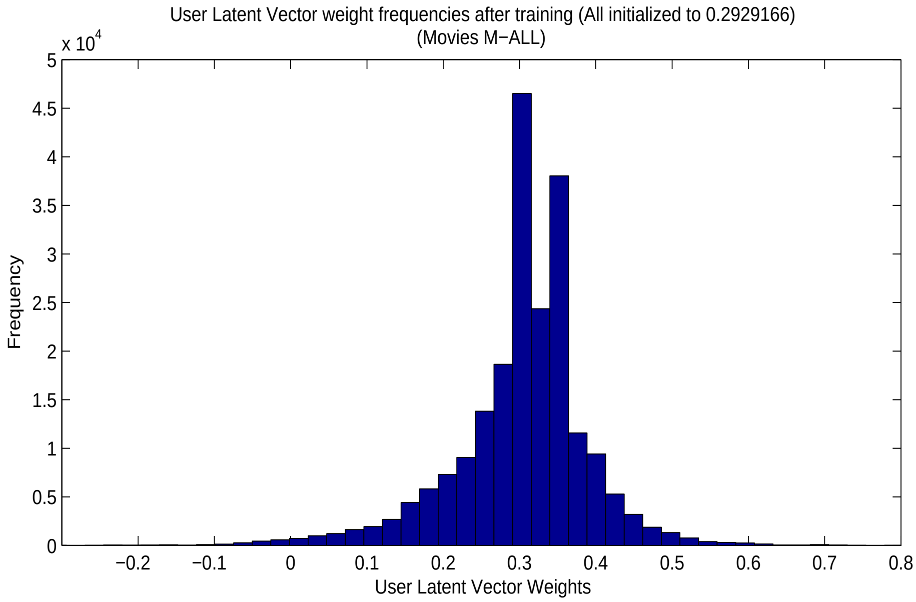

Figure 5.11: The final user latent vector weights after PMF training with INMED method (Movies M-ALL)

The PMF was applied to the dense Movies dataset. After the PMF training, the final product weights were found to have deviated significantly from the initial values (Figures 5.12 and 5.13). The same experimental settings as described in the thesis were used. The kurtosis value of product latent vector weights for M-ALL was 9.4223. However, for the dense Movies dataset the kurtosis value was 2.8398. On the other hand, the kurtosis of user latent vector weights for M-ALL was 6.6794 while it was 3.5489 in the dense Movies dataset.

The results demonstrated that the problem is probably related to the sparseness of the dataset as discussed before. There is a possibility that the training may have been stuck at the local minima; however, the comparisons with different initial values show that the INMED method produces initial points that help the PMF algorithm to converge training close to the global minima.

5.5.2 Findings on the Different Characteristics of RMSE and MAE

The reason why INMED performed better in RMSE but not in MAE in some datasets is as follows: RMSE is a function of 3 characteristics of set of errors [54]. In addition to the average-error magnitude (MAE), RMSE varies with the variability within the distribution of error magnitudes and with the square root of the number of errors [54]. The INMED method was inherently based on finding an initial value that minimized the sum of absolute deviations (corresponding to the mean value) [28] since the objective was to maintain a balanced distribution of error magnitudes among the ratings. The proposed algorithm can be improved to generate initial values that will minimize both RMSE and MAE. A recommended approach for a future study can be to use the least absolute deviation method [11] to find an optimum point in the error space and incorporate this information into the INMED calculation. To clarify the problem, the following experiment was conducted: Table 5.26 shows the RMSE and

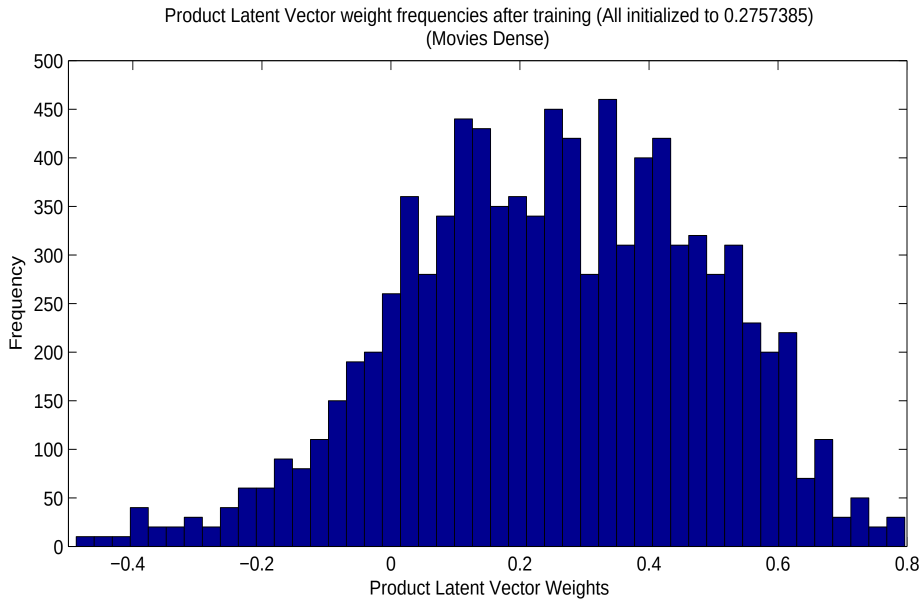

Figure 5.12: The final item (product) latent vector weights after PMF training with INMED method (Movies Dense)

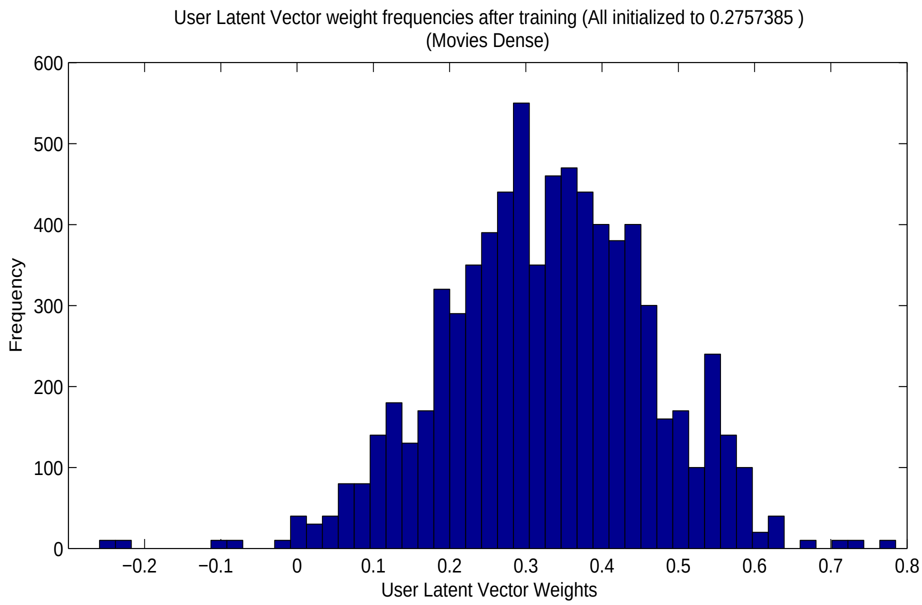

Figure 5.13: The final user latent vector weights after PMF training with INMED method (Movies Dense)

MAE of the predicted ratings for each original rating value (between 1 and 5) using different initialization values in HG-All test dataset. The average number of ratings was obtained by calculating the average number of ratings in the test dataset in 10 folds. For example, in the test dataset, there were approximately 285 ratings equivalent to 1. The results show that while RMSE was minimum for C-0.3 and increased when it was greater than 0.3, MAE was at minimum in the C-0.7 case. However, while MAE of the rating values equivalent to 4 and 5 decreased, the MAE of the rating values equivalent to 1 and 2 increased. This is due to the number of ratings for 4 and 5 constituting a significant part of the dataset; therefore they had the highest impact on the MAE calculation.

5.5.3 Findings on the Effect of User-Dense and Item-Dense Concepts on the Improved INMED Methods on Sparse Datasets

The algorithms used in this study were based on the idea that when a user or item matrix is sparse, it should be initialized with a constant value, and for denser matrices, a random initialization with an interval is necessary. To demonstrate the effectiveness of the proposed approaches, the initialization scheme was changed to initialize a denser matrix with a constant value and a sparser one randomly within the pre-specified interval. In this study, this technique is given the term "opposite scheme". The Improved INMED III, IV and their opposite schemes were analyzed using sparse datasets. The RperU and RperI values and the experiment results are shown in Table 5.27. When the difference between RperU and RperI become greater, the performance of the opposite schemes of the Improved INMED methods reduced. For example, in the dataset HG_1, the RperU and RperI values were 3.38 and 5.07, respectively. The RMSE results of the Improved INMED IV and the opposite scheme were also very close to each other (1.0983 and 1.1014). However HG_5 dataset had RperU and RperI values that were very different from each other (2.42 and 14.55). On this dataset, RMSE results were also different being 1.0997 and 1.1351 for the Improved INMED IV and its opposite scheme, respectively.

5.5.4 Other Findings

- The smaller the rating matrix density and the number of training examples, the better INMED and the Improved INMED based algorithms perform.
- The distribution of the ratings is another important issue in terms of the initialization of latent vectors in PMF. When the skewness of the distribution of the training dataset is close to zero, the mean of the ratings moves closer to the median value. This makes the initialization value of the latent vectors close to zero. In constant initialization approaches, this close-to-zero value makes PMF training difficult to converge. The Improved INMED methods II and IV provide a solution to this problem by moving the initialization values away from zero using the skewness of the dataset.
- In sparse datasets with a limited number of examples and with the mean of ratings above the median, the Improved INMED methods make better predictions than BPMF. However in denser datasets with higher number of examples or with the mean rating value below the median, BPMF RMSE performance is better than that of the Improved INMED methods.

Table 5.26: The distribution of RMSE and MAE among the ratings on Home and Garden category

HG-ALL	Original Rating	RMSE($\mu \pm \sigma$)	MAE($\mu \pm \sigma$)	Avg. # of Ratings
C-0.2	1	2.5360±0.0386	2.4347±0.0414	285.3
	2	1.7556±0.0385	1.6544±0.0421	271.1
	3	0.9467±0.0202	0.8433±0.0213	393.6
	4	0.4672±0.0140	0.3547±0.0090	1355.8
	5	1.0260±0.0133	0.9097±0.0157	2344.5
	All	1.1155±0.0135	0.8792±0.0120	4650.3
C-0.3	1	2.6678±0.0359	2.5699±0.0421	285.3
	2	1.8607±0.0303	1.7769±0.0371	271.1
	3	1.0052±0.0162	0.9320±0.0213	393.6
	4	0.3851±0.0188	0.2843±0.0128	1355.8
	5	0.8816±0.0097	0.7974±0.0124	2344.5
	All	1.0766±0.0151	0.8250±0.0118	4650.3
C-0.4	1	2.7918±0.0373	2.6741±0.0465	285.3
	2	1.9695±0.0248	1.8792±0.0346	271.1
	3	1.0888±0.0182	1.0159±0.0240	393.6
	4	0.4102±0.0154	0.3221±0.0103	1355.8
	5	0.7801±0.0082	0.6935±0.0056	2344.5
	All	1.0772±0.0171	0.8031±0.0118	4650.3
C-0.5	1	2.8693±0.0419	2.7168±0.0520	285.3
	2	2.0470±0.0248	1.9354±0.0374	271.1
	3	1.1656±0.0218	1.0745±0.0275	393.6
	4	0.4943±0.0128	0.3974±0.0086	1355.8
	5	0.7374±0.0121	0.6120±0.0063	2344.5
	All	1.0999±0.0188	0.7948±0.0129	4650.3
C-0.6	1	2.9232±0.0488	2.7407±0.0614	285.3
	2	2.1025±0.0261	1.9727±0.0391	271.1
	3	1.2260±0.0253	1.1196±0.0316	393.6
	4	0.5666±0.0130	0.4584±0.0090	1355.8
	5	0.7243±0.0165	0.5557±0.0084	2344.5
	All	1.1258±0.0203	0.7918±0.0140	4650.3
C-0.7	1	2.9723±0.0545	2.7640±0.0702	285.3
	2	2.1538±0.0267	2.0102±0.0391	271.1
	3	1.2831±0.0265	1.1661±0.0346	393.6
	4	0.6255±0.0126	0.5125±0.0090	1355.8
	5	0.7184±0.0175	0.5067±0.0095	2344.5
	All	1.1501±0.0211	0.7904±0.0153	4650.3
C-0.8	1	3.0857±0.0450	2.8726±0.0592	285.3
	2	2.2451±0.0327	2.1012±0.0424	271.1
	3	1.3578±0.0248	1.2340±0.0329	393.6
	4	0.6629±0.0127	0.5512±0.0102	1355.8
	5	0.6711±0.0139	0.4515±0.0080	2344.5
	All	1.1787±0.0203	0.7916±0.0155	4650.3

Table 5.27: The RMSE Comparison of Improved INMED III and IV results on User-Dense and Item-Dense Concepts.

	RperU	RperI	IINMED III	IINMED III(Opp.Scheme)	IINMED IV	IINMED IV(Opp. Scheme)
M_ALL	8.14	5.49	1.0589±0.0034	1.0586±0.0037	1.0591±0.0038	1.0579±0.0037
M_1	7.45	10.84	1.0483±0.0036	1.0517±0.0035	1.0481±0.0038	1.0522±0.0032
M_3	6.81	19.29	1.0465±0.0047	1.0572±0.0048	1.0450±0.0041	1.0576±0.0045
M_5	6.43	26.13	1.0509±0.0040	1.0734±0.0048	1.0501±0.0041	1.0731±0.0047
HG_ALL	4.59	2.23	1.0805±0.0095	1.0802±0.0094	1.0804±0.0095	1.0801±0.0095
HG_1	3.38	5.07	1.0986±0.0090	1.1014±0.0091	1.0983±0.0091	1.1014±0.0092
HG_3	2.72	10.10	1.0953±0.0135	1.1066±0.0148	1.0953±0.0138	1.1064±0.0147
HG_5	2.42	14.55	1.1000±0.0138	1.1350±0.0138	1.0997±0.0146	1.1351±0.0140
WB_ALL	6.69	2.39	1.1298±0.0096	1.1291±0.0094	1.1298±0.0097	1.1292±0.0095
WB_1	5.22	5.24	1.1418±0.0102	1.1370±0.0103	1.1421±0.0103	1.1370±0.0101
WB_3	4.23	9.94	1.1462±0.0090	1.1644±0.0097	1.1461±0.0088	1.1647±0.0094
WB_5	3.73	14.14	1.1586±0.0147	1.2042±0.0145	1.1587±0.0146	1.2040±0.0147

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, two related problems of PMF have been reported (1) the effects of initialization of the latent vectors on the PMF method for sparse datasets and (2) the impact of different characteristics of datasets on the performance of PMF. For the first problem, different matrix initialization methods for PMF were investigated to find the best approach for very sparse datasets. It was observed that the initialization values of the user and item latent vectors are quite important for sparse datasets since the number of ratings in these datasets is not sufficient. If the initialization values are selected randomly, there is no guarantee that the PMF algorithm will perform its best. In the proposed method, INMED, the user and item latent vectors were initialized with a constant value that was computed using the trimmed mean of the training dataset ratings. As a result, the PMF algorithm made better predictions compared to the other PMF algorithm where user and item matrices were initialized randomly. Then further approaches that were based on the original INMED method were proposed. To evaluate the effects of this method, different datasets with various characteristics were analyzed. It was observed that if the density of the dataset increased, the latent vectors could be initialized with random values generated from a selected interval. The interval boundaries could be determined as their mean being equivalent to the INMED value using the rating matrix density of the datasets. In addition, the user-based and item-based density was investigated and incorporated into the proposed methods. The experiments showed that the methods based on the rating matrix density (Improved INMED I, III) made better or/and comparable predictions than INMED depending on the datasets.

For the second problem, the impact of different characteristics of datasets on the performance of PMF algorithm was investigated. For this study, in addition to the real world datasets, additional datasets were created using two different approaches: sub-sampling and eliminating specific items. Six different distributions were subsampled from three datasets (Douban, Epinions and MovieLens), giving a total of 18 new datasets. As a result, these generated datasets had different distributions as well as a different rating density, number of ratings per user and number of ratings per item.

The results of INMED were found to be better in very sparse datasets but when the number of ratings per user and item increased the performance decreased and the INMED-based methods started to perform similar to or worse than the state-of-the-art methods compared in this study.

As a result of the experiments performed in this study, the following conclusions can be made:

- If the dataset is very sparse with a small number of ratings, constant initialization methods including the INMED based methods can be used. The datasets with a rating matrix density of less than 0.001, the constant based initialization methods (INMED and other constants) make better predictions.

- If the mean of the ratings is close to the median of the rating values, Improved INMED II and IV methods produce better results than the other initialization methods. If the mean and/or trimmed mean value of ratings is close to the median, the original INMED approach produces initialization values around 0. The Improved INMED II and IV methods were proposed as a remedy to the base INMED method and provide better results than other initialization methods, in these cases.

- On cold start users, INMED is better than BPMF and produces comparable results with memory-based methods.

- For sparse datasets with a small number of examples where the mean of the ratings is above the median, the Improved INMED methods make better predictions than BPMF with respect to RMSE. However, in denser datasets with a higher number of examples or a mean rating value below the median, the predictions of BPMF are better than those of the Improved INMED methods.

6.2 Future Work

In this study, a beta distribution was used for the random selection of the initialization values from the specified interval in the proposed methods. The beta distribution parameters were fitted using the concept of rating fuzziness. The random initialization values were placed on user and item latent vectors without considering their rating history. In other words, using the proposed methods it was possible to initialize an item latent vector entry with high values although this item had a low rating average. This information can be incorporated into the approaches to be used in future studies.

Knowing the effect of initialization techniques on PMF can contribute to developing new hybrid methods using other RS approaches. The predictions made using the proposed methods can be used as an input to another RS. Additionally the rating predictions of the proposed methods can be combined with the predictions of other techniques to obtain better results.

REFERENCES

[1] R. Adams, G. Dahl, and I. Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 1–9, Corvallis, Oregon, 2010. AUAI Press.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[3] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez. Collaborative filtering based on significances. *Information Sciences*, 185(1):1–17, 2012.

[4] J. Bobadilla, F. Ortega, and A. Hernando. A collaborative filtering similarity measure based on singularities. *Information Processing & Management*, 48(2):204–217, 2012.

[5] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. Generalization of recommender systems: Collaborative filtering extended to groups of users and restricted to groups of items. *Expert Systems with Applications*, 2011.

[6] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26:225–238, 2012.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46(July 2013):109–132, 2013.

[8] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52, Madison, Wisconsin, 1998. Morgan Kaufmann Publishers Inc.

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

[10] F. Carmagnola, F. Vernero, and P. Grillo. Sonars: A social networks-based algorithm for social recommender systems. *User Modeling, Adaptation, and Personalization*, pages 223–234, 2009.

[11] K. Chen, Z. Ying, H. Zhang, and L. Zhao. Analysis of least absolute deviation. *Biometrika*, 95(1):107–122, 2008.

[12] M. Chen and J. P. Singh. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 154–162, Tampa, Florida, USA, 2001. ACM.

[13] W. Chen and S. Fong. Social network collaborative filtering framework and online trust factors: A case study on facebook. In *Digital Information Management (ICDIM), 2010 Fifth International Conference on*, pages 266–273. IEEE, 2010.

[14] J. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, Faculty of the Graduate School of the University of Maryland, College Park, 2005.

[15] J. Golbeck. Generating predictive movie recommendations from trust in social networks. *Trust Management*, pages 93–104, 2006.

[16] J. Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web (TWEB)*, 3(4):1–33, 2009.

[17] J. Golbeck and J. Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, volume 96, 2006.

[18] Movielens dataset. `http://grouplens.org/datasets/movielens/`. Accessed: 2013-12-15.

[19] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, Berkeley, California, USA, 1999. ACM.

[20] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, Berkeley, California, USA, 1999. ACM.

[21] C. Hwang and Y. Chen. Using trust in collaborative filtering recommendation. *New Trends in Applied Artificial Intelligence*, pages 1052–1060, 2007.

[22] Lastfm dataset. `http://grouplens.org/datasets/hetrec-2011/`. Accessed: 2013-09-20.

[23] Douban dataset. `https://www.cse.cuhk.edu.hk/irwin.king/pub/data/douban`. Accessed: 2013-09-25.

[24] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.

[25] H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[26] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer, 2011.

[27] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[28] D. M. Lane. Median and mean. `http://onlinestatbook.com/lms/summarizing_distributions/mean_median_def.html`. Accessed: 2014-01-01.

[29] F. Liu and H. Lee. Use of social network information to enhance collaborative filtering performance. *Expert Systems with Applications*, 37(7):4772–4778, 2010.

[30] J. Liu, C. Wu, and W. Liu. Bayesian probabilistic matrix factorization with social relations and item contents for recommendation. *Decision Support Systems*, 55(3):838–850, 2013.

[31] P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[32] H. Ma, I. King, and M. R. Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):29, 2011.

[33] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940, Napa Valley, California, USA, 2008. ACM, ACM.

[34] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296, New York, NY, USA, 2011. ACM.

[35] H. Ma, T. C. Zhou, M. R. Lyu, and I. King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems (TOIS)*, 29(2):9, 2011.

[36] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508, 2004.

[37] P. Massa and P. Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI Workshop on Recommender Systems*, pages 29–33, 2006.

[38] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.

[39] P. Massa and P. Avesani. Trust metrics in recommender systems. *Computing with social trust*, pages 259–285, 2009.

[40] P. Massa and B. Bhattacharjee. Using trust in recommender systems: an experimental analysis. *Trust Management*, pages 221–235, 2004.

[41] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[42] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174, San Diego, California, USA, 2005. ACM.

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

[44] I. Porteous, A. U. Asuncion, and M. Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.

[45] B. Pradel, S. Sean, J. Delporte, S. Guérif, C. Rouveirol, N. Usunier, F. Fogelman-Soulié, and F. Dufau-Joel. A case study in a recommender system based on purchase data. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–385. ACM, 2011.

[46] C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

[47] S. Ray and A. Mahanti. Improving prediction accuracy in trust-aware recommender systems. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–9. IEEE, 2010.

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

[49] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887, Helsinki, Finland, 2008. ACM, ACM.

[50] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2008.

[51] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, Hong Kong, Hong Kong, 2001. ACM.

[52] Epinions dataset. `http://www.trustlet.org/wiki/Downloaded_Epinions_dataset`. Accessed: 2013-07-30.

[53] F. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.

[54] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30(1):79, 2005.

[55] L. Wu, E. Chen, Q. Liu, L. Xu, T. Bao, and L. Zhang. Leveraging tagging for neighborhood-aware probabilistic matrix factorization. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1854–1858, Maui, Hawaii, USA, 2012. ACM.

[56] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 211–218. ACM, 2009.

[57] W. Yuan, L. Shu, H. Chao, D. Guan, Y. Lee, and S. Lee. itars: trust-aware recommender system using implicit trust networks. *Communications, IET*, 4(14):1709–1721, 2010.

[58] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *SIAM International Conference on Data Mining (SDM)*, volume 12, pages 403–414, 2012.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: AR, YILMAZ
Nationality: Turkish (TC)
Date and Place of Birth: August 5, 1976, NALLIHAN
Marital Status: Married
Phone: 5052606221

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Syracuse University	2001
B.S.	Uludağ University	1997
High School	Nallıhan High School	1993

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2005-2014	Ankara University	Research Assistant