Multimedia Tools and Applications (2021) 80:18553–18581
https://doi.org/10.1007/s11042-021-10529-4

Embedding metadata using deep collaborative filtering to address the cold start problem for the rating prediction task

Ravi Nahta[1] · **Yogesh Kumar Meena**[1] · **Dinesh Gopalani**[1] · **Ganpat Singh Chauhan**[1]

Received: 6 June 2020 / Revised: 26 December 2020 / Accepted: 5 January 2021 /
Published online: 18 February 2021

Abstract

In recent years, deep learning has yielded success in many research fields including machine translation, natural language processing, computer vision, and social network filtering. The area of deep learning in the recommender system is flourishing. Previous research has relied on incorporating metadata information in various application domains using deep learning techniques to achieve better recommendation accuracy. The use of metadata is desirable to address the cold start problem and better learning the user-item interaction, which is not captured by the user-item rating matrix. Existing methods rely on fixed user-item latent representation and ignore the metadata information. It restricts the model performance to correctly identify actual latent vectors, which results in high rating prediction error. To tackle these problems, we propose a generalized recommendation model named Meta Embedding Deep Collaborative Filtering (MEDCF), which inputs user demographics and item genre as metadata features together with the rating matrix. The proposed framework primarily comprises of Generalized Matrix Factorization (GMF), Multilayer Perceptron (MLP), and Neural Matrix Factorization (NeuMF) methods. GMF is applied to the rating matrix, whereas MLP is applied to metadata. Using NeuMF, the outputs for GMF and MLP are then concatenated and input to a neural network for rating prediction. To prove the effectiveness of proposed model, two metrics are used, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The MEDCF model is experimented on MovieLens and Amazon Movies datasets showing a significant improvement over the baseline methods.

✉ Ravi Nahta
2018rcp9077@mnit.ac.in

Yogesh Kumar Meena
ymeena.cse@mnit.ac.in

Dinesh Gopalani
dgopalani.cse@mnit.ac.in

Ganpat Singh Chauhan
2018rcp9063@mnit.ac.in

[1] Department of Computer Science and Engineering, Malaviya National Institute of Technology, Jaipur, Rajasthan, India

Keywords Recommender systems · Collaborative filtering · Cold start problem · Neural networks · Matrix factorization · Metadata

1 Introduction

In the era of information explosion, people enjoy the convenience of huge data. However, they face the problem of information overload. Recommender systems are a subclass of information filtering systems that play an important role in alleviating this problem by predicting items (or ratings for items) that the user may like. Recommender systems have been adopted by many online services such as social media sites, online news, and e-commerce. In general, there are two methods for recommender systems: Content-Based (CB) techniques and Collaborative Filtering (CF) based techniques. Amongst both, CF is widely used as it predicts (filtering) the interests of a user by collecting preferences or taste information from many users (collaborating). Matrix Factorization (MF) [23, 32] is a popular CF method that transforms rating matrix into a low dimensional concept space which is often referred as user-item latent (hidden) space. The user's interaction on an item is then modeled as the inner product of their latent vectors. The hybrid model is used to take advantage of both the techniques and is found to be more successful than individuals in literature.

MF techniques have been popularized by the Netflix Prize competition [3] and are widely applied to movie recommendation systems. Through time, MF techniques have evolved from Singular Valued Decomposition (SVD) to latent factor models to generalized Factorization Machines (FM) [45]. However, the MF technique's main problem is that it identifies only linear interaction between user and item latent features which may not be adequate to model the actual interaction between them. Another problem with MF algorithms is the well known cold start problem for new users and items. Many pioneering studies have been applied to resolve the above issues using deep learning approaches including Neural Collaborative Filtering (NCF) [22], NCF model with an interaction-based neighborhood [2], neural rating prediction [58], and autoencoder based recommendation [48]. However, when it comes to the CF effect for rating prediction, these studies still use a fixed element-wise product between user and item latent features (vectors).

Cold start problem concerns the issue that the system does not draw any inferences for users or items about which adequate information has not yet been collected. Metadata information can help in alleviating this problem. Metadata information is useful because of the interaction matrix sparsity and it also helps to identify each user's interest in each item. To deal with the cold start problem, hybrid methods were used. A hybrid approach uses metadata as content information for user-item pairs while collaboratively learning these pairs using neural networks function. Applying a neural network learning on metadata information is more prominent than using the MF technique. However, the metadata information (features) embedding using neural networks is getting less investigation in recommendation literature. Some work has been proposed that incorporates metadata information but ultimately predict ratings using dot product between user and item latent vectors. It is well known that one can approximate a neural network to any continuous function [26]. It motivates us to implement collaborative learning between users and items metadata features using neural networks.

Explicit feedback is used in the form of ratings because it is readily available and can be easily utilized since user satisfaction is easily observed in contrast with implicit feedback where user negative preferences are not easily captured. This paper proposes a Meta

Embedding Deep Collaborative Filtering (MEDCF) model that inputs user-item interaction (preference) and their metadata features and outputs the predicted rating a user would give to an item. The Generalized Matrix Factorization (GMF) model identifies the linear interaction between user and item latent features. Multilayer Perceptron (MLP) learning is utilized on the metadata information to identify the non-linear interaction between the user and item, especially in cold start and data sparsity scenarios. MEDCF model is a generalized framework that lets anyone apply MLP with MF. Using Neural Matrix Factorization (NeuMF), the outputs for GMF and MLP are concatenated and input to a neural network for rating prediction. NeuMF learns user preferences towards an item better. The concatenated output acts as a bias term for ratings and the metadata information to prevent overfitting. This paper focuses on the neural networks to exploit the noisy explicit feedback in the form of ratings and, at the same time, use the metadata information to alleviate the cold start problem.

2 Related work

This section briefly summarizes the deep learning based recommendation system approaches that are mostly related to this work. Here, those papers are encouraged that exploit metadata and other auxiliary information which are very helpful in the recommendation task. Most of the current work on the recommender systems that use Neural Collaborative Filtering (NCF) approaches and exploits the metadata information were described in [13, 22, 34, 59, 68].

Previously, Collaborative Filtering (CF) approaches have been extensively applied to explicit feedback given by the users in the form of ratings and reviews. The same approaches are now being applied to implicit feedback such as the purchasing history of the user, Click-Through Rate (CTR), and the number of times the web page is visited. However, gathering and analyzing implicit feedback is not an easy task, although worthier as it is beneficial in minimizing the cold start problem and better user personalization. CF based methods [2, 22] are more popular than Content-Based (CB) methods [29, 52] as these methods depend on other collaborative users to predict the rating without requiring knowledge-gathering of the current user.

To further improve the CF based methods, auxiliary information in the form of metadata features are used. Metadata [34] represents user and item preferences as the latent (hidden) vector of features, which refers to the embedding of user and item in high dimensional latent space. Meta-Prod2vec [59] model represent items uniquely by finding the item similarity interactions using metadata and its attributes and compute low dimensional item embeddings. Metadata is used to regularize these embeddings. It leads to better recommendation tasks on the music dataset. Exploiting MovieLens and Netflix datasets to encapsulate different metadata information [52] increases the quality of the recommendations. Thereafter, metadata with deep learning for user personalization for the movie rating prediction task using the Word2Vec embedding model [69] has been proposed. Recently, research scholar coauthors and collaborators recommendation tasks based on the knowledge graph embeddings [25] were proposed. Further, scholarly metadata such as citation networks and research publications were also exploited. Researchers [7] utilized different metadata views of the item from the user reviews which reveals the item statistics, user opinions, and the item quality. Recent studies [39, 40, 49, 67] have proposed a deep latent factor model for high-dimensional and sparse (HiDS) rating matrices to estimate patterns in large scale matrix. However, these studies do not take into account the metadata information of

user-item interactions. Both the metadata information and rating matrix are analyzed in this study.

The use of metadata and demographic information has proved useful in the literature of recommendations. Content metadata [29] of the movies with the conventional user ratings results in a correct recommendation for a large number of content items. The proposed method is helpful, especially in sparse ratings and cold start environments. To address the cold start problem, the authors [10] generates user embeddings using reinforcement learning by asking the questions to cold start users. Deep Q Network were introduced that generates interview questions of the form "Do you like λ?", where lambda is taken from the set of all movies to be answered by cold start users in the form of ratings from 0 to 5. The network produces subsequent questions dynamically based on the user's answer. Upon completion of the interview, the question-answer pairs are passed into neural networks that provide user embedding. The user and movie embeddings can be used to predict the movie ratings. To deal with cold start items, metadata information is incorporated to recommend the movie based on the Word2Vec algorithm [69]. With metadata information, the proposed method could effectively suggest a new item that has been rarely used. Thereafter, user/item metadata is incorporated into the Poisson Factorization [8] to deal with popularity within an item and cold start problem. The model learns about the user/item couplings and their metadata. In Poisson Factorization, the user ratings obey gamma distribution. This model is well suited for a scalable recommendation. Further, the researchers have found that there are popularity and demographic biases in recommender systems and when it comes to the evaluation task, the use of demographic information is vital for enhancing recommendation accuracy [11]. Therefore, different demographic groups had different utility from the recommender system. Cold start users only have necessary demographic information without previous movie rating information [4]. Using demographic clusters of different sizes increase the prediction accuracy of movie ratings for the MovieLens dataset. Sometimes, the users based on demographic attributes is partitioned and the k-means clustering algorithm is used to cluster the partitioned users based on the user rating matrix [55]. Compared to the traditional collaborative filtering strategy, it eliminates the expensive computations to classify similar users to predict movies. A simple neighborhood selection technique [43] is proposed emphasizing the metadata groups to enhance recommendations and alleviate the cold start problem. Thereafter, many user-profiling approaches [1] for demographic recommender systems were also been examined. The demographic user profile is mostly made up of three attributes, age, gender, and occupation. For cold start problem, an in-depth study [52] is proposed on using various metadata elements including the title, genre, date of production, and the list of directors and actors so as to improve the quality of the movie recommendations. Demographic data [60] in some cases lead to the generation of more accurate predictions. Thereafter, the researchers have found that demographics such as gender and occupation are very useful in recommendation systems for applications such as personalization services and marketing [50]. It is also observed that users who share similar demographics would probably prefer similar genres. This approach can align latent factors across domains that do not share the same users or the same items, integrating user demographics with latent factors in a single framework.

Text reviews proved to be useful in literature, especially in the cold start environment, to achieve better representation learning of user interest and item features. HFT (Hidden Factors as Topics) [42] model combined rating and review information leading to a higher

recommendation accuracy for items with few ratings to address the cold start problem. A new topic model [6] is proposed that gives varying aspect importance using different attention weights by targeting the review text to learn better user preferences and item characteristics. MMALFM [5] employed item images and review texts together with the ratings to tackle non-transparency and cold-start problems for individual user-item pairs. Attentive Aspect-based Recommendation Model (AARM) [17] is proposed to tackle the problems of sparsity in common aspects and the static user's preference on aspects concerning different items. Dual Attention Mutual Learning (DAML) [37] is a novel dual attention mutual learning between ratings and reviews for an item recommendation, which enhances the interpretability of the recommendation model.

The metric learning method and Graph Convolution Network (GCN) based approaches are rapidly developed in recent two years and have become new state-of-the-art for collaborative filtering. The benefit of metric learning is that the issue of dot product similarity will naturally be resolved, which is adopted by matrix factorization (MF) based recommendation models but did not fulfill the triangle inequality property. The GCN based recommendation approaches can exploit and refine the user-item interaction graph structure by propagating embeddings on it. GraphSAGE [19] is an inductive framework that exploits node attributes/features information to efficiently generate node embeddings for previously unseen data for better representation learning on large graphs. Collaborative Metric Learning (CML) [27] learns a combined user-item metric learning to encode users' interests along with the user-user and item-item similarity. Multimodal Attentive Metric Learning (MAML) [36] method employs an attention neural network for each user-item pair to capture diverse user preferences for several items. The item's multimodal attributes are used to capture the attention of users to various aspects of this item and then integrate attentions in a metric-based learning method that predicts user preferences for the item. Neural Graph Collaborative Filtering (NGCF) [63] is a recommendation framework that uses the user-item graph structure by propagating embeddings on it. Specifically, the user-item interactions, which is the bipartite graph structure, are integrated into the embedding process. Light Graph Convolution Network (LightGCN) [21] includes the main component in GCN, which is neighborhood aggregation for collaborative filtering. The improvement is about 16.0% on an average over the state-of-the-art NGCF based framework.

The differences among most of the literature were in their model architecture to find user and item embeddings. One of the significant shortcomings is that most of the proposed models compute the element-wise product on the embeddings to predict the final rating constraining them to learn non-linear interactions between the user and the item. Our proposed method utilizes neural network learning to determine these interactions with user-item metadata features showing state-of-the-art results.

3 Problem formulation

In this section, problem formulation as a preliminary task is represented. For clarity, Table 1 describes the notations list that will be used in this work. In this work, user metadata r denotes user demographic information that contains gender and occupation features. Similarly, item metadata s denotes item (in this work items are the movies) genre feature. Gender feature has two categories, the occupation feature has 21 categories, and the genre feature has 18 categories.

Table 1 Notation list

A	Number of users
B	Number of items
P'	Latent factor matrix for users metadata
Q'	Latent factor matrix for items metadata
y_{ui}	Actual rating of user u on item i
$\hat{y}_{ui}$	Predicted rating of user u on item i
y_{ui}^{G}	Actual binarized score of user u on item i for GMF
$\hat{y}_{ui}^{G}$	Predicted score of user u on item i for GMF
y_{rs}^{M}	Actual binarized score of user metadata r on item metadata s for MLP
$\hat{y}_{rs}^{M}$	Predicted score of user metadata r on item metadata s for MLP
Y	User-item binary interaction matrix
Y'	User-item metadata binary interaction matrix
p_u	Latent vector for user u
q_i	Latent vector for item i
p'_r	Latent vector for user metadata r
q'_s	Latent vector for item metadata s
v_u^U	Binarized one-hot encoded feature vector of user u for GMF
v_i^I	Binarized one-hot encoded feature vector of item i for GMF
w_r^R	Binarized one-hot encoded feature vector of user metadata r for MLP
w_s^S	Binarized one-hot encoded feature vector of item metadata s for MLP
ϕ_x	Mapping function for the x-th neural network layer
W	Weight matrix of neural network layer
h	Weights of the output layer
b	Bias vector
a	Activation function

Subscript u and subscript i denotes a user and an item, respectively. Subscript r and subscript s denotes the user metadata and the item metadata, respectively. Superscript G and superscript M denotes GMF and MLP, respectively

3.1 Matrix factorization

Let u and i be the user and the item, respectively. Let the number of users and items be A and B, respectively. Let us define an interaction matrix $Y \in \mathbb{R}^{A \times B}$ between the user and the item from users implicit feedback as:

$$y_{ui} = \begin{cases} 1, & \text{if an interaction between user } u \text{ and item } i \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here a value of $y_{ui} = 1$ indicates an interaction between the user u and the item i. Similarly, $y_{ui} = 0$ indicates user u has no interaction with item i. This representation using implicit feedback has a problem of inferring user u positive feedback (interest) towards item i; however, negative feedback for the same item cannot be inferred because 0 indicates missing values or unobserved entries. Therefore that poses an inherent lack of negative interest.

Every user and item has its corresponding latent vector of real-valued features. Let p_u and q_i denotes the latent vector for user u and item i, respectively. The predicted rating $\hat{y}_{ui}$ can be obtained by the inner dot product of p_u and q_i as:

$$\hat{y}_{ui} = f\,(u, i|p_u, q_i) = p_u^T q_i = \sum_{k=1}^{K} p_{uk} q_{ik}, \tag{2}$$

where K symbolizes the number of dimensions of the embedded latent space, assuming that dimensions of the latent space are independent of each other and have the same weight when linearly combining the interaction of latent vectors of users and items. Therefore Matrix Factorization (MF) is regarded as the *linear* model of latent factors.

The function 'inner dot product' between the user-item latent vectors can limit the expressive power of MF resulting in a ranking loss [22]. A common solution is to increase latent dimensions K. Still, the generalization of the model is not achieved due to the over-fitting of the data, especially in the sparse environment [45]. So as a consequence of the above limitation, the *non-linear* user-item interaction function using MLP is learnt.

3.2 Learning from content-based metadata information of users and items

Let r and s be the user metadata and the item metadata, respectively. Let us define an interaction matrix $Y' \in \mathbb{R}^{A \times B}$ between the user metadata and the item metadata from users implicit feedback as:

$$y_{rs} = \begin{cases} 1, & \text{if interaction (user metadata } r\text{, item} \\ & \text{metadata } s\text{) is observed;} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Here a value of $y_{rs} = 1$ indicates an interaction between the user metadata r and the item metadata s. Similarly, $y_{rs} = 0$ indicates r has no interaction with s. This representation using implicit feedback has a problem that inferring user metadata r positive feedback towards item metadata s is trivial, however, negative feedback for the same item metadata cannot be inferred because 0 indicates missing values or unobserved entries.

Recommendation problem having implicit feedback is defined as estimating the scores of unobserved entries in Y', that are needed to rank the items. Model-based techniques assume data to be predicted using the model itself. Formally, it can be described as learning $\hat{y}_{rs} = f\,(r, s|\theta)$, where $\hat{y}_{rs}$ denotes predicted score of y_{rs} and θ indicates the parameters of the model and f indicates the function which maps the parameters of the model to $\hat{y}_{rs}$ which is referred to as *interaction function*.

Estimating model parameters θ requires existing machine learning approaches that optimize a certain cost function. Generally two types of cost functions are used : pointwise loss [23, 28] and pairwise loss [46, 53]. Pointwise loss is calculated for regression problems with usually explicit feedbacks [32, 70] by minimizing the Mean Squared Error (MSE) between $\hat{y}_{rs}$ and y_{rs}. To handle the missing entries in the interaction matrix, these entries are treated as negative feedbacks [23], or these are sampled as negative examples. Pairwise learning [46, 66] has the idea that observed interaction items must be ranked higher than the non-observed missing entries. Therefore, the margin between the observed score $\hat{y}_{rs}$ and the unobserved neighborhood input $\hat{y}_{rj}$ is maximized by pairwise learning instead of minimizing the error between $\hat{y}_{rs}$ and y_{rs}.

The MEDCF framework as shown in Fig. 1 calculates function f on the metadata binarized information using neural networks to predict $\hat{y}_{rs}$. Therefore it implicitly supports both pointwise as well as pairwise learning.

4 Proposed model

The model is shown in Fig. 2 as a layered neural network architecture. It comprises of three modules named as Generalized Matrix Factorization model (GMF), Multilayer Perceptron (MLP), and Neural Matrix Factorization (NeuMF). In the GMF module, the interaction matrix is considered as the rating matrix and apply the MF technique using the neural network framework. It computes the inner dot product between the user and item latent vectors to obtain the predicted score $\hat{y}_{ui}^{G}$ (G stands for GMF). In the MLP module, neural network architecture is applied on metadata information of users and items to obtain the predicted score $\hat{y}_{rs}^{M}$ (M stands for MLP). Metadata information is useful because of the sparsity of the interaction matrix and it also helps to identify each user's interest towards each item. Finally, in the NeuMF module, $\hat{y}_{ui}^{G}$ and $\hat{y}_{rs}^{M}$ is concatenated in the NeuMF layer. The concatenated output is fed in a small neural network to learn further non-linear interactions between user and item to obtain the final predicted rating $\hat{y}_{ui}$.

5 Methodology

In this section, the Meta Embedding Deep Collaborative Filtering (MEDCF) framework is described first. Next, the Generalized Matrix Factorization (GMF) method under the MEDCF framework is described. Next, an instance of the MEDCF framework is presented to explore neural networks for Collaborative Filtering (CF) using a Multilayer Perceptron (MLP) that learns the user-item metadata interaction function. At last, the Neural Matrix Factorization (NeuMF) method is described that integrates MLP and GMF under a single framework. NeuMF takes advantage of the linearity of GMF and non-linearity of MLP for modeling user-item non-linear interactions.

5.1 Initial user representation learning using MEDCF framework

To model a user-item metadata interaction y_{rs} and to take advantage of fully connected neural network based CF, the MEDCF framework is used as neural network architecture, as

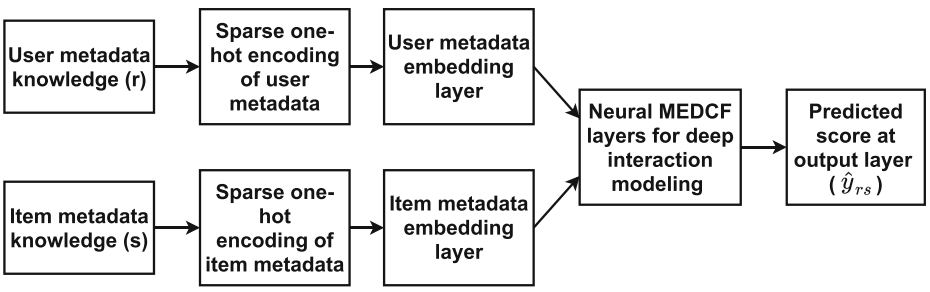

Fig. 1 Meta embedding deep collaborative filtering framework

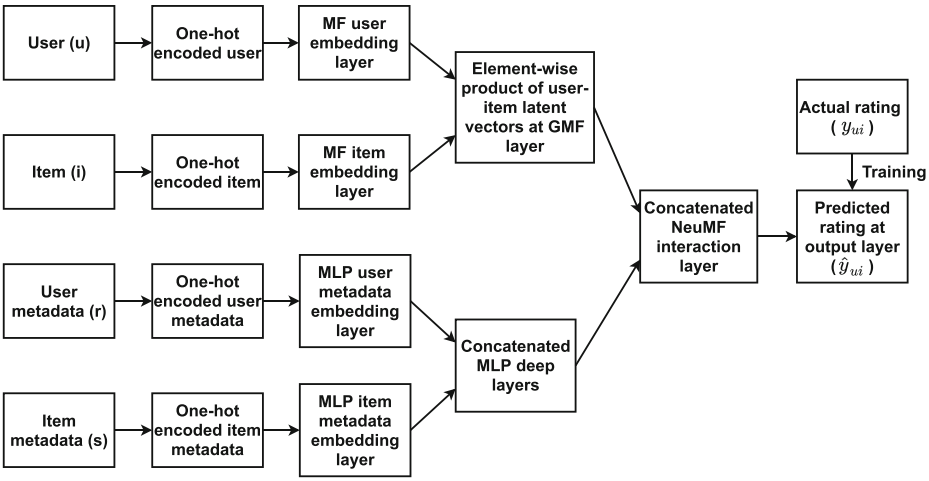

Fig. 2 Neural matrix factorization model under MEDCF framework (a layered neural network view)

shown in Fig. 1 where one layer output feeds as input to the next layer of the neural network. The bottom input layer comprises of feature vector w_r^R and feature vector w_s^S that represents binarized one-hot encoded feature vector of user metadata r and item metadata s, respectively. This input layer can be customized in a variety of ways and applications for modeling users and items metadata features such as content-based, context-aware, and neighborhood-based data. NCF [22] focuses on the collaborative framework where the binarized one-hot encoded version is used as the input features of user and item. It is a generic and most straightforward representation possible that can be customized to any input feature vector. The one-hot encoding vector representation is customized to include content-based metadata information of user and item to address the cold start problem. The user's demographic information and item's genre information is taken as the content features. The input user-item feature vector corresponds to binarized values of the user-item metadata information.

An embedding layer is at the top of the input layer. It is a fully connected layer that transforms the one-hot encoded sparse vector representation into a dense vector representation. These embedded vector representations of users and items metadata signify the latent features. These features are then introduced into the neural network architecture as MEDCF layers. It transforms these features into further low dimensional latent features to learn the complex non-linear interaction between the user and the item metadata. Each layer can be customized to represent a more complex hidden interaction. The last hidden X-th layer dimension determines the model's capability. The final output layer computes the predicted score $\hat{y}_{rs}^M$. Training is executed by minimizing the pointwise error between predicted value $\hat{y}_{rs}^M$ and the desired value y_{rs}^M. Another way of training can be pairwise loss learning, such as Bayesian Personalized Ranking (BPR) [46] loss. However, at this moment, pairwise learning is ignored as main motive is on neural network modeling approach.

Mathematically, MEDCF framework can be formulated as:

$$\hat{y}_{rs}^M = f\left(P'^T w_r^R, Q'^T w_s^S | P', Q', \theta_f\right), \tag{4}$$

where $P' \in \mathbb{R}^{A \times K_1}$ and $Q' \in \mathbb{R}^{B \times K_2}$ symbolizes latent vector matrix for users and items metadata, respectively. K_1 and K_2 denotes number of latent dimensions for user and item metadata, respectively. θ_f denotes the interaction function f which is the model parameter. Here, f is defined to be a multi-layer neural network and mathematically, it is defined as:

$$f\left(P'^T w_r^R, Q'^T w_s^S\right) = \phi_{out}\left(\phi_X\left(\ldots \phi_2\left(\phi_1\left(P'^T w_r^R, Q'^T w_s^S\right)\right)\ldots\right)\right), \tag{5}$$

where ϕ_{out} and ϕ_x denotes the mapping function for the output layer and the x-th MEDCF layer, respectively, and in total, there are X MEDCF layers.

5.2 Learning MEDCF

Learning the model parameters involves existing pointwise approaches [23, 62] that extensively perform regression problems with a mean squared loss given as:

$$L_{sqr} = \sum_{(r,s)\in Y' \cup Y'^-} \left(y_{rs}^M - \hat{y}_{rs}^M\right)^2, \tag{6}$$

where Y' and Y'^- symbolizes a set of observed and unobserved (negative feedback) interactions in the user-item binary metadata matrix, respectively. The quadratic error can be validated by assuming the examples coming from normal distribution [44]; however, it does not match with implicit feedback because the binarized target value of 1 or 0 indicates whether the user-item metadata interaction occurs or not. So, a probabilistic model is applied for learning MEDCF that deals with the binarized output value.

The implicit input feedback data is in the form of 0 or 1 and the output label y_{rs}^M is 1 if user metadata r has interaction with item metadata s otherwise 0. Considering this, MEDCF is build as a probabilistic model that constrains the model output $\hat{y}_{rs}^M$ into the range of [0,1] denoting how likely user metadata r is relevant to item metadata s, and this can be achieved by using a sigmoid activation function for the output layer ϕ_{out}. With this setting, let us represent likelihood function as:

$$p\left(Y', Y'^-|P', Q', \theta_f\right) = \prod_{(r,s)\in Y'} \hat{y}_{rs}^M \prod_{(r,j)\in Y'^-} \left(1 - \hat{y}_{rj}^M\right), \tag{7}$$

The negative logarithm of the above-mentioned likelihood function is:

$$\begin{aligned} L &= -\sum_{(r,s)\in Y'} log\,\hat{y}_{rs}^M - \sum_{(r,j)\in Y'^-} log\left(1 - \hat{y}_{rj}^M\right) \\ &= -\sum_{(r,s)\in Y' \cup Y'^-} y_{rs}^M log\,\hat{y}_{rs}^M + \left(1 - y_{rs}^M\right) log\left(1 - \hat{y}_{rs}^M\right). \end{aligned} \tag{8}$$

It is the loss function to be minimized and to optimize it, the Stochastic Gradient Descent (SGD) algorithm can be used. Equation (8) is equivalent to the *log loss* or *binary cross-entropy loss*. Now using a likelihood model for MEDCF, recommendations is treated as a binary classification task with implicit feedback. Classification based log loss is seldom applied in recommendation theory, although here it is examined which demonstrate its efficiency.

5.3 Generalized matrix factorization (GMF)

A generalization of the Matrix Factorization (MF) technique is shown and how MF can be used as a special case of MEDCF framework. MF is regarded as the popular recommendation technique and many factorization models [45] can be treated as a special case of the MEDCF framework. MEDCF framework is modified and name this modified version as Generalized Matrix Factorization (GMF) model.

The bottom input layer of the MEDCF framework comprises of v_u^U and v_i^I which represents binarized one-hot encoded feature vector of user u and item i, respectively. Applying embedding layer on v_u^U gives latent vector for user u, which symbolizes as p_u. Similarly, applying the embedding layer on v_i^I gives latent vector for item i, which symbolizes as q_i. To deal with uncertainty in the ratings to determine more accurate latent representations, we calculate mean and variance of user and item embedded vector, i.e., $\mu(p_u)$, $\sigma(p_u)$, $\mu(q_i)$, and $\sigma(q_i)$ and then sample a p_u (or q_i) to be fed to the next layer. It makes the whole process non-deterministic and thus, the function learned by a neural network is not a continuous function of the inputs. We use a neat trick here known as the reparameterization trick to get around this problem, wherein we shift the sampling process to an input layer. For one dimensional case, given μ and σ we can sample from $\mathcal{N}(\mu, \sigma)$ by first sampling $\epsilon \sim \mathcal{N}(0, 1)$, and then computing revised $p_u = \mu + \sigma * \epsilon$ (similar computation for q_i) as a final latent vector of the user (item). The function randomness is now associated with ϵ and not the inputs or the model's parameters. Next, the mapping function of the first neural MEDCF layer is introduced as:

$$\phi_1(p_u, q_i) = p_u \odot q_i, \tag{9}$$

where $\odot$ symbolizes the element-wise inner product between p_u and q_i vectors. Next, the vector is transformed to the output layer as:

$$\hat{y}_{ui}^G = a_{out}\left(h^T\left(p_u \odot q_i\right)\right), \tag{10}$$

where h and a_{out} denotes the weights and activation function of the output layer, respectively.

Now, to generalize and extend MF under this MEDCF framework, h is used to learn from the data itself. It allows a version of MF which has a different meaning of hidden dimensions. Using the non-linear function for a_{out} allows another version of MF which has a non-linear interaction between user and item and is more expressive than the linear MF model.

This work incorporate generalized version of MF (i.e., GMF) under MEDCF framework. It utilizes sigmoid non-linear activation function $\sigma(x) = 1/\left(1 + e^{-x}\right)$ as a_{out} and learning weights h from the data itself with the log loss objective function.

5.4 Multilayer perceptron (MLP)

MEDCF framework utilizes user and item metadata latent vectors to represent their features and concatenates them. This design is extensively used and adopted in multi-modal deep learning literature [56, 71]. However, simply concatenating user and item metadata features

is not sufficient to have a CF effect. To solve this, several hidden layers is proposed on top of the concatenated layer to learn the user-item metadata latent features collaboratively. Applying embedding layer on w_r^R to give latent vector for user metadata r which symbolizes as p'_r. Similarly, applying embedding layer on w_s^S to give latent vector for item metadata s which symbolizes as q'_s. This provides flexibility in learning the non-linear complex interactions between p'_r and q'_s vectors in contrast with the GMF that uses simply the element-wise dot product on p_u and q_i vectors. More accurately, the MLP model under MEDCF framework can be summarized mathematically as:

$$\begin{aligned} z_1 &= \phi_1\left(p'_r, q'_s\right) = \begin{bmatrix} p'_r \\ q'_s \end{bmatrix}, \\ z_2 &= \phi_2(z_1) = a_2\left(W_2^T z_1 + b_2\right), \\ &\cdots \\ z_L &= \phi_L(z_{L-1}) = a_L\left(W_L^T z_{L-1} + b_L\right), \\ \hat{y}_{rs}^M &= \sigma\left(h^T z_L\right). \end{aligned} \tag{11}$$

where b_x, a_x, W_x denotes the bias vector, activation function, weight matrix for the x-th layer of MLP, respectively. There are various choices for the activation function selection such as sigmoid, Rectified Linear Unit ($ReLU$), hyperbolic tangent ($tanh$), and many others. These functions are analyzed as follows: 1) The sigmoid function σ is a probabilistic function that restricts its value into the range [0,1] which limits the model performance. It has a problem of saturation meaning that the neuron stops learning when the function value is close to 0 or 1. 2) $tanh$ seems to be a good option [12, 66] however it only slightly alleviates the problem of saturation for sigmoid function because $tanh$ is a rescaled variant of sigmoid ($tanh(x/2) = 2\sigma(x) - 1$). 3) $ReLU$ is well known to be the non-saturated function [15]. It is also treated as a sparse activation function because it can be used when the data is sparse and it also prevents overfitting of the data. The empirical results show that $ReLU$ somewhat preferable than $tanh$, which in turn considerably preferable than sigmoid.

MLP model uses a hierarchy structure of hidden layers meaning that every successive hidden layer has the fewer number (say half) of hidden neuron units than the previously hidden layer neuron units. It is because as going deeper into the neural network, the more complex interaction between the user and item metadata can be determined [24] and therefore more user metadata r likeness or dis-likeness towards item metadata s.

5.5 Fusion of GMF and MLP

Till now, two examples under the MEDCF framework is discussed. One is the GMF that allows the linear function to model the latent features interaction between user and item. Another one is MLP that allows a non-linear function to learn the latent features interaction between user and item metadata. Now, the question is how to fuse GMF and MLP under the MEDCF framework so that both can mutually complement each other on the user-item interaction for the enhanced model?

Algorithm 1 Rating prediction algorithm using meta embedding deep collaborative filtering (MEDCF) model.

Input: $Y \in \mathbb{R}^{A\times B}$, $Y' \in \mathbb{R}^{A\times B}$, v_u^U, v_i^I, w_r^R, w_s^S
Output: Final predicted rating $\hat{y}_{ui}$

procedure GMF(v_u^U, v_i^I)
  for each v_u^U and v_i^I **do**
    Apply embedding layer on v_u^U and v_i^I to give p_u and q_i, respectively
    Apply GMF layer to compute $\hat{y}_{ui}^G \leftarrow p_u \odot q_i \leftarrow p_u^T q_i$
  end for
end procedure
procedure MLP(w_r^R, w_s^S)
  for each w_r^R and w_s^S **do**
    Apply embedding layer on w_r^R and w_s^S to give p_r' and q_s', respectively
    Apply final MLP layer to compute $\hat{y}_{rs}^M \leftarrow \sigma\left(h^T a_L\left(W_L^T\left(a_{L-1}\left(\ldots a_2\left(W_2^T\begin{bmatrix}p_r'\\q_s'\end{bmatrix}+b_2\right)\ldots\right)\right)+b_L\right)\right)$
  end for
end procedure
procedure NeuMF(p_u, q_i, p_r', q_s')
  $\phi^G \leftarrow p_u \odot q_i$
  $\phi^M \leftarrow a_L\left(W_L^T\left(a_{L-1}\left(\ldots a_2\left(W_2^T\begin{bmatrix}p_r'\\q_s'\end{bmatrix}+b_2\right)\ldots\right)\right)+b_L\right)$
  Apply NeuMF layer to merge GMF and MLP outputs to compute final predicted rating as :
  $\hat{y}_{ui} \leftarrow ReLU\left(h^T\begin{bmatrix}\phi^G\\\phi^M\end{bmatrix}\right)$
end procedure
return $\hat{y}_{ui}$

One solution is to presume the embedding layer of GMF and MLP and then combine (concatenate) the outputs of their interaction functions. It is similar to the well-known approach followed in Neural Tensor Network (NTN) [53]. Mathematically, coupling GMF layer with a one-layer MLP can be defined mathematically as:

$$\hat{y}_{ui} = ReLU\left(h^T a\left(p_u \odot q_i + W\begin{bmatrix}p_r'\\q_s'\end{bmatrix}+b\right)\right), \tag{12}$$

Sharing the embedding of GMF and MLP might obstruct the performance of the combined model in the way that both are restricted from using the same length of embedding.

So, to give more flexibility to MEDCF model, separate embedding are used for the two models so that both learn their respective features more conveniently. Further, the outputs of the last hidden layer for GMF and MLP models are concatenated and mathematically it is defined as:

$$\begin{aligned}\phi^G &= p_u \odot q_i,\\ \phi^M &= a_L\left(W_L^T\left(a_{L-1}\left(\ldots a_2\left(W_2^T\begin{bmatrix}p_r'\\q_s'\end{bmatrix}+b_2\right)\ldots\right)\right)+b_L\right),\\ \hat{y}_{ui} &= ReLU\left(h^T\begin{bmatrix}\phi^G\\\phi^M\end{bmatrix}\right).\end{aligned} \tag{13}$$

where p_u and p'_r denotes latent vector for user u and user metadata r, respectively. Similarly, q_i and q'_s denotes latent vector for item i and item metadata s, respectively. $ReLU$ as the activation function is used for MLP layers. The fused model is named as *Neural Matrix Factorization* (NeuMF) under the MEDCF framework which jointly binds the linearity of the GMF model and non-linearity of MLP model to learn the user-item latent interaction using rating and metadata information. The MEDCF learning model is outlined in Algorithm 1.

6 Experimental setup

In this section, datasets, a brief introduction of baseline methods to compare with and the evaluation metrics for comparing model performance with the baselines are discussed.

6.1 Datasets

Experiments are carried out on the two publicly available real-world datasets in order to evaluate the model's performance on the rating prediction task:

(1) **MovieLens.** MovieLens 100K[1] (ML100K) and MovieLens 1M[2] (ML1M) datasets were created by the grouplens research project at the University of Minnesota. These datasets comprise of user's explicit ratings on items ranging from 1 to 5. Three files for the datasets are used. The rating file consists of a tab-delimited list of user id, item id, rating, and timestamp. The user file comprises user demographic information consisting of a tab-delimited list of user id, age, gender, occupation, and zip code. There is a total of 21 occupation categories having values 1 or 0 depending on whether the user corresponds to that occupation or not. The item file consists of movie metadata information consisting of the tab-delimited list of movie id, movie title, release date, video release date, IMDb URL, and genres. There is a total of 18 genre categories having values 1 or 0 depending on whether the movie is of that genre or not. The occupation distribution of the users and genre distribution of the movies for the MovieLens 1M dataset are shown in Figs. 3 and 4, respectively.

(2) **Amazon Movies**.[3] The Amazon Movies (AMovies) dataset was larger and sparse compared to the MovieLens dataset. For the Amazon dataset, minor pre-processing is done to filter the users' having a minimum of 20 interactions and items' having minimum 5 interactions. This step makes it similar to the MovieLens datasets.

The statistics of the MovieLens and Amazon movies datasets (after pre-processing) are mentioned in Table 2. The above datasets have explicit information. This information is converted into implicit information that contains 1 or 0 implying user interaction with the item or not. This implicit information is used for the GMF model. For the MLP model, metadata information of the user and the item is used. From the user file, features or attributes are extracted that are useful in learning the user's interest in the item. So gender and occupation features are extracted as user metadata (demographic) information. Similarly, from the item

[1] https://grouplens.org/datasets/movielens/100K/.
[2] https://grouplens.org/datasets/movielens/1M/.
[3] http://jmcauley.ucsd.edu/data/amazon/.

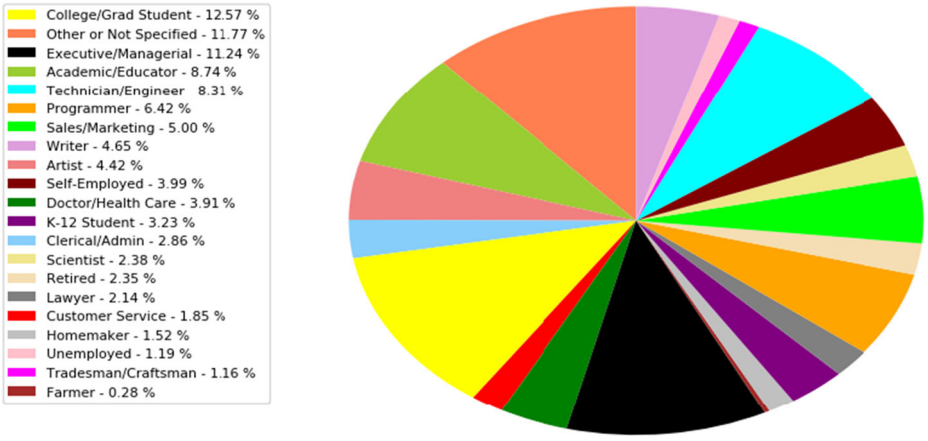

Fig. 3 Occupation distribution for the MovieLens 1M dataset

file, the genre feature is extracted as item metadata. These user-item metadata features are converted into a one-hot vector of 1's and 0's, indicating whether the user or item possesses that feature or not. Next, an interaction matrix Y' is used with each entry 1 or 0 depending on whether there is an interaction between user metadata r and item metadata s or not. Both MovieLens and AMovies datasets are split randomly into a training set (70%), validation set (10%), and the test set (20%).

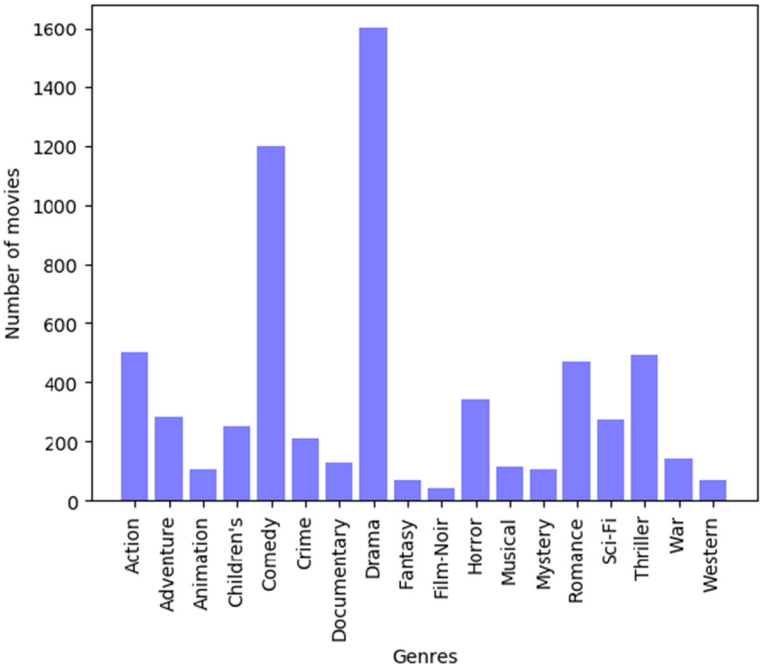

Fig. 4 Genre distribution for the MovieLens 1M dataset

Table 2 Data statistics of two real-world datasets

Dataset	Users	Items	Ratings	Density(%)
ML100K	943	1,682	100,000	6.3047
ML1M	6,040	3,952	1,000,209	4.4685
AMovies	15,067	69,629	877,736	0.0837

6.1.1 Cleaning the MovieLens datasets

For a real-time system scenario, cleaning the dataset is a crucial step that involves extracting the relevant features from the dataset and encode these features in integer values for further processing in neural network architectures. The MovieLens datasets are cleaned by removing irrelevant features such as timestamp, age, zip code, movie title, release date, video release date, and IMDb URL. The gender and occupation features as the user's metadata and genre feature as the item's metadata are adopted since both are helpful in better rating prediction. All the relevant features are binary encoded using categorical encoding provided by Pandas. The Keras embedding layer is applied to the binary encoded features which learn embeddings of the user-item metadata information.

6.2 Baseline methods

The proposed MEDCF model are evaluated and compared with the following baseline methods:

- **Item-Item similarity** uses item-item Collaborative Filtering (CF) approach. The similarity score such as cosine similarity is calculated between the two items. A similar argument with the user-user CF approach, however, it is less popular than item-item CF.
- **Matrix Factorization (MF)** [33] finds low dimensional latent features (hidden dimensions) of liking/disliking user's interest towards an item that depends on the pattern of ratings given by the user.
- **Blind Compressed Sensing** [14, 16] computes the user and item latent factor matrices. The user hidden factor matrix can be dense in contrast with the item hidden factor matrix because a user generally interacts with few items. The item latent factor matrix sparsity may enhance the recommendation accuracy.
- **Matrix Completion** [20] computes the missing entries in a sparse rating matrix. It finds the lowest rank matrix possible and whenever the rank of the filled matrix is pre-known, it finds the matrix of rank r that matches with the filled matrix values.
- **Probabilistic Matrix Factorization (PMF)** [44] is a rating prediction task that only utilizes ratings for CF. It can be considered as a probabilistic version of the SVD model.
- **Singular Value Decomposition (SVD)** [47] calculates the matrix for the given rank and minimizing the sum of squared error for the target matrix.
- **SVD++** [32] achieves better accuracy prediction by incorporating implicit feedback information.
- **Collaborative Topic Regression (CTR)** [61] is a recommendation model that combines CF such as PMF and subject modeling such as Latent Dirichlet Allocation (LDA) to learn both ratings and text documents.
- **Collaborative Deep Learning (CDL)** [65] is a recommendation model that improves rating prediction accuracy by considering documents using Autoencoders.

- **Convolutional MF (ConvMF)** [30] is a model that merges PMF model into Convolutional Neural Network (CNN).
- **additional Stacked Denoising Autoencoder (aSDAE)** [9] is a joint model that learns the user and item latent factors using side information and CF through the rating matrix. It is a hybrid CF that utilizes MF through implicit feedback and user and item side information through stack denoising autoencoder.
- **Multilayer Perceptron (MLP)** [22] is the artificial neural network having non-linear activation functions.
- **Neural Collaborative Filtering (NCF)** [22] integrates MLP model with the MF model on the user-item interaction matrix.
- **Hierarchical Structures (HSR)** [64] is a state-of-the-art algorithm able to capture information from both the rating and structured side information. In this strategy, however, flat information is ignored.
- **Heterogeneous side Information for Recommendation (HIRE)** [38] is a unified recommendation framework that simultaneously captures both flat and hierarchical information mathematically.

Since proposed model targets to find an interaction between users and items, it is compared with already known baseline methods and use the best hyper-parameters settings on the validation set.

6.3 Evaluation metrics

For comparing the model with other baseline methods, Root Mean Squared Error (RMSE) is used as the evaluation metric. It is a standard quantitative measure for supervised regression problems (for example, rating prediction task). It measures the difference in the predicted rating and the actual rating [18]. Mathematically, the RMSE loss is computed as:

$$RMSE = \sqrt{\frac{\Sigma \left(\hat{r}_{ui} - r_{ui}\right)^2}{\#\text{ of ratings}}} \tag{14}$$

Mean Absolute Error (MAE) is used as another metric that is vastly used in past literature [41, 51, 54]. It computes the absolute difference between the actual and predicted score. Conceptually, it gives an idea of how big the error can be think for the predicted score on an average. Mathematically, it is computed as:

$$MAE = \frac{\Sigma |r_{ui} - \hat{r}_{ui}|}{\#\text{ of ratings}} \tag{15}$$

Both RMSE and MAE are losses and should be minimized for better recommendation accuracy.

7 Experimental results

In this section, the quantitative results with error analysis, analysis of the cold start problem, execution time analysis and scalability issues, model parameter learning and need for pre-training, hyper-parameters setting and statistical significance, and the importance of deep learning for this work are discussed.

7.1 Quantitative results with error analysis

The RMSE and MAE values of MEDCF model and the baseline approaches on the MovieLens 100K dataset are shown in Table 3. Note that *Improve* signifies the relative improvement of MEDCF over the best competitor. As it can be seen that the proposed approach consistently and remarkably outcompetes baseline approaches by an apparent margin on the dataset. As an example, MEDCF improves on RMSE 11.303% and on MAE 14.245% over the best competitor HIRE with lagging behind other baseline methods.

For MovieLens 1M dataset also, the model obtains good results over baselines methods. Table 4 shows the rating prediction error of MEDCF and other baselines. The noticeable improvement of MEDCF over the best competitor HIRE is 12.27% on RMSE and 15.128% on MAE which indicates that using user-item metadata as auxiliary information can benefit in a deeper understanding of the latent factors and more accurately predict ratings. MEDCF model shows high expressiveness by combining linear GMF and non-linear MLP models using NeuMF. Also, notice that MLP slightly outperforms MF which limits MF as a special case of MLP. MLP model can further be improved by adding more hidden layers and here only three layers are shown. An overall indication is that the model utilizes a better understanding of the combined metadata and rating information to help improve the rating prediction task. For AMovies dataset, similar results as that of MovieLens datasets were obtained, as shown in Table 5.

The plots between MEDCF model training error versus epoch for the MovieLens 100K and MovieLens 1M datasets are shown in Fig. 5a and b, respectively.

As MEDCF model's last hidden layer decides the potential of the model, it is referred as *explanatory predictors* and the predictors of [8, 16, 32, 64] is evaluated. It is worth noting that large predictors may cause the model to be overfitted and degrades the performance.

Table 3 Experimental results on MovieLens 100K dataset

Model	RMSE	MAE
Item-Item similarity	1.061	0.744
Matrix Factorization	1.128	0.828
Blind Compressed Sensing	0.9409	0.735
Matrix Completion	1.102	0.832
PMF	0.9639	0.756
SVD	0.9521	0.743
SVD++	0.9434	0.738
CTR	0.9836	0.757
CDL	0.9628	0.748
ConvMF	0.9469	0.735
aSDAE	0.9382	0.728
MLP	0.9743	0.729
NCF	0.9319	0.718
HSR	0.9312	0.713
HIRE	0.9289	0.709
Ours	**0.8239**	**0.608**
Improve	11.303%	14.245%

The RMSE and MAE values of our proposed method are indicated in bold

Table 4 Experimental results on MovieLens 1M dataset

Model	RMSE	MAE
Item-Item similarity	0.9196	0.671
Matrix Factorization	0.8790	0.686
Blind Compressed Sensing	0.8789	0.691
Matrix Completion	0.9102	0.719
PMF	0.8971	0.706
SVD	0.8730	0.686
SVD++	0.8620	0.673
CTR	0.8969	0.706
CDL	0.8879	0.691
ConvMF	0.8549	0.676
aSDAE	0.8469	0.672
MLP	0.8773	0.681
NCF	0.8480	0.668
HSR	0.8466	0.664
HIRE	0.8459	0.661
Ours	**0.7421**	**0.561**
Improve	12.270%	15.128%

The RMSE and MAE values of our proposed method are indicated in bold

Without special mention, three hidden layers are used for MLP. For example, if the explanatory predictors size is 8 then the MEDCF layers architecture is $32 \rightarrow 16 \rightarrow 8$ and the

Table 5 Experimental results on AMovies dataset

Model	RMSE	MAE
Item-Item similarity	0.8786	0.632
Matrix Factorization	0.8391	0.648
Blind Compressed Sensing	0.838	0.652
Matrix Completion	0.8704	0.676
PMF	0.8568	0.667
SVD	0.832	0.643
SVD++	0.823	0.634
CTR	0.857	0.666
CDL	0.848	0.653
ConvMF	0.815	0.637
aSDAE	0.8071	0.634
MLP	0.8371	0.642
NCF	0.808	0.629
HSR	0.8067	0.624
HIRE	0.8059	0.621
Ours	**0.7021**	**0.521**
Improve	12.880%	16.103%

The RMSE and MAE values of our proposed method are indicated in bold

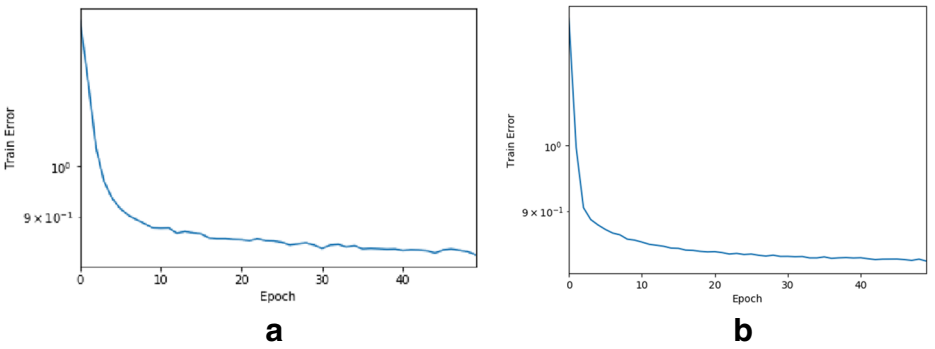

Fig. 5 Plot between MEDCF model training loss versus epoch for the MovieLens datasets. **a** Plot between training error versus epoch for the MovieLens 100K dataset **b** Plot between training error versus epoch for the MovieLens 1M dataset

embedding size is 16. α was set to 0.5 for the NeuMF with pre-training, enabling the pre-trained GMF and MLP to contribute to the initialization of NeuMF equally. It can be seen that MEDCF achieves the best performance over all datasets, substantially outperforming the state-of-the-art NCF and HIRE methods on RMSE by a large margin (on average, the relative improvement over NCF and HIRE is 12.39% and 12.15%, respectively).

Figure 6 shows the performance of MAE for the number of explanatory predictors. Here, the difference in performance between personalized methods is highlighted. For all the datasets, MEDCF significantly outperforms ConvMF and NCF methods with a large predictor of 64. MEDCF shows consistent improvements over all the baselines methods for predictors greater than or equal to 16, showing the effectiveness of the classification-aware log loss for the recommendation task. Also, both linear MF and non-linear MLP models are fused to reveal the high expressiveness of the MEDCF model. For MF methods, the number of explanatory predictors is equal to the latent factors since every feature obtained by the neural network layer corresponds to the latent dimension. The NCF and MLP methods also show fairly good results showing their effectiveness in learning user-item non-linear interactions using neural networks. NCF nearly outperforms MLP over all datasets. MLP and MEDCF methods suffer from overfitting for large predictors as it tends to make the model very complex by having too many parameters. Item-Item and PMF methods have the worst performance for small and large predictors, respectively, for all datasets as both of them consider the linear interactions between the users and items only. CTR and SVD methods also show poor results for large predictors on all datasets. The aSDAE, HSR and

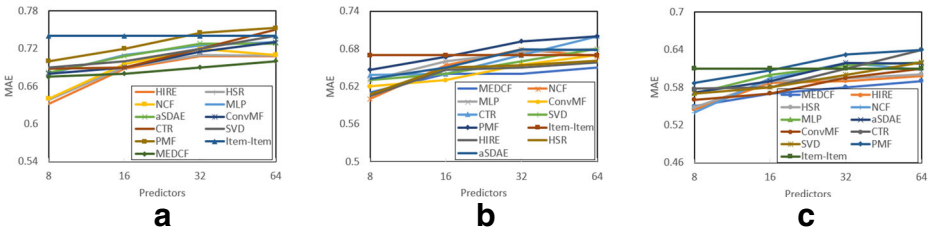

Fig. 6 Performance of MAE w.r.t. the number of explanatory predictors over all the datasets. Predictors denotes the architecture of the neural MEDCF layers. **a** MovieLens 100K **b** MovieLens 1M **c** AMovies

HIRE methods show promising results confirming their effectiveness in learning both the metadata information and the rating matrix in a unified framework.

MEDCF's other two methods, GMF and MLP, also demonstrate excellent performance. Between them, MLP is marginally underperforming GMF. Note that adding more hidden layers (more than five hidden layers slightly overfits the data) MLP can be further improved. Still, the generalization of the overall MEDCF model cannot be guaranteed. Here, the performance of the MEDCF model is shown up to three hidden layers. GMF outperforms ConvMF on all datasets for small explanatory predictors. Although GMF suffers from large predictors being overfitted, its best performance obtained is better than (or equivalent to) ConvMF.

The recommendation is used as a binary classification method to deal with the one-class solution of implicit feedback. By viewing MEDCF as a probabilistic model, the log loss of (8) is optimized. Figure 7 shows the training loss of MEDCF methods of each epoch on MovieLens 1M dataset with predictors of 8. As it can be seen that the training loss of the MEDCF methods slowly decreases with more iterations. In the first 10 iterations the most effective updates occur, and even more iterations may overfit a model. Also, NeuMF achieves the lowest training loss among the three MEDCF methods, followed by MLP, and then GMF. The above findings provide empirical evidence of the validity and efficacy of log loss optimization in order to learn from implicit data.

7.2 Cold start problem analysis

The MEDCF framework examines this work expertly in dealing with the cold start problem on the MovieLens datasets. For the new users and new items (new users are the users who have rated very few movies or have not rated any movie yet, and new items are the items which have been rated by only few users or have not been rated by any user yet), users' demographics and items genre information are exploited as their metadata features. This information serves as an input to the MEDCF framework. In general, the model gains excellent performance over the baseline methods which does not take into account the auxiliary information (e.g., NCF and PMF) of cold start users or items. It implies the model effectiveness which incorporates metadata information to alleviate the cold start problem.

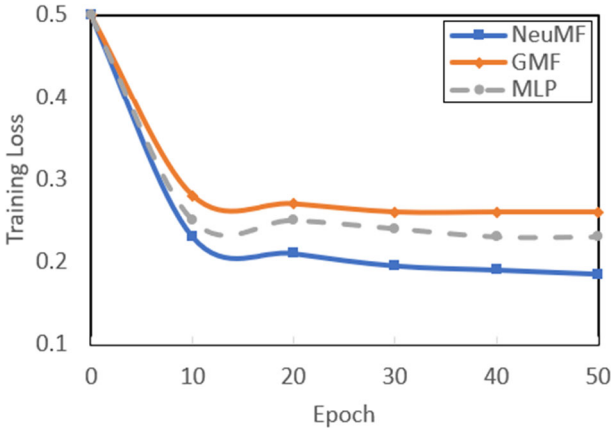

Fig. 7 Training loss (averaged over all instances) of MEDCF methods w.r.t. the number of epochs on MovieLens 1M dataset for predictors=8 which means the the MEDCF layers architecture is $32 \rightarrow 16 \rightarrow 8$ and the embedding size is 16

In Fig. 8, the reductions in MAE resulted from MEDCF is shown on the MovieLens 1M dataset. By MAE reduction, it means the difference between MEDCF's MAE and MF's MAE. Users and items are classified according to the number of explanatory predictors and reductions are plotted for both the groups. It can be seen that the reductions are positive and with a few number of predictors, MEDCF can achieve MAE reduction on all users and items groups. A benefit of MEDCF is that higher reductions are obtained for groups with fewer predictors. This shows that MEDCF can reduce the sparsity issue and help with the cold start problem.

Also, a relationship exists between the effectiveness of MEDCF and the number of users or items predictors. MEDCF reduction in MAE is higher for users or items with a smaller number of predictors. It shows metadata can be valuable information particularly in case of limited users or items information.

7.3 Execution time analysis and scalability

The computational time of proposed MEDCF model and the baseline methods is analyzed. All of these methods run on a single NVIDIA GeForce GTX 1080 GPU machine having a computed capability of 6.1. At the training phase, NCF and MEDCF take about 2-3 seconds per epoch on the MovieLens 100K dataset whereas both takes about 2-4 seconds per epoch on the MovieLens 1M dataset. In less than 25 epochs, all the models typically converge using the early stop criterion. The recommendation prediction results are reasonably fast in the testing phase and are within 1-2 seconds. The MF based techniques (e.g., PMF or SVD) take less running time than the neural networks based techniques. In the training phase, the proposed model takes comparable time with NCF, however, more time than SVD or PMF. The model takes comparable time to other methods during the testing phase. Therefore, the proposed MEDCF model is practical for a movie recommendation in the real scenario.

As the number of users and items increases, the computation increases linearly which is the problem with the recommendation algorithms. The recommendation model which is trained and gives better results on the limited dataset degrades its performance when the dataset is scaled. Therefore, there is a necessity to apply recommendation algorithms successfully as the dataset scales. To handle the scalability issues, the dimensionality reduction technique is used using neural networks. The total neuron units in the successive hidden

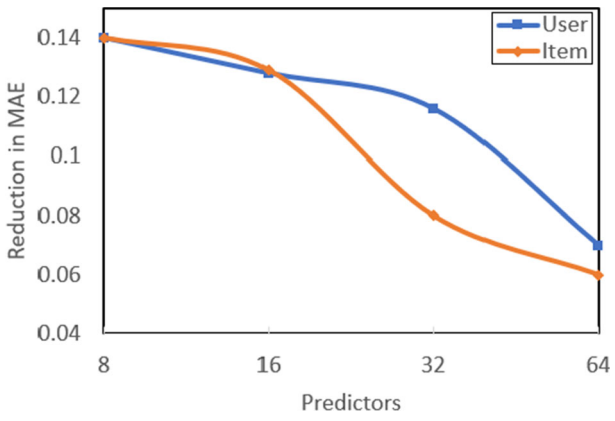

Fig. 8 MEDCF gains different MAE reductions on the MovieLens 1M dataset for users and items with different number of explanatory predictors. Predictors denotes the architecture of the neural MEDCF layers

layer are decreased by half of its previous hidden layer. It ensures that each successive hidden layer learns new latent representations for user-item interactions and removes the noise (i.e., irrelevant information) from the previous hidden layer to extract relevant features. Using experiments, it is found that using the same number of neuron units in each hidden layer will roughly triple the cost per computation (backpropagation) and requires more data, approximately triple the training data and training time.

Collaborative Filtering (CF) is one of the most commonly used algorithms to make rating predictions within a recommender system. More resources are required for CF-based recommendation systems to process information and form recommendations as the number of users and items increases. Most of these resources are consumed to determine similar preferences for users and similar descriptions for items. CF algorithms therefore face a scalability problem that can become an important factor for recommendation task and it is difficult to produce recommendations in real time. In a practical movie recommendation system, it is imperative to solve the scalability issue in order to make recommendations in real time. In response, a neural network learning using metadata features such as user's demographic and item's genre information is proposed. Compared with the raw user-item rating matrix that is used by most of the baseline methods as discussed, the latent dimensions of the user-item metadata interaction matrix are greatly reduced especially in the cold start scenario. All of the above features significantly speed up the training of the classification model and ensure that recommendations can be provided in real time.

To investigate the scalability issues of some of the competitive methods, training and inference time are considered as shown in Table 6. Training time means the time needed to train the model to its best performance. Three variants of dataset size: 25%, 75% and 100% (complete dataset) are considered. With the increase in dataset size, training and prediction time increases significantly for all the methods over all the datasets. HSR has the least training and prediction time for all the datasets. For the training process, the total training time taken by proposed method is less compared to the HIRE method. It can be explained by the fact that the proposed model takes fewer iterations to converge due to the hybrid

Table 6 Training and prediction time with varying dataset size ratios for different models over all the datasets

		AMovies		ML100K		ML1M	
Model	Dataset ratio	Training time(s)	Prediction time(s)	Training time(s)	Prediction time(s)	Training time(s)	Prediction time(s)
HSR	25	601	5	15	0.2	310	2
	75	810	13	25	0.7	405	4
	100	930	20	36	0.8	447	5
HIRE	25	18300	10	120	0.2	3502	4
	75	28810	15	141	0.6	4573	6
	100	3,5370	21	147	0.9	5362	8
Ours	25	14345	10	32	0.2	1814	4
	75	21362	15	68	0.8	2637	7
	100	25962	20	81	1.1	3006	8

All values are rounded off with the nearest integer

approach used for feature modeling. For inference time, the HIRE and proposed method take more time compared to HSR.

7.4 Model parameter learning and need for pre-training

The objective function of non-convexity leads us to the local optimal solution. Parameter initialization has an essential role in the convergence as well as overall performance for the MLP model. Since NeuMF is the combination of both GMF and MLP models, the pre-trained parameters of both the models is used. For this, GMF and MLP models are learnt from scratch until convergence and then use these learned parameters as an initialization for the NeuMF model. Adaptive Moment Estimation (ADAM) [31] is used as an optimizer for pre-training and vanilla SGD for the NeuMF model. The reason to use ADAM is that it uses slow updates for frequent parameters and fast updates for infrequent parameters. Also, ADAM has a fast convergence rate than vanilla SGD. SGD is used in NeuMF because ADAM must save the moment information to update the parameters correctly. As the NeuMF is initialized with the pre-trained parameters, the momentum information is restricted to proceed further. So, it is useless to use NeuMF with the momentum-based optimizers.

7.5 Hyper-parameters setting and statistical significance

Indicators that are widely used to assess the reliability of experimental outcomes such as hyperparameter search or statistical significance is discussed.

The experiments with proposed model are based on Keras[4] neural networks API. All methods of the MEDCF model are learned by optimizing the log-likelihood loss function as given in (8). The ADAM optimizer is used with mini-batch gradient descent. Learning rate of [0.0001, 0.0005, 0.001] and batch size of [128, 256, 512] have been tested. Three hidden layers are used for MLP.

MEDCF shows noticeable improvements over other baselines, and further one-sample paired t-tests is performed to verify that $p < 0.02$ is statistically significant for all improvements. It can be argued that merely relying on significance tests does not guarantee reliable results. Proper setups are the ultimate choice for the best results. MovieLens 1M dataset has most of the results reported in [35, 57]. The reported standard deviation is generally less and the difference is*statistically significant* in the reported results. Significance test results should not be taken as a suitable choice of whether method A is better than method B. Significance test does not find how well the setup of the method is, rather it measures the standard deviation within the setup. Therefore, variance and significance tests must only be considered after having proof that the method applied is well used. The large source of errors occurs while setting up the method used. In some sense, the statistical significance tests are of little use and it often gives a false belief in the experimental results.

7.6 Is deep learning helpful?

There is little work dedicated to user-item interactions with metadata embedding using neural networks. It is interesting to see whether applying neural networks enhance the recommendation accuracy. For this, it is analyzed with increasing the MLP hidden layers. The

[4] https://keras.io/.

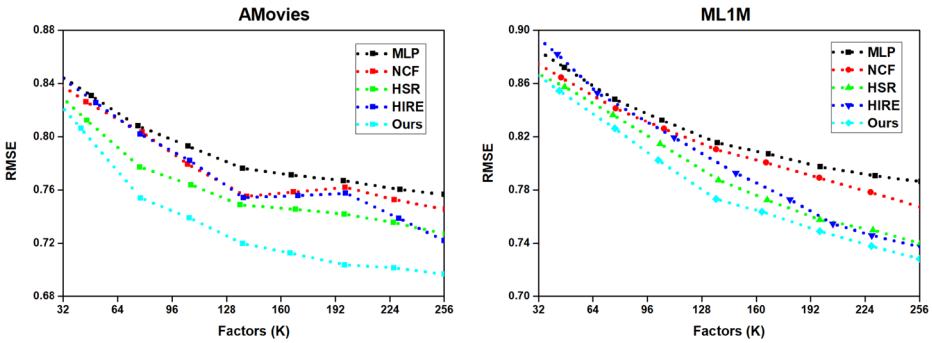

Fig. 9 Performance comparison on RMSE with varying embedding size K on the test set

notation MLP-3 indicates three hidden layers (apart from the embedding layer). As it can be seen that stacking more hidden layers is beneficial to the recommendation task. The results found using MLP-3 is highly encouraging, indicating the effectiveness of using neural networks for the collaborative recommendation model. This improvement is attributed to the non-linearities brought by stacking more hidden layers. Increasing the hidden layers to six or more slightly degrades the performance due to the overfitting of the user-item interactions. To verify further, linear activation functions are used for the hidden layers. It is noticed that the performance degrades due to the inability to capture non-linear interactions between users and items. This observation was the same as if applying the factorization techniques for the rating prediction task.

For MLP-0, which has no hidden layers (only embedding layer is there), the performance degrades which ensures that merely element-wise dot product between the user and item latent vectors is insufficient for modeling their interactions. Thus there is a need for transforming it with hidden layers.

Further, RMSE values are tested with varying factors K. The RMSE of the proposed model outperforms some of the baseline methods as shown in Fig. 9. The results are tested on AMovies and ML1M datasets. As the ML100K dataset also follows similar results as that of the ML1M dataset, it is not shown here. The improvements in the rating prediction task signify proposed model superiority and it is because deep neural networks are employed explicitly to capture the latent and interaction features between users and items. It can be seen that when K equals 256, all the models attain their minimum rating prediction error.

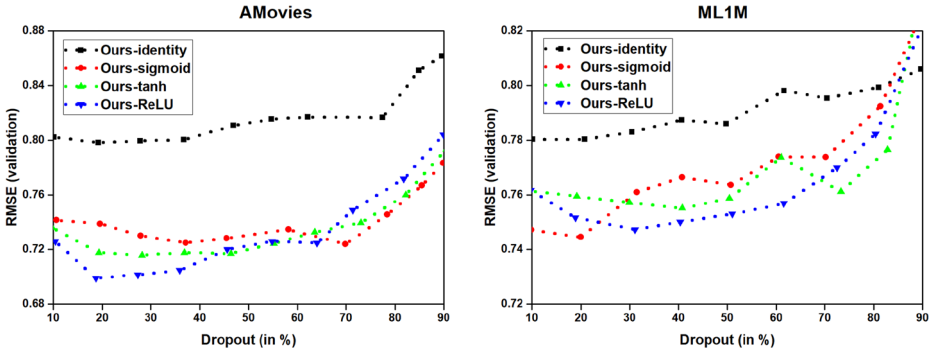

Fig. 10 RMSE validation error $w.r.t$ different activation functions and dropout ratios on the first hidden layer

Figure 10 demonstrates the RMSE validation error w.r.t. the dropout ratios. To prevent overfitting, dropouts as regularizer is used and consider its value through 0.1 to 0.9. The validation error fluctuates as the dropout ratio is increased. It may be because higher dropouts may underfit the model. The proposed model is considered with four different activation functions and concluded that ReLU gives minimal error on smaller dropouts for AMovies dataset and sigmoid gives minimal error on smaller dropouts for the ML1M dataset. The proposed model having dropouts above 0.7 cannot predict the ratings efficiently for both the datasets and therefore, model underfits. Identity function shows approximately linear performance with increasing dropouts but leads to high validation error.

8 Conclusion

The general framework of the MEDCF model is discussed which includes three methods named as GMF, MLP, and NeuMF. Using this model, the historical ratings given by the user to the items is exploited along with the user-item metadata interactions. The MEDCF model learns comprehensive user-item interaction function for rating prediction task that alleviates the cold start problem. This work can serve as the basic guideline on how to incorporate auxiliary information such as metadata and apply it to neural networks for generalized recommender systems. This work integrates the shallower neural network models for collaborative learning to have CF effect. It emphasizes research possibility for deep learning based recommender systems. The work shows the integration of auxiliary as well as rating information using a combined neural network and MF technique. Several experiments is conducted to verify the effectiveness of the model on widely acceptable real-world MovieLens datasets achieving state-of-the-art results.

The future effort aims at augmenting the MEDCF model to accommodate other crucial auxiliary information such as knowledge bases, user reviews, temporal signals, and multimodal data such as still images and visual semantics. This information gains more insight into user preferences for better personalization. Also, the previous personalization models focus on individuals. This work can be extended for a group recommender system used for decision making in social groups.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations

Conflict of Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Human and Animal Rights Statements All authors declared that there is no research involving human and/or animal.

References

1. Al-Shamri MYH (2016) User profiling approaches for demographic recommender systems. Knowl-Based Syst 100:175–187

2. Bai T, Wen JR, Zhang J, Zhao WX (2017) A neural collaborative filtering model with interaction-based neighborhood. In: Proceedings of the 2017 ACM on conference on information and knowledge management. ACM, pp 1979–1982
3. Bennett J, Lanning S, et al. (2007) The netflix prize. In: Proceedings of KDD cup and workshop. Citeseer, vol 2007, pp 35
4. Callvik J, Liu A (2017) Using demographic information to reduce the new user problem in recommender systems. KTH, School of Computer Science and Communication (CSC)
5. Cheng Z, Chang X, Zhu L, Kanjirathinkal RC, Kankanhalli M (2019) Mmalfm: Explainable recommendation by leveraging reviews and images. ACM Trans Inf Syst (TOIS) 37(2):1–28
6. Cheng Z, Ding Y, He X, Zhu L, Song X, Kankanhalli MS (2018) Aˆ 3ncf: An adaptive aspect attention model for rating prediction. In: IJCAI, pp 3748–3754
7. D'Addio RM, Marinho RS, Manzato MG (2019) Combining different metadata views for better recommendation accuracy. Inf Syst 83:1–12
8. Do TDT, Cao L (2018) Coupled poisson factorization integrated with user/item metadata for modeling popular and sparse ratings in scalable recommendation. In: Thirty-second AAAI conference on artificial intelligence
9. Dong X, Yu L, Wu Z, Sun Y, Yuan L, Zhang F (2017) A hybrid collaborative filtering model with deep structure for recommender systems. In: Proceedings of the thirty-first AAAI conference on artificial intelligence, pp 1309–1315
10. Dureddy HV, Kaden Z (2018) Handling cold-start collaborative filtering with reinforcement learning. arXiv:180606192
11. Ekstrand MD, Tian M, Azpiazu IM, Ekstrand JD, Anuyah O, McNeill D, Pera MS (2018) All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In: Conference on fairness, accountability and transparency, pp 172–186
12. Elkahky AM, Song Y, He X (2015) A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: Proceedings of the 24th international conference on World Wide Web, international World Wide Web conferences steering committee, pp 278–288
13. Fernández-Tobías I, Cantador I, Tomeo P, Anelli VW, Di Noia T (2019) Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization. User Model User-Adap Inter 29(2):443–486
14. Gleichman S, Eldar YC (2011) Blind compressed sensing. IEEE Trans Inf Theory 57(10):6958–6975
15. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 315–323
16. Gogna A, Majumdar A (2015) Blind compressive sensing framework for collaborative filtering. arXiv:150501621
17. Guan X, Cheng Z, He X, Zhang Y, Zhu Z, Peng Q, Chua TS (2019) Attentive aspect modeling for review-aware recommendation. ACM Trans Inf Syst (TOIS) 37(3):1–27
18. Gunawardana A, Shani G (2009) A survey of accuracy evaluation metrics of recommendation tasks. J Mach Learn Res 10:2935–2962
19. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems, pp 1024–1034
20. Hastie T, Mazumder R, Lee JD, Zadeh R (2015) Matrix completion and low-rank svd via fast alternating least squares. J Mach Learn Res 16(1):3367–3402
21. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: Simplifying and powering graph convolution network for recommendation. arXiv:200202126
22. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, international world wide web conferences steering committee, pp 173–182
23. He X, Zhang H, Kan MY, Chua TS (2016) Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 549–558
24. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
25. Henk V, Vahdati S, Nayyeri M, Ali M, Yazdi HS, Lehmann J (2019) Metaresearch recommendations using knowledge graph embeddings. In: RecNLP workshop of AAAI conference
26. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
27. Hsieh CK, Yang L, Cui Y, Lin TY, Belongie S, Estrin D (2017) Collaborative metric learning. In: Proceedings of the 26th international conference on world wide web, pp 193–201

28. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE international conference on data mining. IEEE, pp 263–272
29. Kim KS, Chang DS, Choi YS (2019) Boosting memory-based collaborative filtering using content-metadata. Symmetry 11(4):561
30. Kim D, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM conference on recommender systems. ACM, pp 233–240
31. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:14126980
32. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 426–434
33. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
34. Kula M (2015) Metadata embeddings for user and item cold-start recommendations. arXiv:150708439
35. Li D, Chen C, Liu W, Lu T, Gu N, Chu S (2017) Mixture-rank matrix approximation for collaborative filtering. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems. Curran Associates Inc., vol 30, pp 477–485
36. Liu F, Cheng Z, Sun C, Wang Y, Nie L, Kankanhalli M (2019b) User diverse preference modeling by multimodal attentive metric learning. In: Proceedings of the 27th ACM international conference on multimedia, pp 1526–1534
37. Liu D, Li J, Du B, Chang J, Gao R (2019a) Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 344–352
38. Liu T, Wang Z, Tang J, Yang S, Huang GY, Liu Z (2019) Recommender systems with heterogeneous side information. In: The World Wide Web conference, pp 3027–3033
39. Luo X, Yuan Y, Zhou M, Liu Z, Shang M (2019) Non-negative latent factor model based on β-divergence for recommender systems. IEEE Trans Syst Man Cybern Syst, 1–12. https://doi.org/10.1109/TSMC.2019.2931468
40. Luo X, Zhou M, Li S, Hu L, Shang M (2019) Non-negativity constrained missing data estimation for high-dimensional and sparse matrices from industrial applications. IEEE Trans Cyber 50(5):1844–1855
41. Ma H, Yang H, Lyu MR, King I (2008) Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM conference on information and knowledge management. ACM, pp 931–940
42. McAuley J, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on recommender systems, pp 165–172
43. Mittal P, Jain A, Majumdar A (2014) Metadata based recommender systems. In: 2014 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 2659–2664
44. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: Advances in neural information processing systems, pp 1257–1264
45. Rendle S (2010) Factorization machines. In: IEEE International conference on data mining. IEEE, pp 995–1000
46. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp 452–461
47. Sarwar B, Karypis G, Konstan J, Riedl J (2002) Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth international conference on computer and information science. Citeseer, vol 27, pp 28
48. Sedhain S, Menon AK, Sanner S, Xie L (2015) Autorec: Autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on World Wide Web. ACM, pp 111–112
49. Shang M, Luo X, Liu Z, Chen J, Yuan Y, Zhou M (2018b) Randomized latent factor model for high-dimensional and sparse matrices from industrial applications. IEEE/CAA J Autom Sin 6(1):131–141
50. Shang J, Sun M, Collins-Thompson K (2018a) Demographic inference via knowledge transfer in cross-domain recommender systems. In: 2018 IEEE international conference on data mining (ICDM). IEEE, 1218–1223
51. Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 650–658
52. Soares M, Viana P (2015) Tuning metadata for better movie content-based recommendation systems. Multimed Tools Appl 74(17):7015–7036

53. Socher R, Chen D, Manning CD, Ng A (2013) Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems, pp 926–934
54. Srebro N, Rennie J, Jaakkola TS (2005) Maximum-margin matrix factorization. In: Advances in neural information processing systems, pp 1329–1336
55. Sridevi M, Rao RR (2017) Decors: A simple and efficient demographic collaborative recommender system for movie recommendation. Adv Comput Sci Technol 10(7):1969–1979
56. Srivastava N, Salakhutdinov RR (2012) Multimodal learning with deep boltzmann machines. In: Advances in neural information processing systems, pp 2222–2230
57. Strub F, Gaudel R, Mary J (2016) Hybrid recommender system based on autoencoders. In: Proceedings of the 1st workshop on deep learning for recommender systems. ACM, New York, NY, USA, DLRS 2016, pp 11–16. https://doi.org/10.1145/2988450.2988456
58. Tang D, Qin B, Liu T, Yang Y (2015) User modeling with neural network for review rating prediction. In: Twenty-fourth international joint conference on artificial intelligence
59. Vasile F, Smirnova E, Conneau A (2016) Meta-prod2vec: Product embeddings using side-information for recommendation. In: Proceedings of the 10th ACM conference on recommender systems. ACM, pp 225–232
60. Vozalis M, Margaritis KG (2004) Collaborative filtering enhanced by demographic correlation. In: AIAI symposium on professional practice in AI, of the 18th world computer congress
61. Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 448–456
62. Wang M, Fu W, Hao S, Tao D, Wu X (2016) Scalable semi-supervised learning by efficient anchor graph regularization. IEEE Trans Knowl Data Eng 28(7):1864–1877
63. Wang X, He X, Wang M, Feng F, Chua TS (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 165–174
64. Wang S, Tang J, Wang Y, Liu H (2018) Exploring hierarchical structures for recommender systems. IEEE Trans Knowl Data Eng 30(6):1022–1035
65. Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1235–1244
66. Wu Y, DuBois C, Zheng AX, Ester M (2016) Collaborative denoising auto-encoders for top-n recommender systems. In: Proceedings of the ninth ACM international conference on Web search and data mining. ACM, pp 153–162
67. Wu D, Luo X, Shang M, He Y, Wang G, Zhou M (2019) A deep latent factor model for high-dimensional and sparse matrices in recommender systems. IEEE Trans Syst Man Cybernet Syst, 1–12. https://doi.org/10.1109/TSMC.2019.2931393
68. Xiao T, Liang S, Shen W, Meng Z (2019) Bayesian deep collaborative matrix factorization. In: Proceedings of the thirty-third AAAI conference on artificial intelligence (AAAI 2019). AAAI
69. Yoon YC, Lee JW (2018) Movie recommendation using metadata based word2vec algorithm. In: 2018 international conference on platform technology and service (PlatCon). IEEE pp 1–6
70. Zhang H, Shen F, Liu W, He X, Luan H, Chua TS (2016) Discrete collaborative filtering. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 325–334
71. Zhang H, Yang Y, Luan H, Yang S, Chua TS (2014) Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In: Proceedings of the 22nd ACM international conference on multimedia. ACM, pp 187–196