



A deep neural network-based collaborative filtering using a matrix factorization with a twofold regularization

Armienne Noulapeu Ngaffo¹ · Zièd Choukair¹

Received: 1 April 2021 / Accepted: 4 December 2021 / Published online: 25 January 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

In recent years, the ever-growing contents (movies, clothes, books, etc.) accessible and buyable via the Internet have led to the information overload issue and therefore the item targeting problem. Indeed, the huge mass of contents complexifies the identification of items fitting users' expectations. As powerful filtering tools, recommender systems efficiently alleviate the item targeting issue. Collaborative filtering-based methods are among the most influential algorithms adopted in recommender systems. Among collaborative filtering-based methods, model-based approaches are widely used in recent powerful recommendation methods. Due to its efficiency, the matrix factorization technique is spreadly employed in model-based approaches. However, those methods badly deal with issues such as data sparseness and cold-start problems that severely affect the recommendation quality. To overcome these limitations shown by state-of-the-art methods, we propose in this paper a recommender approach that couples the effectiveness of an enhanced matrix factorization technique to the power of a deep neural network model. In the first step, the user's latent factors and item latent factors are extracted from a doubly-regularized matrix factorization process. Thereafter, those latent factors are used to feed a deep learning structure in a forward-propagation process, and a normalized cross-entropy method is used to increase the precision of the deep neural network through a backpropagation process. The end prediction is made by combining results from the matrix factorization step and the deep neural structure. Extensive experiments are conducted on real-world datasets and show that our proposal outperforms other methods in terms of prediction accuracy and recommendation quality.

Keywords Deep neural network · Collaborative filtering · Matrix factorization · Twofold regularization · Recommender system

1 Introduction

Recommender systems (RS) help to tackle the information overload issue since they are useful to quickly and easily identify items matching users' needs [20]. For several years, RS have been widely adopted by various commercial platforms to enhance users' experiences by providing customers with products (movies, books, shoes, etc.) fitting their expectations. RS are mainly organized into content-based methods and Collaborative Filtering (CF)-based

methods [53]. Content-based approaches exploit extra information on items and user profile to perform the recommendation. However, those approaches show poor performances when the target user profile is sparse or when the system does not have enough extra information (users reviews, document content, etc.) on items. Collaborative Filtering (CF) methods are spreadly adopted in RS since they overcome limitations of content-based approaches by using a recommendation model that is built on experiences of a set of users that share similar tastes and needs [34]. For this purpose, interactions between items and users of this set of users help to perform the recommendation without requiring any extra information on items. CF-based methods are categorized into memory-based approaches and model-based approaches. Memory-based algorithms are simple implementable but inaccurate in case of large datasets. Meanwhile, model-based methods are hard-

✉ Armienne Noulapeu Ngaffo
armienne.ngaffo@supcom.tn

Zièd Choukair
choukair@supcom.tn

¹ University of Carthage, Higher School of Communication of Tunis, Mediatron Laboratory, Carthage, Tunisia

masterable but they are highly accurate even in case of large datasets [7].

In the literature, model-based approaches appear as the most widely used algorithms among CF-based methods. They use machine learning techniques to model users' interests. Model-based approaches are famous due to their valuable recommendation performances. However, they badly deal with issues such as data sparsity and cold-start problems that can severely affect recommendation quality. Among model-based methods, Matrix Factorization (MF) is a very popular technique largely adopted in several recent powerful RS. This technique consists in decomposing the original high-dimensional data matrix into low-rank latent factors matrices [22]. However, in order to perform the prediction, this decomposition process involves a linear dot product between latent factors matrices that does not accurately model complex interactions that happen between users and items [1, 11]. Several studies [7, 25, 32, 35] try to tackle this insufficiency of the MF process by adding bias terms in the loss function also known as the interaction function [11]. Despite the incorporation of bias terms in the interaction function, the linearity of the inner product of the matrix factorization remains not completely mitigated. To remedy this, the nonlinearity property of deep neural networks shows encouraging promises.

Thanks to their nonlinearity, deep neural networks appear as a powerful technique to solve limitations of the simple inner product involved in the MF process. More recently, Deep neural networks (DNN) have proved their effectiveness in several domains such as computer vision, speech recognition, to text processing [8, 9, 14, 38] but they are not enough exploited in recommender systems (RS) comparatively to the rich literature on MF-based recommender methods. Some recent studies have proposed recommendation approaches based on neural networks but these methods are built on the baseline matrix factorization model that does not effectively alleviate the issue of the simplicity and the linearity of the inner product of latent factors in the interaction function. Subsequently, some complex interactions and relationships between users and items are not accurately modeled. To remedy these limitations, we propose a model-based recommendation approach that uses a matrix factorization technique in which the simplicity of the inner product of latent factors is alleviated by a twofold regularization of the interaction function. Moreover, the nonlinearity of complex users-items interactions is leveraged by employing deep neural networks (DNN).

The recognized effectiveness of Matrix Factorization-based model remains limited due to the fact that complex interactions between users and items are not accurately modelled by the inner dot product involved during the

factorization process. We aim to address this limitation by leverage the contribution of neural networks. Our proposal is a recommendation method that combines the effectiveness of an improved matrix factorization and the power of deep neural networks. The contribution of this paper is declined as follows:

- A matrix factorization based model is proposed and enhanced through a twofold regularization in order to significantly reduce the linearity impact of the inner product of latent factors, and therefore increase the accuracy of the user-item interactions modelling.
- A deep neural network architecture is built and made of multilayer perceptron and single layer perceptron to effectively solve the linearity issue proper to the dot product of latent factor matrices from the matrix factorization process.
- Extensive experimentations are conducted on several real-world datasets to highlight the effectiveness of the proposed method and the valuable contribution of deep learning in the recommendation task.

The remainder of this paper is structured as follows: section 2 reviews state-of-the-art recommendation methods, Sect. 3 details the proposed recommendation approach. Section 4 presents the conducted experiments and discusses obtained results. Section 5 concludes this paper and shows our perspectives.

2 Related work

According to the literature, Collaborative Filtering (CF)-based methods are very popular in Recommender Systems (RS). They are mainly categorized into memory-based and model-based approaches [22]. In this section, we survey several CF-based methods and promises of the deep learning applied in RS.

2.1 Memory-based methods

Memory-based methods are easy-understandable and simple-implementable [53]. They are mainly based on a neighborhood computation process that evaluates inter-user relationships in order to identify like-minded users. Authors in [4] propose a recommendation model that performs reliable recommendations according to three views of reliability measures such as user-based, item-based, and rating-based reliability measures. According to the authors' method, an initial rating prediction is made and is thereafter improved through a neighborhood improvement mechanism for unreliable predicted ratings. Authors in [58] propose a recommendation model that uses an adaptive similarity measure in order to alleviate the data sparsity

issue. For this purpose, the authors' similarity measure combines online context-based multiple armed bandit mechanism. Authors in [3] propose a social collaborative filtering based on an adaptive neighborhood selection mechanism. A reliability measure evaluates the recommendation credibility, and a confidence model drops unreliable users from the initial neighborhood. The neighborhood is continuously updated in order to refine the prediction. Authors in [2] propose a social recommendation model based on user profiles improved by using virtual ratings. For this purpose, they compute the minimum number of required ratings through a probabilistic mechanism. The authors' method ensures reliable recommendations thanks to reliable user rating profiles. Authors in [16, 56] propose a recommender system that relies on inter-user similarities computed by using the Spearman Rank Correlation Coefficient (SRCC). They perform predictions as average of weighted ratings from like-minded users.

Although simple and easy-affordable, memory-based methods badly deal with large datasets. In addition, they are highly sensitive to the data sparsity issue. Model-based methods have been developed to overcome these limitations.

2.2 Model-based methods

Model-based methods show the advantage of their accuracy and scalability. Several recent high-performant RS [13] use them to efficiently model user-item interactions. Those methods are most often based on Bayesian networks [23, 33, 46, 48, 52], clustering CF [5], latent semantic CF [42] and matrix factorization technique [30, 32, 36, 40]. Among model-based approaches, those based on matrix factorization are the most popular [28, 57]. Matrix factorization (MF) is a dimensionality reduction technique that reduces an original high-rank matrix into low-rank latent factor matrices. Afterwards, an optimization method helps to minimize the interaction function. Considering user-item interactions modeled by low-rank latent factor matrices, authors in [22] propose MF-based methods to alleviate the data sparsity problem while authors in [18] propose a MF-based recommender framework for an online recommendation. Authors in [28, 32, 35] propose a non-negative MF-based model while authors [36] incorporate social information into their recommender framework to effectively feed the MF-based RS. In [29], to address the high computational problem shown by existing recommender models, the authors propose the Nonnegative Latent Factor Model (ANLF) based on Alternating Direction Method (ADM). Their proposal is accurate and highly scalable while showing a low complexity. Authors in [24] develop a trust metric model that is incorporated in the regularization of the matrix factorization process. Similarly, authors in

[10] develop a recommendation model that integrates trust social information in the latent feature extraction process performed by using the Single Value Decomposition (SVD). The authors' proposal is scalable and performs reliable recommendations. Authors in [17] propose an adaptive learning rate function that comes to improve SVD++ recommendation algorithm. The authors' proposal enhances recommendation performances while ensuring high scalability for large datasets.

Despite the efficiency and scalability of MF-based methods, their performances are limited due to the linearity of the dot product of latent factors involved in the interaction function that does not efficiently model complex user-item interactions. To alleviate this limitation, most often bias terms are added to the interaction function. This idea is useful since a rightly defined interaction function contributes to improve the recommendation quality. However, this approach is not enough to completely jugulate insufficiencies of the inner product of latent factors. To address the linearity issue from the inner product of the MF process, some researches exploit the nonlinearity property of deep neural networks.

2.3 Deep learning in recommender systems

For several years, deep learning succeeds in solving complex tasks. Applied to recommender systems, deep learning brings tremendous opportunities by overcoming limitations of state-of-the-art recommender models [55, 59]. Authors in [54] propose a recommendation model based on a deep neural network. The authors perform a matrix factorization by using the Quadric Polynomial Regression in order to reduce the original user data matrix into low-rank latent features matrices. In [11, 19, 50], the authors propose a recommender system based on a fusion of a baseline matrix factorization based model and a deep neural architecture. Authors in [21, 44] couple a Bayesian approach to a neural architecture to improve recommendation performances. Authors in [39] predict visitor shopping intent by using multilayer perceptron and long-short term memory (LSTM) recurrent neural networks. Their proposal is an accurate and scalable purchasing predictor that supports effective recommendations. Authors in [12] overcome flaws of statistical measures such as Pearson Correlation Coefficient (PCC) and cosine similarity by proposing a neural attentive model that learns the relative importance of historical items in a user profile and therefore improves recommendation performances. In [26], the authors propose an app recommendation based on hierarchical neural networks. To accurately model complex user interactions, the authors' model is developed according to different views namely feature-level attention and view-level attention. Authors in [41] propose a deep collaborative

filtering based recommendation model. They use Latent Dirichlet Allocation (LDA) for the feature extraction from user reviews in order to compute user similarities. Thereafter, they perform recommendations based on matrix factorization coupled with deep learning. Authors in [6] mine user sentiments from user reviews on e-commerce websites to improve recommendation performances. For this purpose, they model the fine-grained user-item interactions by using deep neural networks via LSTM encoder in order to enhance sentiment-aware representations. Authors in [45] propose a hybrid neural architecture to predict ratings. The authors’ architecture combines an autoencoder and a multilayer perceptron. The autoencoder extracts latent features from both users and items while the multilayer perceptron model nonlinear and intricate user-item interactions. Authors in [15] propose a deep hybrid recommendation model that is consisted of a Stacked Denoising Autoencoders- Factorization Machine (SDAE-FM) module for the latent feature extraction, a deep neural network module that effectively captures complex nonlinear user-item interactions, and the metric learning module that assesses the relationship between users and items. Besides the efficiency of neural networks and deep learning, some important aspects such as security and privacy need to be handled when using neural structures [27].

With unlimited capabilities, deep learning brings tremendous flexibility in recommender systems since it can be associated with various conventional models such as matrix factorization, factorization machines, and sparse linear models. Furthermore, its nonlinearity property enable the modeling of intricate user-item interactions that are not linear. Indeed, by using nonlinear activation functions, deep neural networks can accurately catch complex user-item interaction patterns and therefore significantly enhance recommendation performances. However, the latent feature extraction is the crucial step that is prior to the capture of nonlinear user-item interactions, and that therefore needs to be effectively performed. The proposed model efficiently merges the effectiveness of an enhanced matrix factorization with the power of deep neural networks in order to refine recommendations. The improved matrix factorization performs the latent feature extraction while the neural architecture proceeds to the modeling of complex user-item interactions.

The next section presents the proposed method.

3 The proposed recommendation method

In this section, we present a recommendation model and its specifications. The proposed model is an association between an improved Matrix Factorization (MF) model and a novel deep neural architecture.

The next subsection details specifications of our proposal.

3.1 dualDeepMF model presentation

The proposed method lays on a doubly-regularized Matrix Factorization (MF) model that is used for latent features representation. This MF model is biased by including reliable user’s neighborhood terms in order to accurately model user-item interactions. Thereafter, we exploit the nonlinearity property of deep neural networks in order to alleviate limitations from the linear dot product performed during the MF process. The latent features issued from the MF process feed a deep neural network (see Fig. 1). This neural network is mainly a Multilayer Perceptron (MLP) that is consisted of an input layer L_{in} fed by latent features, several hidden layers that enable the nonlinearity of the neural architecture, and an output layer L_{out} . Thereafter, at the merging layer L_{merge} , outcomings of the MLP model are combined with findings of the MF model in order to perform the end-prediction, and the predicted score is assessed regarding training instances.

The next subsection details the proposed MF model.

3.2 Matrix factorization model

The system hosts the set U of m users that interact with at most n items belonging to the set I . Users $u \in U$ rate items $i \in I$ by assigning to them values r_{ui} that express the user’s satisfaction intensity. We define the data matrix $R = [r_{ui}]_{m \times n}$ that contains all ratings assigning by users on items. The matrix factorization assume that R matrix can be approximate by low-rank latent feature matrices as follows:

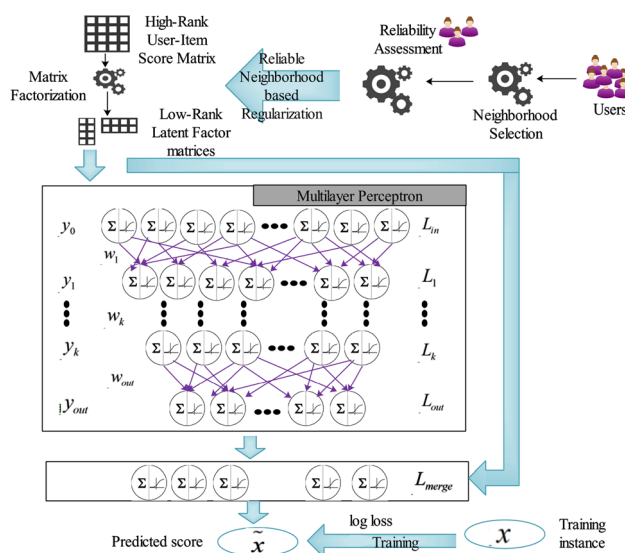


Fig. 1 dualDeepMF Model

$$R \approx PQ \Leftrightarrow r_{ui} \approx \sum_{f \in F} p_{uf} q_{fi}, \tag{1}$$

where $P = [p_{uf}]_{m \times f}$ is the user latent feature matrix, $Q = [q_{fi}]_{f \times n}$ is the item latent feature matrix and $f \ll \min(m, n)$ is the number of latent features. We slightly modify the approximation expression by adding a bias term b_{ui} that compensates for user-interaction variations. The approximation formula is reexpressed as follows:

$$r_{ui} \approx b_{ui} + \sum_{f \in F} p_{uf} q_{fi}, \tag{2}$$

From the factorization process, the loss function is defined as follows:

$$\begin{aligned} \min_{P, Q} \frac{1}{2} \|R - PQ\|_F^2 \\ \Leftrightarrow \min_{P, Q} \frac{1}{2} \sum_{u \in U} \sum_{i \in I} G_{ui} (r_{ui} - b_{ui} - \sum_{f \in F} p_{uf} q_{fi})^2, \end{aligned} \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm, G_{ui} is an indicator function equal to 1 when a user rates an item and 0 otherwise.

A regularization parameter λ is incorporated in the error function in order to alleviate the overfitting problem. For this purpose, the loss function is reexpressed as follows:

$$\begin{aligned} \min_{P, Q} \frac{1}{2} \sum_{u \in U} \sum_{i \in I} G_{ui} ((r_{ui} - b_{ui} - \sum_{f \in F} p_{uf} q_{fi})^2 \\ + \lambda (\sum_{u \in U} \|P_u\|^2 + \sum_{i \in I} \|Q_i\|^2 + \sum_{u \in U} \sum_{i \in I} b_{ui}^2)) \\ \Leftrightarrow \min_{P, Q} \frac{1}{2} \sum_{u \in U} \sum_{i \in I} G_{ui} ((r_{ui} - b_{ui} - \sum_{f \in F} p_{uf} q_{fi})^2 \\ + \lambda (\sum_{u \in U} \sum_{f \in F} p_{uf}^2 + \sum_{i \in I} \sum_{f \in F} q_{fi}^2 + \sum_{u \in U} \sum_{i \in I} b_{ui}^2)), \end{aligned} \tag{4}$$

where P_u and Q_i are respective latent feature vectors of user u and item i .

We perform a second regularization in order to refine the modeling of user-item interactions. For this purpose, we integrate the neighborhood effect into the loss function. Indeed, like-minded users influence each other since they enjoy the same items. Those users are determined after a similarity assessment that is performed by using a weighted Pearson Correlation Coefficient (PCC) computed as follows:

$$sim_{uv} = \frac{JCC_{uv} * \sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}, \tag{5}$$

where \bar{r}_u and \bar{r}_v are rating averages of both users u and v respectively. $JCC_{uv} = \frac{|I_u \cap I_v|}{|I_u| + |I_v| - |I_u \cap I_v|}$ is the Jaccard

correlation coefficient that enables the impact of co-rated items in the PCC measure.

The neighborhood $Near = \{v \in U | sim_{uv} \geq \gamma\}$ is defined as the set of like-minded users that share similar interests in the same items. It can be valuable to select only users that can effectively refine the prediction process. For this purpose, we define a reliability measure that ensures strong user profiles and therefore a reliable neighborhood.

3.2.1 Reliable neighborhood selection

Let C_{uv} be the reliability value between users u and v . C_{uv} expresses how reliable v is according to u . It is computed as follows:

$$C_{uv} = 1 - \frac{\sum_{i \in I_u \cap I_v} \frac{|r_{ui} - r_{vi}|}{|I_u \cap I_v|}}{r_{max}}, \tag{6}$$

where I_u and I_v are sets of items, respectively, selected by u and v , r_{max} is the maximum value of rating. The reliability between u and v decreases when the gap between r_{ui} and r_{vi} increases. It translates a divergence of interests of u and v . Inversely, the reliability between u and v as the gap between r_{ui} and r_{vi} decreases.

We consider users and items, respectively, as nodes and edges of a graph in which users who have co-rated items are linked by an edge. We assume indirect reliable users are linked by an edge. We assume indirect reliable users since if v is reliable to u and user w is reliable to v , then w is indirectly reliable to u [3]. The indirect reliability evaluation enables the data sparsity resilience of our system. We assess indirect reliability scores as follows:

$$C_{uv} = 1 - \frac{d_{uv} + \epsilon}{d_{max}}, \tag{7}$$

where d_{uv} is the trust propagation distance [31] that refers to the number of nodes (users) existing between u and v , d_{max} is the maximum allowable distance between two users, and $\epsilon \ll 1$ is a positive value to avoid a reliability value equal to 0. Looking to the reliability measure in Eq. (7), the reliability between u and v decreases when d_{uv} increases and vice-versa.

The reliability measure is finally updated in order to integrate both direct and indirect aspects of the reliability between users. The reliability measure is reexpressed as follows:

$$C_{uv} = (1 - \frac{\sum_{i \in I_u \cap I_v} \frac{|r_{ui} - r_{vi}|}{|I_u \cap I_v|}}{r_{max}}) (1 - \frac{d_{uv} + \epsilon}{d_{max}}). \tag{8}$$

Given the reliability measure above-computer, the reliable neighborhood is defined as $Near_{rel} = \{v \in U | sim_{uv} \geq \gamma \text{ and } C_{uv} \geq \delta\}$.

3.2.2 Matrix factorization regularization

Like-minded users from the reliable neighborhood tend to enjoy similar items. Since user actions on items are described by latent feature spaces, it can be intuitively considered that the divergence between latent vectors needs to be minimized [51] as follows:

$$\min \frac{1}{2} \sum_{v \in \text{Near}_{rel}} \text{sim}_{uv} \cdot C_{uv} \|P_u - P_v\|_F^2, \quad (9)$$

where P_u and P_v are respective latent feature vectors of users u and v .

Given the reliable neighborhood-based regularization, the loss function is updated as follows:

$$\begin{aligned} & \min_{P, Q} \frac{1}{2} \sum_{u \in U} \sum_{i \in I} G_{ui} \left((r_{ui} - b_{ui} - \sum_{f \in F} p_{uf} q_{fi})^2 \right. \\ & \quad + \lambda \left(\sum_{u \in U} \|P_u\|^2 + \sum_{i \in I} \|Q_i\|^2 + \sum_{u \in U} \sum_{i \in I} b_{ui}^2 \right) \\ & \quad \left. + \omega \sum_{u \in U} \sum_{v \in \text{Near}_{rel}} \text{sim}_{uv} \cdot C_{uv} \|P_u - P_v\|_F^2 \right) \\ \Leftrightarrow & \min_{P, Q} \frac{1}{2} \sum_{u \in U} \sum_{i \in I} G_{ui} \left((r_{ui} - b_{ui} - \sum_{f \in F} p_{uf} q_{fi})^2 \right. \\ & \quad + \lambda \left(\sum_{u \in U} \sum_{f \in F} p_{uf}^2 + \sum_{i \in I} \sum_{f \in F} q_{fi}^2 + \sum_{u \in U} \sum_{i \in I} b_{ui}^2 \right) \\ & \quad \left. + \omega \sum_{u \in U} \sum_{v \in \text{Near}_{rel}} \text{sim}_{uv} \cdot C_{uv} \sum_{f \in F} (p_{uf} - p_{vf})^2 \right). \end{aligned} \quad (10)$$

where ω is an additional parameter that enables the control of the effect of the neighborhood-based regularization.

The interaction function is solved by using the Stochastic Descent of Gradient (SDG) since it is widely adopted as optimization method [57]. This method iteratively updates latent feature spaces regarding the gradient direction of the loss function. The update rules are expressed as follows:

$$\begin{aligned} \frac{\partial D}{\partial P} &= \sum_{u \in U} \sum_{i \in I} G_{ui} \left((r_{ui} - r_{ui}) \sum_{f \in F} q_{fi} + \lambda \sum_{u \in U} \sum_{f \in F} p_{uf} \right. \\ & \quad \left. + \omega \sum_{u \in U} \sum_{v \in \text{Near}_{rel}} \text{sim}_{uv} \cdot C_{uv} \sum_{f \in F} (p_{uf} - p_{vf}) \right), \end{aligned} \quad (11)$$

where D corresponds to the loss function,

$$\frac{\partial D}{\partial Q} = \sum_{u \in U} \sum_{i \in I} G_{ui} \left((r_{ui} - r_{ui}) \sum_{f \in F} p_{uf} + \lambda \sum_{i \in I} \sum_{f \in F} q_{fi} \right), \quad (12)$$

$$\frac{\partial D}{\partial b} = \sum_{u \in U} \sum_{i \in I} G_{ui} \left((r_{ui} - r_{ui}) b_{ui} + \lambda \sum_{u \in U} \sum_{i \in I} b_{ui} \right), \quad (13)$$

where $r_{ui} = b_{ui} + \sum_{f \in F} p_{uf} q_{fi}$.

The optimization process starts with the initialization of P and Q with random positive values. The optimal latent feature matrices that minimize the interaction function are retrieved at the algorithm convergence. Iterations are performed following update rules (in Eqs. (11,12)) and observing a learning rate α .

Algorithm 1 summarizes the latent feature extraction process.

Algorithm 1: Latent Feature Extraction Algorithm

Data:

- R : Original data matrix
- λ, ω : Parameters to avoid overfitting problem
- α : Learning rate
- f : Number of latent features
- P, Q : User and item latent feature matrices
- I, U : Respective sets of items and users

Result: Updated P and Q

```

1 Begin
2 for ( $f \in F$ ), ( $u \in U$ ) and ( $i \in I$ ) do
3   | Initialize  $p_{uf}$  and  $q_{fi}$  with random positive values;
4 end
5 repeat
6   for ( $u \in U$ ) do
7     | for ( $i \in I$ ) do
8       |  $r_{ui} = b_{ui} + \sum_{f \in F} p_{uf} q_{fi}$ ;
9     | end
10  | end
11  for ( $u \in U$ ) do
12    | for ( $i \in I$ ) do
13      | for  $f \in F$  do
14        |  $p_{uf} \leftarrow$ 
15        |  $p_{uf} - \alpha \left( (r_{ui} - r_{ui}) \cdot q_{fi} - \lambda \cdot p_{uf} + \right.$ 
16        |  $\left. \omega \cdot \text{sim}_{uv} C_{uv} \sum_{f \in F} (p_{uf} - p_{vf}) \right)$ ;
17        |  $q_{fi} \leftarrow q_{fi} - \alpha \left( (r_{ui} - r_{ui}) \cdot p_{uf} - \lambda \cdot q_{fi} \right)$ ;
18        |  $b_{ui} \leftarrow b_{ui} - \alpha \left( (r_{ui} - r_{ui}) \cdot b_{ui} - \lambda \cdot b_{ui} \right)$ ;
19      | end
20    | end
21  | end
22 until Convergence;

```

3.3 Deep neural network model

The proposed model is a feedforward neural network that is consisted of an input layer L_{in} , several hidden layers L_k , an output layer L_{out} and a merging layer L_{merge} . The neural network is fed by latent features in order to predict z score in a first step. Thereafter, at the merging layer L_{merge} , the prediction of the multilayer perceptron is combined to outcomes of the doubly-regularized MF model in order to predict \tilde{x} score that is assessed according to the training instance x .

The input layer L_{in} of the neural network is fed by latent features and the input vector is obtained as follows:

$$y_0 = P_u \otimes Q_i, \tag{14}$$

where \otimes is the dot product operator, P_u is the user latent vector and Q_i the item latent vector. The layer L_1 is fed by y_0 and the output vector y_1 of this first hidden layer is computed as follows:

$$y_1 = \rho_1(w_1 y_0 + a_1), \tag{15}$$

where w_1 is the set of weights contained in a matrix between the input layer and the first hidden layer L_1 , a_1 are biases in L_1 layer, and ρ_1 is the activation function that enables the nonlinearity property of the neural network. We use Swish function proposed by Google Brain [37, 49] and that consistently outperforms the most spreadly adopted ReLU function [37]. Unlike ReLU function, Swish activation function is smoother and does not hastily change direction. In addition, it efficiently deals with issues such as vanishing gradients [43]. Swish activation function is defined as follows:

$$\text{Swish}(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1 + e^{-x}}. \tag{16}$$

Generally, outputs of hidden layers L_k are denoted by y_k and obtained as follows:

$$y_k = \rho_k(w_k(\rho_{k-1}(\dots\rho_2(w_2 y_1 + a_2)\dots)) + a_k), \tag{17}$$

where ρ_k , w_k and a_k are, respectively, the activation function, the weight matrix and biases of neurons in the hidden layer L_k . For the output layer L_{out} of the Multilayer Perceptron (MLP), the prediction y_{out} is trivially computed as follows:

$$y_{out} = \rho_{out}(w_{out} y_k + a_{out}), \tag{18}$$

where ρ_{out} is the activation function that is also Swish like for hidden layers.

For the merging layer L_{merge} , the output prediction is performed as a combination between the doubly-regularized MF model and the MLP model. The output at this layer is computed as follows:

$$\tilde{x} = \rho_{merge}(w_{merge}(y_{out} + P_u \otimes Q_i) + a_{merge}), \tag{19}$$

where ρ_{merge} is the Softmax activation function that is appropriated for output of neural network [8]. The prediction of the proposed model is assessed by using the normalized cross-entropy method [47] through the following cost function:

$$E = - \sum_{t=1}^O \frac{x_t}{R_{max}} \log \tilde{x}_t + (1 - \frac{x_t}{R_{max}}) \log(1 - \tilde{x}_t), \tag{20}$$

where R_{max} is the maximum rating values and O is the number of neurons in the merging layer L_{merge} .

The training stage of the proposed model consists to determine optimal weights and biases that minimize the cost function in Eq. (20). For this purpose, we use the gradient descent method through update rules expressed as follows:

$$\begin{aligned} w &\leftarrow w - \gamma \left(\frac{\partial E}{\partial W} \right), \\ a &\leftarrow a - \gamma \left(\frac{\partial E}{\partial A} \right), \end{aligned} \tag{21}$$

where W and A are respectively the weight and bias matrices.

Consider the merging layer L_{merge} , we compute the gradient for weight w_{merge} as follows:

$$\frac{\partial E}{\partial w_{merge}} = \frac{\partial E}{\partial(T_{merge})} \frac{\partial(T_{merge})}{\partial w_{merge}} = \frac{\partial E}{\partial(T_{merge})} y_{out}, \tag{22}$$

where $T_{merge} = w_{merge} y_{out} + a_{merge}$,

$$\begin{aligned} \frac{\partial E}{\partial T_{merge}} &= \frac{\partial E}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial T_{merge}}, \text{ where} \\ \frac{\partial E}{\partial \tilde{x}} &= \frac{1 - \frac{x}{R_{max}}}{1 - \tilde{x}} - \frac{x}{R_{max} \tilde{x}} = \frac{R_{max} \tilde{x} - x}{R_{max} \tilde{x} (1 - \tilde{x})}, \\ \frac{\partial \tilde{x}}{\partial T_{merge}} &= \frac{\partial \text{Soft max}(w_{merge}(y_{out} + P_u \otimes Q_i) + a_{merge})}{T_{merge}} \\ &= \frac{\partial \text{Soft max}(T_{merge} + w_{merge}(P_u \otimes Q_i))}{T_{merge}} \\ &= \tilde{x}(1 - \tilde{x}), \text{ finally} \\ \frac{\partial E}{\partial T_{merge}} &= \frac{R_{max} \tilde{x} - x}{R_{max}}. \end{aligned} \tag{23}$$

For hidden layers L_k , following the same analysis we have:

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial(T_{k+1})} \frac{\partial(T_{k+1})}{\partial T_k}, \tag{24}$$

$$\text{where } \begin{cases} T_k = w_k y_{k-1} + a_k, \\ T_{k+1} = w_{k+1} y_k + a_{k+1} \\ y_k = \text{Swish}(w_k y_{k-1} + a_k) = \text{Swish}(T_k), \\ \frac{\partial T_{k+1}}{\partial T_k} = w_{k+1} (y_k + \frac{y_k}{T_k} (1 - y_k)) \end{cases}$$

Finally, Eq. (24) is reexpressed as follows:

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial T_{k+1}} [w_{k+1} (y_k + \frac{y_k}{T_k} (1 - y_k))]. \tag{25}$$

The gradients for bias a are computed as follows:

$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial T_k} \frac{\partial T_k}{\partial a_k} = \frac{\partial E}{\partial T_k}, \tag{26}$$

with $\frac{\partial T_k}{\partial a_k} = \frac{\partial(w_k y_{k-1} + a_k)}{\partial a_k} = 1$.

Update rules of Eq. (21) are reexpressed as follows:

$$\begin{aligned} w_k &\leftarrow w_k - \gamma \frac{\partial E}{\partial T_k}, \\ b_k &\leftarrow b_k - \gamma \frac{\partial E}{\partial T_k}, \end{aligned} \quad (27)$$

where $\frac{\partial E}{\partial T_k}$ is computed in Eq. (24) for hidden layers and in Eq. (23) for the merging layer.

3.4 The recommendation

Once the training phase achieved, by using the proposed model, the prediction \tilde{r}_{ui} on unselected items is computed as follows:

$$\tilde{r}_{ui} = \arg \max_t (\tilde{x}_t). \quad (28)$$

Predicted items are ranked by decreasing order of ratings. Thereafter, the most relevant items are returned to the end-user.

The next section details the experiments and presents the obtained results.

4 Experiments and results

In this section, performances of the proposed method are assessed comparatively to competing CF-based methods. Experiments are conducted on real-world datasets¹, described in Tab. 1.

The next subsection presents the experiment process.

4.1 Experiments setup

Experimentations are conducted on a computer that hosts an Intel Core i7 (2.4 GHz) typed processor with 16 GB RAM, and that runs Windows 10 Operating System. We develop our algorithm by using TensorFlow and Keras libraries in the Spyder environment since we use Python 3.6 language.

The used datasets contain invocations of movies rated by users according to their relevance. Those datasets are split into a training data part and a test data part. The test part is gradually increased causing a decrease in the training part. In this way, we can assess the data sparseness impact on recommendation precision.

Parameters of the proposed model have been set in order to maximize our proposal's performance. The regularization rate has been set to 0.01. The learning rate has been set to 0.001. We set the number of latent features to 10. We set the size of the reliable neighborhood to 15. In the input layer, the number of units is the sum of latent features that feed the neural network and that are used at the merging

layer. The neural architecture has 3 hidden layers. The two first hidden layers have 32 units for each of them. The third hidden layer has 18 neurons and the merging has 8 neurons.

The next subsection presents the metrics used to evaluate performances of the proposed method.

4.2 Evaluation metrics

We evaluate the prediction precision of the proposed method by using the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) that are spreadly adopted for this concern [57]. Low values of MAE and RMSE indicators express a high prediction precision. MAE and RMSE values are computed as follows:

$$MAE = \frac{\sum_{s \in S_{rec}} |r_{ui} - \tilde{r}_{ui}|}{N}, \quad (29)$$

$$RMSE = \sqrt{\frac{\sum_{s \in S_{rec}} (r_{ui} - \tilde{r}_{ui})^2}{N}}, \quad (30)$$

where r_{ui} is the true rating of u on item i , and \tilde{r}_{ui} the predicted rating of u on item i ; N is the number of recommended items and S_{rec} is the set of recommended items.

We evaluate the recommendation quality by measuring precision and recall indicators. High scores of precision and recall indicators translate a high recommendation quality. Recall measures compute the number of items rightly recommended given the number of items expected to be recommended. Meanwhile, precision measures assess the number of items rightly recommended given the proportion of recommendations. Precision and recall are computed as follows:

$$Precision = \frac{|S_{rec} \cap E|}{S_{rec}}, \quad (31)$$

$$Recall = \frac{|S_{rec} \cap E|}{E}, \quad (32)$$

where S_{rec} is the set of recommendations and E the set of expected items.

4.3 Results and analysis

The dualDeepMF method proposed in this paper is assessed comparatively to state-of-the-art recommendation methods hereafter detailed:

- IPCC is a memory-based method that is based on the evaluation of inter-item similarities [53]. We apply on this approach the trustworthiness evaluation in order to enhance the recommendation quality.
- NeuMF is a neural model-based method that uses the generalized matrix factorization combined to a neural

¹ <https://www.librec.net/datasets.html>.

Table 1 Datasets Specifications

Dataset	Number of users	Number of items	Number of ratings
Movielens	1.000	1.700	100.000
Filmtrust	1.508	2.071	35.497

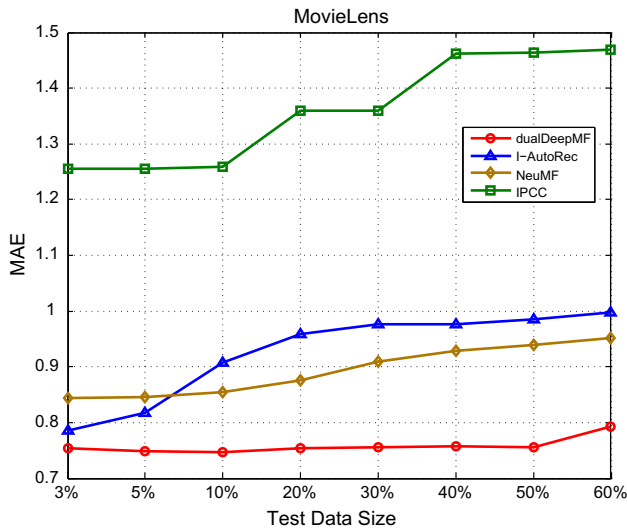


Fig. 2 MAE Performances on MovieLens

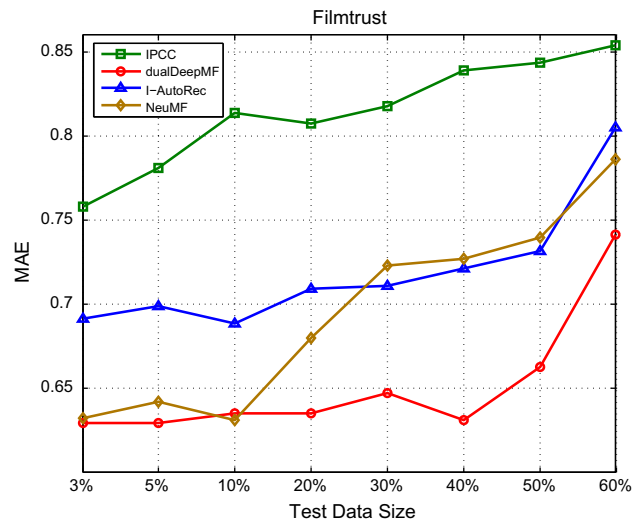


Fig. 4 MAE Performances on Filmtrust

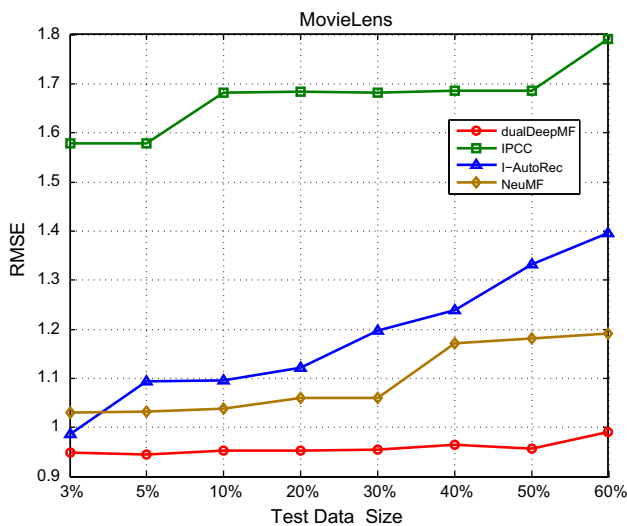


Fig. 3 RSME Performances on MovieLens

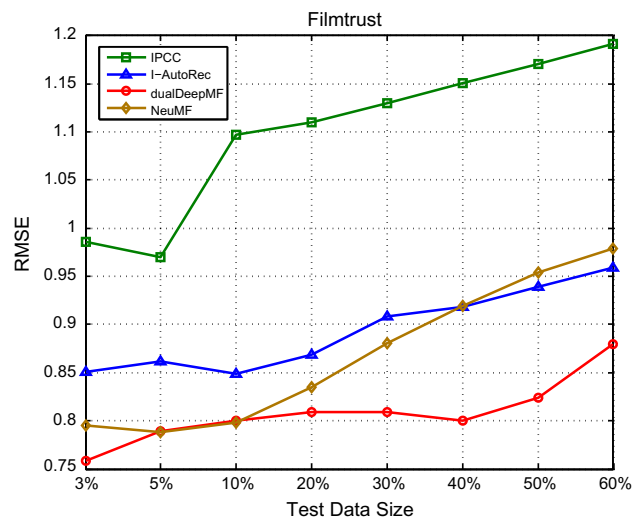


Fig. 5 RMSE Performances on Filmtrust

architecture in order to perform the recommendation [11].

- I-AutoRec is a neural model-based method that performs recommendations by employing an autoencoder to extract latent features that thereafter feed a neural network which makes predictions of unknown items [57].

4.3.1 Assessment of the item prediction accuracy

Using the MovieLens dataset, Figs. 2 and 3 depict MAE and RMSE performances of the proposed method. It can be observed that our proposal outperforms other methods in terms of prediction precision. Using the Filmtrust dataset, Figs. 4 and 5 show that the proposed method has a better prediction accuracy compared to others. In addition, it can

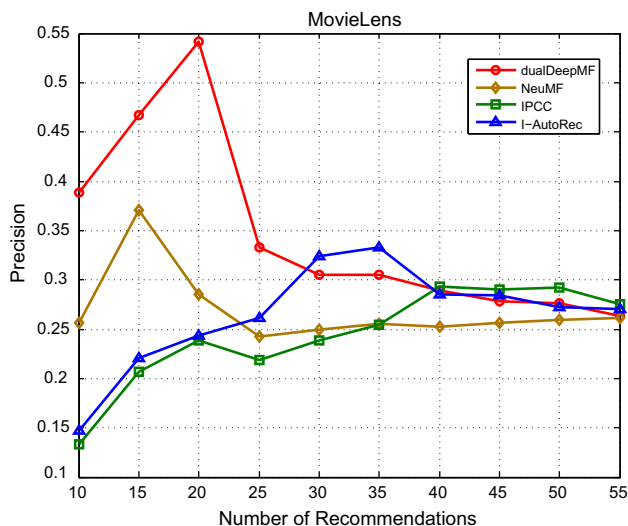


Fig. 6 MAE Performances on MovieLens

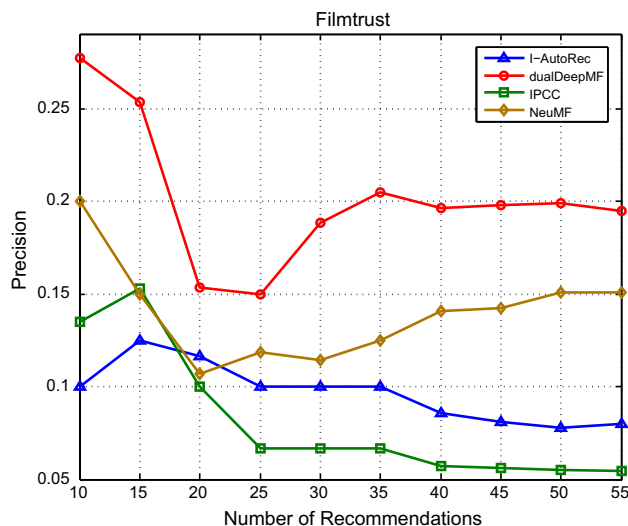


Fig. 8 MAE Performances on Filmtrust

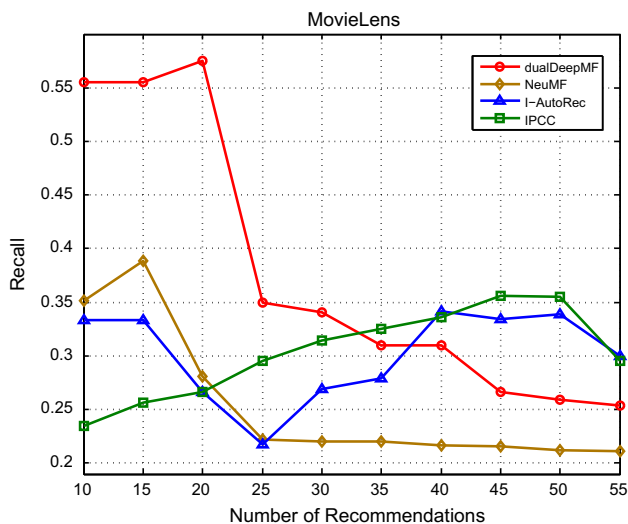


Fig. 7 RSME Performances on MovieLens

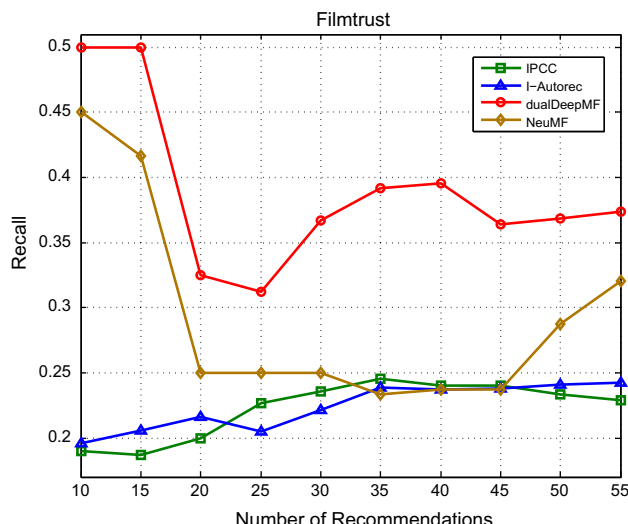


Fig. 9 RMSE Performances on Filmtrust

be observed that the prediction accuracy of all methods globally decreases when the test data part increases. It highlights the robustness and the scalability of our proposal in challenging conditions.

4.3.2 Assessment of the recommendation quality

Using the MovieLens dataset, Figs. 6 and 7 show the precision and recall performances of the proposed method. It can be observed that the precision and recall measures of our proposal are better than those of other methods. Indeed, the recommendation quality of the proposed method is the highest for 20 recommendations. Upper than 20 recommendations, the precision and recall decrease. It can be due to the fact that a number of recommendations upper than 20

include items with poor predicted ratings and that therefore appear as noisy recommended items.

Using the Filmtrust dataset, Figs. 8 and 9 show that the global precision and recall trends of the proposed method are better than those of other methods. The recommendation quality of our proposal is highest for a number of recommendations comprise between 10 and 15. Upper than 15 recommendations, the precision and recall trends decrease. It can be explained by the fact that the additional items behave as noise since they contribute with lower ratings.

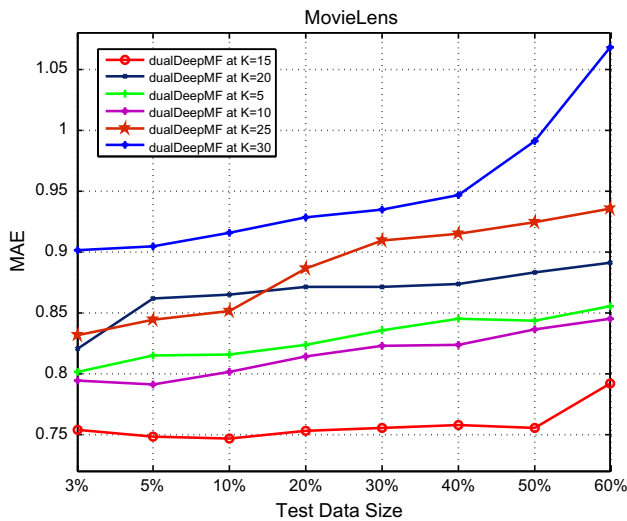


Fig. 10 Impact of the Reliable Neighborhood Size on MAE Performances on MovieLens

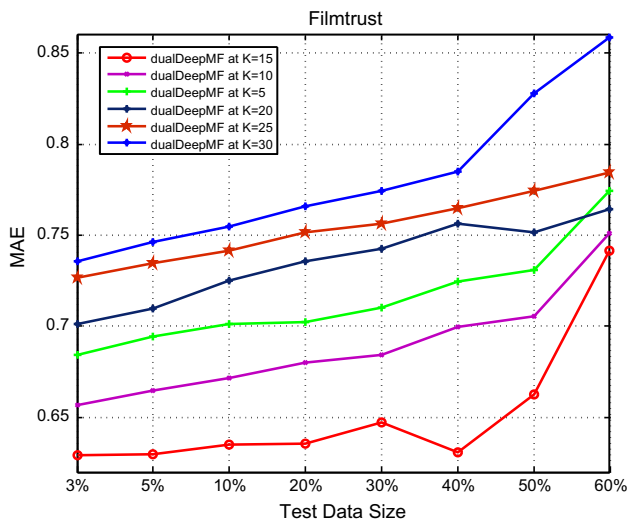


Fig. 11 Impact of the Reliable Neighborhood Size on MAE Performances on Filmtrust

4.3.3 Impact of the reliable neighborhood

Looking at Figs. 10 and 11, it can be observed the impact of the size of the reliable neighborhood using both MovieLens and Filmtrust datasets. It can be observed that for a neighborhood size upper than 15 the prediction accuracy is affected. It can be observed by the fact, for a neighborhood size upper than 15, the reliability of additional users is poor. Therefore, those users are considered as doubtful users since their contribution to the recommendation process affects the prediction accuracy.

5 Conclusion and perspectives

In this paper, we have proposed a recommender system to effectively address the information overload. The proposed model lays on an enhanced matrix factorization (MF) coupled to a novel deep neural architecture. The MF model developed is doubly regularized with both biases and reliable user’s neighborhood to accurately model user-item interactions. Thereafter, the nonlinearity of the proposed deep neural network is used to alleviate the limitations of the linear dot product involved in the MF process. Series of experiments have been performed on real-world datasets and show the effectiveness of our proposal compared to state-of-the-art recommendation methods in terms of accuracy and quality of the recommendation.

In the future, to further refine the recommendation, the proposed method could be extended by mining user’s opinions or by analyzing user’s sentiment through users’ activities on social networks. The proposed model can be extended by mining user’s opinions since words in users’ reviews or comments on social networks can be scored by using a sentiment lexicon to transform words into scores expressing the users’ appreciation level about items. The users’ reviews expressed into scores rating users’ satisfaction about items make ease the implementation of our recommendation proposal to predict users’ expectations about unknown items. The subjects on what users comment can be correlated to items likely to be interesting for users who left comments about them. For this purpose, recurrent neural networks could be useful to explore in order to mine textual reviews and comments left by users concerning selected items. Neural networks offer tremendous promises in the recommendation field. However, security and privacy aspects need to be also considered since some sensible training data can be inferred or the system can be vulnerable and wrongly performs predictions.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ahamed MT, Afroge S (2019) A recommender system based on deep neural network and matrix factorization for collaborative filtering. pp 1–5
2. Ahmadian S, Meghdadi M, Afsharchi M (2018a) Incorporating reliable virtual ratings into social recommendation systems. *Appl Intell* 48(11):4448–4469
3. Ahmadian S, Meghdadi M, Afsharchi M (2018b) A social recommendation method based on an adaptive neighbor selection mechanism. *Inf Process Manag* 54(4):707–725
4. Ahmadian S, Afsharchi M, Meghdadi M (2019) A novel approach based on multi-view reliability measures to alleviate

- data sparsity in recommender systems. *Multimed Tools Appl* 78(13):17763–17798
5. Birtolo C, Ronca D (2013) Advances in clustering collaborative filtering by means of fuzzy c-means and trust. *Expert Syst Appl* 40(17):6997–7009
 6. Da'u A, Salim N (2019) Sentiment-aware deep recommender system with neural attention networks. *IEEE Access* 7:45472–45484
 7. Du R, Lu J, Cai H (2019) Double regularization matrix factorization recommendation algorithm. *IEEE Access* 7:139668–139677
 8. Galushkin AI (2007) *Neural Network Theory*. Springer-Verlag, Berlin, Heidelberg
 9. Goldberg Y (2016) A primer on neural network models for natural language processing. *J Artif Intell Res* 57:345–420
 10. Guo G, Zhang J, Yorke-Smith N (2016) A novel recommendation model regularized with user trust and item ratings. *IEEE Trans Knowledg Data Eng* 28(7):1607–1620
 11. 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, Republic and Canton of Geneva, CHE, WWW '17*, p 173–182, 10.1145/3038912.3052569
 12. He X, He Z, Song J, Liu Z, Jiang YG, Chua TS (2018) Nais: Neural attentive item similarity model for recommendation. *IEEE Trans Knowl Data Eng* 30(12):2354–2366
 13. Hernando A, Bobadilla J, Ortega F (2016) A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowl-Based Syst* 97:188–202
 14. Hong R, Hu Z, Liu L, Wang M, Yan S, Tian Q (2015) Understanding blooming human groups in social networks. *IEEE Trans Multimed* 17(11):1980–1988
 15. Huang Z, Yu C, Ni J, Liu H, Zeng C, Tang Y (2019) An efficient hybrid recommendation model with deep neural networks. *IEEE Access* 7:137900–137912
 16. Jayapriya K, Mary NAB, Rajesh RS (2016) Cloud service recommendation based on a correlated QoS ranking prediction. *J Netw Syst Manag* 24(4):916–943
 17. Jiao J, Zhang X, Li F, Wang Y (2019) A novel learning rate function and its application on the svd++ recommendation algorithm. *IEEE Access* 8:14112–14122
 18. Li K, Zhou X, Lin F, Zeng W, Alterovitz G (2019) Deep probabilistic matrix factorization framework for online collaborative filtering. *IEEE Access* 7:56117–56128. <https://doi.org/10.1109/ACCESS.2019.2900698>
 19. Kapetanakis S, Polatidis N, Alshammari G, Petridis M (2019) A novel recommendation method based on general matrix factorization and artificial neural networks. *Neural Comp Appl* 32(16):12327–34
 20. Kluver D, Ekstrand MD, Konstan JA (2018) Rating-based collaborative filtering: algorithms and evaluation. *Social Inf Access*. https://doi.org/10.1007/978-3-319-90092-6_10
 21. Ko YJ, Maystre L, Grossglauser M (2016) Collaborative recurrent neural networks for dynamic recommender systems. In: *Journal of Machine Learning Research: Workshop and conference proceedings* 63
 22. Lara-Cabrera R, González-Prieto Á, Ortega F, Bobadilla J (2020) Evolving matrix-factorization-based collaborative filtering using genetic programming. *Appl Sci* 10(2):675
 23. Li G, Zhu T, Hua J, Yuan T, Niu Z, Li T, Zhang H (2019a) Asking images: Hybrid recommendation system for tourist spots by hierarchical sampling statistics and multimodal visual bayesian personalized ranking. *IEEE Access* 7:126539–126560
 24. Li W, Zhou X, Shimizu S, Xin M, Jiang J, Gao H, Jin Q (2019b) Personalization recommendation algorithm based on trust correlation degree and matrix factorization. *IEEE Access* 7:45451–45459
 25. Lian D, Xie X, Chen E (2019) Discrete matrix factorization and extension for fast item recommendation. *IEEE Trans Knowl Data Eng*
 26. Liang T, Zheng L, Chen L, Wan Y, Philip SY, Wu J (2020) Multi-view factorization machines for mobile app recommendation based on hierarchical attention. *Knowl-Based Syst* 187:104821
 27. Liu X, Xie L, Wang Y, Zou J, Xiong J, Ying Z, Vasilakos AV (2020) Privacy and security issues in deep learning: a survey. *IEEE Access* 9:4566–4593
 28. Luo X, Zhou M, Xia Y, Zhu Q (2014) An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Trans Industr Inf* 10(2):1273–1284
 29. Luo X, Zhou M, Li S, You Z, Xia Y, Zhu Q (2015) A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method. *IEEE Trans Neural Netw Learn Syst* 27(3):579–592
 30. Ma X, Guo D, Cui L, Li X, Jiang X, Chen X (2019) Som clustering collaborative filtering algorithm based on singular value decomposition. pp 61–65
 31. Massa P, Avesani P (2005) Controversial Users Demand Local Trust Metrics: An Experimental Study on Epinions.Com Community. In: *Proceedings of the 20th national conference on artificial intelligence - Volume 1, AAAI Press, AAAI'05*, pp 121–126, <http://dl.acm.org/citation.cfm?id=1619332.1619354>, event-place: Pittsburgh, Pennsylvania
 32. Ortega F, Hernando A, Bobadilla J, Kang JH (2016) Recommending items to group of users using Matrix Factorization based Collaborative Filtering. *Inf Sci* 345:313–324
 33. Valdiviezo-Diaz P, Ortega F, Cobos E, Lara-Cabrera R (2019) A collaborative filtering approach based on Naive Bayes Classifier. *IEEE Access* 7:108581–108592. <https://doi.org/10.1109/ACCESS.2019.2933048>
 34. Parvin H, Moradi P, Esmaeili S, Qader NN (2019a) A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method. *Knowl-Based Syst* 166:92–107
 35. Parvin H, Moradi P, Esmaeili S, Qader NN (2019) A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method. *Knowl-Based Syst* 166:92–107
 36. Du R, Lu J, Cai H (2019) Double regularization matrix factorization recommendation algorithm. *IEEE Access* 7:139668–139677. <https://doi.org/10.1109/ACCESS.2019.2943600>
 37. Ramachandran P, Zoph B, Le QV (2017) Swish: a self-gated activation function. *arXiv preprint arXiv:171005941:7*
 38. Rastegari M, Ordenez V, Redmon J, Farhadi A (2016) Xnor-net: Imagenet classification using binary convolutional neural networks. pp 525–542
 39. Sakar CO, Polat SO, Katircioglu M, Kastro Y (2019) Real-time prediction of online shoppers' purchasing intention using multi-layer perceptron and lstm recurrent neural networks. *Neural Comput Appl* 31(10):6893–6908
 40. Salakhutdinov R, Mnih A (2007) Probabilistic matrix factorization, *NIPS'07*. Curran Associates Red Hook, NY, USA, pp 1257–1264
 41. Shoja BM, Tabrizi N (2019) Customer reviews analysis with deep neural networks for e-commerce recommender systems. *IEEE Access* 7:119121–119130
 42. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques
 43. Tripathi GC, Rawat M, Rawat K (2019) Swish activation based deep neural network predistorter for rf-pa. pp 1239–1242

44. 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, association for computing machinery, New York, NY, USA, KDD '15, p 1235–1244, 10.1145/2783258.2783273
45. Wang Q, Peng B, Shi X, Shang T, Shang M (2019) Dccr: Deep collaborative conjunctive recommender for rating prediction. *IEEE Access* 7:60186–60198
46. Wen S, Wang C, Li H, Zheng G et al (2019) Parallel naïve bayes regression model-based collaborative filtering recommendation algorithm and its realisation on hadoop for big data. *Int J Inf Technol Manage* 18(2/3):129–142
47. Xue HJ, Dai X, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: *IJCAI*, pp 3203–3209
48. Zhang Y, Meng K, Kong W, Dong ZY, Qian F (2019) Bayesian hybrid collaborative filtering-based residential electricity plan recommender system. *IEEE Trans Industr Inf* 15(8):4731–4741. <https://doi.org/10.1109/TII.2019.2917318>
49. Yang S, Hao K, Ding Y, Liu J (2018) Vehicle driving direction control based on compressed network. *Int J Pattern Recognit Artif Intell* 32(08):1850025
50. Yi B, Shen X, Liu H, Zhang Z, Zhang W, Liu S, Xiong N (2019) Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Trans Industr Inf* 15(8):4591–4601
51. Yin J, Lo W, Deng S, Li Y, Wu Z, Xiong N (2014) Colbar: A collaborative location-based regularization framework for qos prediction. *Inf Sci* 265:68–84
52. Yu J, Xuan Z, Feng X, Zou Q, Wang L (2019) A novel collaborative filtering model for lncrna-disease association prediction based on the naïve bayesian classifier. *BMC Bioinf* 20(1):396
53. Yang Z, Wu B, Zheng K, Wang X, Lei L (2016) A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access* 4:3273–3287. <https://doi.org/10.1109/ACCESS.2016.2573314>
54. Zhang L, Luo T, Zhang F, Wu Y (2018) A recommendation model based on deep neural network. *IEEE Access* 6:9454–9463
55. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. *ACM Computing Surveys (CSUR)* 52(1):1–38
56. Zheng X, Xu LD, Chai S (2017) Qos recommendation in cloud services. *IEEE Access* 5:5171–5177. <https://doi.org/10.1109/ACCESS.2017.2695657>
57. Zheng Z, Xiaoli L, Tang M, Xie F, Lyu MR (2020) Web service qos prediction via collaborative filtering: a survey. *IEEE Trans Services Comput*
58. Zhong S, Ying W, Chen X, Fu Q (2020) An adaptive similarity-measuring-based cmab model for recommendation system. *IEEE Access* 8:42550–42561
59. Zi Y, Li Y, Sun H (2018) Research of personalized recommendation system based on multi-view deep neural networks. pp 514–529

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.