International Journal of Machine Learning and Cybernetics (2021) 12:1993–2005
https://doi.org/10.1007/s13042-021-01288-7

ORIGINAL ARTICLE

DLSA: dual-learning based on self-attention for rating prediction

Fulan Qian[1,2] · Yafan Huang[1,2] · Jianhong Li[1,2,3] · Chengjun Wang[3] · Shu Zhao[1,2] · Yanping Zhang[1,2]

Received: 9 March 2020 / Accepted: 16 February 2021 / Published online: 18 May 2021

Abstract

Latent factor models (LFMs) have been widely applied in many rating recommendation systems because of their prediction rating capability. Nevertheless, LFMs may not fully leverage rating information and lack good recommendation performance. Furthermore, many subsequent works have often used auxiliary text information, such as user attributes, to improve the prediction effect. However, they did not fully utilize implicit information (i.e., users' preferences, items' common features), and additional information is sometimes difficult to acquire. In this paper, we propose a new framework, named dual-learning based on self-attention for rating prediction (DLSA), to solve these problems. Self-attention has a proven ability to learn implicit information about sentences in machine translation, which can be used to mine implicit information in recommendation systems. Additionally, dual learning has shown that the model can generate feedback information when it learns from unlabeled data; therefore, we were inspired to use it in recommendation and obtain implicit information feedback. From the user's perspective, we design a user self-attention model to learn user-user implicit information and create an interactive user-item self-attention mechanism to learn user-item information. We can also obtain item self-attention to utilize item-item information and an item-user self-attention model to acquire item-user information from an item's perspective. The interactive structure of the user-item and item-user can adopt the dual learning mechanism to learn implicit information feedback. Moreover, no auxiliary text information was used in the process. The proposed model combines the power of self-attention for implicit information and dual learning for information feedback in a new neural network architecture. Experiments on several real-world datasets demonstrate the effectiveness of DLSA over competitive algorithms on rating recommendation.

Keywords Rating recommendation · Self-attention distribution · Dual learning

Fulan Qian and Jianhong Li have contributed equally.

✉ Fulan Qian
qianfulan@hotmail.com

Yafan Huang
1172796360@qq.com

Jianhong Li
1659117121@qq.com

Chengjun Wang
cumt1279@163.com

Shu Zhao
zhaoshuzs2002@hotmail.com

Yanping Zhang
zhangyp2@gmail.com

[1] School of Computer Science and Technology, Anhui University, Hefei, China

[2] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Hefei, China

[3] School of Artificial Intelligence, Anhui University of Science and Technology, Huainan, China

1 Introduction

In recent years, rating recommendation systems have attracted much attention from both the research community and industry. Many systems are often based on collaborative filtering (CF) [1, 2], and the core idea is to learn users' preferences against items from past interaction records. Latent factor models (LFMs) based on CF are widely used and have achieved great solutions for rating recommendations. For instance, Koren et al. [3] introduced methods based on matrix factorization (SVD),that can directly obtain preference scores by computing the inner product of user and item vectors. Wu et al. designed the user score probability and item type based on CF [4], and Koren et al. [5] proposed SVD technology to focus on information feedback (i.e., rating information) (SVD++); namely, it obtains expressions of user preferences for items from ratings. However, LFMs also encounter some problems in rating recommendations. Acquiring good recommendation effects with explicit

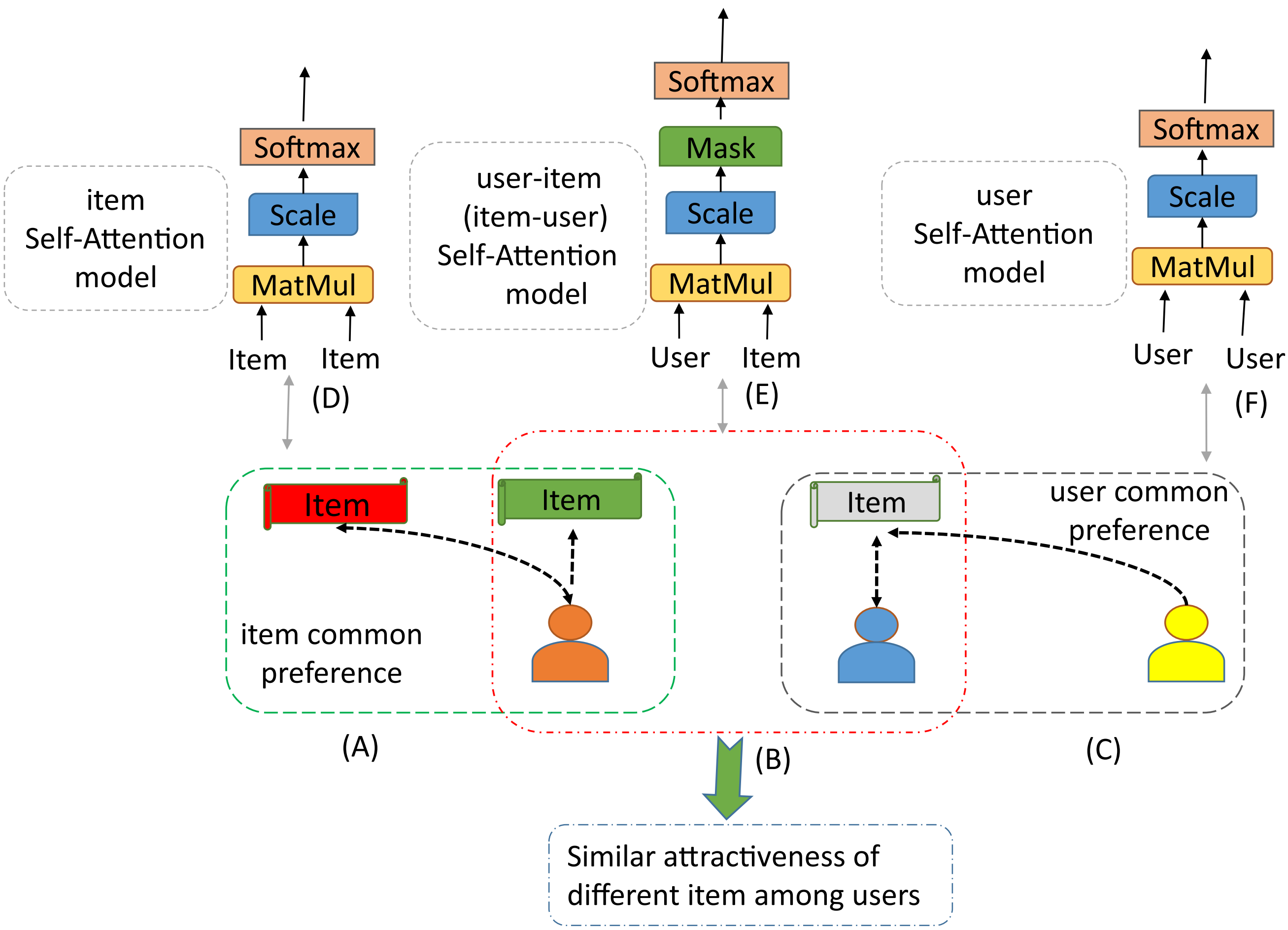

Fig. 1 Illustration of user's preference and self-attention model

feedback is typically difficult in many real applications and has inspired rating recommendation systems to exploit more abundant implicit feedback from rating information. Additionally, LFMs do not consider other inside information, such as that of users who have browsed the same items (Fig. 1c illustrates it).

With the development of technologies, LeCun et al. [6] proposed deep learning to illustrate its powerful capacity for learning feature representation in many fields, such as computer vision and natural language processing. Many researchers have attempted to use these techniques to obtain implicit information and achieve better recommendations. For example, Liu et al. [7] employed CF based on a Restricted Boltzmann Machine (RBM) to combine deep learning and CF. Zhang et al. [8] proposed deep neural networks (DNNs) for rating recommendations, the key idea of which is to learn user preferences from DNNs and quadric polynomial regression. Furthermore, some researchers have utilized text information to improve rating recommendation efficiency. A hybrid collaborative filtering model with a deep structure for recommendation systems has been proposed [9] that uses a stacked denoising autoencoder (SDAE) [10] to acquire users' preferences from side information, i.e., user age and item type. Liu et al. [11] adopted the recommender systems with a heterogeneous side information algorithm (HIRE) to address similar side information to improve recommendation accuracy. Moreover, attribute mapping and autoencoder neural network based matrix factorization initialization methods have been designed [12] (IAI) that use item attributes and an autoencoder to provide recommendations. DeepAE[13] and Wu et al. [14] adopted deep learning to learn user preferences with text information .In essence, these methods utilize deep learning to learn single-rating information and text side information for better recommendations. Similar to LFMs, deep learning methods do not consider other interaction information (e.g. item-item). Moreover, acquiring additional text information is sometimes difficult.

To address the above mentioned challenges, this paper proposes a novel rating recommendation framework **D**ual-**L**earning based on **S**elf-**A**ttention (DLSA), which is illustrated in Fig.3. The core idea is to combine dual learning with self-attention. The self-attention mechanism [15] has been used to find relations inside the sequence, which inspired us to obtain implicit information in the rating information sequence, and dual learning can generate information feedback [16]. Thus, we can learn the preference information of each user through dual learning and self-attention. Specifically, in addition to primitive single-rating information, we propose a user self-attention module for obtaining user-user information. We also design an interactive user-item self-attention mechanism to automatically discriminate the importance of difference items for users. Similarly, we can obtain item self-attention and item-user self-attention to characterize items. Therefore, the user-item and item-user modules can be used by dual learning to generate interactive rating information feedback, which can enhance preference information effectiveness and improve recommendation performance.

The main contributions of this paper can be summarized as follows:

1. By integrating user self-attention based on users, item self-attention based on items and an interaction self-attention model, different implicit information can be extracted from users and items, and thus, more implicit user preferences can be obtained.
2. We adopt the dual learning mechanism to learn information feedback from users and items. Thus, preferences information feedback is obtained, which is beneficial to rating recommendations.
3. We also construct the user-user, item-item, user-item and item-user rating information module, so our model can fully utilize rating information and achieve better results even without additional text information.
4. Finally, we naturally leverage the concept of dual learning, the self-attention framework and the rating information module to improve the recommendation accuracy. Experiments demonstrate the effectiveness of the model and indicate that our model outperforms other methods used for rating recommendation.

The remainder of this paper is organized as follows. In Sect. 2, background knowledge is provided, including information on LFMs, deep learning technologies for recommendation, and notations. We provide a detailed description of dual learning based on the self-attention algorithm in Sect. 3. Section 4 provides the rationale for using dual learning to learn user preferences. Section 5 contains a description of the datasets, measurement metrics, and the experimental results and analysis. We provide a brief conclusion and discussion of future work in Sect. 6.

2 Related work

LFMs and extended algorithms have been very popular, as their performance exceeds other methods in many benchmark datasets, They include probabilistic matrix factorization (PMF) [17], Bayesian Probabilistic Matrix Factorization (BPMF) [18], Nonnegative Matrix Factorization (NMF) [19] and Adaptive nonnegative matrix factorization [20]. All these models learn only from rating information. Then, some other methods [21] that can absorb enriched text information were proposed, namely, factorization machine [22] and content-aware matrix factorization [23]. They are popular due to their good performance, but auxiliary text information is difficult to obtain.

As deep learning techniques have gained much success in many domains, much effort has been made to introduce deep learning techniques to rating recommendations, such as collaborative filtering for the ratings based on deep learning [24], Neural variational collaborative filtering [25], latent factor models based on deep learning [26] (HLFM). For instance, the probabilistic rating autoencoder (PRA) [27] uses an autoencoder to generate latent user feature profiles. Representation learning via a dual-autoencoder for recommendation (ReDa) [28] learns new hidden representations of users and items. The Variational AutoEncoder for directed acyclic graphs (D-VAE) [29] uses graph neural networks [30] to encode directed acyclic graphs. Qian et al. proposed deep hybrid model [31] for rating recommendation (DeepHM), the core idea of which is based on wide and deep learning [32] for APP recommendation. In general, these deep learning models have been applied to feature extraction and prediction, but they consider only interaction information and not user-user and item-item information, namely, users' relations and items' relations.

The attention mechanism has been shown to be very effective in machine translation and question answering systems, and attention-based recommendation models have been designed. For example, attentive collaborative filtering (ACF) [33] integrates LFMs and component-level attention. The attention-drive factor model for explainable personalized recommendation (AFM) [34] combines LFMs and user's attention distributions on different item features. Attention-based methods for recommendation model [35] obtain users' interest by learning context information. Deep attention user-based collaborative filtering for recommendation (Deep-UCF) [36] mines the complex relationships between users and items from historical data. In addition, Peng et al. [37] designed A hybrid model for recommendation supported by self-attention mechanism (HARSAM) to model user interaction data, the core of which is to model user interaction data and

Table 1 The main notations of this paper

Symbol	Description
u	User
i	Item
l	The number of layers
f	Activation
$\odot$	Dot product
m	Pooling result
$\otimes$	Concat operation

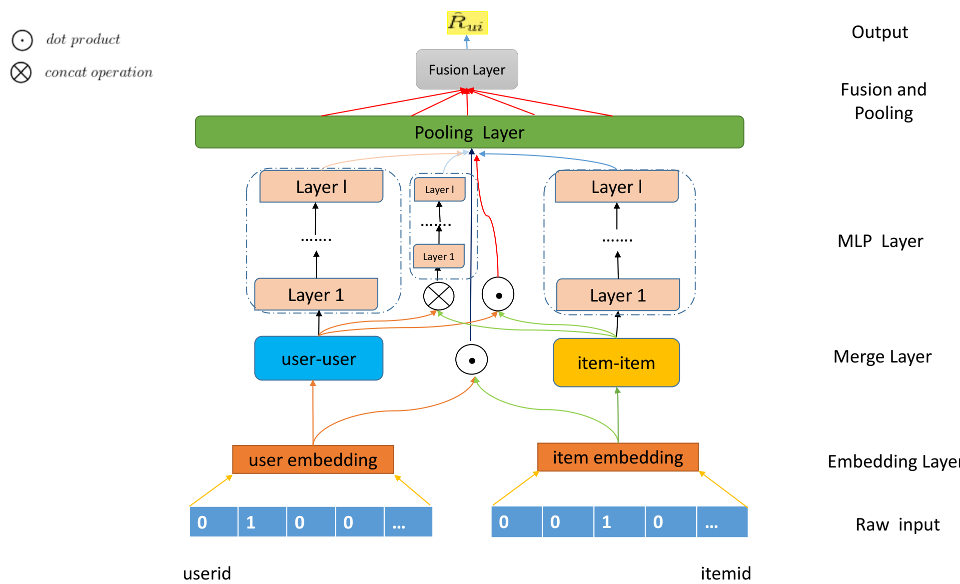

Fig. 2 The basic framework of dual-learning of user and item

acquire user's preference by self-attention. Lv et al. [38] proposed Attention-based item collaborative filtering (AICF) , which adopts an attention mechanism to achieve the weights of items and users. Pang et al. [39] proposed a novel recommender with an attention-based convolutional neural network and factorization machines (ACNN-FM) in which they use different levels of attention mechanisms to extract hidden features of users and items in text information. Moreover, factorization machines [40] have been designed to analyze the association between the hidden features of users and items.

Although attention-based models can gain better results, some of them need text information or consist of only user-item rating information. Can we use an attention mechanism if there is no related text information in the rating recommendation? The answer is "yes". DLSA can not only construct an attention model but also achieve great results even without additional text information. Furthermore, the model fully utilizes rating information, user information and item information.

To better understand the core ideas of DLSA, we describe the main notations used in this paper, as illustrated in Table 1. We use different fonts to represent scalars, vectors, and matrices. Scalars are shown in regular font, matrices in bold font and vectors in bold italic font.

3 Dual-learning based on self-attention

In this section, we first present the dual-learning of user and item (DLUI), which is the framework of DLSA, and then explain how these modules serve as a rating recommendation framework. Finally, we fuse these modules to predict ratings through the DLSA model, which has been trained

3.1 DLUI

Deep learning possesses powerful feature representation capability, so it is easy to assume that multilayer perceptrons (MLPs) can extract user-user, item-item and user-item implicit information. Figure 2 shows the structure of DLUI. We can see that it includes an embedding layer, a merge layer, an MLP layer, a pooling layer, a fusion layer and output.

Apart from rating information, we adopt the merge layer to fuse users for their relation information and do the same for the items. Thus, the neural network-based model learns implicit feedback by utilizing all interactive information. In this section, we provide a detailed introduction for the embedding, merge and MLP layers. The remaining layers are shown in the self-attention model.

Embedding layer DLUI requires user and item information as input. The embedding layer extracts latent representations in vector format from original inputs, the core technology of which is word2vec [41]. Each user and item can be represented by a user-specific embedding and item-specific embedding. Then, we can obtain *uemb* for the user as a set of feature vectors. Similarly, *iemb* for items as a set of feature vectors can be acquired. Here, we use a simple equation to express the embedding layer operation as follows:

$$uemb = embedding\_lookup(userid) \tag{1}$$

$$iemb = embedding\_lookup(itemid) \tag{2}$$

where *embedding_lookup* is the embedding operation.

Merge layer In addition to rating information, user-user relation information and item-item relation information can also reflect implicit user preferences. Figure 1a shows the situation among users in which different items may exist similarly for one user. Therefore, item-item representation could be defined as follows:

$$i_c i = iemb \otimes iemb. \tag{3}$$

Regarding Fig. 1c, it is easy to understand that some people may have similar preferences when they are focused on the same items, so user relations can be formulated by:

$$u_c u = uemb \otimes uemb. \tag{4}$$

MLP The MLP layer is designed to learn implicit information and output, and the following equation shows the calculation process:

$$\left(u_c u\right)_{mlp}^{l+1} = f\left(W_{\left(u_c u\right)_{mlp}^{l}} \left(u_c u\right)_{mlp}^{l} + b_{\left(u_c u\right)_{mlp}^{l}}\right)) \tag{5}$$

$$\left(i_c i\right)_{mlp}^{l+1} = f\left(W_{\left(i_c i\right)_{mlp}^{l}} \left(i_c i\right)_{mlp}^{l} + b_{\left(i_c i\right)_{mlp}^{l}}\right) \tag{6}$$

where $W_{(i_c i)_{mlp}}$, $W_{(u_c u)_{mlp}}$, $b_{(i_c i)_{mlp}}$ and $b_{(u_c u)_{mlp}}$ are the item weight, user weight, item bias and user bias, respectively.

We directly learn single-rating information through a dot product and MLP layer, and the formulation is shown by:

$$u_d i = uemb \odot iemb \tag{7}$$

$$u_c i = \left(u_c u\right) \otimes \left(i_c i\right) \tag{8}$$

$$\left(u_c i\right)_{mlp}^{l+1} = f\left(W_{\left(u_c i\right)_{mlp}^{l}} \left(u_c i\right)_{mlp}^{l} + b_{\left(u_c i\right)_{mlp}^{l}}\right) \tag{9}$$

where $W_{(u_c i)_{mlp}}$ and $b_{(u_c i)_{mlp}}$ represent the weight and bias, respecially.

We can obtain the DLUI prediction formulation when output results have pooled results by Eq. (21), and the expression is as follows:

$$\hat{R}_{dl} = m_{u_d i} + m_{u_c i} + (m)_{ui}^{mlp} + (m)_{uu}^{mlp} + (m)_{ii}^{mlp} \tag{10}$$

where Eqs. (5)–(9) pooled results are $m_{u_d i}$,$m_{u_c i}$, $(m)_{ui}^{mlp}$, $(m)_{uu}^{mlp}$ and $(m)_{ii}^{mlp}$, respectively, and $\hat{R}_{dl}$ is the output prediction rating.

However, DLUI cannot obtain great performance because this method does not have the capacity to capture different attention weights for related items on a specified user; namely, DLUI does not achieve more preference information that is beneficial to rating recommendations. Thus, we need to use another method instead of DLUI. Next, we show the self-attention mechanism.

3.2 Self-attention model

Self-attention has been emphasized in machine translation, and simultaneously, the attention function has been computed on a set of queries to create a matrix **Q**. The keys and values were also assembled into matrices **K** and **V**, respectively. Thus, the model computes the matrix of outputs as:

$$Attention(Q, K, V) = soft\max\left(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V} \tag{11}$$

where **Q**, **K**, **V** represent queries, key and values. $\mathrm{d_k}$ denotes dimension, and softmax is the activation.

Because of the self-attention model, **Q**, **K** and **V** require **Q** = **K** = **V**. It is easy to understand in machine translation that **Q**, **K** and **V** represent the sequence. However, for the triple (*user*, *item*, and *rating*), it cannot denote *user*=*item*=*rating*. *user* and *item* are nodes, and *rating* is the value. Therefore, the self-attention model is not applicable for triples, and we need to redefine it.

3.3 DLSA

The DLSA framework includes user self-attention, item self-attention and interaction self-attention models, as illustrated in Fig. 3. Next, we introduce them one by one.

User self-attention model A user's preferences for one item could be impacted by other people, which motivates us to design a user self-attention model to obtain users' preferences in the user domain. Generally, ratings are often impacted by users, which means that the recommendation model needs to obtain different preferences for one item of different user ratings. We can use different weights to represent user preference. Thus, a user self-attention model has been proposed that can obtain different weights for users. The basic framework is illustrated in Fig. 1f. We do not add a

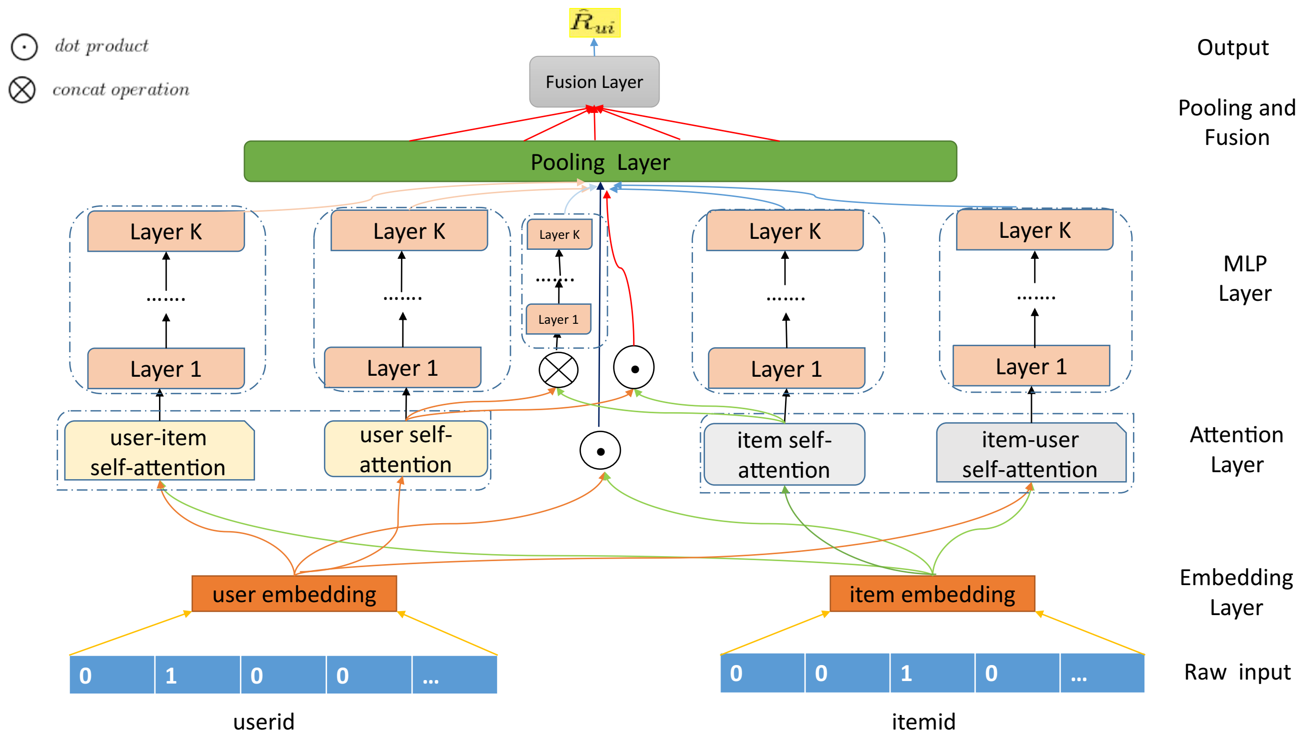

Fig. 3 The key modules of dual learning based on self-attention (DLSA)

mask to the user self-attention model and item self-attention model because the input lengths are equal.

Compared with the self-attention model, the user self-attention model has two of the same input variables, which can incorporate user information by using weights. The equation is defined as follows:

$$u_{att} = soft\max\left(\frac{uemb(uemb)^T}{\sqrt{d_u}}\right) \tag{12}$$

where d_u is the dimension of the user vector.

We design the user self-attention model to ensure that the same item contributes to different user weights. Then, implicit information can be extracted by MLP-learned weights, and the process is formulated as:

$$u_{att}^{l+1} = f\left(W_u^l u_{att}^l + b_u^l\right) \tag{13}$$

where W_u and b_u are the weight matrix and bias of the user, respectively, and f is the activation function.

Item self-attention model Similar to the user attention, we model item-to-item influence, which is rating-aware and relies on the specific targeted user. Fig. 1a illustrates the situation. We design the user-based item self-attention model to output different weights for related items with regard to the target user. The framework is shown in Fig. 1d, and we can represent it as follows:

$$i_{att} = soft\max\left(\frac{iemb(iemb)^T}{\sqrt{d_i}}\right) \tag{14}$$

where d_i represents the dimension of the user. Then, the model can obtain implicit information through the MLP:

$$i_{att}^{l+1} = f\left(W_i^l i_{att}^l + b_i^l\right) \tag{15}$$

where W_i and b_i represent the weight and bias, respectively, of the user.

Correspondingly, item-to-item attention contributes to the effect for objective users, so we assume it is item preference from the user's perspective (comparatively speaking, items rate users).

3.4 Interaction self-attention model

To fully mine interactive implicit information from rating information, we extend self-attention and propose an interaction self-attention learning model. The main idea of the interactive self-attention model is revealed in Fig. 1b. It is important to note that the attention model needs to confirm which model should be applied. We consider two types of models: the user-item self-attention model and the item-user self-attention model. One model focuses on "user", and the other focuses on “item”. They jointly affect user preference and item preference feedback by dual learning. In addition, interaction self-attention models have a mask operation in the case of different lengths of user and item inputs.

User-item self-attention model We obtain users and item vector representations via the embedding layer. Then, the user-item self-attention model operation can output the attention weights of the interaction information. Fig. 1e shows the framework of the user-item self-attention model. Thus, we can obtain the weights of users in different items, and the user-item self-attention operation is defined as follows:

$$att_{u,i} = soft\max\left(\frac{uemb(iemb)^T}{\sqrt{d_u}}\right) \tag{16}$$

The MLP operation outputs implicit information via attention model outputs:

$$att_{u,i}^{l+1} = f\left(W_{u,i}^{l} att_{u,i}^{l} + b_{u,i}^{l}\right) \tag{17}$$

where $W_{u,i}$ and $b_{u,i}$ are the weight and bias, respectively.

Item-user self-attention model The model framework is the same as that in Fig. 1e; it focuses attention on items of different users, and the calculation is as follows:

$$att_{i,u} = soft\max\left(\frac{iemb(uemb)^T}{\sqrt{d_i}}\right) \tag{18}$$

Similarly, the MLP operation can be defined as follows:

$$att_{i,u}^{l+1} = f\left(W_{i,u}^{l} att_{i,u}^{l} + b_{i,u}^{l}\right) \tag{19}$$

where $W_{i,u}$ and $b_{i,u}$ are the weight and bias, respectively.

3.5 Dot product

We also directly obtain user preference information from the dot product and attention models, and two places exist: the embedding layer of the next step and following the self-attention model, which is illustrated in Fig.1. Except for Eq. (5), the rest of these dot product equations are as follows:

$$u_{att}di_{att} = u_{att} \odot i_{att} \tag{20}$$

where udi is the output of the user and item dot product, and $u_{att}di_{att}$ is the dot product result of the user self-attention model and item self-attention model.

3.6 Pooling layer

Based on previous work, we adopt attention and MLP modules to obtain implicit information and user preferences under a rating recommendation. However, it needs to be converted to ratings with sum pooling to be reduced to 1-dimension. The following equation shows the calculation process:

$$m = \sum_{j}^{n} e_j, \forall j = 2, 3, ..., n \tag{21}$$

where e_j represents the j-dimensional vector of vector e, and m is the 1-dimensional pooling output. Based on the pooling layer location of Fig. 3, we can pool the results of Eqs. (5), (14), (16), (18)–(20), and the values are $m_{u_{att}}^{mlp}$, $m_{i_{att}}^{mlp}$, $m_{ui_{att}}^{mlp}$, $m_{iu_{att}}^{mlp}$, m_{ui} and m_{ui}^{att}, respectively.

3.7 Fusion layer and output

The fusion layer is above all these multilayer perceptrons; it combines six embedding vector representations into a single-vector representation and outputs the prediction rating. Therefore, the combined feature after fusion is formulated as:

$$\hat{R}_{ui} = m_{ui} + m_{ui}^{att} + m_{u_{att}}^{mlp} + m_{i_{att}}^{mlp} + m_{ui_{att}}^{mlp} + m_{iu_{att}}^{mlp} + b_i + b_u + b \tag{22}$$

where $\hat{R}_{ui}$ denotes the rating of a user for a specific item and m_{ui} and m_{ui}^{att} are obtained by the pooling layer operation of the user-item dot product and user self-attention and item self-attention model dot product. $m_{ui_{att}}^{mlp}$ and $m_{iu_{att}}^{mlp}$ denote the pooling results of the MLP of the user-item self-attention model and item-user self-attention model, respectively. b_i, b_u and b are the bias of an item, bias of a user and global bias, respectively.

3.8 Learning

To benefit from the field of *dual learning based on self-attention*, we simply apply an l2 loss function and penalty to conduct the final loss as:

$$L = \frac{1}{2}\left(\sum\left(r_{ui} - \hat{R}_{ui}\right)^2 + \alpha\sum(uemb)^2 + \beta\sum(iemb)^2\right) \tag{23}$$

where r_{ui} is the true rating, **uemb** and **iemb** are the vector representations of the user and item, respectively, and α and β are penalty ratios. To optimize the objective function, we use Adam, a variant of stochastic gradient descent (SGD), which dynamically tunes the learning rate during the training process and leads to faster convergence.

4 Why dual learning?

He et al. proposed dual learning for machine translation, and this mechanism was inspired by the following observation: any machine translation has a dual task. They considered that

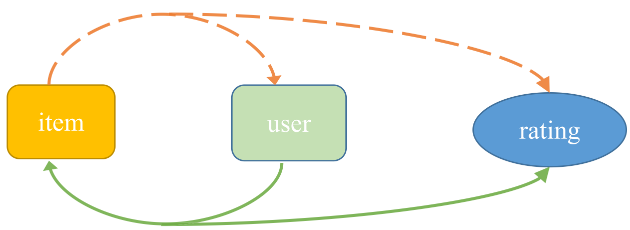

Fig. 4 The dual-learning in rating recommendation

machine translation adopts dual learning to reduce human labeling and learns from unlabeled data. Likewise, in user-item rating (primal) versus item-user rating (dual), the primal and dual tasks can form a loop (user and item) and generate information feedback to train the model, which is illustrated in Fig. 4. Such a framework learns a probabilistic mapping $p(x, y, z)$ from a source user $x = \left(user_1, user_2, ..., user_n\right)$ and item $y = \left(item_1, item_2, ..., item_n\right)$ to target $r = \left(r_1, r_2, ..., r_n\right)$. Then, for $p(x, y, z)$, we consider:

$$p(x, y, r) = p(x, y|r)p(x, y). \tag{24}$$

Similarly,

$$p(y, x, r) = p(y, x|r)p(y, x). \tag{25}$$

For a given triple (x, y, z),$(x, y) \rightarrow r$ and $(y, x) \rightarrow r$ are equal. Because dual learning can utilize implicit information feedback, then $p(x, y, r) + p(y, x, r) > p(x, y, r)$ and $p(x, y, r) + p(y, x, r) > p(y, x, r)$. There, a model based on dual learning can obtain more implicit information and improve the recommendation accuracy.

5 Experiments

In this section, to comprehensively evaluate our proposed model,we conduct experiments to answer the following research questions:

RQ1 How does the DLSA method perform compared with state-of-art models for recommendation?

RQ2 How does the performance of DLSA compare with different metthods for recommendation?

RQ3 Are the key components in DLSA (i.e., dual learning, self-attention) useful for improving the recommendation results?

Experimental settings Our program language is Python 3, and the OS is Ubuntu 16.04.5. The LTS server and memory are 1 TB. The deep learning framework is TensorFlow[1] 1.8, the GPU is NVIDIA GTX 1080Ti, and the CUDA version is 8.0 and CUDNN 6.0. We use the GPU to accelerate the experiment for the purpose of reducing the running time. The code can be found at https://github.com/lijhong/DLSA.

Datasets We conducted the experiments using two public datasets: MovieLens 100K and MovieLens 1M.[2] They are actually the most popular benchmarks for rating recommendation tasks. The characteristics of the two datasets are summarized in Table 2. The ratings in both datasets are from 1 to 5.

MovieLens 100K It contains 100,000 rating records of 943 users in 1682 items, and all the rating scores are positive and not greater than 5. Each user rated at least 20 movies. The higher the rating is, the more the user liked the movie.

MovieLens 1M It includes 1,000,209 rating records from 6040 users on 3952 movies, and each user rated at least 20 movies. As shown in Table 2, as the scale of the dataset increases, the sparseness of the data increases synchronously.

Hyperparameter settings For different datasets, we adopt different hyperparameter settings, as follows: For MovieLens 100K, the hyperparameter settings are as follows: the batch size, dimension, and learning rate are 128, 10, and 0.001, respectively, and the penalty ratio is 0.01. We use the rectified linear unit (*ReLU*) activation function, two layers of neural networks in the MLP and neuron numbers of 100-10. For MovieLens 1M, the hyperparameter settings are as follows: the batch size, dimension, learning rate and penalty ratio are 1000, 10, 0.1, and 0.05, respectively, the activation function is the *ReLU*, and the number of neurons in one neural network layer in the MLP is 10. For each dataset, we employ a split of 4:1 for training and testing, and baselines possess text information and do not contain any additional information; training: testing = 9:1 indicates baselines without any text information.

Evaluation protocol We employ mean absolute error (MAE) and root mean square error (RMSE) as evaluation metrics, which are widely used in recommendation systems, to evaluate the performance of our model.

MAE MAE is a rating-based metric; it is the average the absolute error of the real value and predicted value, which can reflect the actual situation of the prediction error. It is formulated as follows:

$$MAE = \frac{1}{T}\sum_{i=1}^{T}\left|r_{ui} - \hat{R}_{ui}\right| \tag{26}$$

where the parameter T represents the total number of test sizes.

RMSE RMSE is also a rating-based metric and is used to measure the deviation between the prediction rating and the true rating. The purpose is to measure the deviation

[1] https://www.tensorflow.org.

[2] https://grouplens.org/datasets/movielens/.

between the predicted value and the true value. The equation is defined as follows:

$$RMSE = \sqrt{\frac{1}{T}\left(\sum_{i=1}^{T}\left(r_{ui} - \hat{R}_{ui}\right)^2\right)} \tag{27}$$

where parameter T represents the total number of test sizes.

Baseline methods To better answer RQ1 and RQ2, the baselines are divided into methods that possess auxiliary text information and those that do not possess text information; no text information compares methods and attention-based algorithms with the following:

1. **CF-CNN** [42] This method learns to exploit complex latent user-item relations based on convolutional neural networks.
2. **NMC-S** [43] This method inherits the predictive power of neural networks and can be extended to observed samples.
3. **ZMF** [44] This method integrates the user preference information into matrix factorization and decomposes the user-item interaction matrix.
4. **BPMF** Bayesian probabilistic matrix factorization for recommendation.
5. **SVD++** This model fuses the latent factor model and neighbor model for recommendation.
6. **SR** [45] This model exploits users' implicit social relationships for recommendation.
7. **PRA** Probabilistic rating autoencoder.
8. **RLMC** [46] This method provides a new robust noise estimator where we characterize the bias and variance of the estimator in a finite sample setting, and it does not add text information.
9. **RMGCNN** [47] This proposed method matrix completion architecture combines a multigraph convolutional neural network that can learn meaningful graph structures from users and items.
10. **PRMF** [48] This method can automatically learn the dependencies between users to improve recommendation accuracy.
11. **IGPL**[49] This method learns an inductive model for a recommendation system based on the local graph patterns around user-item pairs.
12. **SHCGM** [50] Stochastic Frank-Wolfe for Composite Convex Minimization optimization.
13. **DMF** Deep Matrix Factorization Models for Recommender Systems.
14. **NRR**[51] It is a neural rating regression model that captures specific characteristics of users and items.

Methods based on text information include the following:

Table 2 Statistics of the MovieLens datasets

	MovieLens 100 k	MovieLens 1 M
Users	943	6040
Items	1682	3952
Ratings	100,000	1,000,209
Ratings of per user	106.4	165.6
Rating of per item	59.5	253.1
Rating sparsity	93.7%	95.8%

15. **IAI** [12] This method uses the number of item attribute types, items' attributes, and an autoencoder neural network to achieve great results in recommendation systems. It takes the genres of movies as item attributes.
16. **JICO** [52] This method integrates the joint interaction information and contextual information. It takes three types of user contextual information: gender, occupation, and age.
17. **Hybrid-CRBM** This method is a hybrid conditional restricted Boltzmann machine. This method adopts the movie category as text information.

Methods based on attention mechanism include the following:

18. **HARSAM** [37] Hybrid model for recommendation supported by self-attention mechanism.
19. **DeepUCF** [36] Deep attention user-based collaborative filtering for recommendation
20. **ACNN-FM** [39] Attention-based convolutional neural network and factorization machines.
21. **AICF** [38] Attention-based item collaborative filtering.
22. **ACA-GRU** [35] Attention-based context-aware sequential recommendation model using Gated Recurrent Unit.

5.1 Performance comparison with baselines (RQ1)

Table 3 shows the performance of DLSA on the MovieLens datasets with other comparative methods. First, we can see that DLSA outperforms other comparative methods and achieves great improvements, i.e., at least a 0.3% improvement in MAE and a 0.2% improvement in RMSE at least on MovieLens 100K. The results on MovieLens 1M further verify the superiority of our proposed model, with an improvement in MAE of at least 0.3%. Second, matrix factorization methods such as PMF, SVD++ and BPMF achieve good effects on the two

Table 3 Experimental performance of DLSA compared to 10-cv baselines on the MovieLens datasets

		NRR	ReDa	PMF	NMF	SVD++	PRA	BPMF	ZMF	NMC-S	SR	**DLSA**
MovieLens 100 K	MAE	0.717	0.715	0.788	0.766	0.722	0.760	0.863	–	0.729	–	**0.712**
	RMSE	0.910	0.911	0.975	0.967	0.924	0.965	1.103	–	0.921	–	**0.909**
MovieLens 1 M	MAE	0.691	0.665	0.685	0.718	0.666	0.716	0.676	0.688	0.674	0.675	**0.662**
	RMSE	0.875	**0.847**	0.875	0.916	0.848	0.907	0.869	0.875	0.862	0.850	**0.847**

Table 4 Experimental performance of DLSA compared to 5-cv baseline (existing text information) and attention-based method RMSE metrics on the MovieLens datasets

	MovieLens 100 K	MovieLens 1 M
IAI	**0.902**	0.862
JICO	0.941	0.859
HIRE	0.924	0.861
Hybrid CRBM	**0.902**	0.866
AFM	–	0.858
ACA-GRU	0.907	0.858
AICF	0.912	0.855
ACNN-FM	0.905	0.854
DeepUCF	0.912	0.856
HARSAM	0.941	0.874
DLSA	0.912	**0.853**

Table 5 Experimental performance of DLSA compared to 5-cv baseline (without text information) RMSE metrics on the MovieLens datasets

	MovieLens 100K	MovieLens 1M
RMGCNN	0.924	–
IGPL	0.929	–
DMF	0.940	0.878
PRMF	0.913	0.857
RLMC	0.983	0.936
DeepHM	0.984	0.982
HLFM	**0.901**	0.854
PRA	0.966	0.905
BPMF	1.128	0.874
PMF	0.970	0.881
Q-DNN	0.987	0.936
SHCGM	1.144	–
CF-CNN	–	0.854
DLSA	0.912	**0.853**

datasets but perform poorly compared to ReDa and DLSA, revealing that methods based on deep learning can capture more implicit user information. In addition, PRA and ReDa methods based on deep learning results are better than matrix factorization-based methods, but DLSA can achieve a better recommendation accuracy than ReDa.

In summary, the results illustrate that our proposed method in efficient, so it can significantly improve recommendation performance.

5.2 Performance comparison with methods based on text information and attention mechanism (RQ2)

Deep learning can obtain feature representation from context, and some deep learning-based methods utilize context information to improve recommendation accuracy. We report the experimental results with these methods in Table 4. Bold indicates the best experimental results. As we can see, the IAI and hybrid CRBM achieve the best overall performance on MovieLens 100K, and DLSA achieves great results on Movielens 1M. Given the statistical information in Table 2 and the RMSE values in Table 4, we believe that this result is caused by the size of MovieLens 100K. The smaller the size of the dataset, the less information that can be learned from the dataset, so DLSA cannot obtain better recommendation accuracy.

We also performed comparisons with some algorithms that are not text information-based approaches, and the results are shown in Table 5. It is easy to see that methods based on auxiliary text information can acquire good performance. However, DLSA achieves the best result on MovieLens 1M and the second best result on MovieLens 100K, as shown in Tables 4 and 5, respectively. These findings illustrate that our designed rating information modules including the user-user, user-item, item-item and item-user approaches, as well as the training model, are reasonable. Thus, our proposed model can achieve great results even without auxiliary text information.

In addition, we compared our approach with some attention-based algorithms, and the results are shown in Table 4. It is easy to see that methods based on the attention mechanism can improve recommendation accuracy. Some methods, such as the ACNN-FM achieve a good preference on Movielens 100K. Nevertheless, DLSA can achieve the best result on MovieLens 1M. These findings indicate that our designed rating information modules including user attention, item attention, interactive attention and the training

Table 6 Experimental performance of dual learning,self-attention and DLSA on the MovieLens datasets, namely, biased matrix factorization (BiasM), dual-learning of user and item (DLUI), dual-learning based on self-attention (DLSA), user self-attention model (user self-att), item self-attention model (item self-att) and interaction self-attention model (inter self-att)

		BiasM	User self-att	Item self-att	Inter self-att	DLUI	DLSA
MovieLens 100 K		0.722	0.721	0.720	0.719	0.719	**0.716**
	RMSE	0.920	0.920	0.919	0.917	0.917	**0.912**
MovieLens 1 M	MAE	0.674	0.673	0.671	0.672	0.669	**0.667**
	RMSE	0.861	0.856	0.855	0.856	0.859	**0.853**

model are reasonable. Thus, our proposed model shows a good recommendation performance.

5.3 Dual learning and self-attention (RQ3)

We conduct a study of dual learning and self-attention and show the effects achieved by user self-attention, item self-attention, interaction self-attention, DLSA, DLUI and BiasM in Table 6 (training datasets 80%, and test datasets 20%).

The BiasM method target function is: $\hat{r} = u \odot i + b_u + b_i$; it does not add dual learning or self-attention mechanisms. The dual-learning of user and item (DLUI) is first presented in the DLSA framework, and together they serve as a rating recommendation framework. Apart from rating information, DLUI adopts the merge layer to fuse users for their relation information and uses the same approach for items. We find that compared with BiasM, DLUI shows a relative improvement of 0.3% on MovieLens 100K, which is similar to MovieLens 1M. Thus, a method based on dual learning and rating information modules can achieve good results.

In addition, the self-attention mechanism methods in DLSA (user self-attention model and item self-attention model detail in Sect. 3.3, interactive self-attention model details in Sect. 3.4) have achieved some improvement compared with BiasM on MovieLens 100K and MovieLens 1M. Therefore, self-attention can obtain some useful user preference information and improve recommendation effects. We found that the dual learning and self-attention mechanisms achieve almost identical results, suggesting that both had an impact on the capability of improvement.

Furthermore, by combining the self-attention mechanism and dual learning, DLSA was found to achieve a better solution (compared with the self-attention module and DLUI, the DLSA average improvement is approximately 0.3%). Therefore, the sufficiency and efficiency of our proposed dual-learning based on self-attention is verified.

6 Conclusion

In this paper, an approach is proposed to solve the disadvantages of existing methods in rating recommendations, including matrix factorization models based on LFMs that do not perform well, latent interactive information such as user-user and item-item information that is not used in recommendation models, and auxiliary text information that is not easily acquired in some deep learning recommendation models. In this work, we propose DLSA as a general framework for rating recommendation. To make full use of rating information, we introduce a dual-learning based on the self-attention framework to mine pairwise interaction (i.e., user-item and user-user) information and implicit information feedback. Then, we fuse this implicit information to predict the rating of items. Extensive experiments on MovieLens datasets demonstrate the superior performance of our proposed model compared with state-of-the-art algorithms.

In the future, we will extend DLSA to incorporate review information, which can be utilized to characterize users and items from different perspectives. In this way, negative implicit feedback can be used to gain weights for optimization. We will also consider graph structures, such as user-item-user or item-user-item, which can benefit rating recommendations.

Acknowledgements This work was jointly supported by the National Key Research and Development Program of China (2017YFB1401903), the Natural Science Foundation of China (Nos. 61673020, 61702003, 61876001), and the Natural Science Foundation of Anhui Province (1808085M-F175). The authors would also like to thank the anonymous reviewers for their valuable comments.

References

1. Adomavicius G, Kwon YO (2011) New recommendation techniques for multicriteria rating systems. IEEE Intell Syst 22(3):48–55
2. Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the tenth international world wide web conference, pp 285–295

3. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
4. Wu C, Wu H, Luo C, Wu Q, Liu C, Wu Y, Yang F (2019) Recommendation algorithm based on user score probability and project type. Eur J Wirel Comm 2019:80
5. 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, pp 426–434
6. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
7. Liu X, Ouyang Y, Rong W, Zhang X (2015) Item category aware conditional restricted Boltzmann machine based recommendation. In: International conference on neural information processing, pp 609–616
8. Zhang L, Luo T, Zhang F, Wu Y (2018) A recommendation model based on deep neural network. IEEE Access 6:9454–9463
9. Xin D, Lei Y, 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. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11(12):3371–3408
11. 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
12. Zhao J, Geng X, Zhou J, Sun Q, Xiao Y, Fu Z (2019) Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems. Knowl Based Syst 166:132–139
13. Yan W, Wang D, Cao M, Liu J (2019) Deep auto encoder model with convolutional text networks for video recommendation. IEEE Access 7:40333–40346
14. Bi JW, Liu Y, Fan ZP (2019) A deep neural networks based recommendation algorithm using user and item basic data. Int J Mach Learn Cybern 11(4):763–777
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: 2017 annual conference on neural information processing systems, pp 5998–6008
16. Xia Y, He D, Qin T, et al(2016) Dual learning for machine translation. In: 2016 annual conference on neural information processing systems, pp 820–828
17. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: Proceedings of the twenty-first annual conference on neural information processing systems, pp 1257–1264
18. Salakhutdinov R, Mnih A (2008) Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: 2008 ICML proceedings of the twenty-fifth international conference, pp 880–887
19. Lee DD, Seung HS (2001) Algorithms for nonnegative matrix factorization. In: 2000 NIPS papers from neural information processing systems, pp 556–562
20. Corso D, Gianna M, Romani F (2019) Adaptive nonnegative matrix factorization and measure comparisons for recommender systems. Appl Math Comput 354:164–179
21. Zhao X, Zhu Z, Zhang Y, James C (2020) Improving the estimation of tail ratings in recommender system with multi-latent representations. In: Thirteenth ACM international conference on web search and data mining, 2020 WSDM, Houston, TX, USA, pp 762–770
22. Rendle S (2010) Factorization machines. In: the 10th IEEE international conference on data mining, pp 995–1000
23. Lian D, Liu R, Ge Y, Zheng K, Xie X, Cao L (2017) Discrete content-aware matrix factorization. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 325–344
24. Wang H, Wang N, Yeung D (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015, pp 1235–1244
25. Deng X, Zhuang F, Zhu Z (2019) Neural variational collaborative fltering with side information for top-K recommendation. Int J Mach Learn Cybern 10:3273–3284
26. Mongia A, Jhamb N, Chouzenoux E, Majumdar A (2020) Deep latent factor model for collaborative filtering. Signal Process 169:107366
27. Liang H, Baldwin T (2015) A probabilistic rating auto-encoder for personalized recommender systems. In: 2015 CIKM the 24th ACM international conference on information and knowledge management, pp 1863–1866
28. Zhuang F, Zhang Z, Qian M, Shi C, Xie X, He Q (2017) Representation learning via dualautoencoder for recommendation. Neural Netw 90:83–89
29. Zhang M, Jiang S, Cui Z, Garnett R, Chen Y (2019) D-vae: A variational autoencoder for directed acyclic graphs. In: 2019 NeurIPS annual conference on neural information processing systems, pp 1586–1598
30. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. IEEE T Neural Networ 20(1):61–80
31. Qian F, Li J, Zhao S, Zhang Y (2019) Rating recommendation based on deep hybrid model. J Nanjing Univ Aeronaut Astronaut 51(5):592–598
32. Cheng H, Koc L, Harmsen J (2016) Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems, pp 7–10
33. Chen J, Zhang H, He X, Nie L, Liu W, Chua TS (2017) Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In: Proceedings of the 40th international acm sigir conference on research and development in information retrieval, pp 335–344
34. Chen J, Zhuang F, Hong X, Ao X, Xie X, He Q (2018) Attention-driven factor model for explainable personalized recommendation. In: 2018 SIGIR 41st international ACM SIGIR conference on research & development in information retrieval, pp 909–912
35. Yuan W, Wang H, Yu X, Liu N, Li Z (2020) Attention-based context-aware sequential recommendation model. Inf Sci 510:122–134
36. Chen J, Wang X, Zhao S, Qian F, Zhang Y (2020) Deep attention user-based collaborative filtering for recommendation. Neurocomputing 383:57–68
37. Peng D, Yuan W, Liu C (2019) HARSAM: A hybrid model for recommendation supported by self-attention mechanism. IEEE Access 7:12620–12629
38. Lv Y, Zheng Y, Wei F, Wang C, Wang C (2020) AICF: Attention-based item collaborative filtering. Adv Eng Inform 44:101090
39. Pang G, Wang X, Hao F, Xie J, Wang X, Lin Y, Qin X (2019) ACNN-FM: A novel recommendation with attention-based convolutional neural network and factorization machines. Knowl Based Syst 181:104786
40. Xue H, Dai X, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: Proceedings of the 26th international joint conference on artificial intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017, pp 3203–3209

41. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: 27th annual conference on neural information processing systems 2013, 5–8 Dec 2013, pp 3111–3119
42. Jia Y, Wang X, Zhang J (2019) Collaborative filtering via learning characteristics of neighborhood based on convolutional neural networks. In: Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data, pp 1–4
43. Nguyen D, Tsiligianni E, Deligiannis N (2018) Extendable neural matrix completion. In: 2018 IEEE international conference on acoustics, speech and signal processing, ICASSP 2018, Canada, 15–20 April 2018, pp 6328–6332
44. Li M, Dai J, Zhu F, Zang L, Hu S, Han J (2019) A fuzzy set based approach for rating bias. In: The thirty-third AAAI conference on artificial intelligence, pp 9969–9970
45. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the Forth international conference on web search and web data mining, 2011 WSDM, Hong Kong, China, pp 287–296
46. Sabetsarvestani Z, Kiraly F, Miguel R, Rodrigues D (2018) Entry-wise matrix completion from noisy entries. In: 2018 EUSIPCO 26th European signal processing conference, pp 2603–2607
47. Monti F, Bronstein M, Bresson X (2017) Geometric matrix completion with recurrent multi-graph neural networks. In: Annual conference on neural information processing systems 2017, 4–9 Dec 2017, Long Beach, CA, USA, pp 3697–3707
48. Liu Y, Zhao P, Liu X, Wu M, Duan L, Li XL (2017) Learning user dependencies for recommendation. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI 2017, 19–25, pp 2379–2385
49. Zhang M, Chen Y (2019) Inductive graph pattern learning for recommender systems based on a graph neural network. arXiv:1904.12058
50. Locatello F, Yurtsever A, Fercoq O, Cevher V (2019) Stochastic Frank-Wolfe for composite convex minimization. In: Annual conference on neural information processing systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada, pp 14246–14256
51. Li P, Wang Z, Ren Z, Bing L, Lam W (2017) Neural rating regression with abstractive tips generation for recommendation. In: Proceedings of the 40th international acm sigir conference on research & development in information retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017, pp 345–354
52. Bai P, Ge Y, Liu F, Lu H (2019) Joint interaction with context operation for collaborative filtering. Pattern Recogn 88:729–738

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