8-2021

Cosine-based explainable matrix factorization for collaborative filtering recommendation.

Pegah Sagheb Haghighi
University of Louisville

COSINE-BASED EXPLAINABLE MATRIX FACTORIZATION FOR COLLABORATIVE FILTERING RECOMMENDATION

By

Pegah Sagheb Haghighi
M.Eng., Information Technology,
University of Guilan, Guilan, Iran

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Science and Engineering
University of Louisville
Louisville, Kentucky

August 2021

COSINE-BASED EXPLAINABLE MATRIX FACTORIZATION FOR COLLABORATIVE FILTERING RECOMMENDATION

By

Pegah Sagheb Haghighi
M.Eng., Information Technology,
University of Guilan, Guilan, Iran

A Dissertation Approved On

August 4, 2021

by the following Dissertation Committee:

Dr. Olfa Nasraoui, Dissertation Director

Dr. Hichem Frigui

Dr. Nihat Altiparmak

Dr. Antonio Badia

Dr. Cara Cashon

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dr. Olfa Nasraoui for her invaluable support, expertise and patience. Your guidance helped me see new opportunities when things would get a bit discouraging.

My dissertation committee for their support and guidance.

The University of Louisville and the Department of Computer Science and Engineering for awarding me the dissertation completion scholarship and providing me with the financial support.

My lab mates for always being there for me and offering me advice.

My amazing friends Fadoua Khmaissia and Wiem Safta for always supporting me and making me feel special. Your presence was very important in this process.

My parents for their love and encouragement.

Finally, I would like to thank my fiancé, William Rigby, for keeping things going and having a sympathetic ear when I needed.

ABSTRACT

COSINE-BASED EXPLAINABLE MATRIX FACTORIZATION FOR COLLABORATIVE FILTERING RECOMMENDATION

Pegah Sagheb Haghighi

August 4, 2021

Recent years saw an explosive growth in the amount of digital information and the number of users who interact with this information through various platforms, ranging from web services to mobile applications and smart devices. This increase in information and users has naturally led to information overload which inherently limits the capacity of users to discover and find their needs among the staggering array of options available at any given time, the majority of which they may never become aware of. Online services have handled this information overload by using algorithmic filtering tools that can suggest relevant and personalized information to users. These filtering methods, known as Recommender Systems (RS), have become essential to recommend a range of relevant options in diverse domains ranging from friends, courses, music, and restaurants, to movies, books, and travel recommendations. Most research on recommender systems has focused on developing and evaluating models that can make predictions efficiently and accurately, without taking into account other desiderata such as fairness and transparency which are becoming increasingly important to establish trust with human users. For this reason, researchers have been recently pressed to develop recommendation systems that are endowed with the increased ability to explain why a recommendation is given, and hence help users make more informed decisions. Nowadays, state of the art Machine Learning (ML) techniques are being

used to achieve unprecedented levels of accuracy in recommender systems. Unfortunately, most models are notorious for being black box models that cannot explain their output predictions. One such example is Matrix Factorization, a technique that is widely used in Collaborative Filtering algorithms. Unfortunately, like all black box machine learning models, MF is unable to explain its outputs.

This dissertation proposes a new Cosine-based explainable Matrix Factorization model (CEMF) that incorporates a user-neighborhood explanation matrix (NSE) and incorporates a cosine based penalty in the objective function to encourage predictions that are explainable. Our evaluation experiments demonstrate that CEMF can recommend items that are more explainable and diverse compared to its competitive baselines, and that it further achieves this superior performance without sacrificing the accuracy of its predictions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The use of information filtering tools that discover or suggest relevant and personalized items has become essential to avoid information overload. For instance, recommender systems assist users by providing them with personalized suggestions. They recommend items in a variety of domains (music, movies, books, travel recommendation, etc).

While black-box Machine Learning (ML) models are increasingly being applied to make predictions in the field of recommendations, the demand for explainability has been increasing. The most accurate recommendation models tend to be black boxes that cannot justify why items are recommended to a user. The ability of a recommender system to explain the reasoning of its recommendation can serve as a bridge between humans and recommender systems. Explanations can serve different purposes such as building trust, improving transparency and user satisfaction and helping users make informed decisions.

Black-box models which include Latent factor models have been the state of the art in Collaborative Filtering recommender systems. For instance, Matrix Factorization (MF) [2] which has proved to be highly accurate in various domains in particular recommendation systems learns hidden interactions between entities to predict possible user ratings for items.

Unfortunately, a common challenge with black-box methods is the difficulty of explaining their predictions to users using the latent dimensions. To solve this problem, some explainable recommendation algorithms have recently been proposed. For instance, Explicit factor models (EFM) [3] generate explanations based on the explicit features extracted from users' reviews. Explainable Matrix Factorization (EMF) [4,5] is another algorithm that uses an explainability regularizer or soft constraint in the objective function of classical matrix factorization. The constraining term tries to bring the user and explainable item's latent

factor vectors closer, thus favoring the appearance of explainable items at the top of the recommendation list. However, the soft constraint has the disadvantage of using the Euclidean distance which is known to suffer from the curse of dimensionality, a common issue with rating data that tends to be high dimensional and sparse.

To overcome the limitations of using the Euclidean distance, we propose a new objective function for the Matrix Factorization model used in Collaborative Filtering that computes top-n recommendations that are both accurate and explainable. We use a cosine based distance as an explainability regularizer or a soft constraint in the objective function, and call the new model Cosine-based Explainable Matrix Factorization (CEMF). Like MF and EMF, this model does not require using any additional external data modalities such as content, to generate explanations.

1.1 Problem Statement

Matrix Factorization (MF) is one of the most widely used state of the art techniques in Machine learning, where it is mainly used for representation learning. MF has also been successfully used in Collaborative Filtering (CF) recommender systems to predict missing ratings. Unfortunately, like all black box machine learning models, MF is unable to explain its outputs. Hence, while predictions from a Matrix Factorization-based recommender system tend to be highly accurate, it is generally impossible for the user to understand the reasons behind a recommendation.

Our aim is to design an explainable recommendation system whose predictions can be explained using only the data that was used as input to learn the model, and no additional data, hence ensuring transparency. In addition, we aim for a method that can incorporate explainability scores from the Neighbor Style Explanation (NSE) because this style has been previously validated when used to explain the outputs of CF recommendation systems [6], [7], [5] and [8]. Thus, it is out of the scope of this study to demonstrate that this explanation style is effective to users in promoting explanations. Instead, what we aim to achieve is a model for recommendations that is able to generate explainable recommendations without

sacrificing its accuracy

Finally, by explainable, we mean that the top recommendations should have higher Explainability scores. The latter can be measured using metrics such as Mean Explainability Precision (MEP) [5].

1.2 Research Contributions

1. We propose a new Cosine-based explainable Matrix Factorization model (CEMF) that incorporates a user-neighborhood explanation matrix (NSE) [4, 7] along with the cosine distance in the objective function to assist in explaining its predictions. This model overcomes the drawback of the Euclidean distance metric in higher dimensions. We chose to use the NSE explainability scores because (1) they rely on the same data (namely rating data and not any external data) that is used as input to learn the recommendation model, thus ensuring the transparency of explanations, and (2) they have been previously validated in user studies that found the explanation scores to be correlated with user satisfaction [4].

2. We assess the accuracy and explainability of our method based on several evaluation metrics such as the RMSE, MAP, nDCG and MEP.

3. We define a new explainability metric that considers the list of explainable items in the top-n recommendation as well as the true ranks of the items according to the scores for each user in the test set. We call this metric MEP-TR, which is an acronym for Mean Explainability Precision with true rating.

4. The proposed model can recommend more diverse items in its top-n recommendations because it tends to be less popularity biased.

1.3 Research Questions

The research questions that we attempt to answer, in relation to the contributions above, can be stated as:

1. Do the CEMF recommendations have higher explainability than the baseline methods?
2. Is the CEMF model more accurate than the baseline methods?
3. Do the CEMF recommendations have less popularity bias than the baseline methods?
4. Do CEMF recommendations recommend "simultaneously" more accurate and explainable items in its top-n list?

1.4 Document Organization

The rest of this document is organized as follows. Chapter 2 reviews the related work and background on recommender systems and explanations. Chapter 3 presents our proposed Cosine-based Explainable Matrix Factorization model for Collaborative Filtering. Chapter 4 presents the experimental results. Finally, Chapter 5 concludes the dissertation.

CHAPTER 2

LITERATURE REVIEW

Over the past few decades, there has been an increase in the amount of information and number of users on the Internet. Online services such as Facebook, Netflix, Spotify, LinkedIn, Amazon have changed the way users communicate with them. This led to the necessity of relying on data filtering tools that discover or suggest relevant and personalized items to the users.

Recommender systems (RS) offer valuable means to assist users by providing them with personalized services to cope with information overload. They recommend relevant or interesting items in different domains such as music, movies, books, travel recommendation. In addition, recommender systems can add business value to online service providers by increasing the number of cross-sells and up-sells [9]. Recommender systems strive to estimate the level of interest of a user in a particular item and then recommend to the user the most interesting items. The user's interest level can be in the form of scalar, binary or unary responses [10].

This chapter will review the fundamental concepts and relevant background including latent factor and recent neural network models used for recommendation systems. We also review the background on explainable recommendation systems.

2.1 Recommendation Systems

The main purpose of recommender systems is to recommend relevant items to users. Therefore, the recommendation problem can be defined as follows [11]:

Let U be the set of all users and I be the set of all items, where m and n are the number of users and items, respectively. The set of ratings recorded in the system is denoted

by R, and S is the set of possible values for a rating. When a user $u \in U$ rates an item $i \in I$, the rating is represented by the notation r_{ui}. The goal is to learn a function $f : U \times I \rightarrow S$, where S predicts the rating of an item for a user.

The main types of recommender systems are collaborative filtering, content-based filtering and hybrid methods. The key idea is to build a user-item matrix and then match users with similar interests by computing the similarities.

2.1.1 Content-based Filtering

Content-based filtering is based on the similarity between the items or users which is usually calculated by considering their associated features. A tangible example is when a user has given a high rating to a movie that falls into a thriller genre, then the system learns to generate recommendations from the same movie genre. Therefore, it is assumed that content for each item is available.

The main advantage of these approaches is that they have the potential to generate new items recommendation even if the users have not rated any items. In other words, the model does not require to have additional information or data about other users since the recommendations are user independent. In addition, the system is more transparent because the model's reasoning or explanation is based on the item's content. However, recommender systems that are based solely on content may suffer from the problems of limited content. This is because they are dependent on the rich description of items. Over-specialization [12, 13] is another challenge that content-based filtering approaches can face. This means that these systems tend to recommend items very similar to what users have already liked or purchased in the past. Therefore, the model is not capable of expanding on the users' existing interests.

2.1.2 Collaborative Filtering

Collaborative Filtering (CF) is the most well-known recommendation [14,15]. Unlike content based filtering, in CF, information gathered by users' interests or behavior towards

items is used to produce recommendations [15]. The key idea in predicting user's preferences is to allow users to collaborate with one another. In other words, this approach considers the ratings of not only user u but also other users in the system.

Collaborative Filtering techniques overcome some of the challenges of content-based approaches. For example, CF improves the quality of a recommendation by taking the peer opinions into account instead of using the contents of the items. This can also be helpful when the contents of items is not available or difficult to obtain.

The two fundamental issues with CF methods are: i) data sparsity [16] and ii) cold start problem [17, 18]. Data sparsity is defined as having insufficient rating data which occurs when users rate a very limited portion of available items. As a result, a sparse user-item rating matrix is created. This lack of sufficient ratings or feedback data can prevent CF models to find users with similar preferences. The cold start problem refers to a situation in which it is difficult to generate accurate recommendations due to the presence of new items or users.

In general, CF algorithms are categorized into two groups: *memory-based* and *model-based methods*:

- **Memory-based Methods:** These methods are categorized as user-based or item-based, thus use the rating matrix to measure the similarity between users or items. In these approaches, the ratings of items given by users are directly used to predict ratings for new items [19]. These systems employ statistical approaches to search out a group of users that have a common history with the target user [20]. Typical examples are neighborhood-based CF (item-based/user-based CF algorithms with pearson/vector cosine correlation) and item-based/user-based top-n recommendations. In memory-based Collaborative Filtering approaches, the steps required to recommend items to users are to calculate similarities between users, find the nearest neighbors, predict ratings, rank all items based on ratings and finally recommend items to users [19]. Similar to the user-based Collaborative Filtering, the following steps are required in item-based Collaborative Filtering: Calculate similarities between items, find closest

item neighbors for the target item, Predict ratings for the target item [19].

The main advantage of these approaches is the easy implementation. However, the main shortcoming is that they are dependent on human ratings and also the accuracy of the model is not satisfactory when the data is sparse [21].

In the memory-based method, neighborhood-based approaches have shown a promising result in predicting ratings by considering users whose ratings are similar to the target user, or items that are similar to the target item. [14]. The goal of recommendation systems can be described as, given a $user_u$ and $item_i$, determine the most likely $rating_{ui}$. Neighborhood methods compute the weighted average ratings differently. Some examples are listed below:

Pearson Correlation (PC): The similarity $PC_{u,v}$ between two users u and v is measured as follows [10]:

$$PC(u,v) = \frac{\sum_{i \in I}(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{ui} - \bar{r}_u)^2 \sum_{i \in I}(r_{vi} - \bar{r}_v)^2}} \tag{2.1}$$

where the $i \in I$ is the set of items that both users u and v have rated and $\bar{r}_u$ is the average rating of the co-rated items of user v.

Likewise, to compute the similarity between items i and j using pearson correlation, the following computation is performed [14]:

$$PC(i,j) = \frac{\sum_{u \in U}(r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{ui} - \bar{r}_i)^2 \sum_{u \in U}(r_{uj} - \bar{r}_j)^2}} \tag{2.2}$$

where r_{ui} is the rating that user u has given to item i, $\bar{r}_i$ is the average rating of item i by all those users.

Cosine Similarity: Cosine similarity which is also known as vector-based similarity is defined as a similarity between two items or users with all of their ratings and the angle between these two vectors. User ratings can be shown as an n-dimensional

vector, and the similarity between users is achieved through the user's rating vector angle. For instance, The cosine similarity between two users u and v is computed using the following formula [14]:

$$cos(u,v) = \frac{\sum_{i \epsilon I_{u,v}} r_{u,i} \quad r_{v,i}}{\sqrt{\sum_{i \epsilon I_u} {r_{ui}}^2 \quad \sum_{i \epsilon I_v} {r_{vi}}^2}} \tag{2.3}$$

where I_{uv} is the set of items that both users u and v have rated, $r_{u,i}$ and $r_{v,i}$ are ratings that user u and v have given to an item i, respectively. Similarly, the cosine similarity between two items j and k is as follows [14]:

$$cos(j,k) = \frac{\sum_{u \epsilon U} r_{u,j} \quad r_{u,k}}{\sqrt{\sum_{u \epsilon U} {r_{uj}}^2 \quad \sum_{u \epsilon U} {r_{uk}}^2}} \tag{2.4}$$

where U is the set of users that rated both items j and k, and $r_{u,j}$ and $r_{u,k}$ are the ratings that user u has given to items j and k.

Nearest Neighbors: k-Nearest Neaighbors (kNN) uses the existing data to find clusters of similar users. In this method, the top k neighbors that have the closest similarity to the target user are selected. The main drawbacks of kNN in the recommender system domain are low scalability [22] and vulnerability to high levels of sparsity [23]

- **Model-based Methods:** This approach makes predictions by training the user ratings. In other words, the user ratings is used to learn low dimensional factors of users and items. [24, 25]. A few examples of model-based CFs include the cluster-based CF [26], Bayesian Classifiers [27], rule-based approaches [28] and Matrix Factorization model (MF) [2]. Among these methods, MF has become the popular choice. In MF model, a low rank representation of the rating matrix is learned which is in turn used to predict new ratings between users and items. MF can also be formu-

lated from a probabilistic perspective which is known as Probabilistic Matrix Factorization(PMF) [29]. In general, the model-based CFs deal quite well with sparsity. However, they are often time-consuming to build or update.

2.1.3 Hybrid Recommendation

To overcome certain limitations of CF and content-based methods, hybrid recommendation approaches are used [30]. These models are based on the combination of two or more recommendation models (e.g. content and collaborative filtering techniques) to benefit from their complementary advantages. For instance, to overcome the sparsity and high dimensionality problem, a hybrid model was developed in [31]. They proposed a neighbor user finding approach which is derived from the clustering method. In this approach, different sub-spaces of rated items for different categories such as Interested, Neither Interested Nor Uninterested, and Uninterested are extracted. They also proposed a new similarity method to compute the similarity value. Another hybrid recommendation model proposed in [32] is based on Matrix Factorization approach which uses the hypergraph topological theory to take contextual information, user features, features, and similarities of ratings from users into account.

2.2 Related Work in Recommender Systems (Non-Neural Latent Factor-Based Approaches)

2.2.1 Matrix Factorization (MF)

Ratings in Recommender Systems are often placed in a matrix in which one dimension represents users and the other represents items. This matrix is often sparse due to the high portion of missing values. The main two purposes of matrix factorization for Recommender Systems are to discover the underling latent features and to predict missing values of the matrix. The key idea is to take the original matrix $\mathbf{R}$, and decompose it into two smaller matrices $\mathbf{P}$ and $\mathbf{Q}$ that approximate the original one when multiplied together [2]:

$$\mathbf{R}_{u\times i} \approx \mathbf{P}_{u\times k}\ \mathbf{Q}^{T}_{i\times k}$$

k is the number of hidden factors which is also the hyperparameter of the model. Considering user-item rating predictions, the u^{th} row of P, denoted by p_u is the feature vector for the u^{th} user. Similarly, the ith row of Q, denoted by q_i is the feature vector for the ith item. $p_u q_i^t$ describes the interaction between user u and item i.

To learn the feature vectors (p_u and q_i), the following objective function must be minimized [2]:

$$\sum_{(u,i \in R)} (r_{ui} - P_u Q_i^T)^2 + \lambda(||p_u||^2 + ||q_i||^2) \tag{2.5}$$

where R is the set of user-item pairs. The algorithm learns the model by fitting the previously known values which is observed ratings in this case. In order to avoid overfitting, the hyperparameter λ is used to control the effect of regularization [33]. This value is usually determined by cross-validation.

2.2.2 Probabilistic Matrix Factorization (PMF)

PMF is based on the probability theory which assumes Gaussian noise on the ratings data [29,34]. Given the feature vectors for users and items, the conditional distribution over the corresponding ratings was defined as follows [29, 35]:

$$p(R|P, Q, \sigma^2) = \prod_{u=1}^{m} \prod_{i=1}^{n} [N(R_{ui}|P_u^T Q_i, \sigma^2)]^{I_{ui}} \tag{2.6}$$

where P and Q are the latent user and item feature matrices, and R_{ui} is the rating value of user u for item i, P_u and Q_i denote the user and item specific latent feature vectors, respectively.

The prior distributions over P and Q are given by [29]:

$$p(P|\sigma_P^2) = \prod_{u=1}^{m} N(P_u|0, \sigma_P^2 I) \tag{2.7}$$

$$p(Q|\sigma_Q^2) = \prod_{i=1}^{n} N(Q_i|0, \sigma_Q^2 I) \tag{2.8}$$

where I_{ui} is an indicator variable and is equal to 1 if user u rated item i and 0 otherwise, and $N(x|\mu, \sigma^2)$ is the Guassian distribution with mean μ and variance σ^2.

2.2.3 Generalized Matrix Factorization (GMF)

A variant of matrix factorization was proposed in [36]. In their proposed method, they assume that the user latent vector p_u be $P^T v_u^U$ and the item latent vector q_i be $Q^T v_i^I$. They define the mapping function as follows:

$$(p_u, q_i) = p_u \odot q_i \tag{2.9}$$

where $\odot$ denotes the element-wise product of vectors. They also project the vector to the output layer:

$$\hat{y_{ui}} = a_{out}(h^T(p_u \odot q_i)) \tag{2.10}$$

where a_out and h denote the activation function and edge weights of the output layer, respectively. They also discuss that if they consider the activation function to be the identity function and h to be a uniform vector of 1, the MF model is recovered. In their proposed method, they used the sigmoid function as the activation function and learned h from data with the log loss. The sigmoid function is defined by the following formula:

$$\sigma(x) = 1/(1 + e^{-x}) \tag{2.11}$$

2.2.4 Singular Value Decomposition(SVD)

Sarwar et al. [37], used Singular Value Decomposition (SVD) to improve the performance of recommender systems. This method is well-established in the field of Information Retrieval [38] to examine the dimensionality reduction problem.

In the context of collaborative filtering, matrix $R_{m \times n}$ which is the rating matrix can be factored into three pieces U, Σ, and V^T. The matrix U and V are two orthogonal matrices of size $m \times k$ and $k \times n$, respectively where k is the rank of the matrix R. Σ is a diagonal matrix of size $k \times k$ containing the singular values of the matrix R. U and V are also called as left-singular vectors and right-singular vectors of R, where U and V are equivalent to RR^T and R^TR, respectively [39].

2.3 Related Work in Recommender Systems (Neural Approaches)

Neural Networks (NN) and deep learning (DL) have increasingly become popular to train machines to learn information which in turn enables the machine to accurately predict an output. One of the areas that NNs and DL models have gained attraction is in recommender systems. These networks are capable of modelling the non-linearity (depending on the activation function) in data. Traditional methods such as Matrix Factorization are linear models in which the user and item latent factors are linearly combined to build the user-item interaction [40]. The following section explains some of the neural network architectures used in recommender systems.

2.3.1 Multi-Layer Perceptron (MLP)

The first and simplest form of a fully connected neural network is the Multi-Layer Perceptron model (MLP) [41]. It is a feed-forward network with one or more hidden layers between the input and output layer. In the context of recommender systems, [42] proposed a wide and deep learning for App recommendation in which the deep component utilizes MLP on feature embeddings and a wide component is a linear transformation on input features. Similarly, [36] devised a framework that combined MLP with MF. In their architecture, the role of the MLP is to learn interactions between users and items.

2.3.2 Autoencoder (AE)

Recently, a growing body of research has involved using autoencoders for collaborative filtering. An Autoencoder is a family of neural networks which is trained to reconstruct the input. This reconstruction is obtained at the output layer by using the representation learned in the hidden layer. The network consists of two parts: i) *an encoder* ii) *a decoder*. The encoder and decoder may each have a deep architecture which consists of multiple hidden layers. It is worth noting that the output layer has the same dimension or number of neurons as the input layer. Traditionally, the goals of autoencoders were to reduce dimensionality or learn latent features [43].

In a general form, the input r is encoded to a hidden layer representation h, where the encoder function is $h = f(r)$. The decoder then produces the reconstruction r' using the function $r' = g(h)$. In other words, x' represents the reconstruction of the input r. The general architecture is presented in figure 2.1:

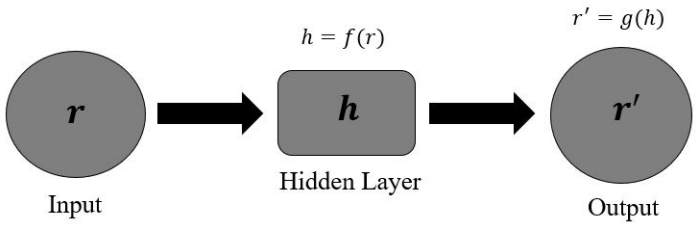


Figure 2.1: Structure of Autoencoders

In [44], an autoencoder model for collaborative filtering was devised to predict movie ratings. AutoRec [45] is another autoencoder based recommender system model which considers only user (r^u) or item (r^i) partially observed ratings as input, and attempts to reconstruct them in the output layer. They developed two variants of the model since it takes either user or item rating vectors as inputs. The U-AutoRec model takes a set of user vectors as inputs, whereas the I-AutoRec's inputs are a set of item vectors. Figure 2.2 illustrates the architecture of the AutoRec and their model worked as follows:

Given a hidden layer, the encoder section of an Autoencoder takes the ratings data

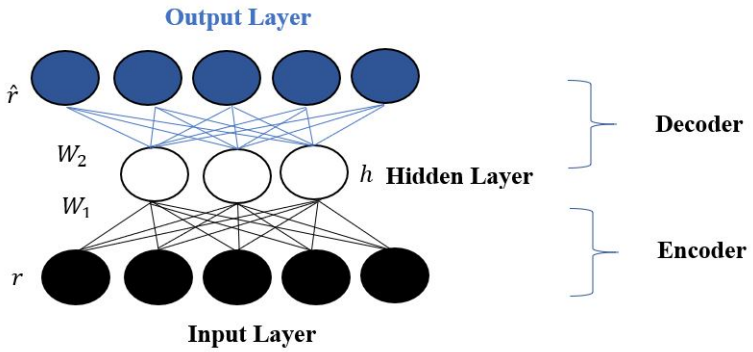


Figure 2.2: An Autoencoder Architecture for Collaborative Filtering

$\boldsymbol{r}$ of each user and maps it to a latent representation $\boldsymbol{h}$, see Eq.(1). User preferences are encoded in a sparse matrix of ratings $\boldsymbol{R} \epsilon \mathbb{R}^{m \times n}$, where m and n are the number of users and items, respectively. $\mathcal{U} = \{1, .., m\}$ represents the set of all users and $\mathcal{I} = \{1, .., n\}$ represents the set of all items. Each user $u \epsilon \mathcal{U}$ is represented by a sparse vector $\boldsymbol{r}^{(u)} = \{R_{u1}, ..., R_{un}\}$, and each item $i \epsilon \mathcal{I}$ is represented by a sparse vector $\boldsymbol{r}^{(i)} = \{R_{1i}, ..., R_{mi}\}$. The hidden or latent representation is given by

$$\boldsymbol{h} = \sigma(\boldsymbol{W}_1.\boldsymbol{r} + \boldsymbol{b}) \tag{2.12}$$

where σ is the activation function - in this case the sigmoid function ($\frac{1}{1+e^{-x}}$), $\boldsymbol{W}_1$ is a weight matrix, and $\boldsymbol{b}$ is a bias vector.

The decoder maps the latent representation $\boldsymbol{h}$ into a reconstruction output $\hat{\boldsymbol{r}}$ given by

$$\hat{\boldsymbol{r}} = \sigma'(\boldsymbol{W}_2.\boldsymbol{h} + \boldsymbol{b}') \tag{2.13}$$

where σ' is an activation function - in this case Identity. The goal of the Autoencoder is to minimize the reconstruction error by minimizing a reconstruction loss function as follows

$$\min \sum_{u} \parallel \boldsymbol{r}^i - \hat{\boldsymbol{r}}^i \parallel_O^2 + \frac{\lambda}{2}.(\parallel \boldsymbol{W}_1 \parallel_F^2 + \parallel \boldsymbol{W}_2 \parallel_F^2) \tag{2.14}$$

where λ is the regularization term coefficient, and $||.||$ is the Frobenius norm. Also, $||.||_O^2$ indicated that only observed ratings are considered. It is worth noting that the U-AutoRec model can be defined in a similar way.

Another work proposed by [46] made use of autoencoder for user or item based model which only supports side information for either users or items. Their model achieved a good performance and could overcome the cold start problem. In a different work, Strub and Mary [47] proposed a Stacked Denoising AutoEncoders neural network (SDAE) with sparse inputs for recommender systems. Wu et al. [48] introduced the Collaborative Denoising Auto-Encoder (CDAE) which has one hidden layer that encodes a latent vector for the user. Zhang et al. [49] developed an Autoencoder model which learns feature representations from side information to improve recommendations.

2.3.3 Restricted Boltzmann Machines (RBM)

Neural Networks (NN) have increasingly become popular to train machines to learn information which in turn enables the machine to accurately predict an output. One of the areas that NNs have gained attraction is in recommender systems. These networks are capable of modelling the non-linearity (depending on the activation function) in data. Traditional methods such as Matrix Factorization are linear models in which the user and item latent factors are linearly combined to build the user-item interaction [40]. An example of such model in collaborative filtering is Restricted Boltzmann Machines (RBM).

RBM is a bipartite graphical model that consists of a visible and a hidden layer [50]. As shown in Figure 2.3 Each layer in this two layered neural architecture is composed of a number of nodes called units and there is no connection between units of the same layer. Let v and h be the visible and output units, respectively. Then W is the weights between these two units. Since RBM is an energy-based model, The energy is computed as follows [51]:

$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} v_j h_i W_{ij} - \sum_{i=1}^{n} b_i h_i - \sum_{j=1}^{m} \mu_j v_j \tag{2.15}$$

where m and n are the number of visible and hidden units, respectively. W_{ij} is the

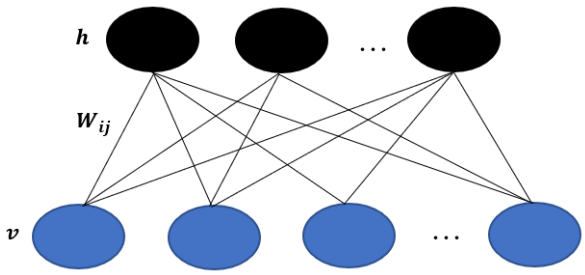


Figure 2.3: RBM Architecture

weight between h_i and v_j; and μ_j and b_i are biases for visible and hidden units, respectively. The conditional probabilities for hidden and visible units are defined as follows [51]:

$$p(v_j = 1|h) = \sigma(\mu_j + \sum_{i=1}^{n} h_i W_i j) \tag{2.16}$$

$$p(h_i = 1|v) = \sigma(\underline{i} + \sum_{j=1}^{m} v_j W_{ij}) \tag{2.17}$$

where $\sigma(x)$ is the logistic function.

2.4 Evaluation of Recommender Systems

The evaluation of recommender systems is generally performed in two ways [6]: i) *offline evaluation* and ii) *online evaluation*. In an offline evaluation, part of the data is used as a training set to build the model, and the rest is used as a test set to test the prediction regarding users' tastes. This method does not involve real user interaction, and thus it is an inexpensive approach [52]. In contract, an online evaluation requires participation of real users on real tasks. It is the most desirable testing method but can be costly [53].

2.5 Explanations for Recommender Systems

Most research in the RS area has focused on developing and evaluating a model that can efficiently compute recommendations. However, the ability to effectively explain why a recommendation is given to a user is another important aspect of a RS. In a definition

given by [7], the main contribution of explanation is to give an opportunity to users to make more informed and accurate decisions rather than to promote recommendations. Building an explanation facilitates recommender systems to benefit users by removing the Black Box form with no given insight to users. Some of the benefits of why recommender systems should provide an explanation can be summarized below:

Justification: which contributes to the user's understanding of the logic behind a recommendation so as to make a decision about a particular recommendation [54].

Efficiency: Explanation can help users make a quick decision on the recommended items. This can be improved by allowing a user to make a comparison between competing choices [55, 56]. For instance, in the domain of smart phones, competing options are "Battery-Life and price".

Satisfaction: Good explanation can increase user's satisfaction or acceptance. The availability of longer descriptions of an item can have a positive impact on the usefulness of the recommender system [57]

Trust: In studies conducted by [57,58], it has been shown that trust can increase the chance of users returning to recommender systems. An explanation does not compensate for a poor recommendation, but it can help the user understand why a bad recommendation has been offered [59].

Persuasiveness: Persuasion measures the likelihood of a user purchasing an item with the presence and absence of explanation facility. A persuasive explanation can also influence previous evaluations of an item [59, 60].

Transparency: Greater visibility and interactivity with the system can help user

satisfaction, build confidence in users and improve performance. It is also worth noting that transparency can be linked to accuracy [59].

2.5.1 Explanation Styles

Most recommender systems are black boxes that do not explain the reasoning behind a particular item recommendation [61]. Thus, the only way that a user can evaluate the quality of a recommendation is to purchase or experience the item. However, the goal of a good recommender system is not to just promote an item to a user, but to increase the user's satisfaction.

The ability of a recommender system to explain the reasoning of its recommendation and making it more transparent can increase user's acceptance and trust. So far, several recommender systems offer explanation [62,63]. The three different explanation styles studied by [7] are listed below:

Keyword Style Explanation (KSE): Keyword Style Explanation(KSE) is a common technique used to explain content-based recommendations. KSE looks for the strongest match between the content of an item and a user. For instance, LIBRA (Learning Intelligent Book Recommending Agent), a content-based system designed by [7], gives the user the option of clicking on the *explain* column to see where the keyword came from. It basically shows the number of occurrence of the word and its strength.

Billsus and Pazzani [63], also designed a news agent that used a keyword style explanation approach which synthesizes speech to read news stories to a user. They explored the use of "concept feedback" to build explanations for its recommendations. They constructed their explanation based on a short-term or long-term model of the user's tastes.

Symeonidis, et al. [54] proposed a movie recommender systems called MoviExplain, which gives the user the ability to check the reasoning behind an explanation. The accuracy of their system was measured using precision and recall.

Neighbor Style Explanation (NSE): A user may be curious to know how his/her neighbors have rated a recommended item. This can be achieved by computing a neighbors' ratings for the recommended item and group them into three different categories: Bad (ratings 1 and 2), Neutral (rating 3), and Good (ratings 4 and 5) [7]. Herlocker et al. [61], used the same approach and found it to give the promising result from the stand point of promotion. In contrast, the KSE and ISE introduced in [1, 7] perform better from a satisfaction perspective.

Influence Style Explanation (ISE): A table of rated items by the user is presented [1, 7]. As opposed to KSE or NSE, ISE does not rely on the underlying recommendation algorithm. It computes the content influence and collaborative influence score separately and then average the two. In order for the result to be interpretable by the user, they set a fixed range of $[-100, 100]$ for the final influence.

In addition to the above-mentioned explanation styles, a few other explanation styles were discussed in [64] that are listed below:

Collaborative-based Explanation Style: In the Collaborative-based Explanation style, similar users are found based on their ratings and the most common format for this style is "Customers Who Bought This Item Also Bought ...". In other words, we can say that this type of explanation is based on similar neighbors.

Content-based Style Explanation: It focuses on each user's behaviour by considering item features that a user has rated. In other words, the similarity of items is computed. For instance, tag preference for a target user is a form of content-based explanation.

Case-Based Reasoning Style Explanation (CBR): This explanation style eliminates the features of items and instead focuses on the similar items used to make recom-

mendation. Basically, it computes the similarity between recommended items. It is worth noting that these items are considered as cases.

Knowledge and Utility-based Style Explanation: In this style, the input is the information about user's interests. Then, the recommender system will find a match based on user's tastes and preferences. There is an overlap between this style, Cased-based Reasoning and the content-based style.

Demographic Style Explanation: In this style, the input to the recommender system is the demographic information of the user. Then, the recommendation algorithm finds users that are demographically similar to the target user.

2.5.2 Recent Works in Explainable Latent Factor Models

Recently, Abdollahi and Nasraoui [5] proposed an Explainable Matrix Factorization based Collaborative Filtering technique.

As discussed in Section 2.2.1, given a feedback matrix $\boldsymbol{R} \epsilon \mathbb{R}^{m \times n}$, where m and n are the number of users and items, respectively, the goal of the Matrix Factorization model is to learn the user latent embedding matrix P and item latent matrix Q so as to generate the predicted ratings. We can assume that $R_{u,i}$ represents the rating of user u for item i. To make a recommendation for user u, the recommender system needs to predict missing values of the u-th row of the rating matrix $\boldsymbol{R}$. Learning the latent factors can be achieved by computing the dot product $P.Q^T$ which is the approximation of the rating matrix $\boldsymbol{R}$. The objective function used in Matrix Factorization is the minimization of the regularized squared distance [2]

$$min \sum_{(u,i \in \mathcal{R})} (r_{ui} - p_u q_i^T)^2 + \frac{\lambda}{2}(||p_u||^2 + ||q_i||^2) + \frac{\beta}{2}||p_u - q_i||^2 W_{ui} \tag{2.18}$$

where R is the set of user-item pairs. In Explainable Matrix Factorization (EMF) [5],

an explainability constraint/regularizer is added to the Matrix Factorization (MF) objective function with the goal of enforcing the user and item latent vectors to be close to each other if item i is explainable to user u. The objective function is as follows:

$$min \sum_{(u,i \in R)} (r_{ui} - p_u q_i^T)^2 + \lambda(||p_u||^2 + ||q_i||^2) + \beta||p_u - q_i||^2 W_{ui} \tag{2.19}$$

They presented two neighbor-based explanation methods: NSE (Neighbor Based Style Explanation) which is user-based and considers the active user's neighbors in the latent space. ISE (Influence Style Explanation) which is item-based and uses the recommended item's neighbors in the latent space. This approach is the same as the NSE but instead of computing the similarity between users, the similarity between the neighboring items is calculated. To compute the explainability of an item for a user, they considered a bipartite graph $G = (V, E)$, where V denotes the set of users U and item I, and E is the edge between a user $u \in U$ and an item $i \in I$. They computed the explainability of item i for user u, as follows [1]:

$$W_{u,i} = \begin{cases} Expl_{u,i}, & \text{if } Expl_{u,i} \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{2.20}$$

where θ denotes a threshold above which item i is explainable for user u.

Another recent work from same authors [5] proposed similar method that uses Restricted Boltzmann Machines (RBM). They added an additional visible layer, m with n nodes, where n denotes the number of items. Each node has a value between 0 and 1 which defines the explainability score of an item to the user. The probability distributions were defined as follows:

$$p(h_j = 1|v, m) = \sigma(a_i + \sum_{i=1}^{n} v_i W_{ij} + \sum_{i=1}^{n} m_i D_{ij}) \tag{2.21}$$

$$p(v_i = 1|h) = \sigma(b_i + \sum_{j=1}^{f} h_j W_{ij}) \tag{2.22}$$

$$p(m_i = 1|h) = \sigma(c_i + \sum_{j=1}^{f} h_j D_{ij}) \tag{2.23}$$

where a, b, and c are biases, f and n are the number of hidden and visible units, respectively. W and D are the weights.

An explainable model was proposed by [65] which uses L-1 norm as a soft explainability constraint. The authors also added novelty to the objective function. Their model considers both novelty and explanability in the Matrix Factorization. They mentioned that Euclidean distance (L2 norm) adds a much higher weight to the outliers by squaring the difference, which is unfair and can dominate smaller weights for normal data points. However, since Manhattan distance (L1 norm) has a constant magnitude for the gradient, it minimizes errors equally.

The above-mentioned works use only ratings as their input. Apart from ratings, another proposed model [8] incorporates Knoweldge-based Graphs (KG) and semantic inferences to build an accurate and explainable recommendation system using Matrix Factorization (MF).

2.5.3 Explanation Metrics

In the literature, a few metrics have been introduced to measure the quality of explanation and they are listed below:

1. **Mean Explainability Precision (MEP):** [1] defined a mean explainability precision metric. It is defined as a fraction of explainable items in the top-n list for a user u.

$$xP = \frac{|\{i : i \in top - n, Exp_{u,i} > \theta\}|}{|top - n|} \tag{2.24}$$

where mean xP can be achieved by taking the average of all the values of xP

2. **Mean Explainability Recall(MER):** The explainability recall metric is defined as follow [1]:

$$xR = \frac{|\{i : i \in top - n, Exp_{u,i} > \theta\}|}{|Exp_{u,i} > \theta|} \tag{2.25}$$

3. **Explainability F-score:** This metric can be computed by combining xP and xR. x_{Fscore} is the harmonic mean of xP and xR [1].

$$x_{F-score} = 2(\frac{xP.xR}{xP + xR}) \tag{2.26}$$

2.6 Summary

In this chapter, we reviewed the fundamentals of recommender system approaches. We formally defined recommendation problems and discussed some of the commonly used techniques such as content-based, collaborative filtering, and latent factor methods. We also reviewed some of the explanation techniques used in this field. In the next chapter, we present our proposed model for Cosine-based Explainable Matrix Factorization.

CHAPTER 3

PROPOSED WORK

3.1 Introduction

Over the last few years, there has been a surge of interest in adding explanation to the black-box machine learning models used in recommendation systems. In particular, Matrix Factorization (MF) is a simple but powerful model to generate accurate recommendations but lacks explanation and transparency [66]. In general, while black-box models are more accurate but less interpretable or explainable, white box models can easily produce explainable results but they are less powerful in terms of achieving higher accuracy. For instance, linear regression or decision tree models are less accurate but more explainable [67]. We tackle this problem by generating explainable predictions in the top-n recommendations by adding a new explainable soft constraint to the matrix factorization objective function. In addition, our proposed method incorporates explainability scores from the Neighbor Style Explanation (NSE) that was previously validated in [6], [7], [5] and [8]. Thus, it is out of the scope of this study to demonstrate that this explanation style is effective to users in promoting explanations. Instead, what we aim to achieve is a model for recommendations that is able to generate explainable recommendations without sacrificing its accuracy.

In this chapter, we propose a methodology to build a Cosine-based Explainable Matrix Factorization (CEMF) model. In Section 3.2, we discuss the motivation for our model. In Section 3.3 we present our proposed model and discuss a desirable feature that is a by-product of our method, namely reduced bias toward popular items. In Section, 3.4, we discuss the explanation scores used in this work. In Section 3.5, we analyze the popularity bias impact of explainability term on predicted ratings. Finally, we summarize the chapter in Section 3.6.

3.2 Motivation

Explainable Matrix Factorization (EMF) [5] is a state of the art explainable Matrix Factorization algorithm that uses the Euclidean distance in formulating a soft explainability penalty or constraint that is key to the explainability of the model. The disadvantages of the Euclidean distance are as follows:

- The L-2 norm (Euclidean distance) only considers the magnitude or the length. In other words, it measures the distance between the two vectors' end points.
- The Euclidean distance weighs all dimensions equally [68] and in general, it is not a desirable metric for high-dimensional spaces due to the curse of dimensionality [69,70].

In order to overcome the above-mentioned problems, we propose a new explainability penalty term that relies on the Cosine based distance between the user and item's latent features. This is because the Cosine similarity is scale invariant as it computes the angle between two vectors. In other words, it is insensitive to the scaling with respect to the origin. In addition, it gives more weight to dimensions having high values (peaks in histogram) and is more robust against the "curse of dimensionality" [69].

3.3 Proposed Objective Function for Cosine-based Explanation in Matrix Factorization

To understand the role of the Cosine metric, we can assume that if the norm of p_u is small, then the user u has shown little interest in most of the items, and if the norm of q_i is a small value, then the item i is less popular among the users. The same applies to this scenario where the value for $||p_u||$ is high which means the user u has shown more interest in most items and if $||q_i||$ is high, then the item i is more popular. These are all global information that are carried with the norm of either $||p_u||$ or $||q_i||$. The role of the Cosine metric is to find the pairwise interaction between users and items. For example, if the value of $cos(p_u, q_i)$ is small, then this means that user u does not like the latent features

TABLE 3.1: Summary of Notations

Symbol	Description
P	user latent embedding matrix
Q	item latent embedding matrix
$U_{j,x}$	set of users given rating x to item j
$\mathcal{R}^{test}$	ratings in the test set
$\boldsymbol{W}$	explainability matrix
$\boldsymbol{R}$	rating matrix
$\mathcal{R}$	set of user-item pairs
N_u	set of users who are most similar to user u
r_{ui}	rating that user u has given to item i
W_{ui}	explainability matrix of user u and item i

associated with item i.

Recall from Section 2.5.2 that the explainability soft constraint in the objective function for EMF used the L-2 norm as shown in Equation 3.1.

$$min \sum_{(u,i \in \mathcal{R})} (r_{ui} - p_u q_i^T)^2 + \frac{\lambda}{2}(||p_u||^2 + ||q_i||^2) + \frac{\beta}{2}||p_u - q_i||^2 W_{ui} \tag{3.1}$$

where $\mathcal{R}$ is the set of user-item pairs. As discussed in Section 3.2, there are known issues with the Euclidean distance metric. To overcome these limitations, we replace the L-2 norm used in the last term of the objective function in Equation 3.2 by the Cosine-based distance. For clarity, all notations used in this chapter are summarized in Table 3.1.

$$\min \sum_{(u,i \in \mathcal{R})} (r_{u,i} - p_u q_i^T)^2 + \frac{\beta}{2}(||p_u|| + ||q_i||)^2 + \frac{\lambda}{2}(1 - \cos(p_u, q_i))^2 W_{ui} \tag{3.2}$$

where $\cos(p_u, q_i)$ is given by

$$cos(p_u, q_i) = \frac{p_u . q_i^T}{||p_u|| ||q_i||} \tag{3.3}$$

To minimize the objective function, we use stochastic gradient descent [71].

In order to obtain the gradients of the objective function with respect to the latent factors, we first obtain the derivatives of $\cos(p_u, q_i)$ with respect to p_u and q_i. These are shown in Equation 3.4 and Equation 3.5:

$$\frac{\partial \cos(p_u, q_i)}{\partial p_u} = \frac{\partial}{\partial p_u} \left(\frac{p_u . q_i^T}{||p_u|| ||q_i||} \right) = \frac{q_i}{||p_u|| ||q_i||} - \frac{(p_u . q_i^T) p_u}{||p_u||^3 ||q_i||} \tag{3.4}$$

$$\frac{\partial \cos(p_u, q_i)}{\partial q_i} = \frac{\partial}{\partial q_i} \left(\frac{p_u . q_i^T}{||p_u|| ||q_i||} \right) = \frac{p_u}{||p_u|| ||q_i||} - \frac{(p_u . q_i^T) q_i}{||q_i||^3 ||p_u||} \tag{3.5}$$

This allows us to derive the following update equations for the latent factor vectors.

$$p_u = (2(r_{u,i} - p_u q_i^T) q_i - \beta p_u - \lambda(1 - (\frac{q_i}{||p_u|| ||q_i||} - \frac{(p_u . q_i^T) p_u}{||p_u||^3 ||q_i||})) W_{ui} \tag{3.6}$$

$$q_i = (2(r_{i,j} - p_u q_i^T) p_u - \beta v_i - \lambda(1 - \frac{p_u}{||p_u|| ||q_i||} - \frac{(p_u . q_i^T) q_i}{||q_i||^3 ||p_u||})) W_{ui} \tag{3.7}$$

Algorithm 3.1 summarizes the steps of the Cosine-based Explainable Matrix Factorization (CEMF) method.

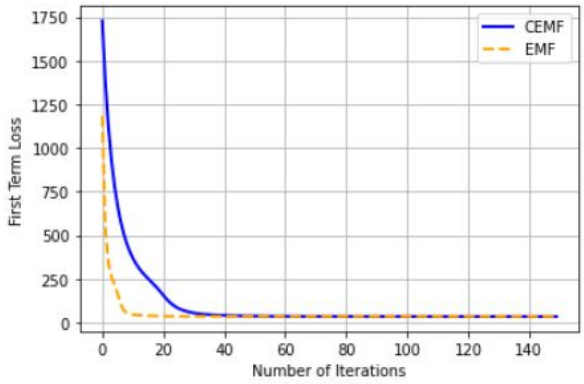
Figure 3.1, shows the convergence of both EMF and CEMF models. The first loss term is the regular matrix factorization objective function in Equation 3.2. The second term is the explainability soft constraint (term 3) in both models EMF and CEMF in Equation 3.1 and Equation 3.2, respectively, while the total loss is the total loss from the entire objective function in Equations 3.1 and 3.2.

Algorithm 3.1 Cosine-based Explainable Matrix Factorization (CEMF)

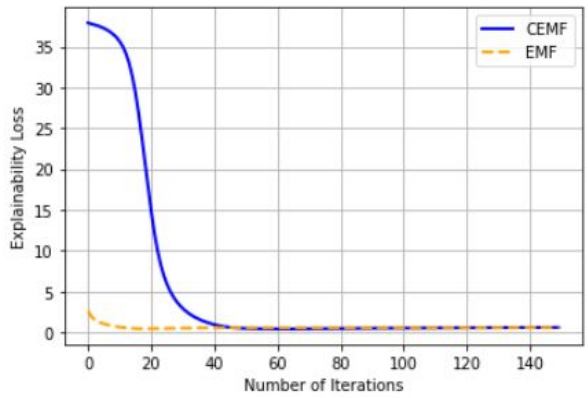
Input: Training set R, number of factors h, number of neighbors N. **Output:** Latent factor matrices: P and Q

1. **for** each user u**:**
   (a) calculate the set of neighbors N_u using the Cosine similarity
2. **end for**
3. **for** each user-item pair (u, i)**:**
   (a) calculate explainability score $W_{u,i}$ using Equation 3.10
4. **end for**
5. initialize latent factor matrices P and Q
6. **for** each rating $r_{u,i}$ from the training set**:**
   (a) Calculate p_u and q_i using the update rule in Eq. 3.6 and 3.7
7. **end for**

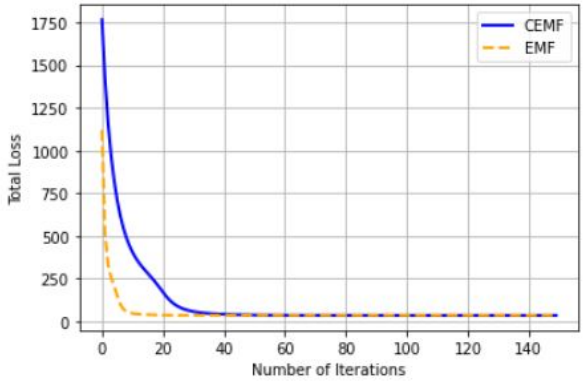
As it is shown in Figure 3.1, we can see that like EMF, CEMF loss decreases with each iterations.



(a)



(b)



(c)

Figure 3.1: (a) First term loss vs number of iterations (b) Second term (Explainability) loss vs number of iterations (c) Total loss vs number of iterations. The loss for each term decreases with each iteration. This shows that the models are learning to predict the ratings, while encouraging the recommendation of explainable items.

3.4 Explanation Scores

We pre-compute the explanation scores offline based on a collaborative filtering neighborhood rationale. The nearest neighbors are determined based on the Cosine similarity. The explainability score for the NSE (Neighborhood Style Explanation) [4, 7, 61] is given by:

$$Exp.Score_{(i,j)} = \boldsymbol{E}(r_{i.j}|N_i) = \sum_{x \epsilon X} x \times Pr(r_{i,j} = x | u \, \epsilon \, N_i) \tag{3.8}$$

where

$$Pr(r_{i,j} = x|N_i) = \frac{|N_i \cap U_{j,x}|}{|N_i|} \tag{3.9}$$

$r_{i,j}$ is the rating user i gave to item j, $U_{j,x}$ is the set of users who have given the same rating x to item j, and N_i is the set of neighbors for user i.

The explainability matrix, $\boldsymbol{W}$, is a thresholded version of the neighborhood explanation scores [4], computed as follows

$$W_{u,i} = \begin{cases} Exp.Score_{(i,j)} & \text{if } Exp.Score_{(i,j)} > \theta \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

where θ is a user-defined threshold value to consider whether item j is explainable to user i.

$Exp.Score_{(i,j)}$ is the expected value calculated in Equation 3.8 and 3.9.

3.4.1 Rationale for the Explainability Scores

To evaluate and measure the user satisfaction for the NSE form of explanations, the authors in [1, 7, 61] performed user-studies. For example, in [1], they used an online application to recommend movies from the MovieLens dataset. They followed the same idea in [7, 61] and divided their pre-computed explainability scores into three categories of low, medium, and high explainability. The users were asked to rate at least 10 movies which allows the recommendation system to know users' tastes. Then recommendations were

generated for each user with an explanation. An example of their explanation is shown in Figure 3.2. After that, users were asked to answer questions related to the explainability they received. Those who were shown explanations from the high group were more satisfied compared to those who were shown medium or low for their recommendations.
The explainability scores that are used in our model as additional inputs were thus previously validated in [1, 7, 61].

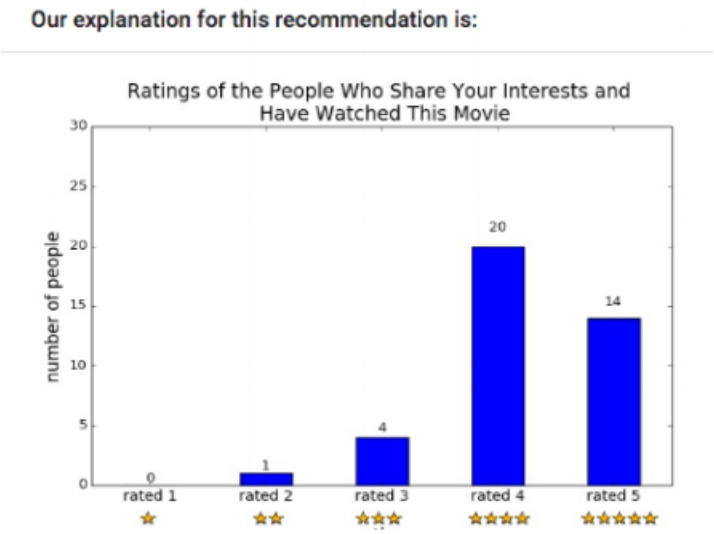


Figure 3.2: An example of NSE explanation from the user-study in [1]

3.5 Analysis of the Impact of the Explainability Term on Predicted Ratings and Popularity Bias

Popularity bias is one of the challenges that collaborative filtering recommender systems face [72]. Popularity bias happens when a recommender system recommends popular items and ignores the unpopular ones also known as "long-tail" items [73]. The popularity distribution of items and the Long Tail items in the MovieLens 100K data is shown in Figure 3.3. As shown, the number of items that are highly rated or the popular items are lower compared to the number of unpopular items.

To compare the popularity bias on the predicted ratings, we investigate the difference between the L-2 norm and Cosine distance in EMF and CEMF, respectively. We first

re-write the equations of the explainability penalty term for both L-2 norm and Cosine distances as shown in Equations 3.11 for EMF and 3.12 for CEMF

$$Exp - penalty_{(EMF)} = ||p_u - q_i||^2 W_{ui} = (||p_u||^2 + ||q_i||^2 - 2p_u.q_i)W_{ui} \tag{3.11}$$

$$Exp - penalty_{(CEMF)} = (1 - cos(p_u, q_i))^2 W_{ui} = (1 - \frac{p_u.q_i}{||p_u||||q_i||})^2 W_{ui} \tag{3.12}$$

Let the Exp-penalty be denoted by Ex_{EMF} and Ex_{CEMF} for EMF and CEMF, respectively. We can re-write the above equations to extract the predicted ratings $p_u.q_i$ as follows:

$$p_u.q_i = \frac{1}{2}(||p_u||^2 + ||q_i||^2 - \frac{Ex_{(EMF)}}{W_{ui}}) \tag{3.13}$$

$$p_u.q_i = 1 - (\sqrt{\frac{Ex_{CEMF}}{W_{ui}}})||p_u||.||q_i|| \tag{3.14}$$

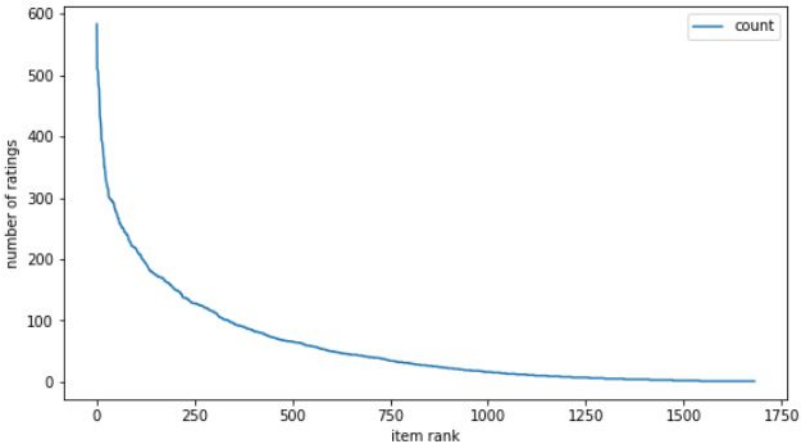


Figure 3.3: Popularity distribution of items in MovieLens 100K

As discussed in Section 3.2, it is expected for popular items to have higher norms. This is also the same when we have popular users or those who rated more items. In order to see the effect of popularity in our model, we identified the top 25 most and least popular

items denoted by MPI and LPI, respectively. These are items with either higher or lower $||q_i||$. We also selected the top 25 most and least popular users and denoted them by MPU and LPU, respectively. These are users with higher or lower $||p_u||$. For each of these users and items, we find their respective learned p and q from each model EMF and CEMF. We then use their average norm and apply the final values of the p and q in Equation 3.13 and Equation 3.14. We assume W_{ui} to be a constant since we are interested in measuring the impact of popularity on the predicted ratings to compare the bias.

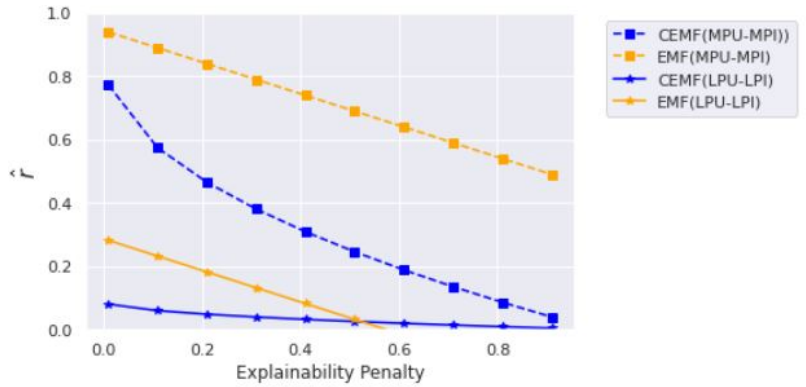
As seen in Figure 3.4a, the negative slope is expected for both models since the predicted ratings should decrease with higher explainability penalty. However, in CEMF the slope is significantly affected for the most popular user-items. Thus, the predicted ratings fall more sharply for more popular items that are not explainable. In the case of CEMF, the predicted ratings fall sharply if the popular item is not explainable for a popular user, meaning that the bias is reduced. Recall that when the explainability penalty for a popular item is high, the item is less explainable, and as expected, this results in a lower predicted rating. When the popular item has a lower explainability penalty, then this means that the item is more explainable and as expected its predicted rating is higher.

By comparing the two models, we can see that EMF predicts a higher rating for most popular items compared to the least popular items. The predicted rating decreases with the explainability penalty until it reaches an explainability penalty close to 0.6, where the predicted rating is almost 0 for both models. This is expected as non-explainable items are discouraged from being recommended. However, CEMF results in a predicted rating of 0 for both least and most popular items while EMF yields drastically different ratings.

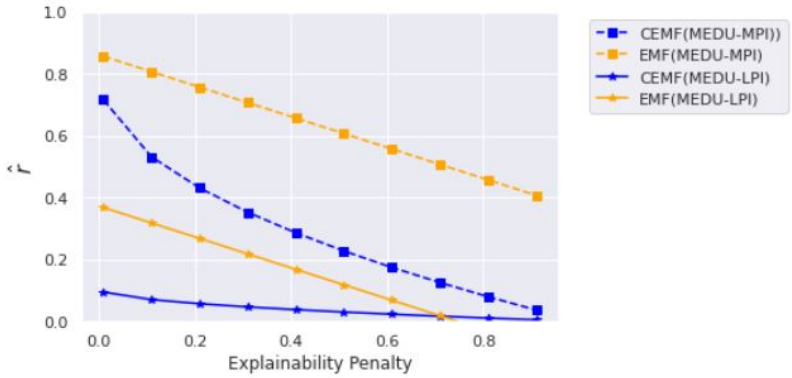
We also looked at the effect of Explainability penalty on the users who are not among the most or least popular users and in our case, we call them Medium users (MEDU). We found p_u for the EMF and CEMF models considering the 25 medium users, then took the average of $||p_u||$. Then, we used the obtained $||p_u||$ and the $||q_i||$ from both most popular (MPI) and least popular items (LPI) in Equation 3.13 and Equation 3.14. Figure 3.4b

shows the same trend as in the previous scenario Figure 3.4a indicating less bias towards most popular items for CEMF compared to EMF.

This shows that in contrast to EMF, CEMF treats both least popular and most popular items that are not explainable (i.e, with very high explainability penalty) the same. Like EMF, CEMF remains biased towards most popular items (albeit less than that of EMF) and this bias is pronounced if the items are highly explainable (i.e., they have low explainability penalty). As the explainability term increases and items become less explainable, however, the popularity bias decreases gradually, until it vanishes completely for non explainable items. This reveals an interesting impact of explainability on popularity bias in both EMF and CEMF. However, for CEMF, this bias vanishes for non-explainable items. This is a desirable feature because in a nutshell, it imposes a high cost on popular items to be favoured in being recommended, namely that they have to be explainable.



(a)



(b)

Figure 3.4: (a) Prediction vs. Explainability penalty for most popular items (MPI) and least popular items (LPI)/ Most popular users (MPU) and least popular users (LPU) (b) Prediction vs. Explainability penalty for medium users (MEDU) and most popular items (MPI)/least popular items (LPI). CEMF yields a reduced bias towards popular items compared to EMF.

3.6 Summary

In this chapter, we proposed a novel explainable matrix factorization framework based on the cosine distance in the penalty term. We discussed the rationale and motivation for our approach to build an explainable recommendation model and also also showed that it succeeds to reduce popularity bias. In the next chapter, we present experiments that will evaluate our method and compare it with other baseline models.

CHAPTER 4

EXPERIMENTAL EVALUATION

4.1 Introduction

In this chapter, we give a brief description of the data and metrics used to evaluate our model. Then, we present and analyze the experimental evaluation of the Cosine-based Explainable Matrix Factorization method presented in Chapter 3 by comparing it with baseline models in terms of accuracy, explainability and popularity bias.

4.2 Data and Experimental Setting

We compared CEMF with three competitive baseline models that are all in the same family of techniques. Matrix Factorization (MF) [2] and Generalized Matrix Factorization (GMF) [36] are two state of the art non-explainable recommendation models, while Explainable Matrix Factorization (EMF) [5] is a state of the art explainable recommendation model. We tested all models on the MovieLens benchmark data. This dataset consists of 100K ratings on a scale of 1-5, given by 943 users to 1682 movies, where each user has rated at least 20 movies. It is worth noting that the dataset sparsity is about 93.695%. The rating data was randomly split into 90% training and 10% testing subsets and the ratings were normalized between 0 and 1. For tuning the hyper-parameters, we split the training set into 80% training and 10% validation. We used grid-search and 5-fold cross validation for hyper-parameter tuning. The hyper-parameters were chosen separately for each algorithm based on the lowest RMSE value. After tuning, the hyper-parameters are as follows: the number of hidden units = 5, the learning rate $\alpha = 0.001$, the regularizer $\beta = 0.001$, the explainability coefficient $\lambda = 0.3$, batch size = 256, and we use the Adams optimizer in SGD. After learning the hyper-parameters, we train the model on the entire training set

which includes both the training and validation set. Finally, we evaluate our model on the left-out test set.

4.3 Research Questions

Our experiments attempt to answer the three research questions, listed below, along with their respective hypotheses.

- **Research Question 1:** Do the CEMF recommendations have higher explainability than the baseline methods?

  Hypothesis Our hypothesis can be stated as follows: Recommending an item using the CEMF model has higher explanation in its top-n recommendation when compared with other Matrix Factorization models.

  The Null Hypothesis: The mean Explainability Precision metric performance of Cosine-based Explainable Matrix Factorization (CEMF) is less than or equal to the competitive baseline models used in this research.

  The Alternate Hypothesis: The mean Explainability Precision performance of the CEMF model will be better than that of the baseline methods used in this research.

- **Research Question 2:** Is the CEMF model more accurate than the baseline methods?

  Hypothesis Our hypothesis can be stated as follows: Recommending an item using the CEMF model is more accurate when compared with other Matrix Factorization

models.

The Null Hypothesis: The mean performance of Cosine-based Explainable Matrix Factorization (CEMF) in terms of accuracy metrics RMSE, MAP or nDCG is less than or equal to the competitive baseline models used in this research.

The Alternate Hypothesis: The mean performance of the CEMF model in terms of accuracy metrics RMSE, MAP or nDCG will be better than that of the baseline methods used in this research.

- **Research Question 3:** Do the CEMF recommendations have less popularity bias than the baseline methods?

Hypothesis Our hypothesis can be stated as follows: The CEMF model has higher popularity bias in its top-n recommendation list when compared with other Matrix Factorization models.

The Null Hypothesis: The mean Diversity metric of Cosine-based Explainable Matrix Factorization (CEMF) is less than or equal to the competitive baseline models used in this research when considering the popularity bias.

The Alternate Hypothesis: The mean Diversity metric of the CEMF model will be better than than that of the baseline methods used in this research when considering the popularity bias.

- **Research Question 4:** Do CEMF recommendations recommend "simultaneously" more accurate and explainable items in its top-n list?

Hypothesis Our hypothesis can be stated as follows: Recommending an item using the CEMF model is more accurate and explainable at the same time when compared with other Matrix Factorization models.

The Null Hypothesis: The mean performance of Cosine-based Explainable Matrix Factorization (CEMF) in terms of the metric, MEP-TR is less than or equal to the competitive baseline models used in this research.

The Alternate Hypothesis: The mean performance of the CEMF model in terms of the metric, MEP-TR will be better than that of the baseline methods used in this research.

4.4 Metrics and Evaluation on Test Data

4.4.1 Evaluating the Accuracy of the Model

To assess the rating prediction accuracy in all the baseline models and the Cosine-based Explainable Matrix Factorization method (CEMF), we compute three different metrics:

- RMSE measures the reconstruction error or rating prediction.
- MAP emphasizes the top of the recommendation list when checking for accurate recommendations.
- nDCG explicitly evaluates the accuracy of the ranking of the top n recommended items.

Hence the three metrics, RMSE, MAP and nDCG, can be considered as increasingly demanding metrics for validating the relevance of the recommendation lists.

First, we compute the Root Mean Squared Error ($RMSE$), shown in Eq. 4.1

$$RMSE = \sqrt{\frac{1}{|\mathcal{R}^{test}|} \sum (r_{i,j} - \hat{r}_{i,j})^2} \tag{4.1}$$

where $|R^{test}|$ is the total number of ratings in the test set, $r_{i,j}$ is the true rating and $\hat{r}_{i,j}$ is the predicted rating for user i and item j.

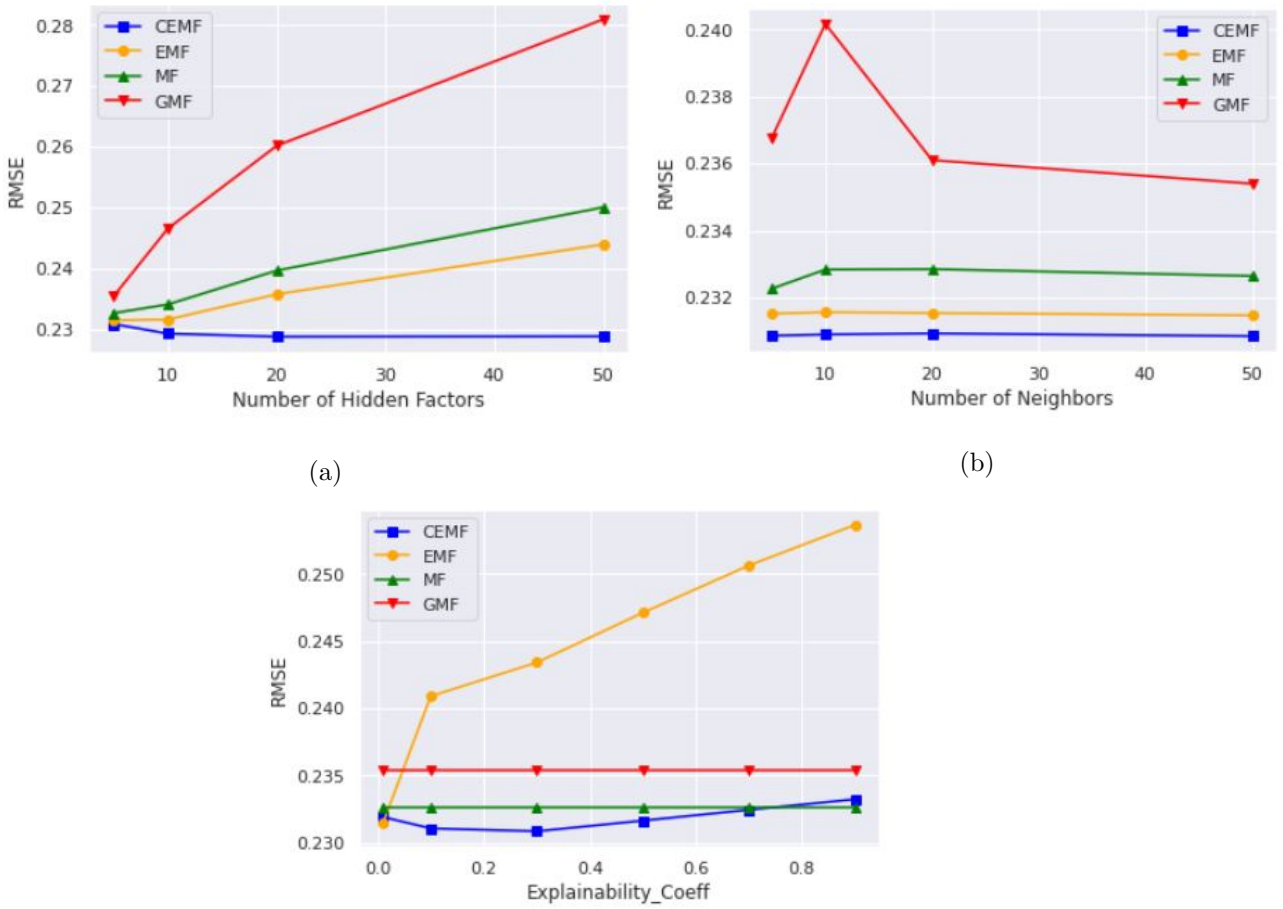
To study the effect of the explainability coefficient, number of neighbors and number of hidden factors on the RMSE result, in each experiment we varied one parameter, while fixing all the other parameters. Figures 4.1a, 4.1b, and 4.1c show the RMSE results vs. the number of hidden factors, number of neighbors and explainability coefficient, respectively. As shown in Figure 4.1a and 4.1b, as we increase the number of hidden factors and number of neighbors, CEMF does not change much in terms of RMSE and remains lower than the baseline methods. Also, as we vary the explainability coefficient, the lowest RMSE for CEMF is achieved when the explainability coefficient is 0.3 (Figure 4.1c).

We also performed a significance test on the results of our method, as shown in Table 4.1. CEMF's superior accuracy result is statistically significant at p-value < 0.05. In order to perform the t-test, we ran each experiment 10 times using the initial tuned hyper-parameters and compared the mean performance metrics of each model with that of CEMF.

TABLE 4.1

RMSE significance test results. Bold indicates significantly better performance.

Model 1	**Model 2**	**P-value**
MF	**CEMF**	1.16e-07
EMF	**CEMF**	7.71e-05
GMF	**CEMF**	1.68e-05



(a)

(b)

(c)

Figure 4.1: (a) RMSE vs. number of hidden factors (b) RMSE vs. number of neighbors (c) RMSE vs. explainability coefficient λ.

To evaluate the top-n recommendation relevance, we compute the Mean Average Precision (MAP@10) using Eq. (10).

$$MAP@n = \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|I|}\frac{1}{m}\sum_{n=1}^{N}P_u(n).rel_u(n) \tag{4.2}$$

where m is the number of relevant items for user i, n is the top-n recommended items and n is 10 in our case, P is the precision of the model and $rel_i(n)$ is a binary value, which is either 0 or 1 that determines whether an item is relevant to user i or not.

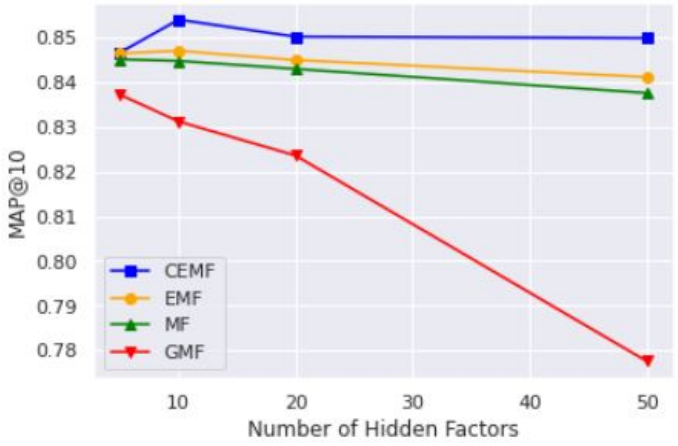
To study the effect of the number of hidden factors and neighborhood size (used for generating the Explainability Matrix (W)), we varied both values and measured the resulting MAP@10 as shown in Figure 4.2. In terms of the number of neighbors and explainability coefficient, there is a drastic drop for EMF as the value of these parameters increase. MF and GMF are steady because there is no number of neighbors or explainability term defined in their objective function. In general, MAP@10 has a more consistent performance in CEMF compared with the baseline models.

In terms of MAP@10, as shown in Table 4.2, the improvement in performance in CEMF when compared with the baseline methods, was only significant when compared with MF and GMF. However, it was not statistically significant when compared with EMF.

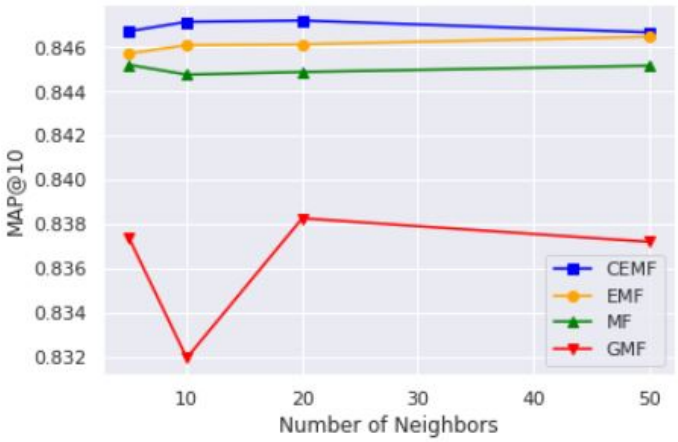
TABLE 4.2

MAP@10 significance test results. Bold indicates significantly better performance.

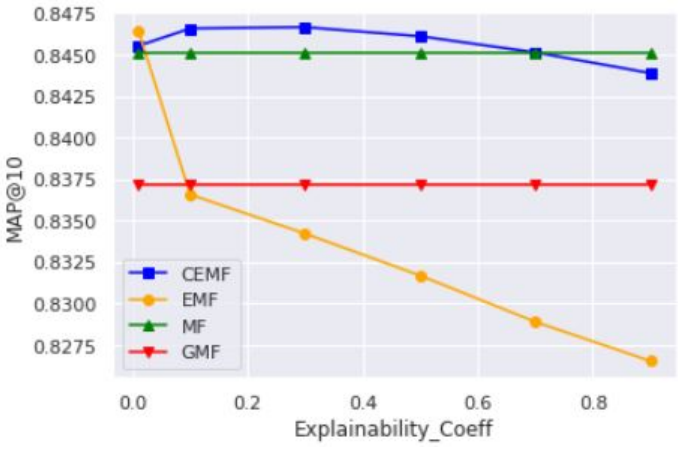
Model 1	**Model 2**	**P-value**
MF	**CEMF**	0.0065
EMF	CEMF	0.5696
GMF	**CEMF**	5.75e-09



(a)



(b)



(c)

Figure 4.2: (a) MAP vs. number of hidden factors (b) MAP vs. number of neighbors (c) MAP vs. explainability coefficient λ.

Finally, we compute the Normalized Discounted Cumulative Gain (nDCG@10) [74] which assigns higher scores to the items in the top ranks, and is given by:

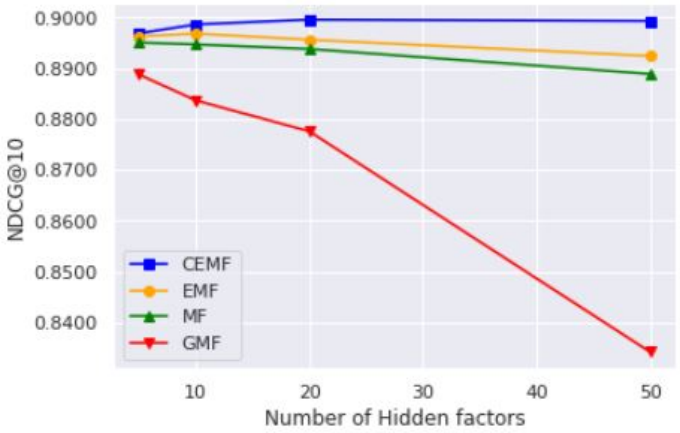
$$nDCG@n = Z_n \sum_{j=1}^{N} \frac{2^{r_j} - 1}{log_2 j + 1} \tag{4.3}$$

where Z_n is a normalizer to ensure that the perfect ranking has a value of 1; r_j is the graded relevance of item at position j.

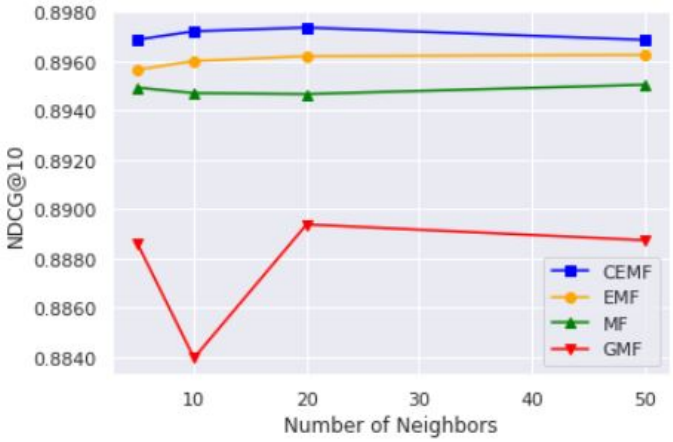
TABLE 4.3

nDCG significance test results. Bold indicates significantly better performance.

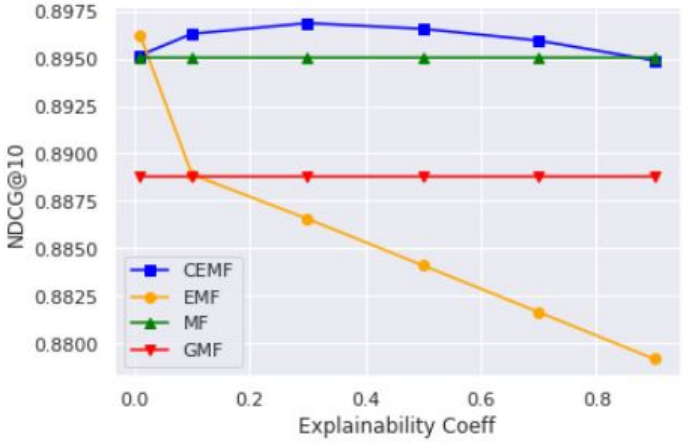
Model 1	**Model 2**	**P-value**
MF	**CEMF**	0.0011
EMF	CEMF	0.1012
GMF	**CEMF**	9.61e-09



(a)



(b)



(c)

Figure 4.3: (a) nDCG vs. number of hidden factors (b) nDCG vs. number of neighbors (c) nDCG vs. explainability coefficient λ.

4.4.2 Evaluating the Explainability of the Model

We compute two metrics, MEP and MEP-TR to evaluate explainability. Evaluating both accuracy and explainability can be achieved by looking at both MEP (4.4) and MAP (4.2) at the same time. MEP-TR (4.5) has the advantage of taking into account accuracy at the same time as explainability, as will be shown below.

The Mean Explainability Precision (MEP) proposed in [5], measures the number of explainable items in the top-n recommended list and is computed as follows:

Let $\boldsymbol{L}^u_{rec}$ be the top-n recommended list for user u and $\boldsymbol{L}^u_{exp}$ be the set of explainable items for user u.

$$MEP@n = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}_{exp} \cap \mathcal{L}_{rec}|}{|\mathcal{L}_{rec}|} \tag{4.4}$$

where $\mathcal{L}_{exp}$ is the set of explainable items for user u, $\mathcal{L}_{rec}$ is the set of items in the top-n recommendation list for user u, $|\mathcal{L}_{exp} \cap \mathcal{L}_{rec}|$ is the number of explainable items present in the top-n recommendation list for user u, and $|\mathcal{U}|$ is the number of users in the test-set.

As shown in (4.4), the MEP metric only considers the number of explainable items in the top-n recommendation without taking into account whether those items are actually liked by the user. To overcome this drawback, we defined another metric called MEP-TR which considers if the set of items in the top-n recommendations appear in the ground truth rank test set for each user. We call this metric MEP-TR and computed it as follows:

$$MEP - TR@n = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}_{exp} \cap \mathcal{L}_{rec} \cap \mathcal{L}_{TrueRank}|}{|\mathcal{L}_{rec}|} \tag{4.5}$$

where $\mathcal{L}_{TrueRank}$ refers to the set of items that are ranked in top-n in the ground truth data

In Figure 4.4, we investigate the effect of the explainability threshold in Equation 3.10 on the explainability metric MEP for the top 10 recommended items by varying the number

TABLE 4.4

MEP@10 significance test results (Explainability Threshold $\theta > 0.1$). Bold indicates significantly better performance.

Model 1	**Model 2**	**P-value**
MF	**CEMF**	3.57e-09
EMF	**CEMF**	7.00e-12
GMF	**CEMF**	0.0050

TABLE 4.5

MEP@10 significance test results (Explainability Threshold $\theta > 0.4$). Bold indicates significantly better performance.

Model 1	**Model 2**	**P-value**
MF	**CEMF**	4.06e-09
EMF	**CEMF**	5.62e-17
GMF	**CEMF**	0.003

of neighbors. We can see that while CEMF maintains its higher explainability in comparison with the baseline models, the explainability decreases for all methods as we impose more restrictions (higher threshold) on the explainability score. This happens because a higher threshold reduces the number of explainable items relative to a user (resulting in more zeros in the explainability matrix $\boldsymbol{W}$). Table 4.4 and Table 4.5 show the significance test of CEMF compared with the baseline models, where we used the explainability threshold of 0.1 and 0.4 for all models, respectively. As shown, MEP in CEMF is statistically significant at p-value < 0.05. CEMF's improvement in MEP is higher and even more significant for higher explainability threshold. Therefore, we can say that the mean performance of CEMF is better than the mean performance of the baseline methods using the MEP metric.

Figure 4.5 shows MEP-TR for the top 10 recommended items by varying the number of neighbors for different explainability thresholds. As shown in Figure 4.5a, there is an overlap between EMF and CEMF when the explainability threshold is greater than 0. However, the difference between these two models becomes significant as we restrict the threshold and CEMF starts performing better and recommending more explainable items

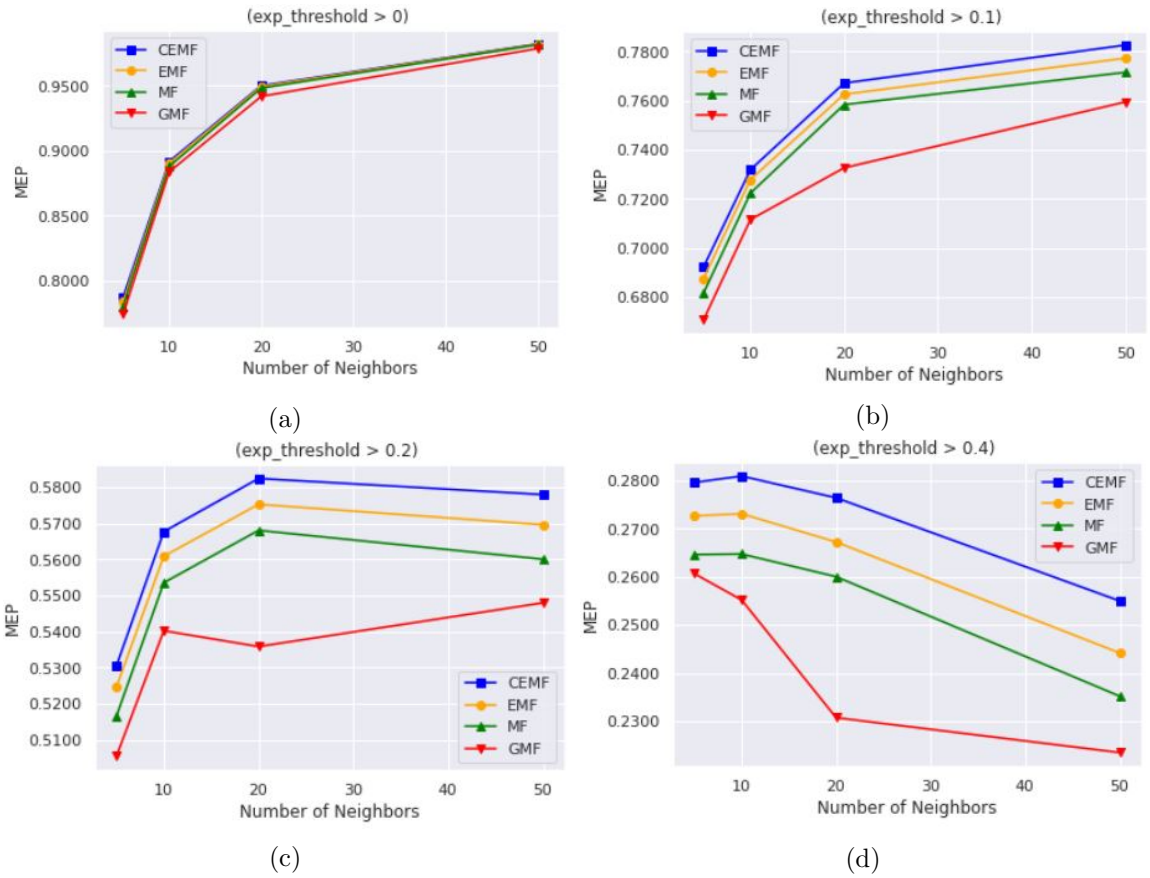


Figure 4.4: MEP vs number of neighbors (a) ($\theta > 0$), (b) ($\theta > 0.1$), (c) ($\theta > 0.2$), (d) ($\theta > 0.4$). CEMF's performance gain increases for very highly explainable recommendations. Hence its recommendations lists contain a greater proportion of highly explainable items ($\theta > 0.4$).

in its top 10 recommendations. MEP-TR for CEMF is also higher when compared with MF and GMF.

Also, notice that the value for MEP-TR is always lower compared with MEP and this is because MEP-TR considers the true rank as well as the explainability of the item in the top-n recommendation.

Table 4.6 and Table 4.7 show the significance test of CEMF compared with the baseline models, where we used the explainability threshold of 0.1 and 0.4 for all models, respectively. As shown, MEP-TR in CEMF is statistically significant at the p-value < 0.05.

CEMF's improvement in MEP is higher and even more significant for higher explainability threshold. Therefore, we can say that the mean performance of CEMF is better than the mean performance of the baseline methods using the MEP-TR metric.

TABLE 4.6

MEP-TR@10 significance test results (Explainability Threshold > 0.1). Bold indicates significantly better performance.

Model 1	**Model 2**	**P-value**
MF	**CEMF**	2.23e-09
EMF	**CEMF**	3.79e-06
GMF	**CEMF**	0.00096

TABLE 4.7

MEP-TR@10 significance test results (Explainability Threshold > 0.4). Bold indicates significantly better performance.

Model 1	**Model 2**	**P-value**
MF	**CEMF**	1.03e-09
EMF	**CEMF**	1.87e-14
GMF	**CEMF**	8.61e-11

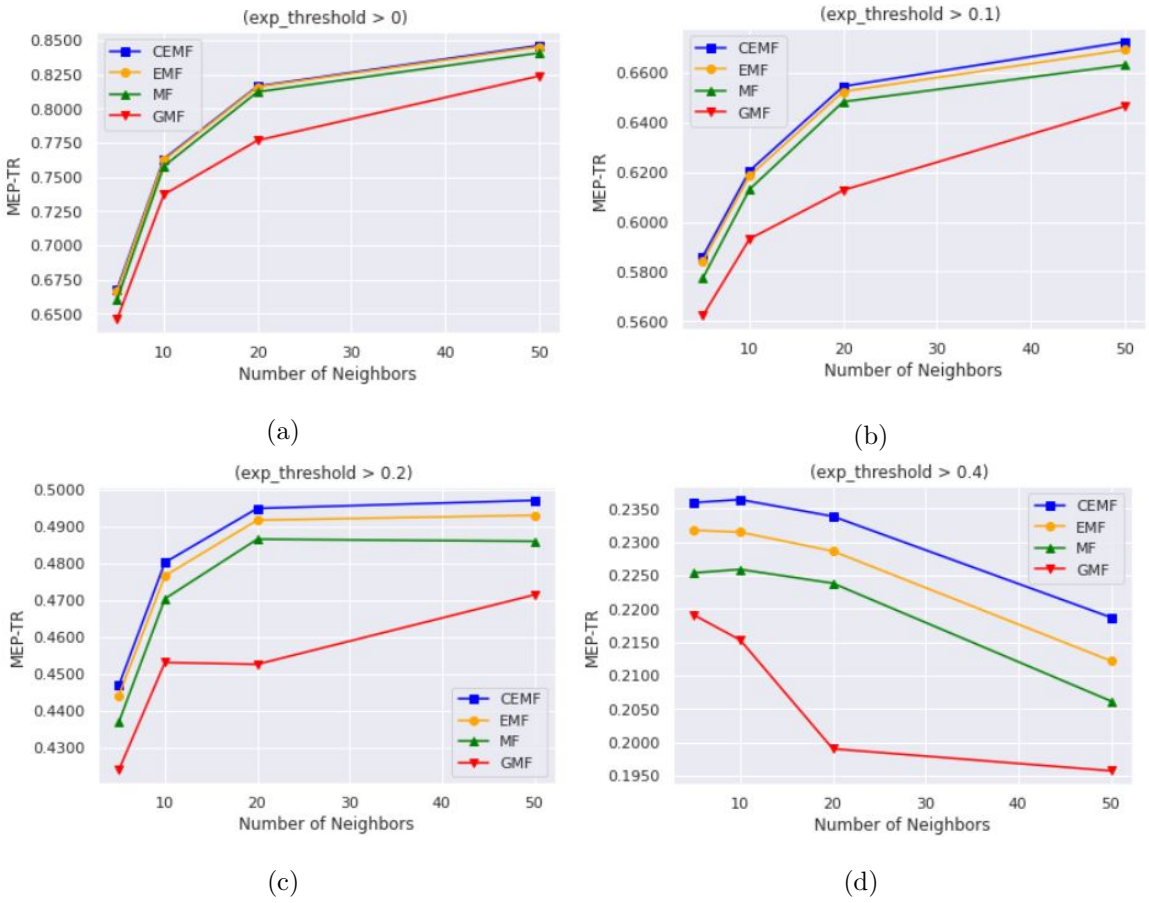


Figure 4.5: MEP-TR vs number of neighbors (a) ($\theta > 0$), (b) ($\theta > 0.1$), (c) ($\theta > 0.2$), (d) ($\theta > 0.4$). CEMF's performance gain increases for very highly explainable recommendations. Hence its recommendations lists contain a greater proportion of highly explainable items ($\theta > 0.4$).

Figure 4.6 shows MEP for various numbers of hidden factors and with different explainability threshold. As can be seen, CEMF model maintains its higher explainability in comparison with the baseline models. In addition, as the explainability threshold increases, the the explainability score increases too. This confirms that the Cosine-based distance metric can handle the higher dimensions better.

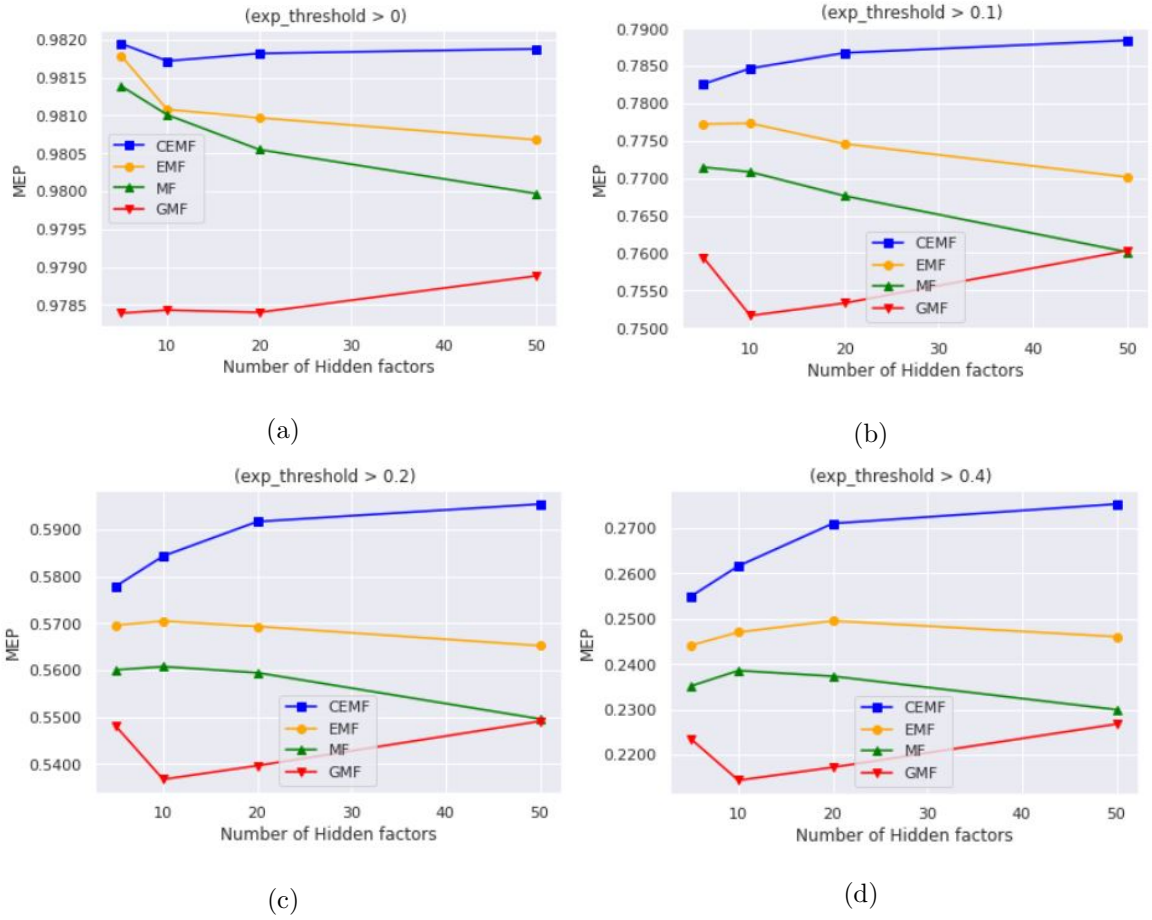


Figure 4.6: MEP vs number of hidden factors (a) ($\theta > 0$), (b) ($\theta > 0.1$), (c) ($\theta > 0.2$), (d) ($\theta > 0.4$). CEMF's performance gain increases for very highly explainable recommendations. Hence its recommendations lists contain a greater proportion of highly explainable items ($\theta > 0.4$).

Figure 4.7 shows MEP-TR by varying the number of hidden factors for different explainability thresholds. As it can be seen, CEMF model maintains its higher explainability in comparison with the baseline models. In addition, as the explainability threshold increases, the the explainability score increases too. This is because Cosine-based distance metric can handle the higher dimensions better.

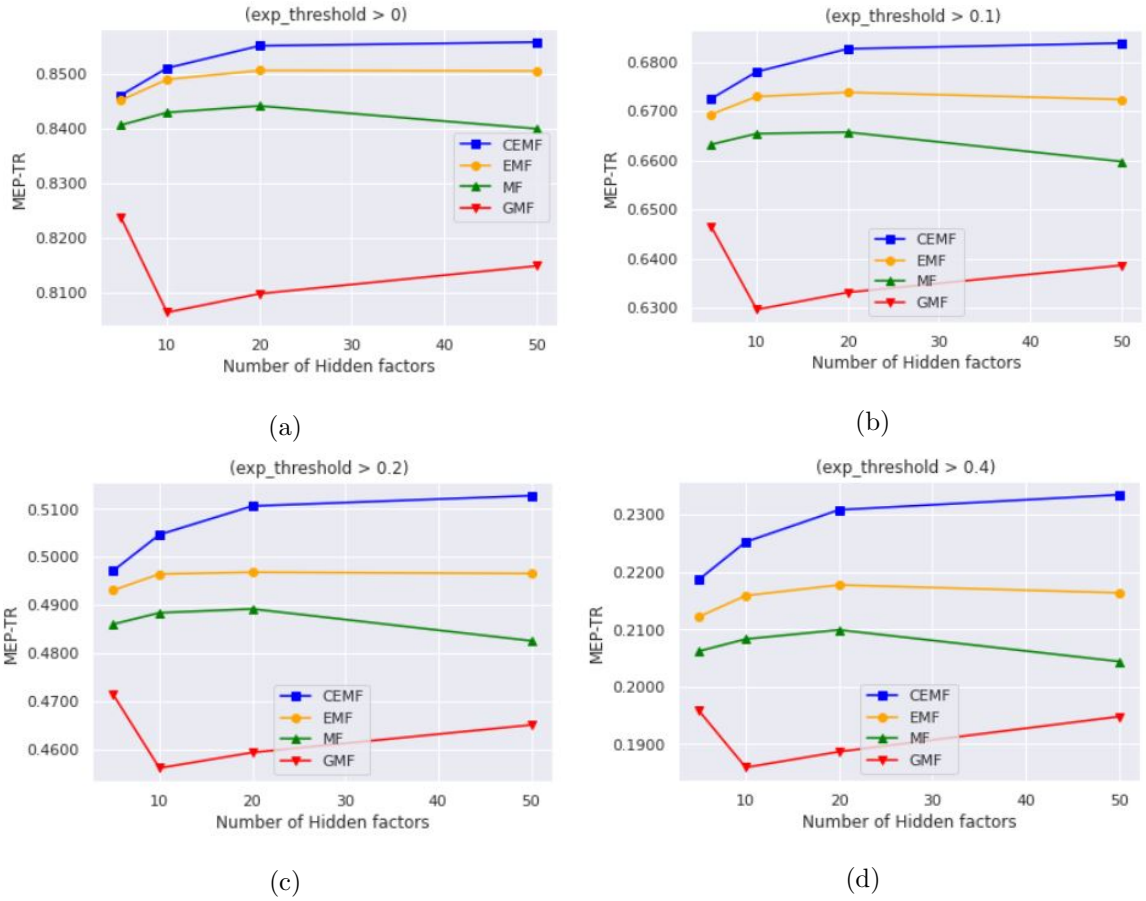


Figure 4.7: MEP-TR vs number of hidden factors (a) ($\theta > 0$), (b) ($\theta > 0.1$), (c) ($\theta > 0.2$), (d) ($\theta > 0.4$).

4.5 Measuring Recommendation Diversity

In addition, we looked at the number of unpopular or novel items that were in the top-n recommendations for each model. The probability that the user u has observed item i [75] is given by

$$P_{ui} = \frac{N_i}{N_U} \tag{4.6}$$

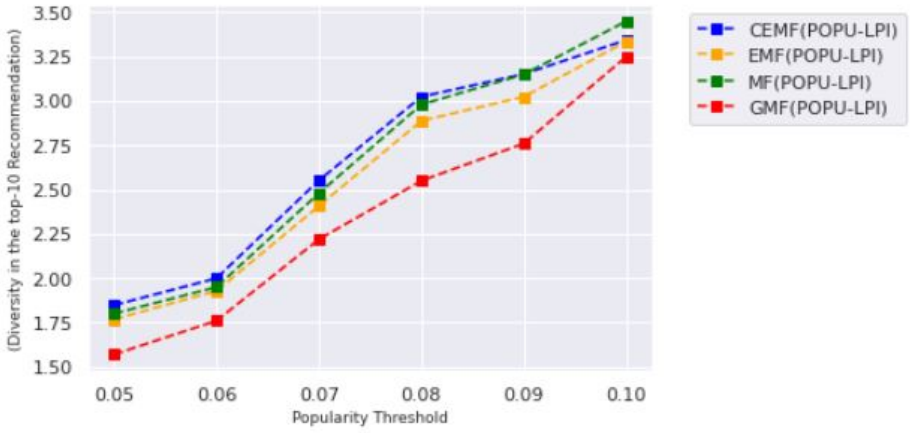
where N_i represents how many times item i has been rated and N_U denotes the total number of users.

Given a popularity threshold θ_p shown in Equation 4.7 [75], we counted the number of unpopular items in the set D_u for each user u in the top-n recommendation. Then, we computed the average of this count across all users [75] as given by Equation 4.8.

$$D_u = \{i \in top - n : P_{u,i} < \theta_p\} \tag{4.7}$$

$$Diversity = \frac{1}{U} \sum_{u \in U} |D_u| \tag{4.8}$$

Figure 4.8 shows the Diversity for the top-25 popular users who has given more ratings to items, by varying the popularity threshold. As we can see, the number of unpopular or diverse items recommended to the users in their top-10 recommendation is higher for CEMF.



(a)

Figure 4.8: Diversity for the top-10 recommendation vs. the popularity threshold for CEMF and EMF.

Table 4.8 shows the significance test p-values of CEMF's diversity compared with

other baseline models. We used the populariy threshold of 0.08 for all models. As shown, Diversity in CEMF is statistically significant at p-value < 0.05 when compared with EMF and GMF. Therefore, we conclude that the mean performance of CEMF is better than the mean performance of the baseline methods in terms of the Diversity metric.

TABLE 4.8

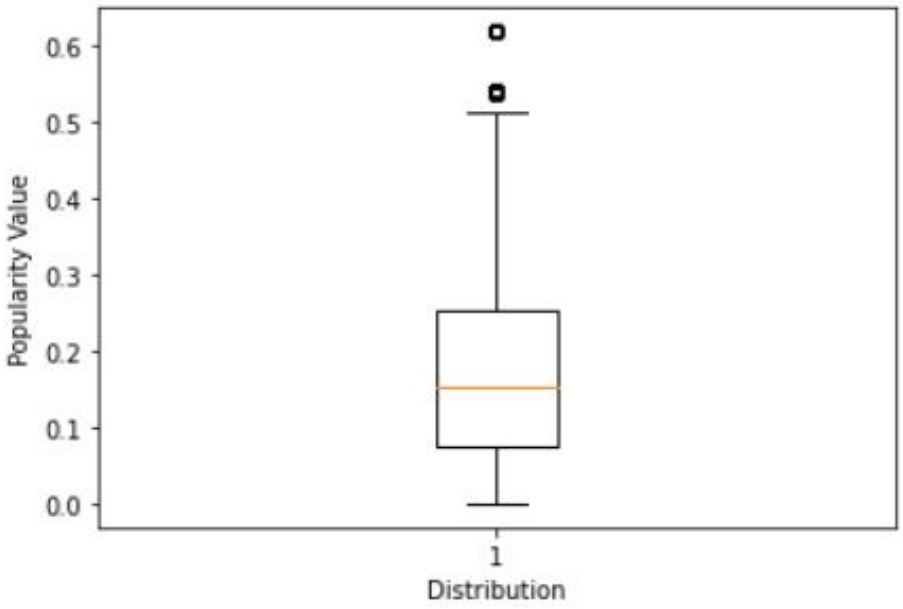
Diversity significance test results for the Top-10 Recommendations (Popularity Threshold < 0.08). Bold indicates significantly better performance.

Model 1	**Model 2**	**P-value**
MF	CEMF	0.80
EMF	**CEMF**	0.03
GMF	**CEMF**	0.007

4.6 Examples

In this section, we show some actual examples of the Neighbor Style Explanation for two randomly selected sample users, denoted as Sample User A and Sample User B. For each user, we show the sample user's top-5 rated movies, from the training set, and the movies recommended to the user using the Cosine-based Explainable Matrix Factorization (CEMF) and the Explainable Matrix Factorization models (EMF). We consider a user to like a movie if the rating is 3 or above, on a scale of 1-5.

Table 4.9 shows the top-5 movies rated by Sample user A, from the training data. Table 4.10 and 4.11 show the top-5 recommendations for the MF and GMF models, respectively. Table 4.12 and 4.13 show the top-5 recommended movies from the CEMF and EMF models. The NSE explanations for CEMF and EMF are shown in Figure 4.12 and 4.13, respectively. Note that MF and GMF are not explainable and there is no explanation available for them. We also show a checkmark symbol (✓) under the Diversity column if the item's popularity is below the threshold of 0.08. This threshold was chosen based on the distribution of the popularity threshold, shown in Figure 4.9. We are interested in the unpopular items or those with popularity below the 25^{th} percentile based on this box-plot.



(a)

Figure 4.9: Popularity distribution box-plot

TABLE 4.9

Top-5 movies rated by Sample user A, from the training data.

Top Rated Movies - Genre
It Happened One Night (1934) - Comedy
Sting, The (1973) - Comedy/Crime
Babe (1995) - Children's/Comedy/Drama
Shining, The (1980) - Horror
Bridge on the River Kwai, The (1957) - Drama/War

TABLE 4.10

Top-5 recommended items for Sample User A. Movies are ranked in descending order of ratings (MF Model). **MF has no explanations**. **Unpopular items have popularity below** 0.08.

Top Recommended Movies - Genre	**Diversity**
Halloween: The Curse of Michael Myers (1995) - Horror/Thriller	-
Die xue shuang xiong (Killer, The) (1989) - Action/Thriller	-
Cinderella (1950) - Animation/Children/Musical	-
Amityville Curse, The (1990) - Musical	-
Sabrina (1954) - Comedy/Romance	-

TABLE 4.11

Top-5 recommended items for Sample User A. Movies are ranked in descending order of ratings (GMF Model). **GMF has no explanations**. **Unpopular items have popularity below** 0.08.

Top Recommended Movies - Genre	**Diversity**
American in Paris, An (1951) - Musical/Romance	✓
Good, The Bad and The Ugly, The (1966) - Action/Western	-
Army of Darkness (1993) - Action/Adventure/Comedy/Horror/Sci-Fi	-
Gone with the Wind (1939) - Drama/Romance/War	-
Annie Hall (1977) - Comedy/Romance	-

TABLE 4.12

Top-5 recommended items for Sample User A. Movies are ranked in descending order of ratings (CEMF Model). **The explanations are shown in Figure 4.10. Unpopular items have popularity below** 0.08**.**

Top Recommended Movies - Genre	**Explanation**	**Expl. Score**	**Diversity**
Pollyanna (1960) - Children's/Comedy/Drama	6% of users who share your interests liked this movie	0.075	✓
Die xue shuang xiong (Killer, The) (1989) - Action/Thriller	24% of users who share your interests liked this movie	0.200	-
Sabrina (1954) - Comedy/Romance	34% of users who share your interests liked this movie	0.345	-
Halloween: The Curse of Michael Myers (1995) - Horror/Thriller	8% of users who share your interests liked this movie	0.085	-
Pump Up the Volume (1990) - Drama	42% of users who share your interests liked this movie	0.405	✓

TABLE 4.13

Top-5 recommended items for Sample User A. Movies are ranked in descending order of ratings (EMF Model) **The explanations are shown in Figure 4.11. Unpopular items have popularity below** 0.08**.**

Top Recommended Movies - Genre	**Explanation**	**Expl. Score**	**Diversity**
Halloween: The Curse of Michael Myers (1995) - Horror/Thriller	8% of users who share your interests liked this movie	0.085	-
Casino (1995) -Drama	8% of users who share your interests liked this movie	0.075	-
Die xue shuang xiong (Killer, The) (1989) - Action/Thriller	24% of users who share your interests liked this movie	0.200	-
Abyss, The (1989) - Action/Adventure/Sci-Fi/Thriller	20% of users who share your interests liked this movie	0.185	✓
Sabrina (1954) - Comedy/Romance	34% of users who share your interests liked this movie	0.345	-

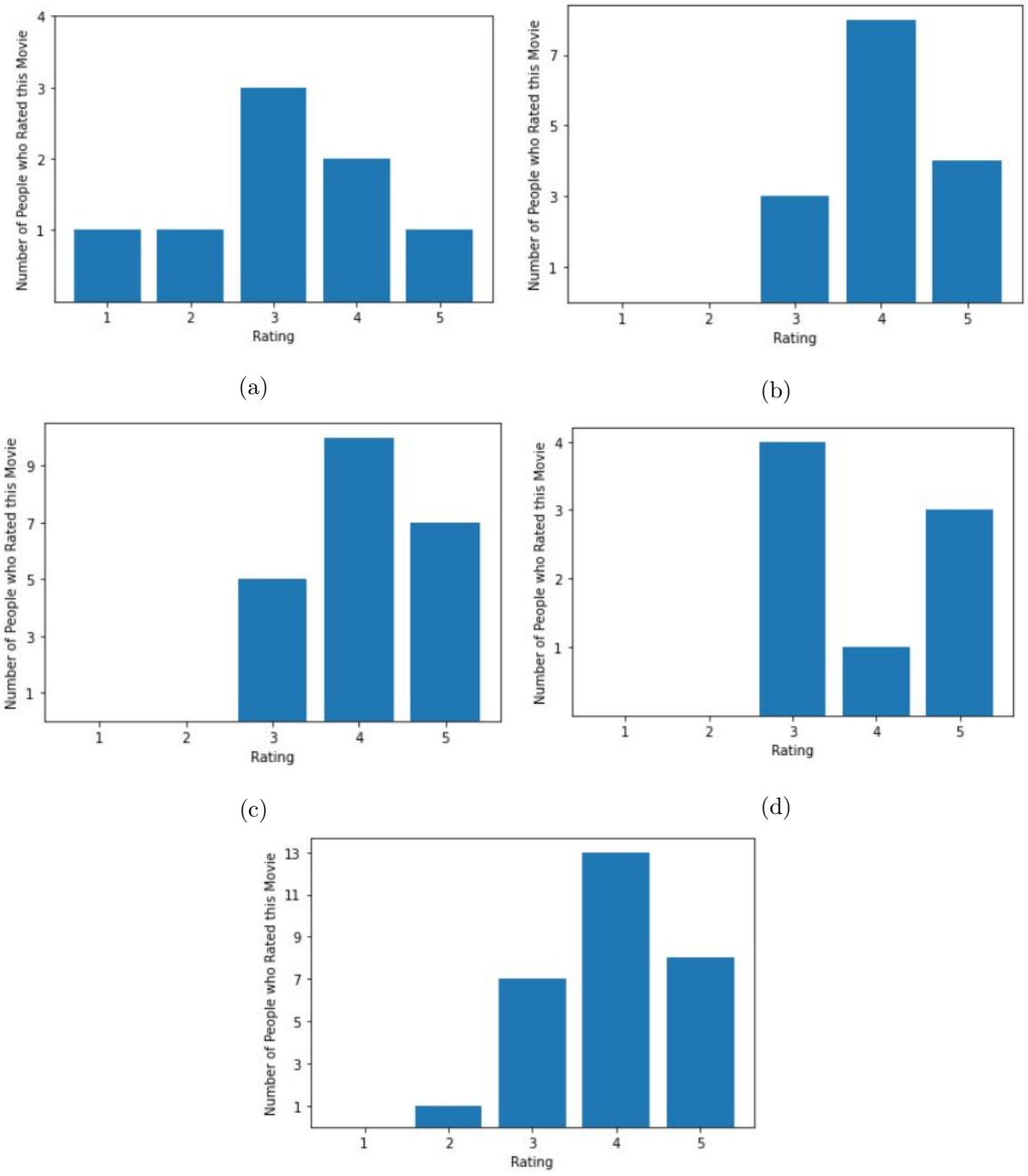


Figure 4.10: NSE Explanation for movies recommended to Sample User A based on Table 4.12 using CEMF model, The movies are shown in the order of their recommendations from (a)-(e).

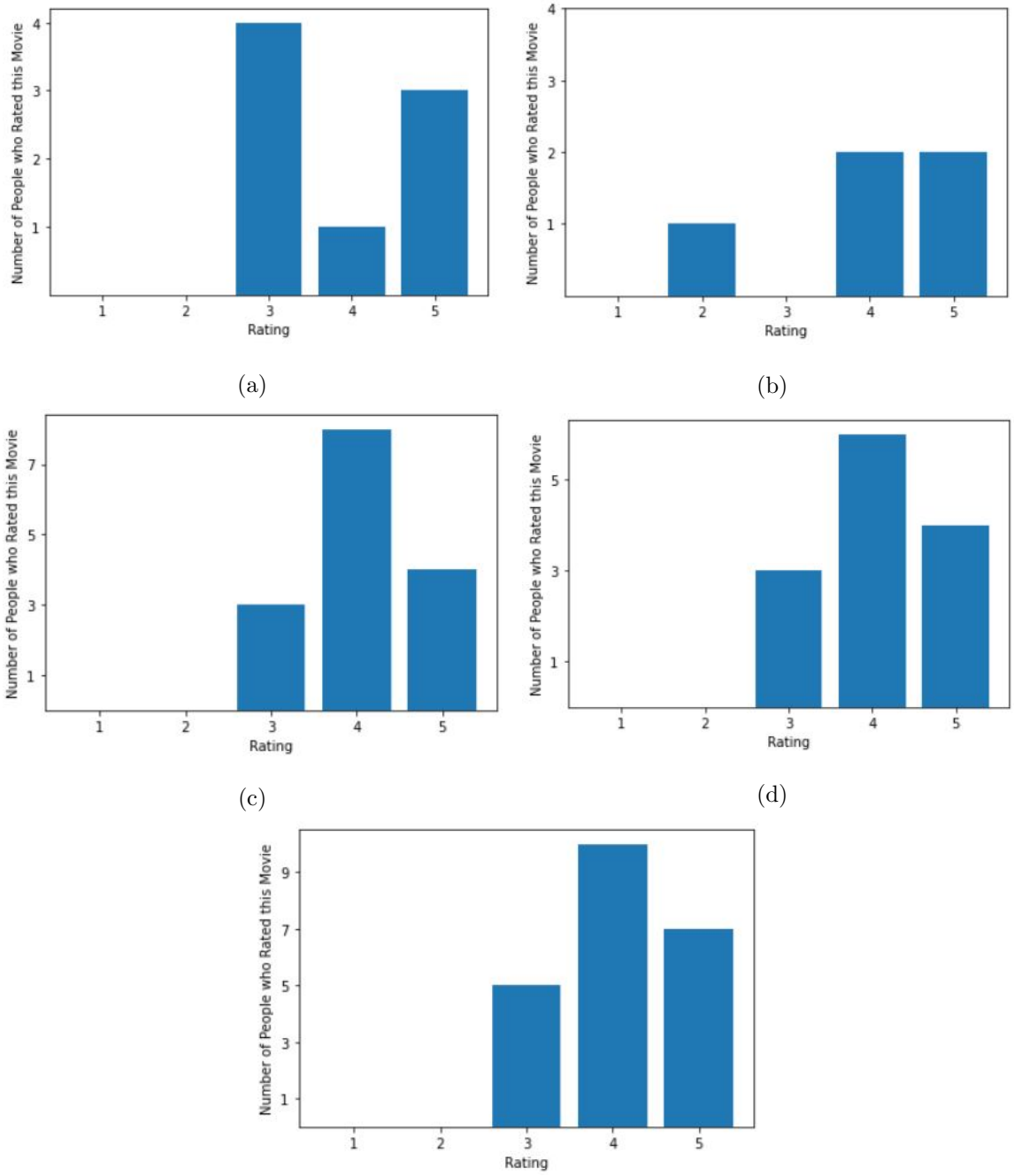


Figure 4.11: NSE Explanation for movies recommended to Sample User A based on Table 4.13 using EMF model, The movies are shown in the order of their recommendations from (a)-(e).

Table 4.14 shows the top-5 movies rated by Sample user B, from the training data. Table 4.10 and 4.11 show the top-5 recommendations for the MF and GMF models.

Table 4.17 and 4.18 shows the top-5 recommended movies for the CEMF model and EMF model, respectively. The NSE explanations for CEMF and EMF are shown in a visual format in Figure 4.17 and 4.18, respectively. MF and GMF are not explainable and there is no explanation available for them.

As shown in Table 4.17 and 4.18, there is more explanation evidence (higher percentages of similar users who like the recommended items) in CEMF than in EMF, for the top-5 recommendations. In general, explanations could provide additional evidence to scrutinize possible biases and missing data and the explanations can possibly unveil issues by scrutinizing the model. For instance, The second and the fourth movies recommended to the sample user in the top-5 recommendations for the EMF models are low. This is also true for the fifth recommended item in the CEMF model.

TABLE 4.14

Top-5 movies rated by Sample user B, from the training data.

Top Rated Movies - Genre
English Patient, The (1996) - Drama/Romance/War
2001: A Space Odyssey (1968) - Drama/Mystery/Sci-Fi/Thriller
Apollo 13 (1995) - Action/Drama/Thriller
Monty Python and the Holy Grail (1974) - Comedy
They Made Me a Criminal (1939) - Crime/Drama

TABLE 4.15

Top-5 recommended items for Sample User B. Movies are ranked in descending order of ratings (MF Model). **MF has no explanations**. **Unpopular items have popularity below** 0.08.

Top Recommended Movies - Genre	**Diversity**
Three Colors: Red (1994) - Drama	-
Three Colors: White (1994) - Drama	-
Killing Fields, The (1984) - Drama/War	-
Glengarry Glen Ross (1992) - Drama	-
Thin Man, The (1934) - Mystery	-

TABLE 4.16

Top-5 recommended items for Sample User B. Movies are ranked in descending order of ratings (GMF Model). **GMF has no explanations**. **Unpopular items have popularity below** 0.08.

Top Recommended Movies - Genre	**Diversity**
Jerry Maguire (1996) - Drama/Romance	✓
Mrs. Doubtfire (1993) - Comedy	-
Star Trek: The Wrath of Khan (1982) - Action/Adventure/Sci-Fi	-
Jurassic Park (1993) - Action/Adventure/Sci-Fi	-
Three Colors: Red (1994) - Drama	-

TABLE 4.17

Top-5 recommended items for Sample User B. Movies are ranked in descending order of ratings (CEMF Model). **The explanations are shown in Figure 4.12**, **Unpopular items have popularity below** 0.08**.**

Top Recommended Movies-Genre	**Explanation**	**Expl. Score**	**Diversity**
Shining, The (1980) - Horror	38% of users who share your interests liked this movie	0.450	-
Butch Cassidy and the Sundance Kid (1969) - Action/Comedy/Western	56% of users who share your interests liked this movie	0.600	-
Vertigo (1958)- Mystery/Thriller	80% of users who share your interests liked this movie	0.615	✓
Bliss (1997) - Drama/Romance	66% of users who share your interests liked this movie	0.570	✓
Three Colors: Red (1994) - Drama	24% of users who share your interests liked this movie	0.230	-

TABLE 4.18

Top-5 recommended items for Sample User B. Movies are ranked in descending order of ratings (EMF Model). **The explanations are shown in Figure 4.13**. **Unpopular items have popularity below** 0.08.

Top Recommended Movies - Genre	**Explanation**	**Expl. Score**	**Diversity**
Three Colors: Red (1994) - Drama	24% of users who share your interests liked this movie	0.230	-
Three Colors: White (1994) - Drama	14% of users who share your interests liked this movie	0.105	-
Butch Cassidy and the Sundance Kid (1969) - Action/Comedy/Western	38% of users who share your interests liked this movie	0.600	-
Adventures of Pinocchio, The (1996) - Adventure/Children's	68% of users who share your interests liked this movie	0.010	-
Shining, The (1980) - Horror	0% of users who share your interests liked this movie	0.450	-

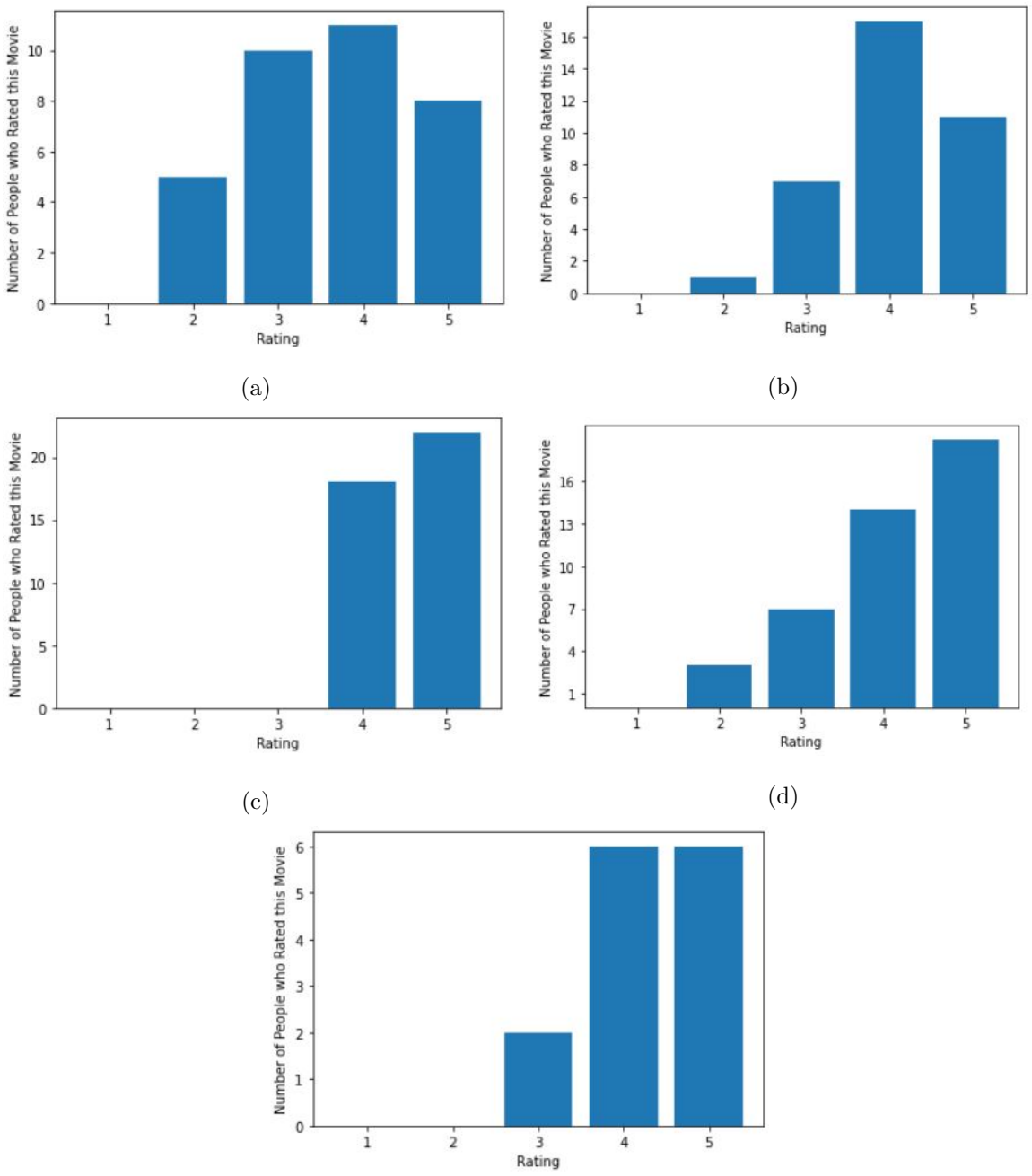


Figure 4.12: NSE explanations for movies recommended to Sample user B based on Table 4.17 using CEMF. The movies are shown in the order of their recommendations from (a)-(e).

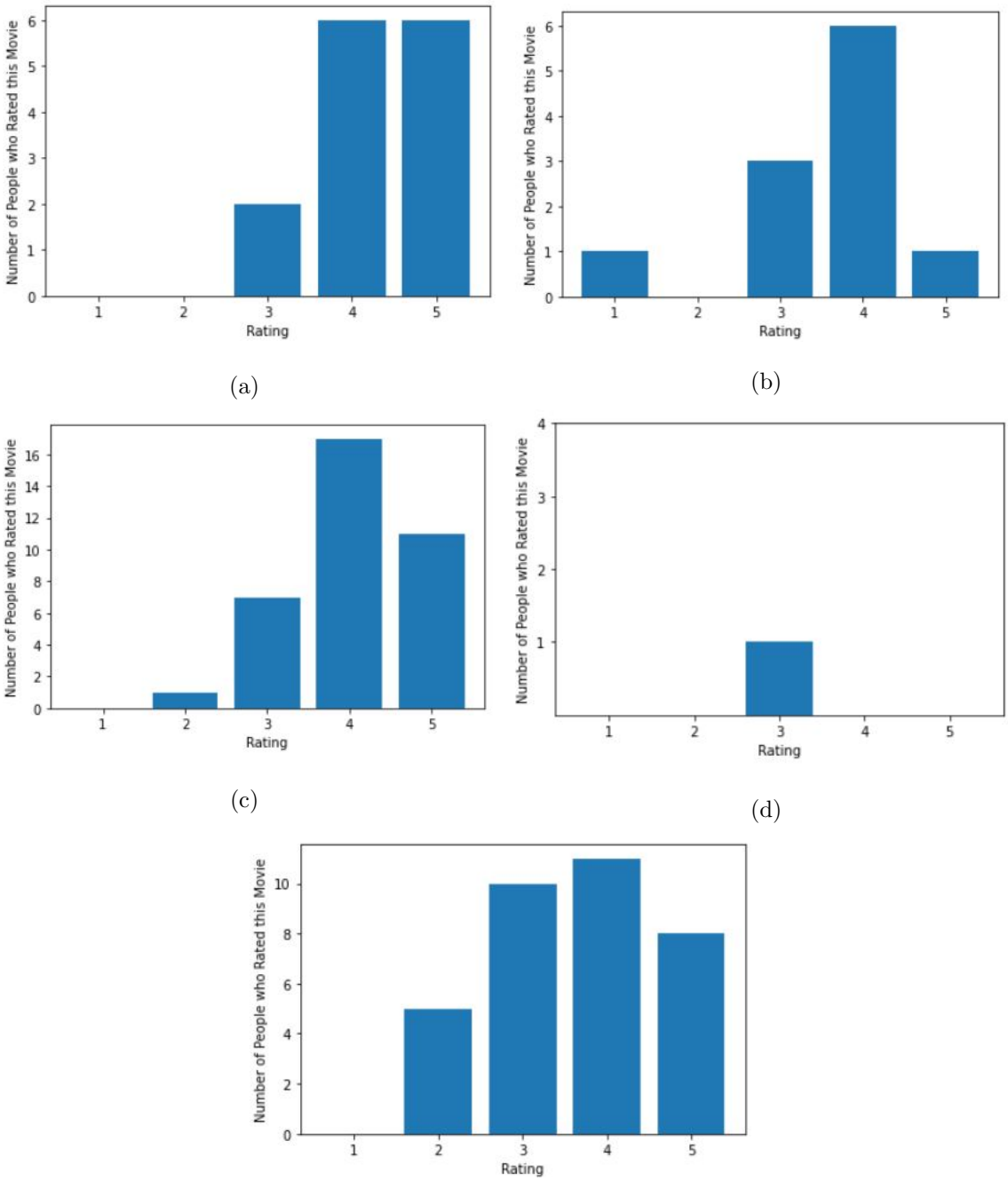


Figure 4.13: NSE explanations for movies recommended to Sample user B based on Table 4.18 using EMF. The movies are shown in the order of their recommendations from (a)-(e).

4.7 Analysis of Results

Based on the results, our proposed method (CEMF) can recommend more explainable items in the top-n recommendation considering both the MEP and MEP-TR metrics. In terms of the RMSE metric, CEMF has the lowest error. The significance tests also show that CEMF's better performance was significant at p-value < 0.05 for MEP, MEP-TR, and RMSE for all methods. It was also significant for nDCG@10 and MAP@10 when compared with MF and GMF.

Tables 4.19 and 4.20 show the results of comparing CEMF with the baseline models on the test data. The best results are shown in bold.

TABLE 4.19

Results on the test data. The best results are in Bold.

Models	**RMSE**	**nDCG**	**MAP**
MF	0.2326 ± 0.0005	0.8950 ± 0.0011	0.8451 ± 0.0012
EMF	0.2314 ± 0.0001	0.8962 ± 0.0007	0.8464 ± 0.0006
CEMF	$\mathbf{0.2308 \pm 0.0003}$	0.8968 ± 0.0008	0.8466 ± 0.0008
GMF	0.2353 ± 0.0017	0.8887 ± 0.0017	0.8372 ± 0.0019

TABLE 4.20

Results on the test data. The best results are in Bold ($\theta > 0.1$).

Models	**MEP**	**MEP-TR**
MF	0.7714 ± 0.0018	0.6632 ± 0.0017
EMF	0.7772 ± 0.0007	0.6693 ± 0.0012
CEMF	$\mathbf{0.7825 \pm 0.0005}$	$\mathbf{0.6724 \pm 0.0007}$
GMF	0.7594 ± 0.0198	0.6465 ± 0.0257

TABLE 4.21

Results on the test data. The best results are in Bold ($\theta > 0.4$).

Models	**MEP**	**MEP-TR**
MF	0.23517 ± 0.0031	0.20617 ± 0.0019
EMF	0.24416 ± 0.0008	0.2122 ± 0.0006
CEMF	$\mathbf{0.25498 \pm 0.0006}$	$\mathbf{0.21871 \pm 0.0006}$
GMF	0.22345 ± 0.025	0.19572 ± 0.019

4.8 Summary

In this chapter, we presented experimental results to evaluate the proposed CEMF method. We compared our approach to the baseline methods EMF [5], MF [2] and Generalized Matrix Factorization (GMF) [36]. Our proposed method showed improvements in terms of accuracy and explainability. Apart from the explainability metric that was previously proposed by [5], we used a new explainability metric called MEP-TR which adds more restriction to find the number of recommended items that are explainable by also considering their ground truth rank. We also showed that the CEMF model can recommend more diverse (unpopular) items in the top-n recommendation.
Our experiments helped to answer the following questions:

1. Do the CEMF recommendations have higher explainability than the baseline methods?
2. Is the CEMF model more accurate than the baseline methods?
3. Do the CEMF recommendations have less popularity bias than the baseline methods?
4. Do CEMF recommendations recommend "simultaneously" more accurate and explainable items in its top-n list?

Our statistical analysis shows that CEMF's performance metrics were favorable (higher), with a significance level that is below the cut-off value of 0.05, for the explainability, diversity, and accuracy metrics, particularly compared to the most competitive (because it is also explainable) comparable (because it is also latent-factor based) baseline, EMF. Therefore, we reject the null-hypothesis in favor of the alternative hypothesis and conclude that CEMF has a significant performance advantage in terms of explainability, accuracy and diversity.

CHAPTER 5

CONCLUSION

Many accurate models in Collaborative Filtering recommendation systems are black-boxes and there is a lack of transparency in the model's predictions. In attempting to overcome the concern about the opacity of black box models, we proposed a Cosine-based Explainable Matrix Factorization model (CEMF) which incorporates the cosine distance in the explainability penalty term. The model is more explainable and by explanation, we mean that the top recommendations have higher Explainability scores where Explainability scores are measured according to the NSE style. The NSE explainability style has been validated in [4, 7, 61]. The Explainability scores were also previously validated in user studies that found them correlated with user satisfaction with explanations [4].

We compared our results with three competitive baseline models that are all in the same family of latent factor techniques. Matrix Factorization (MF) and Generalized MatrixFactorization (GMF) are two state of the art non-explainable recommendations, while Explainable Matrix Factorization (EMF) is a state of the art explainable recommendation technique.

We computed three different metrics to evaluate the accuracy and ranking of the model and two explainable metrics to evaluate the explainability of the model. In addition, we evaluated our proposed model using the diversity metric to see if the proposed model can recommend more diverse and unpopular items in its top-n recommendation. CEMF achieved a significantly improved explainability without compromising accuracy. In terms of explainability, at low explainability thresholds, it performed similarly to the other baseline models, whereas at higher thresholds, CEMF started gaining a significant advantage. This means that at higher thresholds, CEMF can recommend more explainable items in its top-n

recommendations. It is worth noting that we evaluated the model for each metric for a wide range of tested hyper-parameters, and there was no sacrifice in terms of MAP, RMSE and nDCG to pay for the increase in explainability of CEMF.

We performed the t-test for each metric to determine if there is a significant difference between the means of our proposed model and each of the baseline methods. The difference in MEP, MEP-TR and RMSE for CEMF and all baseline models was significant. The difference in MAP@10 and nDCG@10 for CEMF was signifianct when compared with MF and GMF.

In addition, recommender systems suffer from popularity bias, which means that the system recommends more popular items and ignores the unpopular ones. Apart from the explainability metric that was the main focus of this research, the CEMF model could recommend more diverse items in its top-10 recommendations compared with the EMF model. We also evaluated our proposed model in terms of diversity and the difference in diversity metric for the CEMF model was significant when compared with GMF and EMF. Although, this difference was not significant when the CEMF was compared with MF, MF is not an explainable model.

New avenues to explore in the future include:

1. Adding data from a different domain such as reviews. We used the collaborative filtering technique that uses only the ratings data. However, We would want to emphasize that this type of explanation may not be perfect for everyone and some users may expect to see more information about the content of the recommended items. For instance, information about the genre, actors or actresses. These type of information may hurt the transparency but may be appealing to some users.

2. Conducting online user studies to evaluate the user satisfaction. For instance, we can study if different users are satisfied with a reduction in popularity bias, as well as improved explainability.

3. Investigating additional explanation styles such as semantics used in [8]. For instance,

the explanations can be mined from the semantic knowledge graph about users, items and semantic attributes which can in turn overcome both the black-box and the cold-start issues in recommendation systems. In addition, users may have different explanation style preferences. Using tag-boosted techniques used in [76] can provide users with tag-based explanations that can be considered as an explanation style for each user's top-n recommendation.

REFERENCES

[1] Behnoush Abdollahi, "Accurate and justifiable: new algorithms for explainable recommendations.," 2017.

[2] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[3] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 83–92.

[4] Behnoush Abdollahi and Olfa Nasraoui, "Using explainability for constrained matrix factorization," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 79–83.

[5] Behnoush Abdollahi and Olfa Nasraoui, "Explainable matrix factorization for collaborative filtering," in *Proceedings of the 25th International Conference Companion on World Wide Web*, 2016, pp. 5–6.

[6] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[7] Mustafa Bilgic and Raymond J Mooney, "Explaining recommendations: Satisfaction vs. promotion," in *Beyond Personalization Workshop, IUI*, 2005, vol. 5, p. 153.

[8] Mohammed Alshammari, Olfa Nasraoui, and Scott Sanders, "Mining semantic knowledge graphs to add explainability to black box recommender systems," *IEEE Access*, vol. 7, pp. 110563–110579, 2019.

[9] Dokyun Lee and Kartik Hosanagar, "Impact of recommender systems on sales volume and diversity," 2014.

[10] Christian Desrosiers and George Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender systems handbook*, pp. 107–144. Springer, 2011.

[11] Satchidananda Dehuri, *Intelligent Techniques in Recommendation Systems: Contextual Advancements and New Methods: Contextual Advancements and New Methods*, IGI Global, 2012.

[12] Panagiotis Adamopoulos and Alexander Tuzhilin, "On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 153–160.

[13] Zeinab Abbassi, Sihem Amer-Yahia, Laks VS Lakshmanan, Sergei Vassilvitskii, and Cong Yu, "Getting recommender systems to think outside the box," in *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 285–288.

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[15] Paul Resnick and Hal R Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[16] Miha Grčar, Dunja Mladenič, Blaž Fortuna, and Marko Grobelnik, "Data sparsity issues in the collaborative filtering framework," in *International Workshop on Knowledge Discovery on the Web*. Springer, 2005, pp. 58–76.

[17] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[18] Zi-Ke Zhang, Chuang Liu, Yi-Cheng Zhang, and Tao Zhou, "Solving the cold-start problem in recommender systems with social tags," *EPL (Europhysics Letters)*, vol. 92, no. 2, pp. 28002, 2010.

[19] Rui Chen, Qingyi Hua, Yan-Shuo Chang, Bo Wang, Lei Zhang, and Xiangjie Kong, "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018.

[20] SongJie Gong, HongWu Ye, and HengSong Tan, "Combining memory-based and model-based collaborative filtering in recommender system," in *2009 Pacific-Asia Conference on Circuits, Communications and Systems*. IEEE, 2009, pp. 690–693.

[21] Xiaoyuan Su and Taghi M Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[22] Xin Luo, Yunni Xia, and Qingsheng Zhu, "Incremental collaborative filtering recommender based on regularized matrix factorization," *Knowledge-Based Systems*, vol. 27, pp. 271–280, 2012.

[23] Jesus Bobadilla and Francisco Serradilla, "The effect of sparsity on collaborative filtering metrics," in *Proceedings of the Twentieth Australasian Conference on Australasian Database-Volume 92*. Citeseer, 2009, pp. 9–18.

[24] Yue Shi, Martha Larson, and Alan Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–45, 2014.

[25] Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, and Yi-Hsuan Yang, "Collaborative similarity embedding for recommender systems," in *The World Wide Web Conference*, 2019, pp. 2637–2643.

[26] Manh Cuong Pham, Yiwei Cao, Ralf Klamma, and Matthias Jarke, "A clustering approach for collaborative filtering recommendation using social network analysis.," *J. UCS*, vol. 17, no. 4, pp. 583–604, 2011.

[27] Koji Miyahara and Michael J Pazzani, "Collaborative filtering with the simple bayesian classifier," in *Pacific Rim International conference on artificial intelligence*. Springer, 2000, pp. 679–689.

[28] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2nd ACM conference on Electronic commerce*, 2000, pp. 158–167.

[29] Andriy Mnih and Russ R Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[30] Michael J Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial intelligence review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[31] Hamidreza Koohi and Kourosh Kiani, "A new method to find neighbor users that improves the performance of collaborative filtering," *Expert Systems with Applications*, vol. 83, pp. 30–39, 2017.

[32] Xiaoyao Zheng, Yonglong Luo, Liping Sun, Xintao Ding, and Ji Zhang, "A novel social network hybrid recommender system based on hypergraph topologic structure," *World Wide Web*, vol. 21, no. 4, pp. 985–1013, 2018.

[33] Thanh Tran, Kyumin Lee, Yiming Liao, and Dongwon Lee, "Regularizing matrix factorization with user and item embeddings for recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 687–696.

[34] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Procedia Computer Science*, vol. 49, pp. 136–146, 2015.

[35] Zhijun Zhang and Hong Liu, "Application and research of improved probability matrix factorization techniques in collaborative filtering," *Int J control autom*, vol. 7, no. 8, pp. 79–92, 2014.

[36] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[37] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Application of dimensionality reduction in recommender system-a case study," Tech. Rep., Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[38] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[39] Arkadiusz Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, 2007, vol. 2007, pp. 5–8.

[40] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[41] Marvin Minsky and Seymour A Papert, *Perceptrons: An introduction to computational geometry*, MIT press, 2017.

[42] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al., "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, `http://www.deeplearningbook.org`.

[44] Yuanxin Ouyang, Wenqi Liu, Wenge Rong, and Zhang Xiong, "Autoencoder-based collaborative filtering," in *International Conference on Neural Information Processing*. Springer, 2014, pp. 284–291.

[45] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[46] Florian Strub, Romaric Gaudel, and Jérémie Mary, "Hybrid recommender system based on autoencoders," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 11–16.

[47] Florian Strub and Jeremie Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs," 2015.

[48] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, pp. 153–162.

[49] Shuai Zhang, Lina Yao, Xiwei Xu, Sen Wang, and Liming Zhu, "Hybrid collaborative recommendation via semi-autoencoder," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 185–193.

[50] Jérôme Tubiana, Simona Cocco, and Rémi Monasson, "Learning compositional representations of interacting systems with restricted boltzmann machines: Comparative study of lattice proteins," *Neural computation*, vol. 31, no. 8, pp. 1671–1717, 2019.

[51] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791–798.

[52] Francesco Ricci, Lior Rokach, and Bracha Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*, pp. 1–35. Springer, 2011.

[53] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma, "How good your recommender system is? a survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2019.

[54] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos, "Moviexplain: a recommender system with explanations," in *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 317–320.

[55] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth, "Thinking positively-explanatory feedback for conversational recommender systems," in *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop*, 2004, pp. 115–124.

[56] David McSherry, "Explanation in recommender systems," *Artificial Intelligence Review*, vol. 24, no. 2, pp. 179–197, 2005.

[57] Rashmi Sinha and Kirsten Swearingen, "The role of transparency in recommender systems," in *CHI'02 extended abstracts on Human factors in computing systems*, 2002, pp. 830–831.

[58] Alexander Felfernig and Bartosz Gula, "An empirical study on consumer behavior in the interaction with knowledge-based recommender applications," in *The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International*

Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06). ieee, 2006, pp. 37–37.

[59] Nava Tintarev and Judith Masthoff, "A survey of explanations in recommender systems," in *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 2007, pp. 801–810.

[60] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl, "Is seeing believing? how recommender system interfaces affect users' opinions," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003, pp. 585–592.

[61] Jonathan L Herlocker, Joseph A Konstan, and John Riedl, "Explaining collaborative filtering recommendations," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000, pp. 241–250.

[62] Raymond J Mooney and Loriene Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of the fifth ACM conference on Digital libraries*, 2000, pp. 195–204.

[63] Daniel Billsus and Michael J Pazzani, "A personal news agent that talks, learns and explains," in *Proceedings of the third annual conference on Autonomous Agents*, 1999, pp. 268–275.

[64] Nava Tintarev and Judith Masthoff, "Designing and evaluating explanations for recommender systems," in *Recommender systems handbook*, pp. 479–510. Springer, 2011.

[65] Ludovik Coba, Panagiotis Symeonidis, and Markus Zanker, "Personalised novel and explainable matrix factorisation," *Data & Knowledge Engineering*, vol. 122, pp. 142–158, 2019.

[66] Yongfeng Zhang and Xu Chen, "Explainable recommendation: A survey and new perspectives," *arXiv preprint arXiv:1804.11192*, 2018.

[67] Eyal Shulman and Lior Wolf, "Meta decision trees for explainable recommendation systems," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 365–371.

[68] Jonghyun Choi, Hyunjong Cho, Jungsuk Kwac, and Larry S Davis, "Toward sparse coding on cosine distance," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 4423–4428.

[69] Sahar Sohangir and Dingding Wang, "Improved sqrt-cosine similarity measurement," *Journal of Big Data*, vol. 4, no. 1, pp. 1–13, 2017.

[70] Ludovik Coba, *Ratings in Recommender Systems: Decision Biases and Explainability*, Ph.D. thesis, -, 2020.

[71] Shun-ichi Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[72] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher, "Managing popularity bias in recommender systems with personalized re-ranking," in *The thirty-second international flairs conference*, 2019.

[73] Yoon-Joo Park and Alexander Tuzhilin, "The long tail of recommender systems and how to leverage it," in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 11–18.

[74] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1661–1670.

[75] Pablo Castells, Saúl Vargas, and Jun Wang, "Novelty and diversity metrics for recommender systems: choice, discovery and relevance," 2011.

[76] Olurotimi Seton, "Multi-style explainable matrix factorization techniques for recommender systems," 2021.

CURRICULUM VITAE

NAME: Pegah Sagheb Haghighi

ADDRESS: Computer Science and Engineering Department
J.B Speed School of Engineering
University of Louisville
Louisville, KY 40292
United States of America.

EDUCATION:

Ph.D., Computer Science & Engineering, August 2021
University of Louisville, *Louisville, KY, USA.*
M.Eng. in Information Tehcnology, September 2013
University of Guilan, *Guilan, Iran*
B.Sc in Information Technology, May 2008
University of Mumbai, *Mumbai, India*

WORK EXPERIENCE:

Data Science Intern, *Spacee*, Addison, TX, USA
Data Science Intern, *Arrow Electronics*, Denver, CO, USA
Teaching Assistant, *University of Louisville*, KY, USA

AWARDS:

Doctoral Dissertation Completion Award, 2021
Grad Cohort Scholarship Award, 2021
CSE Doctoral Award, 2020
Richard Tapia Celebration of Diversity in Computing Scholarship, 2020
ICLR Travel Award, 2019
Raymond I. Fields Award, 2018
Grace Hopper Conference Scholarship Award, 2018
University of Louisville Doctoral Fellowship, 2016

PUBLICATIONS:

1. **Haghighi, P. S.**, Seton, O., and Nasraoui, O. (2019). *An explainable autoencoder for collaborative filtering recommendation.* arXiv preprint arXiv:2001.04344
2. Khmaissia, F., **Haghighi, P. S.**, Jayaprakash, A., Wu, Z., Papadopoulos, S., Lai, Y., and Nguyen, F. T. (2020). *An unsupervised machine learning approach to assess the zip code level impact of covid-19 in nyc.* arXiv preprint arXiv:2006.08361.