FILLING MISSING RATINGS IN PRIVACY-PRESERVING COLLABORATIVE FILTERING SYSTEMS

Master of Science Thesis

Mehmet ÖZCAN

Eskişehir, 2018

FILLING MISSING RATINGS IN PRIVACY PRESERVING COLLABORATIVE FILTERING SYSTEMS

Mehmet ÖZCAN

MASTER OF SCIENCE THESIS

Department of Computer Engineering
Supervisor : Assist. Prof. Dr. Alper BİLGE

Eskişehir
Anadolu University
Graduate School of Science
January, 2018

FINAL APPROVAL FOR THESIS

This thesis titled “Filling Missing Ratings in Privacy-Preserving Collaborative Filtering Systems” has been prepared and submitted by Mehmet ÖZCAN in partial fullfillment of the requirements in “Anadolu University Directive on Graduate Education and Examination” for the Degree of Master of Science in Computer Engineering Department has been examined and approved on 04/01/2018.

	Commitee Members	**Signature**
Member (Supervisor)	: Assist. Prof. Dr. Alper BİLGE	....
Member	: Assoc. Prof. Dr. Cihan KALELİ	....
Member	: Assoc. Prof. Dr. Ayhan İSTANBULLU	....

....

Director

Graduate School of Science

ABSTRACT

FILLING MISSING RATINGS IN PRIVACY-PRESERVING COLLABORATIVE FILTERING SYSTEMS

Mehmet ÖZCAN
Computer Engineering Department
Anadolu University, Graduate School of Science, January, 2018
Supervisor: Assist. Prof. Dr. Alper BİLGE

Collaborative filtering is an influential personalized recommendation technique deducing like-minded users from their ratings and producing predictions for them. However, the first controversial issue with this technique is that people may share a lot of individual information with collaborative filtering systems, which brings serious privacy risks. Privacy-preserving collaborative filtering algorithms are mainly contrived to deal with this privacy challenge. Missing values in the collected data set is another major issue in collaborative filtering systems. Users usually do not rate all items; conversely, they rate only a limited number of them because there are too many items to rate. Accordingly, there exists insufficient information to locate similar users correctly and generate accurate predictions. There are readily available methods in the literature constituted to overcome this problem. While some of these methods try to impute the missing values by only using the available data, the others utilize auxiliary data. The objective of this study is to apply some of the missing data imputation methods using no auxiliary data on several privacy-preserving collaborative filtering algorithms in order to boost the recommendation quality. Existing missing data imputation methods are modified in such a way that they can be applied to perturbed data. Several experiments are performed using a real data set to show how effective the methods are in privacy-preserving collaborative filtering systems.

Keywords: Privacy, Collaborative Filtering, Missing Value.

ÖZET

GİZLİLİK TABANLI ORTAK FİLTRELEME SİSTEMLERİNDE KAYIP DEĞERLEMELERİ İŞLEME

Mehmet ÖZCAN
Bilgisayar Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Ocak, 2018
Danışman: Yrd. Doç. Dr. Alper BİLGE

Ortak filtreleme kullanıcıların değerlemelerini kullanarak benzer kullanıcıları tespit eden ve onlar için tahmin üreten etkili bir kişiye özgü öneri tekniğidir. Fakat bu teknikle ilgili tartışmalı birinci mesele, ciddi mahremiyet risklerini de beraberinde getiren birçok kişisel bilginin ortak filtreleme sistemleri ile paylaşılmasıdır. Gizliliği koruyan ortak filtreleme algoritmaları temelde bu gizlilik sorununu çözmek için geliştirilmiştir. Toplanan veri setindeki kayıp değerler ise ortak filtreleme sistemlerinin bir başka temel sorunudur. Ortak filtreleme sistemlerinde çok fazla ürün olduğu için genellikle kullanıcılar bu ürünlerin hepsini oylayamadığı gibi aksine az ve sınırlı sayıda ürünü oylamaktadırlar. Bunun sonucunda ise benzer kullanıcıları bulup onlar için doğru tahminler üretmek için yetersiz miktarda bilgi elde edilmektedir. Çok boşluklu kullanıcı-ürün matrisi ortak filtreleme algoritmalarının genel performansını olumsuz etkilemektedir. Literatürde bu problemi ortadan kaldırmak için oluşturulmuş hali hazırda bazı yöntemler mevcuttur. Bunlardan bazıları elde bulunan verileri kullanarak kayıp değerleri doldururken diğerleri ise yardımcı verilerden faydalanırlar. Benzer problem gizlilik-tabanlı ortak filtreleme algoritmalarında da mevcuttur. Bu çalışmanın amacı, çeşitli gizlilik temelli ortak filtreleme algoritmaları üzerinde bazı yardımcı veri kullanmayan kayıp değer doldurma metotlarını uygulayarak öneri kalitesini arttırmaktır. Hali hazırdaki metotlar maskelenmiş seyrek veriye uygulanacak şekilde değiştirilecektir. Gerçek veri setleri ile deneyler yapılarak önerilen kayıp değerlemeleri işleme yöntemlerinin gizlilik-tabanlı ortak filtreleme sistemlerinde ne kadar başarılı oldukları tespit edilecektir.

Anahtar Sözcükler: Gizlilik, Ortak Filtreleme, Kayıp Veri.

04/01/2018

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Anadolu University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

....

Mehmet ÖZCAN

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CBS	**:** Cluster-Based Smoothing
CF	**:** Collaborative Filtering
IBCF	**:** Imputation-Boosted Collaborative Filtering
IM	**:** Imputation with Mean
KNN	**:** *k*-Nearest Neighbor
MAE	**:** Mean Absolute Error
NB	**:** Naïve Bayes
PMM	**:** Predictive Mean Matching
PPCF	**:** Privacy-Preserving Collaborative Filtering
RPT	**:** Randomized Perturbation Technique
RS	**:** Recommender System
SVD	**:** Singular Value Decomposition

1. INTRODUCTION

Growing considerably in recent years, the Internet takes part in every aspect of our lives together with the facilities it provides. It takes out the time and space constraints that one can conclude most of its intended jobs at anytime and anywhere instantly. It is also enabled to reach devilish information which is another benefit gained by use of the Internet. People are increasingly resorting to guidance of the Internet rather than taking support of someone. However, there is a drastic issue that accessing what you need among extraordinary amount of data and numerous commodity is not so easy, which is referred to as information overload problem. People get a complication while searching a book to read, movie to watch, or anything to buy due to the variety of options. At this point, recommender systems play a substantial role in resolving such issue.

A recommender system (RS) aims to create pointed recommendations to a collection of users on products that they might interest [1]. Individual as a user maintains some input to the system first. Such input to the system may belong to one of the following categories: ratings (votes for items), demographic information (age, gender, education, etc.), content data (textual documents like comments on items) or contextual information (identity of people around, date, season, temperature, etc.) [2, 3]. Then the system handles these gathered inputs with some data mining techniques and generate the requested outputs by the user. The outputs may be in the form of a prediction or top-*N* recommendations [2]. While prediction represents the degree of interest for an item, top-*N* recommendations are the suggested item lists which are considered to be the active user will most be interested in.

1.1. Collaborative Filtering

Used widely on e-commerce sites, collaborative filtering (CF) is one of the most efficient RS techniques. It has been used in an e-mail system named Tapestry first to help users find out the right documents to read via one another's reviews [4]. Typically in a CF system, users log in and rate some items among a collection. At the background, ratings of each user are collected in the server in order to construct a user-item matrix where each row represents a user and each column represents an item. Hence a rating value in *i*'th row and *j*'th column of the matrix corresponds to the rating given by *i*'th user for *j*'th item. When a user asks for a prediction or recommendation, the system makes some calculations on the user-item matrix and sends feedback to the user.

The idea behind the technique is that it deduces like minded users by making use of their votes for items and generates personalized recommendations for individuals. It is assumed that having similar activities before, users have similar tastes. Therefore, they are going to like similar items in the future, as well.

Although CF is a widely used personalized recommendation technique, there are numerous challenges they face.

1.1.1 Sparsity

Data sparsity is one of the major problems in CF which implies there exists insufficient information to create high-quality recommendations in the dataset [5]. In a CF system, there exists an extensive number of items while users tend to rate a small fraction of them as they do not want to spend much time or just have an idea about a limited number of them. Causing plenty of missing ratings, such situation is the main reason for sparsity. In order to handle missing values, densification of the dataset is needed. Favorable techniques utilized to densify the dataset are based on either dimensionality reduction [6] or imputation such as Transfer learning [5], imputation-boosted, and content-boosted CF [7, 8].

1.1.2 Scalability

There may be millions of users and items available in a typical CF system. In these conditions finding similarities between users or items turns out an increase in computation time which causes a poor online performance. Moreover, considering a lot of items together with a high rate of missing data, number of commonly rated items is not sufficient to find out the neighborhoods. On challenging these two critical problems, dimensionality reduction [9] and clustering [10] methods are preferred mostly.

1.1.3 Accuracy

A CF system must provide decent recommendations for users which is possible by predicting user preferences on items precisely. The accuracy of predictions can be assessed by generating predictions on available ratings and comparing them with the actual ones. Widespread measurement metrics used in CF are Mean Absolute Error (MAE), Normalized Mean Absolute Error, Root Mean Squared Error, and Mean Squared Error for accuracy of prediction and ROC curve, Precision, Recall, and F1-score for the reliability of recommendations [11]. Indicating error rate of the predictions, MAE is used in the experiments.

1.1.4 Online Performance

CF systems work online so that when a user asks for a recommendation, it returns intended result dynamically to the user. The system has to provide recommendations as quickly as possible since the user does not willing to wait meanwhile. That is why the system must produce instant predictions online. Model-based CF approach is an efficient way of having a decent online performance for large datasets [12].

1.1.5 Cold start

A user newly registered to a RS has only a small number of votes which makes it difficult to analyze neighborhood of that user as the user profile has not been understood well. Such phenomena is referred to as cold-start user problem [13]. Similarly, when a brand new item is attached to the system, it is rated by only a limited number of users. Accordingly, it is rarely recommended for the users. Similarly, this is referred to as cold-start item problem [14]. For a cold-start user problem, it is possible to exploit the user's implicit feedbacks such as user clicks or the time consumed on a web page. Against cold-start item problem, content boosted CF is an efficient approach.

1.1.6 Black and gray sheep

Some users in a RS may have a unique profile that does not match with another user in an opposite or positive way. Applying neighbor selection for such a user does not bring correct revelation about that user. Such phenomena is referred to as black sheep challenge [15]. In a gray sheep situation, a user has similarities with more than one group of users at almost the same extent [15]. It is a difficult job to discover the characteristics of such users as they do not belong to a group precisely. This kind of problems result in a poor prediction accuracy which traditional CF cannot deal with. Several hybrid CF techniques are utilized for attempting a solution for this problem.

1.1.7 Shilling attacks

There may be certain competitors of an organization having CF system particularly when the organization is commercial. In this case, the competitors persumably attempt to deflect the ratings in their own benefit. This kind of attacks coming from competitors are called as shilling attacks which causes an RS generate poor recommendations [16]. The simplest form of these attacks is inserting fake lowest ratings for the items that the attacker wants to decrease the popularity. Except, there are numerous types of shilling attacks which can make an RS confuse unusually. Therefore a CF system has to be designed robust against these attacks. As a precaution, CF systems must enable versatile shilling attack detection algorithms to get rid of fake ratings.

1.1.8 Synonymy

Almost the same items might have more than one entries in a CF system which is referred to as synonymy challenge. In this scenario, the system treats these items in a completely distinct manner. As a consequence, these items would get different rating values coming from users. The most dangerous situation here is that the system explores a dissimilarity between these items [15]. To sum up, the more the synonymy occurs, the worse the precision becomes.

1.1.9 Overfitting

Overfitting or over specialization is the term that there is no variety in recommendations of the system for a user that those items which are already rated and have high scores are constantly served [17]. By this way, those items get new scores from that users and recommended highly again to new users which is a vicious circle. The other items goes useless for recommendation meanwhile. With the intention to cope with the issue, hybrid CF approaches mobilizing content information of items and/or demographic information of users can be preferred.

1.1.10 Privacy

Privacy has a critical place among the challenges of CF. When a user votes for an item, the user also shares a knowledge about themselves. There is a possibility that the knowledge is put in a process against the user. Profiling, price discrimination, government surveliance are the several threats expected a user to face [18]. Whether it is clearly indicated with an agreement between vendor and user that the data submitted by user is approved as private and will not be shared with third parties, the vendors would rather sell it in a case of bankruptcy to rivals. Resolving the issue, numerous solutions are studied under the topic of Privacy-Preserving Collaborative Filtering.

1.2. Privacy-Preserving Collaborative Filtering

The term privacy-preserving corresponds to protecting confidential data from others who is malevolent. Confidential data of a user in terms of a CF system refers to the rating values the user leave and the list of rated items by the user. Goal of a Privacy-Preserving Collaborative Filtering (PPCF) system can be described as producing meaningful recommendations for users while protecting the privacy of them.

Preserving privacy in CF is expressed first in 2002 by John Canny [19]. He indicated the reasons why it is needed to protect privacy in CF and proposed a framework to ensure privacy. Since then, many approaches have been studied on about PPCF. The best way in order to comprehend the aspects of PPCF is to investigate it under three captions namely individual privacy, corporate privacy, and techniques utilized for preserving privacy.

1.2.1 Individual privacy

Collaborative filtering is an influential tool serving personalized recommendations for individuals in order to help them reach appropriate products. However, individuals are hesitant to use CF systems due to privacy concerns.

1.2.1.1 Peer-to-peer collaboration

Users may not want to share their confidential data with a server due to privacy concerns. In order to provide CF services for those users, framework based on peer-to-peer data collaboration emerges, which removes any need of server security. In this case, a group of users compose an aggregate appropriate for getting CF recommendations without deeply disclosing their individual data.

1.2.1.2 Central server

Likewise in a traditional CF approach, there is a central server which holds whole data coming from users except that the central server collect individuals' ratings in a way that it does not violate privacy. In terms of protecting individual privacy, there are two key issues to be watched out by the server [20]. Firstly, the server must not be able to understand how much an item interested by a user. In this context, the server cannot reach actual rating values of users in the system. Secondly, the server must not be able to know the items rated by any user. Without understanding such confidential knowledge, the system must be able to deliver decent referrals.

1.2.2 Corporate privacy

In contemplation of boosting quality of CF services, two or more vendors may attempt to share their data collected from their users. They can elaborate user-item matrices with the data gathered from each others which is an effective way to beat down sparsity. Nonetheless, those vendors hesitate to make a communion by virtue of security reasons. These vendors must be withheld obtaining actual ratings from one another's. In the view of resolving this issue, confidentiality of the collaboration can be ensured.

1.2.2.1 Two-party collaboration

Collaboration of two parties works when there is an intersection available between the two datasets. There must be at least some users or items common. Assuming the data gathered from two vendors A and B as a whole, it is presumably distributed horizontally, vertically, or arbitrarily between A and B.

The case of having the same items while having different users represents a horizontally distributed data. Merging horizontally distributed data is a good idea for those newly established a CF system and having a few users. In this way, the system gets the capability of finding out more similar users as neighbors. Increasing the number of users derives various rated items which enable the systems to generate recommendations diversely.

Users who are registered to CF systems of both vendors votes for different items available in two systems separately. Such phenomena is considered as the vertically distributed data form. The systems having an insufficient number of items on implying a user profile are better to share their data with each others when they have common users. However, they must not violate the privacy while sharing.

It is more familiar to be encountered with an entity of arbitrarily distributed data. Regarding that distribution, two vendors in the same sector have some common users and items in their systems. Those users vote for some items in system A, while vote for some others in system B. Gathering the datasets of A and B together compounds a denser dataset which is favorable for those two vendors regarding prediction accuracy.

1.2.2.2 Multi-party collaboration

More than two vendors may want to collaborate with the objective of generating mighty predictions. There can be the horizontal or vertical distribution of data among multiple parties likewise between two parties. It is possible for those parties to utilize one another's data without damaging the confidentiality.

1.2.3 Techniques utilized in preserving privacy

Several techniques are contributing to protecting individual privacy which aggravates unauthorized parties access confidential data. Such techniques applied in PPCF are described briefly in this section.

1.2.3.1 Data obfuscation

Data obfuscation is one of the most popular techniques used in PPCF schemes. Intention of using this technique is to turn actual ratings of users into an undistinguishable format for others. The goal is achieved by systematically adding noise to confidential data such that CF systems still be able to create personalized recommendations via aggregation. The methods of data obfuscation having a high vogue in PPCF can be specified as randomized perturbation [21], obfuscation by substitution [22], obfuscation by permutation [23], and randomized response [24].

1.2.3.2 Cryptography

Having a widespread use in many fields, cryptography is utilized in PPCF schemes, as well. In PPCF, the actual data is ciphered completely with some mathematical operations. Then the CF framework is modified in a way that is suitable for generating consistent predictions without deciphering the data. Homomorphic encryption is the cryptographic technique which is mostly adopted in PPCF systems [19].

1.2.3.3 Anonymization and aggregation

Instead of altering the data, anonymization focuses on obscuring who the data belong. A PPCF system carries out CF on an aggregate comprising of individuals' actual data but is not able to learn rating profile of a user. k-anonymity method as used in [25] and peer-to-peer architectures proposed in [19, 26] are the popular examples of anonymization and aggregation.

1.2.3.4 Trust-agent based methods

Regarding CF, the implication of trust is the grade of a user showing how much the user's votes can be taken as a reference when generating a prediction. Promoting third parties is a compromising way of assessing the trust levels of users. Agents as secure third parties play an important role in granting trust with protecting privacy. A prospering agent has the capabilities of autonomy, mobility, co-operation, development by learning, security, and so on [27]. Those recommender systems professedly collaborating with anonymized agents get an advantage on improving accuracy and privacy. Various issues and approaches on trust-agent based systems are discussed elaborately in [28, 29]. In addition, [30] is a fine example demonstrating technicality of working and architecture of an agent.

2. PRELIMINARIES & RELATED WORK

2.1. Categories of CF Techniques

CF techniques are mainly scrutinized under the following three categories: memory-based, model-based, and hybrid techniques.

2.1.1 Memory-based CF

Memory-based CF systems generate predictions by practicing a prediction algorithm directly on the user-item matrix or a sample of it. There are user based and item based approaches within this category. For a user based approach, it is possible to summarize CF process in three main steps:

1) Similarity Weighting: Each user is given a weight representing what extent they are similar to the active user via a similarity measurement. Similarity measurements frequently utilized for similarity weighting in CF can be listed as follows: Pearson's Correlation Coefficient [31], Cosine Similarity [32], Jaccard Similarity [33], Mean Squared Difference [34], Spearman's Rank Correlation [35], Constrained Pearson Correlation [36], and Adjusted Cosine Similarity [37].

2) Neighbor Selection: After users are weighted, most similar ones are chosen to be used in prediction algorithm. Two basic ways of the selection are *k*-nearest neighbor (KNN) method and threshold-based selection method. While *k* users with the largest weights are selected in KNN; in the threshold based one, a threshold value is assigned first and users having higher similarity weight than the threshold are selected as neighbors.

3) Prediction: Once the neighbors are determined, a prediction is generated indicating how much the target item will be liked by the active user. Utilized prediction algorithms in CF are prominently based on weighted average and regression [38].

In an item-based approach, it is assumed that if an item is liked by a user, then the items similar to that item most probably will interest the user, as well. The steps remain the same for an item-based approach except that they are based on items. In the similarity weighting step, items are weighted according to their similarity degrees with others. Consequently, a weight matrix which holds the weight values between items is estimated. In neighbor selection step, commonly, most similar *k* items with the target item are specified as neighbor items. In the prediction step, item-based implementation of a prediction algorithm is utilized in order to generate a prediction value.

The most powerful speciality of memory-based CF techniques is their splendid accuracy. When compared to the other CF techniques, memory-based techniques have significantly higher predictive performance. Furthermore, to include a new user or item does not bring any cost of additional operation for the prediction procedure since memory-based CF algorithms run entirely online.

On the other hand, memory-based CF techniques have some critical disadvantages. Because memory-based CF algorithms run dynamically, online performance gets lowered. Accordingly, the user experience is influenced in a wrong way. Moreover, memory-based techniques suffer from scalability issue that they are handling a large dataset; they cannot detect the neighborhood correctly. Such a mistake in neighbor selection results with an inaccurate prediction.

2.1.2 Model-based collaborative filtering

Characteristic of a model based CF algorithm is that it utilizes a model derived by the dataset to generate predictions. This kind of algorithms comprise of two phases named offline phase and online phase. While in the offline phase, model construction process is carried out; in the online phase, a prediction algorithm is performed on previously constructed model to generate a prediction value. In the model construction progress, some data mining techniques are used such as Singular Value Decomposition, Bayesian models and some of the classification, clustering and regression methods [12]. Prediction algorithms utilized in model-based CF techniques vary since they are specific to the constructed model.

Whereas time complexity of model designation is so high, this progress is done within some periods when the server is available for this job. Thereby, model construction does not deteriorate the online performance of the CF system. Conversely, since the model is constructed offline, the online workload is reduced which results in a higher online performance than memory-based CF techniques. Another advantage of the model-based approach is that it can overcome large datasets. Utilized data mining techniques in model construction are good at finding out meaningful relationships on big data which enables a model-based approach to generate scalable predictions. Model-based CF approach can address sparsity issue exploiting several dimensionality reduction techniques as well.

Along with the mentioned advantages, there are several disadvantages appeared in model-based CF techniques. First of all, adding a new user or item is not convenient, inasmuch as they are needed to be adapted to the system by model construction which is a time and resource consuming process. Secondly, there may also be loss of information due to the data mining techniques applied. Such fact results in a lower prediction accuracy with the CF system.

2.1.3 Hybrid collaborative filtering

Hybrid CF systems are developed in the view of combining the strengths of both memory-based and model-based approaches. Such systems practice traditional memory-based CF together with taking advantage of either data mining techniques used in model-based approach or content-based RS.

Together with the data mining techniques, CF can deal with high dimensional data thanks to Singular Value Decomposition, Principal Component Analysis, Latent Semantic Indexing and so on. Data mining techniques are also good at extracting influential patterns from big data which leads to more accurate predictions when combined with the power of CF on analyzing similarities between users and items.

Content-based methods make use of the description of items to make recommendations [39]. When compared to CF, they are not accurate in generating personalized predictions to the extent that CF does. Nevertheless, when these methods are attached to CF, a significant improvement can be observed with the accuracy. Content-based methods provide a utility of extra content info about items which is particularly useful for turning a sparse dataset into a denser one. Note that, with a consistently imputed dataset, CF can provide more incisive recommendations to the users.

2.2. Imputation Boosted Collaborative Filtering

Bringing out a decrease in coverage and accuracy, data sparsity is among the major problems of recommender systems. Missing value imputation is one of the prevalently utilized solutions in order to deal with such issue. According to the imputation-boosted approach, a preprocessing operation is performed on the dataset that several missing values in the dataset are exchanged with newly assigned ones. Then the system works with the new form of the dataset. In a CF scenario, the missing values are the null ratings in a user-item matrix and imputation of missing values refers to filling missing ratings.

Missing value imputation methods can be inquisited through groups of two concerning calling out auxiliary data. While methods in the first group transfer an extra knowledge from the other available dataset, the methods of the second group try to fill missing values by extracting knowledge from the available dataset (related works belong to those groups are mentioned respectively with the following two paragraphs).

Chujai et al. [8] determines frequent item sets with respect to items' genre information and users' demographic information via an association rule mining algorithm called apriori. Then, average values for missing ratings are calculated by combining frequent item set information with rating frequency of the genres. Hwang et al. [40] has made use of users' trust information for estimating missing ratings before applying CF. In an attempt for imputation, reliable neighbors are determined via trust network and the missing value is filled with average ratings of them for interested item. Xia et al. [41] has filled missing values with two new approaches which are average ratings of the users with the same age and occupation. In study [42], Pan et al. have intended to take advantage of an auxiliary dataset in binary form to alleviate sparsity by transferring knowledge from it. Transfer by Collective Factorization has been proposed which jointly factorizes the data matrices in three parts: user specific latent feature matrix, item specific latent feature matrix and two data-dependent core matrices.

Su and Khoshgoftaar [7] has employed both user mean imputation and extended Bayesian multiple imputation on memory-based CF with several neighbor selection methods. The results represent that extended Bayesian multiple imputation significantly enhances accuracy. Su et al. [43] has prosecuted a comparative study that practices nine types of machine learning classifiers, predictive mean matching (PMM) and an ensemble classifier arranged with seven classifiers that yield the best results out of nine for imputation. Those imputation methods have not only been implemented on raw ratings but also on users' demographic information. It results in this study that IBCF using naïve Bayes provides relatively better results. Moreover, IBCF using naïve Bayes (NB), IBCF using ensemble classifier and IBCF using PMM deliver higher accuracy on raw data rather than the user contents. Similarly in [44], Su et al. have performed a comparison-based study for numerous IBCF techniques such as IBCF with PMM, IBCF with Linear Regression, IBCF with item mean, IBCF with NB, content-boosted CF, and a mixture IBCF. While an imputation is made utilizing content information of users via NB in the content-boosted CF, there is used a divide-and-conquer strategy based on NB and item mean in the mixture IBCF. In study [45], Ren et al. have proposed auto adaptive imputation method to overcome these issues so that it determines the key set of missing values to impute automatically with respect to the active user and the target item. The perception of selecting the key set is that users related to the active user and items related to the target item contain the most informative ratings for prediction. Auto-adaptive imputation method is applied from both user and item aspects in the provided collaborative filtering algorithm in order to dig out the effects of user activity and item popularity. Ranjbar et al. [46] proposes to enhance multiplicative update rules algorithm for CF via imputation with 4 types of average ratings: User average, item average, total ratings' average and a hybrid average which is constituted by linear combinations of user and item averages. Huang and Gong [47] have modified ROUSTIDA algorithm for imputation and tested on a user-based CF technique. Gong [48] has prompted user based and item based k-means clustering algorithms to impute vacant ratings. After detection of the clusters, missing values are imputed with the centroids of those clusters which is a smoothing approach. Zhang and Li [49] have designed a rough sets based imputation as well with the object of mitigating the sparsity. Xue et al. [50] has introduced a scalable CF algorithm taking advantage of a cluster based smoothing. In their smoothing algorithm, users are separated into clusters first. A missing rating is imputed with an average of deviation from mean ratings of users in attached cluster given for that item. Ma et al. [51] have made a missing data imputation for CF again via CF approach. For a missing value, a user-based and an item-based CF prediction values are generated first. Then a unified prediction value is created totalizing those previously generated values with specified percentages. Abdelwahab et al. [52] have alleviated data sparsity by employing an iterative prediction method depending on spectral clustering. The imputation has been done again with a combination of a user based and an item based predictions. Distinctly, the effort of imputation is repeated until a stable dataset is obtained.

For the sake of preserving individual privacy, predictive performance is sacrificed in PPCF systems based on data obfuscation. Ratings of the users are modified in order the server cannot infer individuals' profiles. However, the modification causes a decrease on the accuracy. For the fact that it is possible to get an improved accuracy in CF via several imputation techniques, they can also be utilized to regain the accuracy loss in

PPCF schemas.

3. CATEGORIES of PPCF USING RANDOMIZED PERTURBATION TECHNIQUE

3.1. Randomized Perturbation

Randomized perturbation technique (RPT) is a successful method used to achieve privacy in central server based PPCF schemas. Instead of sending pure ratings to the server, individuals disguise their ratings in the client side first, then sends their disguised data to the server. Because of getting the ratings as perturbed, the server cannot reach actual ratings which is the crucial point in privacy preserving. While having perturbed ratings, the server is still able to generate recommendations with decent accuracy by an aggregate calculation.

3.1.1 Perturbation phase

The perturbation codes running at the client side are given with the Algorithm 3.1.

Outline of the perturbation process represented by Bilge and Polat [53] is introduced step by step below:

1)The server decides a σ_{max} value which is the maximum standard deviation that clients can use while creating random values.

2)The server decides a β_{max} value which is the maximum percentage of missing ratings to be filled by clients.

3)The server specifies data distribution techniques which clients can prefer to use when creating random numbers.

4)Each user finds mean and standard deviation of their ratings and calculates z-scores.

5)Each user defines σ and β values in a range of [0,σ_{max}] and [0,β_{max}] respectively.

6)Each user selects $\%\beta$ of their unrated items and finds *k* which is the total count of selected and rated items.

7)Each user creates *k* random numbers as their mean is 0 respecting σ value with a distribution selected among which the server provided.

8)Each user adds those random numbers to their rated and selected unrated items (which are assumed as 0) and sends them to the server.

After each user sends their disguised ratings to the server, a disguised user-item matrix is constructed to apply a PPCF algorithm.

Algorithm 3.1 *Data Disguising by RPT*

Symbols used in the algorithm:
Z_u : z-score normalized rating vector of user u
β_{max}: Maximum percentage of missing ratings to be filled
σ_{max}: Maximum standart deviation of random values
Z'_u : Perturbed vector to be sent to the server

```
**rpDisguise**($Z_u, \beta_{max}, \sigma_{max}$)
$tech = randi(2);$                // Uniform:1 & Gaussian:2;
$beta = rand * \beta_{max};$
$sigma = rand * sigma_{max};$
$emptyindices = find(Z_u == 0);$
$ratedindices = setdiff(length(Z_u), emptyindices);$
$emptycount = length(emptyindices);$
$ratedcount = length(Z_u) - emptycount;$
$fcount = round((emptycount * beta)/100);$
$fillingindices = randsample(emptyindices, fcount);$
$pertcount = ratedcount + fcount;$
$pertindices = [ratedindices\ fillingindices];$
$if(tech == 1)\{$
    $alpha = sqrt(3) * sigma;$
    $rands = -alpha + ((2 * alpha) * rand(1, pertcount));\}$
$if(tech == 2)$
    $rands = sigma. * randn(1, pertcount);$
$Z'_u(pertindices) = Z_u(pertindices) + rands;$
$return\ Z'_u;$
```

3.1.2 Prediction phase

3.1.2.1 Memory based approach

In the memory-based approach, similarity weights between users must be assigned first in order to detect similar users referred to as neighbors. Polat and Du [54] proposed to find the similarity weights by multiplication of users' perturbed z-score vectors. Let a and u be the active user and another user respectively. Similarity between *a* and *u* is calculated with Equation 3.1.

$$w_{au} = \sum_{j=1}^{m} z'_{aj} * z'_{uj} \tag{3.1}$$

Note that, *m* is the total number of items. Let us extend Equation 3.1:

$$w_{au} = \sum_{j=1}^{m} (z_{aj} + r_{aj}) * (z_{uj} + r_{uj}) = \sum_{j=1}^{m} (z_{aj} * z_{uj}) + (z_{aj} * r_{uj}) + (r_{aj} * z_{uj}) + (r_{aj} * r_{uj}) \tag{3.2}$$

Since the random numbers are added by a distribution with mean 0, $\sum_{j=1}^{m} (z_{aj} * r_{uj}) \approx 0$, $\sum_{j=1}^{m} (r_{aj} * z_{uj}) \approx 0$ and $\sum_{j=1}^{m} (r_{aj} * r_{uj}) \approx 0$. Consequently, Equation 3.2 can be simplified as:

$$w_{au} = \sum_{j=1}^{m} z'_{aj} * z'_{uj} \approx \sum_{j=1}^{m} z_{aj} * z_{uj} \tag{3.3}$$

Hereby, sum of products with disguised z-scores gives approximately the same result with sum of products with actual z-scores.

$$\sum_{j=1}^{m} z_{aj} * z_{uj} = \sum_{j=1}^{m} \frac{V_{aj} - \mu_a}{\sigma_a} * \frac{V_{uj} - \mu_u}{\sigma_u} = \frac{\sum_{j=1}^{m} (V_{aj} - \mu_a) * (V_{uj} - \mu_u)}{\sigma_a * \sigma_u} \tag{3.4}$$

Equation 3.4 demonstrates that sum of products with z-score values is resulted in the formula of Pearson's Correlation Coefficient [31]. In the equation, V_{aj} denotes vote of active user for *j*'th item. V_{ij} similarly denotes vote of *i*'th user for *j*'th item. μ_a and μ_i represents mean ratings of active user and *i*'th user respectively. Additionally, σ values correspond to standart deviations of related users' ratings.

After the similarity weights are obtained, *k* nearest neighbors are selected among the collection of users. According to weighted average formula, the prediction on a target item q is calculated with Equation 3.5.

$$p'_{aq} = \frac{\sum_{u}^{k} w_{au} * z_{uq}}{w_{au}} \tag{3.5}$$

Since the ratings are z-score normalized, p'_{aq} in the equation must be denormalized to achieve the final prediction value as formulized in Equation 3.6

$$p_{aq} = \mu_a + \sigma_a * p'_{aq} \tag{3.6}$$

Having disguised z-score values, the server is able to calculate p'_{aq}. However, μ_a and σ_a values are only known by active user at the client side. Hence after finding p'_{aq} the server sends the result to active user. Final prediction value is obtained in the client side by denormalizing p'_{aq} as in Equation 3.6.

3.1.2.2 Model-based approach

In the representative Singular Value Decomposition (SVD) based CF schema, user-item matrix is expressed with the multiplication of three matrices namely U, S and V^T. Therefore, denoting the user-item matrix with *A*, there must be created three matrices such that $A = U \times S \times V^T$. With respect to the pedestrals of SVD based CF [6], following steps must be accomplished:

1) $X = A^T \times A$ must be calculated.

2) Eigenvalues and eigenvectors are extracted from *X*.

3) A diagonal matrix *S* is constructed with the *y* largest eigenvalues.

4) The matrix *V* is constructed by corresponding eigenvectors which selected eigenvalues belong to.

5) Finally the matrix *U* is implicated with the formula $U = S^{-1} \times A \times V$

As a consequence of those steps *U*,*S* and *V* matrices are obtained with the sizes of $n \times y$, $y \times y$ and $m \times y$ respectively. When generating a prediction for user *i* on item *j* in a user-item matrix, Equation 3.7 is utilized.

$$p_{ij} = U_i * S * V_j^T \tag{3.7}$$

Assuming that the procedures are done on a z-score normalized user-item vector, denormalization likewise in Equation 3.6 must be applied to reach up the final prediction.

When procedures of SVD is employed on a data disguised by RPT we must still be able to calculate $A^T \times A$ correctly in order to get the right eigenvalues and eigenvectors. For a disguised user-item matrix A', the entries of $A'^T \times A'$ is calculated in Equations 3.8 and 3.9 [55]:

$$(A'^T \times A')_{fg} = \sum_{u=1}^{n} (z_{uf} + r_{uf}) * (z_{ug} + r_{ug}) \tag{3.8}$$

$$(A'^T \times A')_{fg} = \sum_{u=1}^{n} (z_{uf} * z_{ug}) + (z_{uf} * r_{ug}) + (r_{uf} * z_{ug}) + (r_{uf} * r_{ug}) \tag{3.9}$$

In the equation, $\sum_{u=1}^{n} (z_{uf} * r_{ug}) \approx 0$, $\sum_{u=1}^{n} (r_{uf} * z_{ug}) \approx 0$ and $\sum_{u=1}^{n} (r_{uf} * r_{ug}) \approx 0$ where $f \neq g$. Hence, the result is Equation 3.10:

$$(f \neq g) \Rightarrow (A'^{T} \times A')_{fg} = \sum_{u=1}^{n} z_{uf} * z_{ug} \tag{3.10}$$

In the case of $f = g$ the equation can be expressed as follows:

$$(A'^{T} \times A')_{ff} = \sum_{u=1}^{n} (z_{uf} + r_{uf})^2 = \sum_{u=1}^{n} z_{uf}^2 + 2 * \sum_{u=1}^{n} (z_{uf} * r_{uf}) + \sum_{u=1}^{n} r_{uf}^2 \tag{3.11}$$

Since $\sum_{u=1}^{n} (z_{uf} * r_{uf}) \approx 0$, the Equation 3.11 can be simplified in Equation 3.12:

$$(A'^{T} \times A')_{ff} \approx \sum_{u=1}^{n} z_{uf}^2 + \sum_{u=1}^{n} r_{uf}^2 \tag{3.12}$$

While $(A'^{T} \times A')_{ff}$ must end up with $\sum_{u=1}^{n} z_{uf}^2$ there is an additional effect of random numbers as $\sum_{u=1}^{n} r_{uf}^2$. In exchange to get rid of the noise, the equation can be modified as in Equation 3.13:

$$(A'^{T} \times A')_{ff} \approx \sum_{u=1}^{n} z_{uf}^2 + \sum_{u=1}^{n} r_{uf}^2 - (n * \sigma_r^2) \approx \sum_{u=1}^{n} z_{uf}^2 \tag{3.13}$$

In the equation 3.13, σ_r^2 is the standart deviation of random numbers which are added by users to withold the server learning their actual ratings. The server defines maximum standart deviation (σ_{max}) of the random numbers and each client specifies their own σ between σ_{max} and 0. Therefore, the value of σ_r is unknown by the server. Considering an aggregation, the calculation can be finalized with Equation 3.14:

$$(A'^{T} \times A')_{ff} \approx \sum_{u=1}^{n} z_{uf}^2 + \sum_{u=1}^{n} r_{uf}^2 - (n * (\sigma_{max} \div 2)^2) \approx \sum_{u=1}^{n} z_{uf}^2 \tag{3.14}$$

In conclusion, the server is able to approximately infer $A^T \times A$ from A'. Thus, it can compute the p'_{aq} as expressed in Equation 3.7 and the active user gets eventual prediction value via Equation 3.6.

3.1.2.3 Hybrid approach

k-means clustering-based CF as a hybrid CF approach can be adapted to generate predictions without jeopardizing privay. In such approach, function of executing a clustering algorithm is to form a neighborhood. Therefore, CF can challenge scalability problem. In order to form the clusters by *k*-means clustering, distances or similarities between users must be calculated. Since similarity weights can be computed with Pearson's

Correlation privately with aforementioned methods, it can also be utilized in the clustering procedure. Formation of neighborhood via *k*-means clustering can be illustrated with the following steps:

1) Centroids of *k* clusters are initalized by stating *k* users as centers.

2) Similarities between users and those centroids are calculated as expressed with Equations 3.3 and 3.4.

3) Each user is assigned to a cluster based on the similarity with centroids. They are attended to the clusters that they have the highest correlation with the clusters' centroids.

4) Cluster centroids are updated regarding to newcomers into the clusters. For *i*'th cluster's centroid C_i, formulization of the update process is shown in Equation 3.15.

$$C_{iq} = \frac{\sum_{u=1}^{n_i} z'_{uq}}{n_i} \tag{3.15}$$

Here q is referred to as an item which is a dimension of the cluster's centroid. *u* is the user that belong to that cluster. n_i is the total number of users in related cluster. Expansion of the equation in Equation 3.16:

$$C_{iq} = \frac{\sum_{u=1}^{n_i} z'_{uq}}{n_i} + \frac{\sum_{u=1}^{n_i} r_{uq}}{n_i} \approx \frac{\sum_{u=1}^{n_i} z_{uq}}{n_i} \tag{3.16}$$

Hence the centroids of the clusters are found out precisely from disguised user-item matrix.

5) If there is a change in the members of the clusters, go to step 2.

6) Return clusters and their centroids.

Remember that clusters are constructed when the server is offline in order to boost online performance of the CF system. When a new user enters to the system, it is discovered by the system which cluster the user belong by measuring the similarities with the cluster centroids. After its cluster is decided as online, the same procedures with the memory-based CF are followed. *k* nearest neighbors are selected among the users from the related cluster. Then $p_a q'$ is calculated as in Equation 3.5. It is denormalized by the active user with respect to Equation 3.6 and the final prediction is obtained.

4. APPLICATIONS of IMPUTATION TECHNIQUES

4.1. Cluster-Based Smoothing

Cluster-based smoothing (CBS) is an imputation technique which is applied successfully in CF frameworks [50, 56]. It presents a scalable solution handling the sparsity problem. Given a user-item matrix, CBS operates as follows:

1) Users are divided into clusters. *k*-means clustering is a convenient way of clustering the users.

2) For a missing value which is intersection of user *u* and item *q*, users in the same cluster with *u* which have ratings for *q*'th item are chosen.

3) Deviation from mean ratings of the selected users are calculated for item *q* and average of them is taken.

4) The average value is added to the mean of user *u*'s ratings. The result is substituted with the missing value.

The steps above are practiced for all the missing values in the dataset.

4.2. Predictive Mean Matching

Predictive mean matching (PMM) is a multiple imputation technique proposed by Rubin and Schenker in 1986 [58]. It is also utilized in CF techniques [43, 44] and the prediction accuracy is enhanced. According to PMM, missing values in a recipient are imputed with the predictive means retrieved from closest donors. Application of PMM on user-item matrix is represented as follows:

1) Determine some parameters drawn from a multivariate Gaussian distribution by expectation maximization algorithm.

2) Predictions are created for each incomplete rating of the user (recipient) using linear regression with respect to the predefined parameters via the user's available ratings. For all the users who rated the items that recipient rated (donors), the predictions are generated with the same way.

3) Recipients are matched with their closest donors with respect to Mahalanobis distance.

4) Missing values are imputed with the generated predictions from recipient's closest donor.

4.3. Imputation with PPCF

Ensuring privacy for CF systems brings extra costs in tow that number of operations increases together with predictive accuracy may decrease or online performance may get lowered etc. Especially RPT causes a fall of in the quality of recommendations in a CF system. While negative effects of added random numbers are minimalized with aggregate functions, they can't be removed totally. There is also a dilemma between privacy and accuracy metrics for RPT. The server can provide a higher privacy by adjusting σ and β parameters however the accuracy decreases in that extent. On the purpose of boosting the accuracy while providing satisfactory privacy, imputation techniques can be utilized.

In a RPT based PPCF system, the server does not have a permission to access actual ratings of the users. Since the server has disguised ratings, imputation techniques working with a whole dataset can inevitably be operated on perturbed data. Inasmuch as the data is perturbed by RPT, imputation procedures must be performed remarking added randomness.

Investigating how CBS applied for a disguised user-item matrix, the steps of CBS are discussed with respect to a dataset perturbed with RPT. As a clustering is applied in the first step, the users must be able to clustered correctly. *k*-means clustering by Pearson's Correlation Coefficient can be applied successfully on perturbed dataset as aforementioned in Chapter 3. After the users are selected with respect to step 2, average of deviations are calculated with Equation 4.1:

$$\Delta R_{uq} = \frac{\sum_{u=1}^{n_c} \left(z'_{uq} - \mu'_u \right)}{n_c} \tag{4.1}$$

$\mu'_u = \mu_u$ since added random numbers are generated with zero mean.

$$\Delta R_{uq} = \frac{\sum_{u=1}^{n_c} \left(z_{uq} + r_{uq} - \mu_u \right)}{n_c} \approx \frac{\sum_{u=1}^{n_c} \left(z_{uq} - \mu_u \right)}{n_c} \tag{4.2}$$

Accordingly, average of the deviation from mean ratings is calculated correctly. Seeing that mean of a user's perturbed ratings is equal to the mean of undisguised ratings, the average is added directly to the mean value of user *u*'s ratings which gives the imputation value.

In PMM, it is difficult to select closest donors correctly on disguised data that the results are affected by added random numbers. By using wrong donors, consistency of the imputation gets lowered. In order to select the right donors from disguised ratings, the users can be clustered priorly and predictive mean matching is practiced for each cluster separately. Note that the users can be clustered via their disguised ratings as mentioned before.

In the prediction process of PPCF systems that use imputation methods, the ratings of users are subject to a number of particular changes as described in Figure 4.1. With a view to generate prediction via combining imputation techniques with PPCF, the following steps must be put into practice:

1) Each user normalizes their ratings with z-score normalization. Denoting μ_u and σ_u are mean and standard deviation of u's ratings respectively, normalization on item q is signified with Equation 4.3.

$$z_{uq} = \frac{V_{uq} - \mu_u}{\sigma_u} \tag{4.3}$$

2) Each user disguises own ratings with respect to the received parameters σ_{max} and β_{max} from the server.

2) The server collects the modified ratings from users and constructs a perturbed user-item matrix.

3) The server employs an imputation technique on the perturbed user-item matrix and gets an imputed dataset.

4) The server implements one of the memory-based, model-based, or hybrid PPCF algorithms and creates a prediction value p'_{aq} for active user a on target item q.

5) The server sends the resulted p'_{aq} to active user.

6) Active users denormalize p'_{aq} and gets the final prediction value via Equation 3.6.

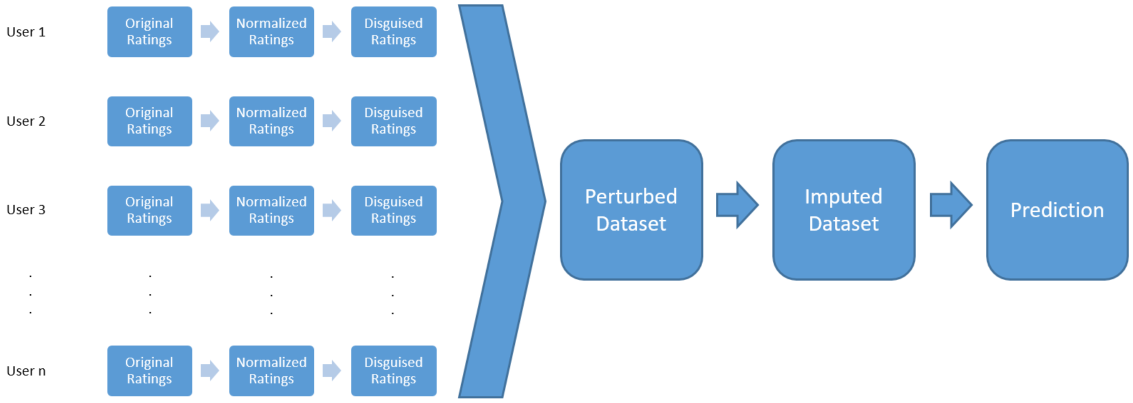

Figure 4.1. *Evolution of ratings towards production of prediction.*

5. EXPERIMENTS

5.1. Experimental Design

It is aimed with the experiments to enhance predictive accuracies of PPCF systems via imputation techniques. In that context, several imputation techniques are performed on dataset perturbed by RPT and tested with representative memory-based, model-based and hybrid PPCF schemas. Clustering-based smoothing (CBS), predictive mean matching (PMM), and imputation with mean (IM) methods are executed within the imputation techniques. For representative memory-based, model-based and hybrid PPCF schemas, user-based, SVD-based, and clustering-based algorithms are selected and implemented respectively.

In CBS method, *k*-means clustering is practiced as a clustering strategy. Experiments are repeated for changing values of *k* as 1, 3, 5, and 10 respectively. When the value of *k* equals to 1, it means that smoothing is done without clustering.

For implementation of PMM, fastpmm function in Multiple Imputations by Chained Equations (MICE) package of R is utilized. Maximum iteration number is fixed to 50 for the imputation which is its default value. PMM method is executed on groups of users separately which are clustered via *k*-means clustering. The values of *k* is varied as 1, 2, 3, 5, and 10. Taking *k* as 1 indicates that PMM is applied on the whole dataset.

IM method is performed based on users. Missing values of each users are straightforwardly filled by mean ratings of corresponding users.

In all the imputation techniques, missing values to be imputed are selected with a certain percentage. For each user, the count of missing values to be imputed is computed with respect to the predefined percentage. Then missing values selected randomly by the count and those are tried to be imputed. The percentage is changed from 20% to 100% incrementing by 20%. For all filling percentages except 100, experiments are repeated 10 times to ensure the randomization.

For disguising by RPT, σ_{max} and β_{max} are defined as 2 and 0 respectively. σ and β are determined separately for each user. Once a perturbed dataset is constructed, the same dataset is used in every experiment.

User-based PPCF algorithm uses KNN methodology for neighbor selection. That *k* value is specified as 50 for all user-based experiments. SVD-based PPCF method needs a y value indicating count of the selected eigenvalues. Value of y is fixed to 10. In clustering-based PPCF, users are divided into 2, 3, 5 and 10 clusters respectively via *k*-means clustering and the experiments are held for each condition. 50-nearest neighbors are utilized for prediction algorithm (the same as user-based PPCF).

The experiments are realized on Movielens 100k dataset [59] which has 100.000 ratings on movies. There are 943 users and 1682 movies. Rating scale of the system is integer values between 1-5. By means of hold-out method, 100 users are selected as test users and remaining 843 are treated as train users. Predictions are generated for all available ratings by treating them as missing one by one.

5.2. Evaluation Criteria

Performance of the systems are evaluated with mean absolute error (MAE) which is one of the most utilized error metrics used in CF. In terms of CF, the formula is predicated as follows:

$$MAE = \frac{\sum_{i=1}^{n} |r_i - p_i|}{n} \tag{5.1}$$

In the equation, r_i and p_i represents actual rating and prediction values respectively. *n* is the total number of the generated predictions. Note that, lower MAE means higher accuracy in terms of CF.

Additionally, *t*-test results are demonstrated within the experimental results. *t*-test is a statistical methods comparing two samples based on their means and reveals rate of association between them. In the experiments, aim of employing *t*-test is to measure whether or not there is a significant change in the accuracy of predictions when applied an imputation. Samples used in *t*-test are constructed from MAE values of each user. Since there are 100 test users, degree of freedom is defined as 99. There is expected a significant change when compared to the results without utilizing an imputation method. There is also obtained results without privacy with the experiments. When compared to those results, the change in the predictive accuracy must be insignificant to prove that the accuracy loss by added random numbers is retrieved. For the results of *t*-tests, 0.05 is defined as threshold that *p* values less than 0.05 implies that there is a significant change.

5.3. Experimental Results

In user-based algorithm, MAE results for with and without privacy are measured as 0.8642 and 0.7699 respectively. When imputation methods are utilized, accuracies are changed as in Figure 5.1. Percentages of improvement on user-based PPCF reaches to 7.48% 8.26% and 4.93% with IM, CBS, and PMM respectively. Best results are yielded when cluster count is 1 in CBS imputation. Besides, PMM brings a higher accuracy when applied separately on more clusters. Experimental results show that trying to fill all missing values is a correct choice for the user-based PPCF schema.

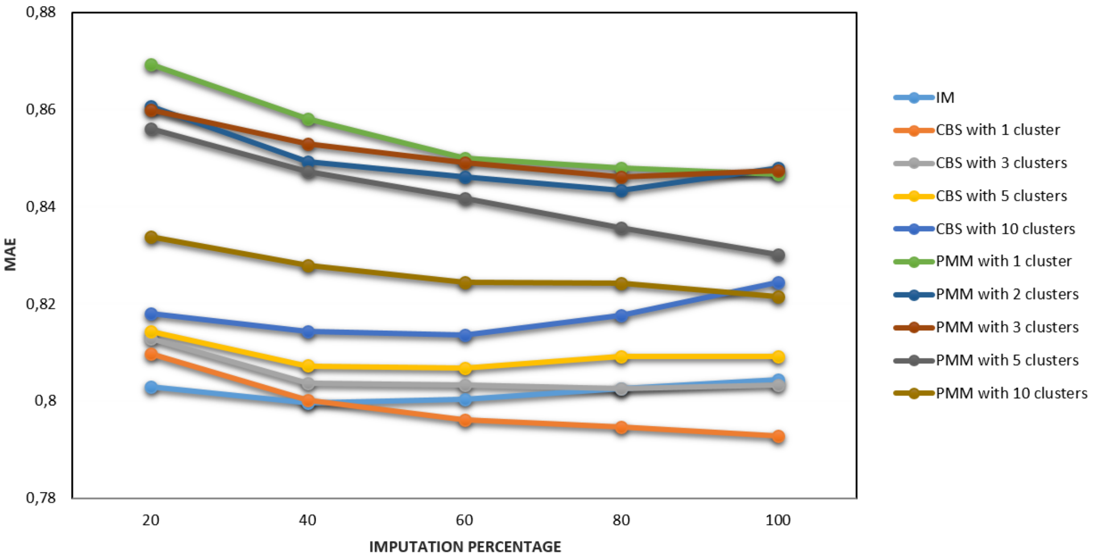

Figure 5.1. *Comparison of MAE results for imputed user-based PPCF by varying imputation percentage*

Results of *t*-tests on user-based algorithms are demonstrated in table 5.1. IM and CBS introduce a significant improvement on user-based PPCF schema. When compared with the results obtained from user-based CF (without privacy) increase in MAE is insignificant.

Table 5.1. *t-test results on user-based algorithm with corresponding p values.*

Imputation Method	Significance Rate of Improvement	Significance Rate of Loss
IM	0,0102	0,1289
CBS	0,0092	0,1976
PMM	0,0876	0,0192

The results of MAE on SVD-based CF and SVD-based PPCF are 0.7952 and 0.8199 respectively. Effects of the imputation techniques on accuracy are demonstrated with figure 5.2. The results show that it is not appropriate to use imputation with CBS and PMM methods owing to the fact that they deteriorate the accuracy. However, IM brings a slight improvement for the accuracy by 0.2927%.

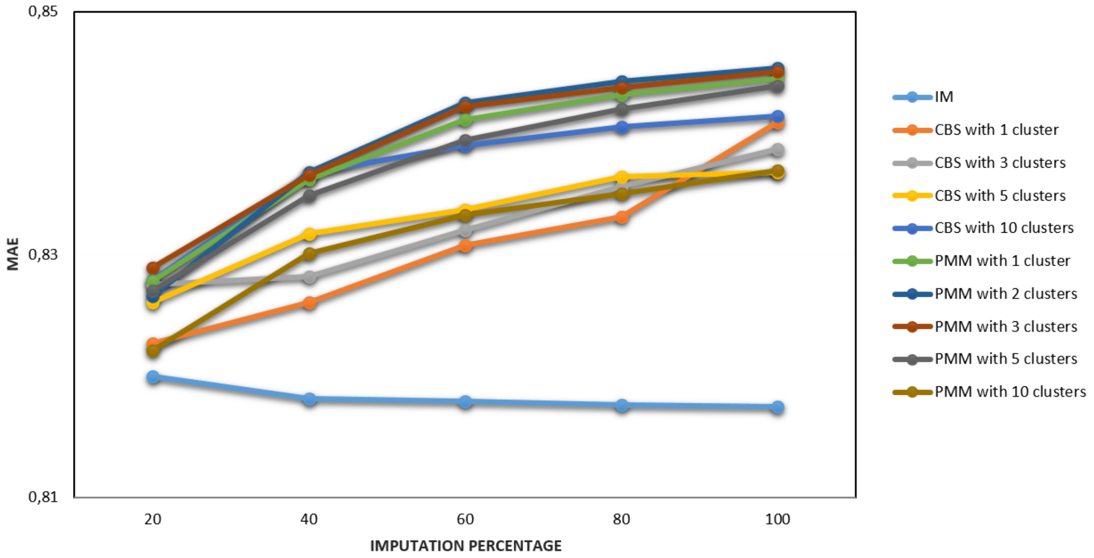

Figure 5.2. *Comparison of MAE results for imputed SVD-based PPCF by varying imputation percentage*

Within the frame of clustering-based algorithms the results are searched seperately by varying cluster counts. When 2-cluster based algorithm is employed, MAE results become 0.7741 and 0.8803 for without and with privacy. Effects of the imputation techniques on accuracy are demonstrated for 2 clusters with figure 5.3. Improvement percentages of accuracy reaches up to 8.24%, 9.10%, and 5.99% for IM, CBS and PMM respectively in terms of MAE. CBS with 1 cluster and 80% filling percentage yields the best result with value of MAE as 0.8002. *t*-test results are referred with table 5.2.

Table 5.2. *t-test results on clustering-based algorithm as number of clusters is 2 with corresponding p values.*

Imputation Method	Significance Rate of Improvement	Significance Rate of Loss
IM	0,0055	0,0849
CBS	0,0064	0,1597
PMM	0,0356	0,0151

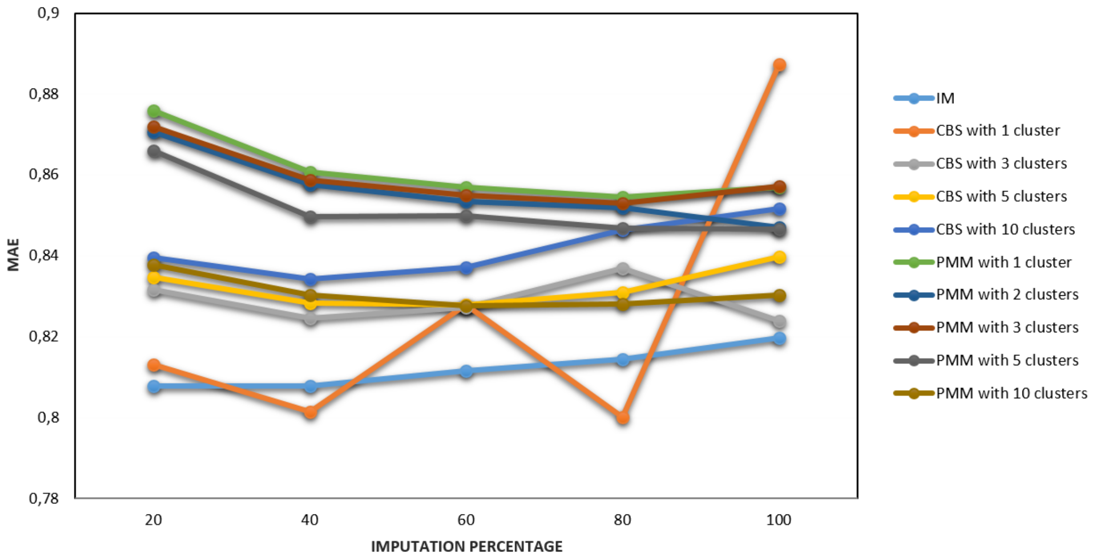

Figure 5.3. *Comparison of MAE results for imputed clustering-based PPCF with 2 clusters by varying imputation percentage*

For 3-cluster based algorithm the resulting MAEs are 0.7824 and 0.9033 for without and with privacy. The effects of the imputation techniques on accuracy are shown in figure 5.4. Improvement percentages of accuracy reaches up to 11.24%, 10.85%, and 7.88% for IM, CBS and PMM respectively in terms of MAE. For 3-cluster-based PPCF algorithm, IM with 60% percentage of filling yields the best result with 0.8018 MAE. *t*-test results are referred with table 5.3.

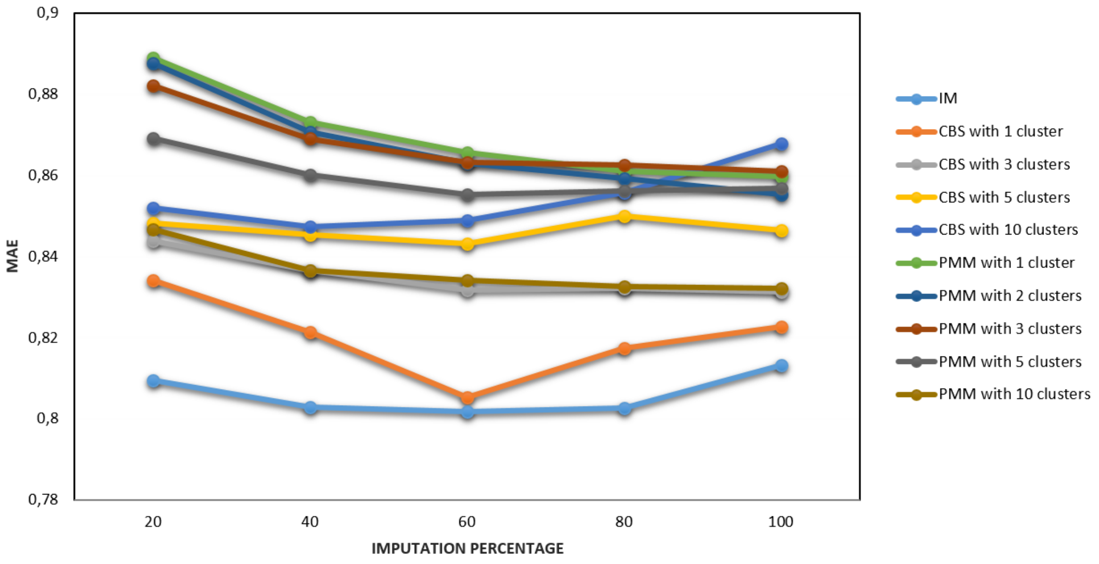

Figure 5.4. *Comparison of MAE results for imputed clustering-based PPCF with 3 clusters by varying imputation percentage*

Table 5.3. *t-test results on clustering-based algorithm as number of clusters is 3 with corresponding p values.*

Imputation Method	Significance Rate of Improvement	Significance Rate of Loss
IM	0,0006	0,1036
CBS	0,0192	0,1071
PMM	0,0132	0,0102

In 5-cluster based algorithm MAE values are evaluated as 0.7921 and 0.9099 for without and with privacy. The effects of the imputation techniques on accuracy are shown in figure 5.5. Improvement percentages of accuracy reaches up to 10.62%, 9.63%, and 7.66% for IM, CBS and PMM respectively in terms of MAE. For 5-cluster-based PPCF algorithm, IM with 20% percentage of filling yields the best result with 0.8133 MAE. *t*-test results are referred with table 5.4.

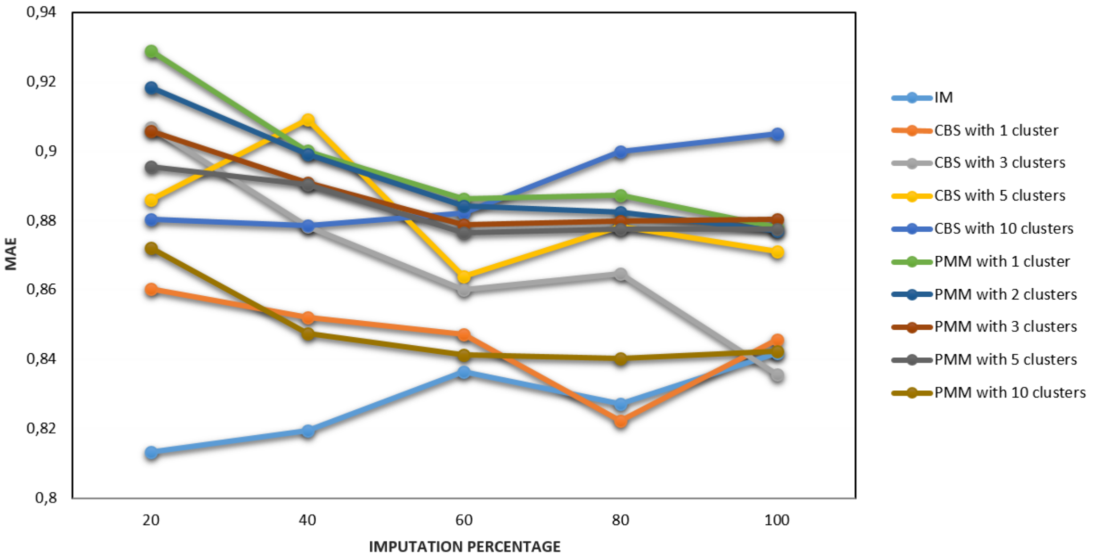

Figure 5.5. *Comparison of MAE results for imputed clustering-based PPCF with 5 clusters by varying imputation percentage*

Table 5.4. *t-test results on clustering-based algorithm as number of clusters is 5 with corresponding p values.*

Imputation Method	Significance Rate of Improvement	Significance Rate of Loss
IM	0,0008	0,2013
CBS	0,0591	0,0831
PMM	0,0060	0,0540

Regarding 10-cluster based algorithm, MAE results are 0.8440 and 1.0705 for without and with privacy. The effects of the imputation methods on accuracy are represented in figure 5.6. Improvement percentages of accuracy reaches up to 22.44%, 18.81%, and 18.98% for IM, CBS and PMM respectively in terms of MAE. For 10-cluster-based PPCF algorithm, the best result is obtained as 0.8303 in terms of MAE by IM with 100% percentage of filling. Related *t*-test results are referred with table 5.5.

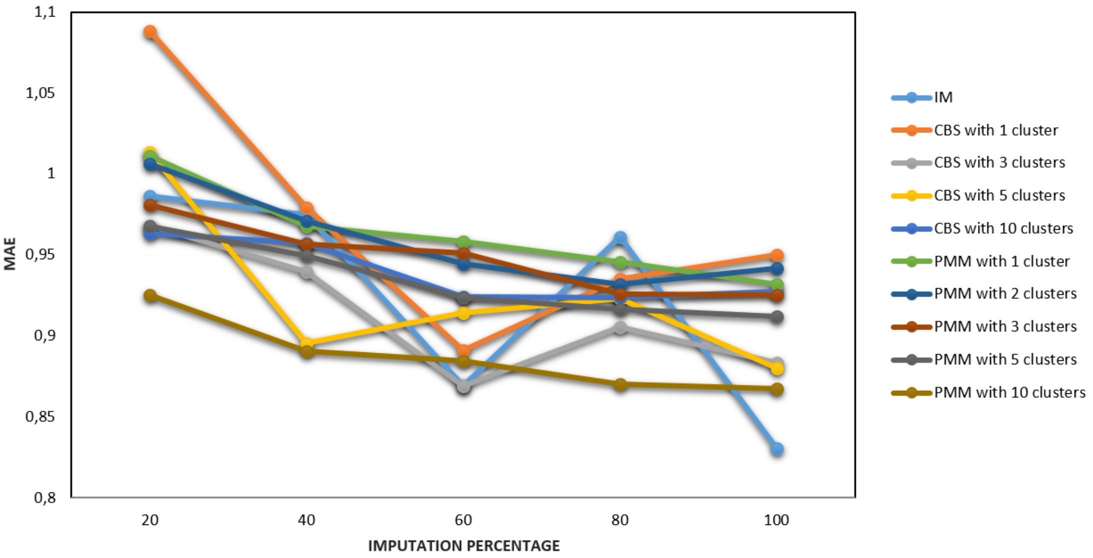

Figure 5.6. *Comparison of MAE results for imputed clustering-based PPCF with 10 clusters by varying imputation percentage*

Table 5.5. *t-test results on clustering-based algorithm as number of clusters is 10 with corresponding p values.*

Imputation Method	Significance Rate of Improvement	Significance Rate of Loss
IM	0,0000	0,8007
CBS	0,0006	0,4863
PMM	0,0001	0,6371

In PPCF schemas, the best result is obtained in user-based algorithm by utilizing CBS with one cluster and filling percentage as 100% in the overall. In order to see success of CBS on user-based approach by varying privacy levels, several experiments are held on various β_{max} and σ_{max} parameters. Results of the experiments are introduced in figures 5.7 and 5.8. Firstly, σ_{max} is fixed to 2 and β_{max} is sampled from 0 to 100 with increasing by 10. The value of MAE minimalized for β_{max} = 10 with 0.7876 when CBS is utilized. Secondly, β_{max} is fixed as 10 and σ_{max} is varied as 0.5, 1, 2, and 4. While user-based PPCF without applying CBS yields MAE results from 0.7630 to 1.1091, when CBS utilized, the MAE results are increased from 0.7645 to 0.8352.

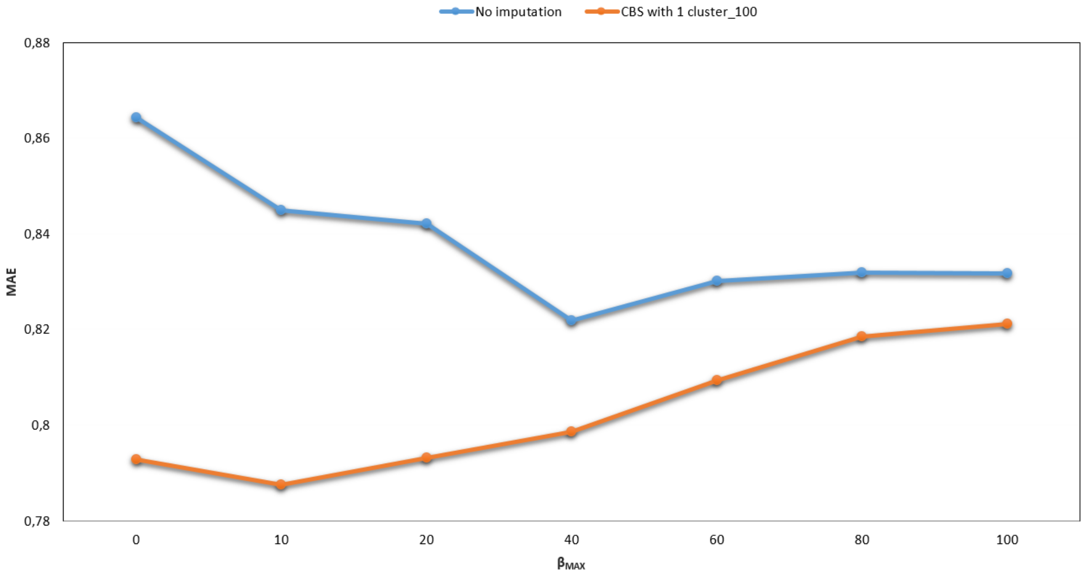

Figure 5.7. *MAE results on user-based PPCF by varying β_{max} value comparing CBS imputation with no imputation. Cluster count and filling percentage are adjusted as 1 and 100 respectively for CBS.*

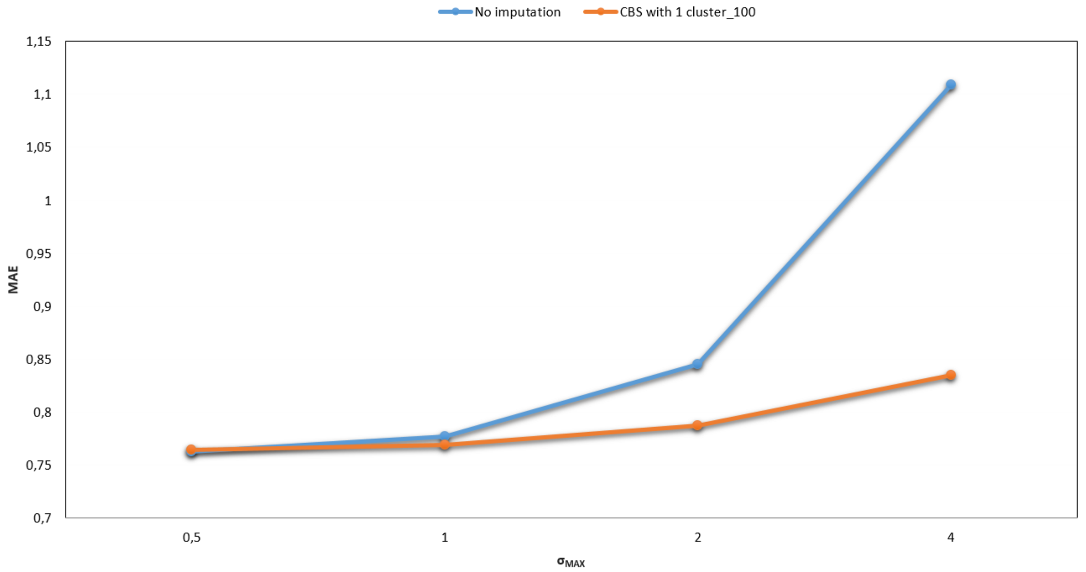

Figure 5.8. *MAE results on user-based PPCF by varying σ_{max} value comparing CBS imputation with no imputation. Cluster count and filling percentage are adjusted as 1 and 100 respectively for CBS.*

6. CONCLUSIONS

The effort of the thesis is to improve existing privacy-preserving collaborative filtering schemas in terms of accuracy by utilizing several imputation techniques. Novelly, cluster-based smoothing and predictive mean matching algorithms are adapted to privacy-preserving collaborative filtering systems in this context. User-based, singular value decomposition based and clustering-based privacy-preserving collaborative filtering algorithms are selected as representative memory-based, model-based and hybrid privacy-preserving collaborative filtering techniques. Effects of imputation techniques on those privacy-preserving collaborative filtering schemas are investigated separately. All of the experiments held on a movie recommendation dataset by dividing it into train and test sets via holdout method.

For user-based privacy-preserving collaborative filtering schema, all the imputation methods provide an enhancement of the accuracy. Cluster-based smoothing yields the best results compared to imputation by mean and predictive mean matching. Imputing all missing ratings via cluster-based smoothing is selected as the most successful way of handling missing values in terms of accuracy. Cluster-based smoothing and imputation with mean provide a significant improvement in the accuracy. Moreover, with the help of those methods accuracy loss due to privacy can be reduced to an insignificant level with respect to *t*-test.

For singular value decomposition based privacy-preserving collaborative filtering schema, the accuracy of the system is slightly improved via imputation with mean approach. The results demonstrate that the more missing values are filled, the higher accuracy obtained by utilizing imputation with mean. Cluster-based smoothing and predictive mean matching methods can be considered as they are not appropriate for singular value decomposition based privacy-preserving collaborative filtering.

For clustering-based privacy-preserving collaborative filtering schema, experimental results revealed that all kinds of applied imputation techniques are able to ensure a significant improvement on the accuracy. There is not a certain percentage of missing values to fill in order to get the most accurate results that the percentages of the best results obtained by imputation techniques vary.

Minimum error on the privacy-preserving collaborative filtering schemas is achieved on user-based privacy-preserving collaborative filtering by 1 cluster-based smoothing strategy. In order to measure the effort of such strategy, several experiments are held on different privacy parameters. When the users are permitted to fill most of their unrated items randomly, count of missing values to be imputed diminishes. Hence, accuracy improvement effect of the imputation method decreases. Designating privacy levels, standard deviation of the added random numbers is tested, as well. It is observed with the results that there is a critical fall in the accuracy at increasing levels of privacy on user-based privacy-preserving collaborative filtering. However, the accuracy can be kept in decent values even with the higher levels of privacy via imputation techniques.

To sum up, it is revealed with the study that imputation methods can be utilized in privacy-preserving collaborative filtering schemas in order to boost predictive accuracy. Accordingly, individuals may prefer either to receive more consistent recommendations with the same privacy level or to improve their privacy with a stable recommendation quality.

Due to negative effects of privacy-preserving collaborative filtering on the accuracy, studies to enhance privacy-preserving collaborative filtering schemas tend to expand in the future. Various imputation methods must be investigated to find out their applicability on privacy-preserving collaborative filtering systems within this scope. In model-based privacy-preserving collaborative filtering schemas, numerous techniques are utilized to construct the model. Such techniques are based on various data mining strategies having different characteristics. Hence, designating model-specific imputation technique is expected to be studied. Imputing certain missing values is a widespread approach to improve the effectiveness of imputation methods. Accordingly, selecting the right missing values on perturbed data to impute is an open issue.

REFERENCES

[1] Melville P., Sindhwani V. (2011). Recommender systems. *Encyclopedia of Machine Learning*. Springer US. pp. 829-838.

[2] Vozalis, E., Margaritis, K. G. (2003). Analysis of recommender systems' algorithms *The 6th Hellenic European Conference on Computer Mathematics & its Applications*. pp. 732-745.

[3] Jain, S., Grover, A., Thakur, P. S., & Choudhary, S. K. (2015). Trends, problems and solutions of recommender system. *Computing, Communication & Automation (ICCCA), 2015 International Conference on IEEE* pp. 955-958.

[4] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM 35(12),* pp. 61-70.

[5] Pan, W., Xiang, E. W., Liu, N. N., & Yang, Q. (2010). Transfer Learning in Collaborative Filtering for Sparsity Reduction. *AAAI, 10*, pp. 230-235.

[6] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study. *Minnesota Univ Minneapolis Dept of Computer Science.*

[7] Su, X., Khoshgoftaar, T. M., & Greiner, R. (2008). Imputed neighborhood based collaborative filtering. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01 IEEE Computer Society,* pp. 633-639.

[8] Chujai, P., Rasmequan, S., Suksawatchon, U., & Suksawatchon, J. (2014). Imputing missing values in Collaborative Filtering using pattern frequent itemsets. *Electrical Engineering Congress (iEECON), 2014 International IEEE,* pp. 1-4.

[9] Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications, 92,* pp. 507-520.

[10] Patil, N. D., & Bhosale, D. S. (2017). Providing highly accurate service recommendation for semantic clustering over big data. *International Research Journal of Engineering and Technology 4(2),* pp. 1889-1892.

[11] Gunawardana, A., & Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research, 10(Dec),* pp. 2935-2962.

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

[13] Nadimi-Shahraki, M. H., & Bahadorpour, M. (2014). Cold-start Problem in Collaborative Recommender Systems: Efficient Methods Based on Ask-to-rate Technique. *CIT. Journal of Computing and Information Technology, 22(2),* pp. 105-113.

[14] Sahebi, S., & Cohen, W. W. (1997). Community-based recommendations: a solution to the cold start problem. *Proceedings of WOODSTOCK'97, 22(2),* pp. 40-44.

[15] Singh, S., & Aswal, M. S. (2016). Towards a framework for web page recommendation system based on semantic web usage mining: A case study. *Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on IEEE,* pp. 329-334.

[16] Gunes, I., Kaleli, C., Bilge, A., & Polat, H. (2014). Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review* pp. 1-33.

[17] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM, 40(3),* pp. 66-72.

[18] Cranor, L. F. (2004). I didn't buy it for myself. *Designing personalized user experiences in eCommerce,* pp. 57-73.

[19] Canny, J. (2002). Collaborative filtering with privacy. *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on IEEE,* pp. 45-57.

[20] Renckes, S., Polat, H., & Oysal, Y. (2012). A new hybrid recommendation algorithm with privacy. *Expert Systems, 29(1),* pp. 39-55.

[21] Polat, H., & Du, W. (2005). Privacy-preserving collaborative filtering. *International journal of electronic commerce, 9(4),* pp. 9-35.

[22] Parameswaran, R., & Blough, D. (2005). A robust data obfuscation approach for privacy preservation of clustered data. *Workshop on Privacy and Security Aspects of Data Mining,* pp. 18-25.

[23] Parameswaran, R., & Blough, D. M. (2007). Privacy preserving collaborative filtering using data obfuscation. *In Granular Computing, 2007. GRC 2007. IEEE International Conference on IEEE,* pp. 380-386.

[24] Polat, H., & Du, W. (2006). Achieving private recommendations using randomized response techniques. *Advances in Knowledge Discovery and Data Mining,* pp. 637-646.

[25] Chen, X., & Huang, V. (2012). Privacy preserving data publishing for recommender system. *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual* pp. 128-133.

[26] Canny, J. (2002). Collaborative Filtering with Privacy via Factor Analysis. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 238-245.

[27] Borking, J. J., Van Eck, B. M. A., Siepel, P., & Verhaar, P. J. A. (1999). *Intelligent software agents and privacy,* The Hague: Registratiekamer.

[28] Fasli, M. (2007). On agent technology for e-commerce: trust, security and legal issues. *The Knowledge Engineering Review, 22(1),* pp. 3-35.

[29] Such, J. M., Espinosa, A., & García-Fornes, A. (2014). A survey of privacy in multiagent systems. *The Knowledge Engineering Review, 29(3),* pp. 314-344.

[30] Aimeur, E., Brassard, G., Fernandez, J. M., Onana, F. S. M., & Rakowski, Z. (2008). Experimental demonstration of a hybrid privacy-preserving recommender system. *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on IEEE,* pp. 161-170.

[31] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd annual*

international ACM SIGIR conference on Research and development in information retrieval, pp. 230-237.

[32] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295.

[33] Candillier, L., Meyer, F., & Fessant, F. (2008). Designing specific weighted similarity measures to improve collaborative filtering systems. *Industrial Conference on Data Mining, Springer Berlin Heidelberg.*, pp. 242-255.

[34] Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval, 5(4)*, pp. 287-310.

[35] Mulla, N., & Girase, S. (2012). A new approach to requirement elicitation based on stakeholder recommendation and collaborative filtering. *International Journal of Software Engineering & Applications, 3(3)*, pp. 51-60.

[36] Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems, 23(6)*, pp. 520-528.

[37] Lemire, D., & Maclachlan, A. (2005, April). Slope one predictors for online rating-based collaborative filtering. *Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics.*, pp. 471-475.

[38] Lin, H., Yang, X., Wang, W., & Luo, J. (2014). A Performance Weighted Collaborative Filtering algorithm for personalized radiology education. *Journal of biomedical informatics, 51*, pp. 107-113.

[39] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review, 13(5-6),* pp. 393-408

[40] Hwang, W. S., Li, S., Kim, S. W., & Lee, K. (2014, April). Data imputation using a trust network for recommendation. *Proceedings of the 23rd International Conference on World Wide Web, ACM* pp. 299-300

[41] Xia, W., He, L., Gu, J., & He, K. (2009, August). Effective collaborative filtering approaches based on missing data imputation. *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on IEEE,* pp. 534-537.

[42] Pan, W., Liu, N. N., Xiang, E. W., & Yang, Q. (2011). Transfer learning to predict missing ratings via heterogeneous user feedbacks. *IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 22(3),* pp. 2318-2323

[43] Su, X., Khoshgoftaar, T. M., Zhu, X., & Greiner, R. (2008). Imputation-boosted collaborative filtering using machine learning classifiers. *Proceedings of the 2008 ACM symposium on Applied computing* pp. 949-950.

[44] Su, X., Khoshgoftaar, T. M., & Greiner, R. (2008, May). A Mixture Imputation-Boosted Collaborative Filter. *the 21st Conference of Florida Artificial Intelligence Research Society (FLAIRS'08),* pp. 312-316.

[45] Ren, Y., Li, G., Zhang, J., & Zhou, W. (2013). Lazy collaborative filtering for data sets with missing values. *IEEE transactions on cybernetics, 43(6),* pp. 1822-1834.

[46] Ranjbar, M., Moradi, P., Azami, M., & Jalili, M. (2015). An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems. *Engineering Applications of Artificial Intelligence, 46,* pp. 58-66.

[47] Huang, C. B., & Gong, S. J. (2008). Employing rough set theory to alleviate the sparsity issue in recommender system. *Machine Learning and Cybernetics, 2008 International Conference on IEEE, 3,* pp. 1610-1614.

[48] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. *JSW, 5(7),* pp. 745-752.

[49] Zhang, S., Li, C., Ma, L., & Li, Q. (2013). Alleviating the sparsity problem of collaborative filtering using rough set. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering, 32(2),* pp. 516-530.

[50] Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., & Chen, Z. (2005). Scalable collaborative filtering using cluster-based smoothing. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 114-121

[51] Ma, H., King, I., & Lyu, M. R. (2007, July). Effective missing data prediction for collaborative filtering. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 39-46.

[52] Abdelwahab, A., Sekiya, H., Matsuba, I., Horiuchi, Y., & Kuroiwa, S. (2009, December). Collaborative filtering based on an iterative prediction method to alleviate the sparsity problem. *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services,* pp. 375-379.

[53] Bilge, A., & Polat, H. (2012). An improved privacy-preserving DWT-based collaborative filtering scheme. *Expert Systems with Applications, 39(3),* pp. 3841-3854.

[54] Polat, H., & Du, W. (2003). Privacy-preserving collaborative filtering using randomized perturbation techniques. *Data Mining, 2003. ICDM 2003. Third IEEE International Conference* pp. 625-628.

[55] Polat, H., & Du, W. (2005, March). SVD-based collaborative filtering with privacy. *Proceedings of the 2005 ACM symposium on Applied computing* pp. 791-795.

[56] Rahmawati, A., Wibowo, A. T., & Wulandari, G. S. (2015). Cluster-Smoothed with Random Neighbor Selection for Collaborative Filtering. *Computer, Control, Informatics and its Applications (IC3INA), 2015 International Conference on IEEE* pp. 154-158.

[57] Rubin, D. B. (1987). Multiple Imputation for Nonresponse in Surveys. *Wiley Series in Probability and Statistics.*

[58] Rubin, D. B., & Schenker, N. (1986). Multiple imputation for interval estimation from simple random samples with ignorable nonresponse. *Journal of the American Statistical Association, 81,* pp. 366-374.

[59] MovieLens 100K Dataset, `https://grouplens.org/datasets/movielens/100k/` Accessed: January, 2018

CURRICULUM VITAE

Name-Surname	: Mehmet ÖZCAN
Birth Place and Year	: Osmangazi/BURSA, 1990
E-mail	: mehmet_ozcan@anadolu.edu.tr

Education

- Bachelor's Degree in Computer Engineering (English)
  Anadolu University, Eskişehir, Turkey 3.50/4.00 G.P.A June 2014
- Sukru Senkaya Anatolian High School
  Concentrationin English and Germany June 2008

Career

- Research Assistant, Anadolu University, Eskişehir, Turkey 2015 - ongoing

Papers Submitted to International Meetings

- Ozcan, M., Goz, F. (2017) ”An Educational Mobile City Learning Application for Kids.”, *The Eurasia Proceedings of Educational Social Sciences, 7,* pp. 24-29.
- Goz, F., Ozcan, M. (2017) ”An Entertaining Mobile Vocabulary Learning Application.”, *The Eurasia Proceedings of Educational Social Sciences, 7,* pp. 63-66.
- Ozcan, M., Temel, T. (2016) ”New Recommendation System Using Naive Bayes for E-Learning.”, *International Conference on Education and Science*, pp. 730-732.

International Book Chapters

- Ozcan, M., Goz, F., Temel, T. (2016) ”New Recommender System Using Naive Bayes for E-Learning.”, W., Wu, S., Alan & M., T., Hebeci (Eds.), *Research Highlights in Education and Science,* pp. 62-68.