Expert Systems With Applications 183 (2021) 115439

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Learning attention embeddings based on memory networks for neural collaborative recommendation ☆

Yihao Zhang [a,*], Xiaoyang Liu [b]

[a] School of Artificial Intelligence, Chongqing University of Technology, Chongqing, 400054, China
[b] School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054, China

ARTICLE INFO

Keywords:
Memory networks
Collaborative filtering
Attention embeddings
Behavioral patterns
Recommender systems

ABSTRACT

Recently, deep learning has dominated the recommender system, as it is able to effectively capture nonlinear and nontrivial user–item relationships, and perform complex nonlinear transformations. However, there are still some issues with respects to the existing methods. Firstly, they always treat user–item interactions independently, and may fail to cover more complex and hidden information that is inherently implicit in the local neighborhood surrounding an interaction sample. Secondly, by quantifying the dependence degree of user–item sequences, it demonstrates that both short-term and long-term dependent behavioral patterns co-exist. Unfortunately, typical deep learning methods might be problematic when coping with very long-term sequential dependencies. To address these issues, we propose a novel unified neural collaborative recommendation algorithm that capitalizes on memory networks for learning attention embedding from implicit interaction (NCRAE). Particularly, the attention is capable of learning the relative importance of different users and items from user–item interaction sequences, which provides a better solution for concentrating on inputs and helps to better memorize long-term sequential dependencies. Extensive experiments on three real-world datasets show significant improvements of our proposed NCRAE algorithm over the competitive methods. Empirical evidence shows that using memory networks for learning attention embeddings of users' implicit interaction yields better recommendation performance.

1. Introduction

Recommender systems have taken more and more places in our lives from e-commerce to online advertisement, which exploit historical user ratings or auxiliary information to make recommendations on items to the users (Wei et al., 2017). Generally, recommender systems are characterized by algorithms aimed at suggesting relevant items to users, which seek to predict the "rating" or "preference" that a user would give to an item. In most cases, the effectiveness of recommendations is restricted by existing user–item interactions and model capacity (Li et al., 2018).

In the past years, collaborative filtering (CF) has been the most popular and successful method, establishing the relevance between users and items from past interactions by assuming similar users will consume similar items (Chen et al., 2019; Yang et al., 2017). CF method predicts users' personalized preference from user–item interactions, which plays a central role especially in the phase of candidate generation (He et al., 2018). As a classic CF method, matrix factorization (MF) works by decomposing the user–item interaction matrix into the product of two lower dimensionality rectangular matrices (Agarwal & Chen, 2009). Although MF is able to provide good recommendation quality, its ability to use only explicit numerical ratings as user–items interactions constitutes a limitation. In order to exploit all available interactions both explicit and implicit, SVD++ is proposed to mine various implicit interactions as well by taking also into account user and item bias (Cao et al., 2015). Asymmetric SVD also is devised for constructing a latent space from the training data, which combines the advantages of SVD++ while being a model-based algorithm, thus able to consider new users with a few ratings without needing to retrain the whole algorithm (Pu & Faltings, 2013). In addition, the trust-aware MF algorithm is also developed to take dual advantages of both social

☆ This work is the results of the research project funded by the National Natural Science Foundation of China under Grant No. 61702063.
The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.
* Corresponding author.
E-mail addresses: yhzhang@cqut.edu.cn (Y. Zhang), lxy3103@cqut.edu.cn (X. Liu).

https://doi.org/10.1016/j.eswa.2021.115439
Received 27 July 2020; Received in revised form 12 June 2021; Accepted 12 June 2021
Available online 17 June 2021

relations and discrete techniques for fast recommendation (Guo et al., 2019).

Recently, deep learning methods have dominated the landscape of algorithmic research in recommender system (Dacrema et al., 2019), and they have been dramatically changing recommendation architecture and significantly improving the performance of recommender systems. It is generally believed that any recommendation architectures using a variant of stochastic gradient descent to optimize the distinguishable objective function are deep learning techniques. Existing methods mainly utilize a neural user–item coupling learning for collaborative filtering, which jointly learns explicit and implicit couplings between users and items (Zhang et al., 2018). Besides, the neural network is used to generalize the traditional matrix factorization algorithm via a non-linear neural architecture, aiming to solve the key problem of collaborative filtering in recommendation on the basis of implicit feedback (He et al., 2017). In summation, deep learning methods can effectively capture nonlinear and nontrivial user–item relationships, and perform complex nonlinear transformations in recommendation model over traditional linear models (Zhang, Yao et al., 2019).

In the recommender system based on deep learning, since the attention machine can learn the relative importance of different users and items from the interaction sequence, the incorporation of attention into the recommendation model has also become an emerging trend. A two-level attentive mechanism is introduced in collaborative filtering model to cope with the challenging item and component-level implicit feedback in multimedia recommendation (Chen et al., 2017). Memory-based attention is used for collaborative ranking, which introduces a latent relation vector learned via attention to collaborative metric learning algorithm (Tay et al., 2018). By applying the attention mechanism to recommender systems, it can filter out uninformative content and select the most representative items while providing good interpretability. However, existing composite architectures incorporate the attention factor ignoring the integration of vanilla attention learned from user–item interactions in a nonlinear fashion. We propose to represent users' and items' attention embeddings with the memory network in the pre-training, which utilizes the user–item pairs and the negative items not liked by the user to train attention model. The external memory permits encoding rich feature representations, and the neural attention mechanism infers users' and items' specific contribution factor.

On other hand, analyzing millions of user sequences and quantifying the degree of long-term sequential dependence in these sequences, demonstrates that both short-term and long-term dependent behavioral patterns co-exist (Tang et al., 2019). The long-term sequential dependencies refer to the dependencies between interactions that are far from each other in a sequence. For example, given a shopping sequence $S_1 = \{Apen, milk, shoes, mirror, bag, egg, apple, \ldots, ink\}$, which consists of a basket of items that are purchased successively by a user. Obviously, the pen and the ink are highly dependent even though they are far from each other. RNNs are usually adopted to model long-term dependencies, whereas they use more strict factorization assumptions for computational reasons (Ko et al., 2016). Therefore, to leverage the rich attention vector from user–item interaction for recommendation model, we propose a unified neural collaborative recommendation algorithm that capitalizes on memory networks for learning attention embeddings from users' implicit feedback (NCRAE). The memory component encodes the complex relations of user–item interactions, the neural attention mechanism provides various higher weights to specific users who share similar preferences. Finally, the embeddings from user–item ratings and the attention embedding vectors learning from user–item interactions by memory networks are feed into the attention networks for recommendation, and multiple layer networks are stacked in the model to further improve the performance. Accordingly, the major contributions of this paper are as follows:

(1) We design a co-attention mechanism that capitalizes on memory networks for learning attention embeddings from user–item interactions. The attention is capable of learning the relative importance of different users and items from user–item interaction sequences without relying additional content or context information, which provides a better solution and helps the network to better memorize inputs. Specially, the attention mechanism makes it easy to memorize very long-term sequential dependencies in neural networks and concentrate on important parts of inputs.

(2) We elaborately design a unified neural collaborative recommendation algorithm by explicitly incorporating users' and items' attention. These attention weights learned by the memory network provide supporting evidence for deeper recommendation architectures, which aim to capture higher order complex user–item interactions. The unified algorithm is augmented with an external memory and neural attention, which learns an adaptive nonlinear weighting of users' and items' for further improving recommendation performance.

(3) We conduct comprehensive experiments on three public datasets, and the results demonstrate that the NCRAE algorithm is better than other competitive algorithms. Particularly, it shows that integrating attention embeddings generated by memory networks in the unified recommendation model is effective to improve the performance. Experimental results also show that our algorithm can capture short-term and long-term dependence well, and is effective to better memorize long-term sequential dependencies.

2. Related work

In recommender systems, early research works mainly focus on collaborative filtering (CF) methods. CF methods make recommendations by learning user–item historical interactions, either through explicit (e.g., user's ratings) or implicit feedback (e.g., browsing information). However, CF methods usually suffer from data sparsity and cold-start problem. Many works attempt to leverage additional information for recommendation, such as social information (Zhao et al., 2015), content information (Fu et al., 2019), heterogeneous information (Shi et al., 2018), and generative adversarial networks (Gao et al., 2021). Recently, deep learning methods have been employed in recommender systems for extracting refined latent features from the user–item interaction data (He et al., 2017), which aim to learn multiple levels of representations and abstractions from data. One of the major motivation for exploit the deep learning methods in recommender system is that they are end-to-end differentiable and likely to provide suitable inductive biases catered to the input data type (Zhang, Yao et al., 2019). As such, the sequential structures of sessions or click-logs in recommender systems are highly suitable for the inductive biases provided by recurrent or convolutional models (Tang & Wang, 2018).

Although the recommender system based on deep learning techniques has made great progress, some issues still exist in the long-term dependencies problem. For example, the typical LSTM can solve the long memory problem theoretically, it is still problematic when coping with long-term dependencies in recommender system (Zhang, Yao et al., 2019). However, integrating an attention mechanism into deep learning methods (such as LSTM) enables it to process longer and noisier inputs. Pi et al. (2019) use machine learning algorithms and online service systems for CTR prediction tasks, which can cope with the challenge of long-term sequential user behavior modeling. Liu et al. (2018) propose a short-term attention and memory priority-based model for capturing users' general interests, while taking into account users' current interests from the short-term memory of the last-clicks. Wu et al. (2019) propose a neural news recommendation model, which uses a CNN network to learn hidden representations of news articles based for the news representation model, and learn users representation based on news articles clicked by them. Cheng et al. (2018) develop an aspect-aware recommender model to capture the varying aspect attentions that a user pays to different items. Hu et al. (2018) develop a deep neural network with the co-attention mechanism for leveraging rich meta-path based context for top-N recommendation. Xiao et al.

(2017) propose a attentional factorization machine to learn the importance of each feature interaction from data via a neural attention network, which aims to improve the performance by discriminating the importance of different feature interactions. Wang, He et al. (2019) propose a knowledge graph attention network explicitly modeling the high-order connectivity in an end-to-end fashion, which connects two items with one or multiple linked attributes to provide more accurate, diverse, and explainable recommendation. In summation, introducing an attention mechanism can yield a better solution and help the neural network to better memorize inputs.

On the other hand, analyzing the current recommendation algorithms based on attention mechanism, most neural methods rely on additional content or context information to learn the attention weights (Hu et al., 2018). Wang et al. (2018) propose an attention-based embedding model for context embedding to weight each observed item in a transaction without assuming order. However, these additional or auxiliary information may not be available in collaborative filtering. In order to make full use of the implicit feedback of user–item interactions, the memory network architecture could be employed to address implicit feedback in the collaborative filtering method. Weston et al. (2014) propose a new class of learning models called memory networks, which reason with inference components combined with a long-term memory component and learn how to use these jointly. Sukhbaatar et al. (2015) design a recurrent attention model over a possibly large external memory, and require significantly less supervision during training, thus making it more generally applicable in realistic settings. Wang, Ren et al. (2019) use parallel memory modules to exploit the collaborative information contained in neighborhood sessions to improve recommendation performance of the current session. Guo et al. (2020) propose a neighbor-guided memory-based neural network for session-aware recommendation. Motivated by the success of the memory network, we propose the NCRAE algorithm, which elaborately design a novel neural collaborative recommendation algorithm to integrate the attention embeddings learned through memory network for unified recommendation. The attention embeddings are capable of learning the relative importance of different users and items from user–item interaction sequences without relying additional content or context information. It is different from the collaborative memory networks (CMN) (Ebesu et al., 2018) to unify the two classes of collaborative filtering models in a nonlinear fashion for end-to-end recommendation. Our algorithm is augmented with an external memory and neural attention to learn an adaptive nonlinear weighting of users' and items', and is supposed to further improve the recommendation performance.

3. Neural collaborative recommendation based on attention

In this section, we present the proposed neural collaborative recommendation model based on attention embeddings, which utilizes the memory networks to learning attention vectors from user–item interactions. Fig. 1 shows the overall architecture of our proposed NCRAE algorithm, which consists four components: the input module, the pre-trained attention module, the unified recommendation module and the prediction module. Particularly, the input module pre-processes the data and transforms them into the user–item interactive format; the pre-trained attention module learns the memory embeddings from the user–item pairs and the negative items; the unified recommendation module integrates the attention embeddings and user–item interactive data in a nonlinear fashion for recommendation; and the prediction module is developed via a classification function to determine the recommendation list.

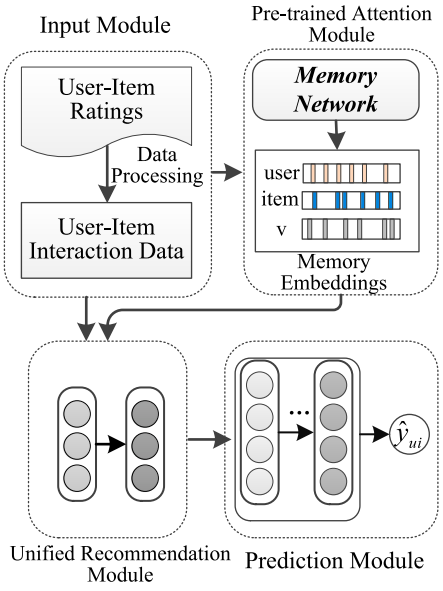

Fig. 1. The overall architecture of the recommendation algorithm (NCRAE).

3.1. The input module

The input module takes the user–item interaction pair and the negative items not liked by the user. Formally, given m users $U = \{u_1, u_2, \ldots, u_m\}$ and n items $I = \{I_1, I_2, \ldots, I_n\}$, we define each user–item interaction pair $r_{u,i}$ in the user implicit feedback matrix $R \in R^{m \times n}$ as follows: $r_{u,i} = 1$ when the user u has rated the item i, and $r_{u,i} = 0$ otherwise, and then they are used to encode the user embeddings and item embeddings. Here the value of 1 in the matrix R indicates the interaction result between each user and an item, such as whether a user has purchased or rated an item. Meanwhile, the user–item interaction pair $[u_p, i_q]$ and negative items that are not liked by the user $(u_m, [i_p, i_q, \ldots, i_k])$ are fed into the memory networks for training user and item attention embeddings from these implicit feedback (Ebesu et al., 2018). The attention embeddings are capable of representing the relative importance of different users and items of the user–item interaction sequences, and help the network to better memorize the inputs.

3.2. The pre-trained attention module

The attention mechanism shares similar intuition with that of the visual attention found in humans, which pays attention to only the most important parts of the target and ignoring the unimportant parts. In this paper, we exploit memory networks to train the user and item attention embeddings for better representation learning, which is supposed to more efficiently memorize long-term sequential dependencies in neural networks and concentrate on important parts of inputs. More specifically, in the networks, the memory component encodes the complex user–item interaction relations, and the neural attention mechanism infers users' and items' specific contribution factors and provides various weights to users who share similar preferences. Afterwards, the user and item embeddings learnt from memory networks are fed into the attention network for the recommendation model.

In attention-based recommendation models, inputs are usually weighted with various scores. Therefore, calculating the attention scores turns to be the heart of attention-based recommendation models. According to the different ways to calculate attention scores, the attention models are usually categorized into standard vanilla attention and co-attention (Zhang, Yao et al., 2019). The former learns attention

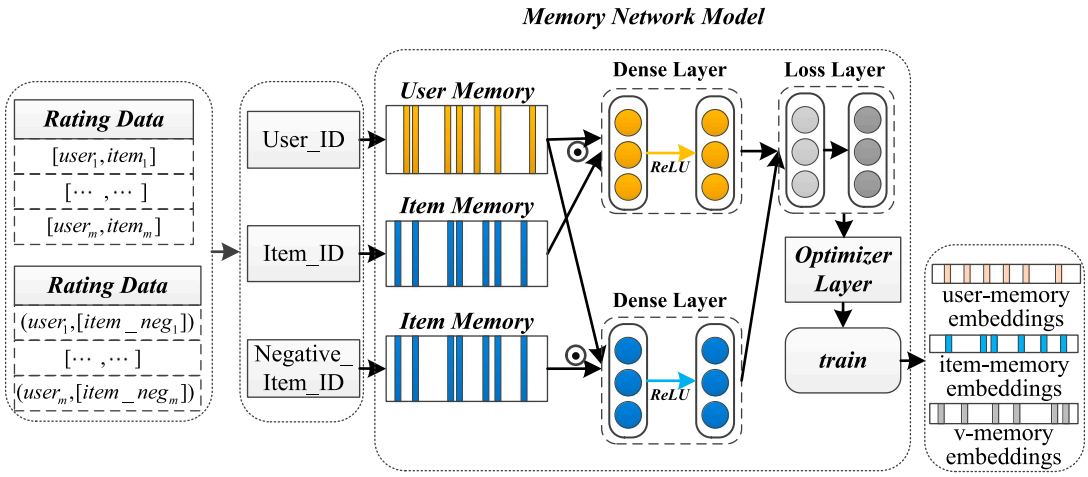

Fig. 2. The memory networks model for calculating the attention embeddings.

Table 1
The meaning of notations.

Notation	The meaning of notations
u	A user
i	An item
U	A user memory matrix
I	An item memory matrix
m	The number of users
n	The number of items
d	The dimensionality of each memory cell
q_{ui}	A user preference vector
q_{uiv}	Each dimension of the vector q_{ui}
p	The probability vector of the inputs
e_u	The memory slot of user u
e_i	The memory slot of item i

scores utilizing a parameterized context vector, while the latter learns attention scores from two sequences.

In the recommender systems, a set of items liked by a user naturally reflects the user's preference. However, it is difficult to summarize diverse items liked by user i into a user attention. As such, we utilize a memory component and an attention mechanism to learn deep adaptive user representations, and derive the relevance between users and items from past interactions. Although the memory component stores all item information of user i, our goal is to learn the deep representation of user attention weight based on their liked items, and the item attention weight based on users that liked these items. The architecture of our proposed memory networks model is depicted in Fig. 2.

In our attention mechanism model, For the convenience of the following description, we define the meaning of each notation as shown in Table 1. The memory component contains a user memory matrix $U \in R^{m\times d}$ and an item memory matrix $I \in R^{n\times d}$. Given an user input set $\{x_1, x_2, \ldots, x_m\}$ stored in memory, it will be converted into the user memory vector $\{x_1, x_2, \ldots, x_m\} \longrightarrow \{u_d\}$ of dimension d to make up the user memory matrix U, which is computed by embedding each $\{x_i\}$ in a continuous space. For example, using an embedding matrix A (of size $d \times V$). The query q is also embedded using another matrix B with the same dimensions as A to obtain an internal state S. In the embedding space, we compute the match between S and each memory $\{x_1, x_2, \ldots, x_m\}$ by taking the inner product as

$$p_i = Softmax(S^T x_m), \tag{1}$$

where $Softmax(V_i) = e^{V_i} / \sum_j e^{z_i}$, and p is a probability vector over the inputs.

In the embedding representation, each user u is embedded in a memory slot $e_u \in U$ encoding the user's specific preferences, and each item i is embedded in another memory slot $e_i \in I$ encoding the item's specific attributes. We compute the similarity of the target user p and the user q, which represents the agreement level of the two users in the neighborhood of given item i. The calculation of similarity is defined as

$$\phi_{piq} = e_p{}^T \bullet e_q + e_i{}^T \bullet e_q, \quad \forall q \in N(i), \tag{2}$$

where $N(i)$ represents the set of all users who have provided implicit feedback for item i. In Eq. (2), the first term computes compatibility between the target user and his neighborhood who have rated item i. The second term introduces the level of confidence user q supports the recommendation on item i.

Then, we compute the attention weight of a given user to weigh the importance of each user's unique contribution to its neighborhood as

$$\varphi_{piq} = \frac{\exp(\phi_{piq})}{\sum_{k\in N(i)} \exp(\phi_{pik})} \qquad \forall\, q \in N(i), \tag{3}$$

Finally, we construct the response vector O from the memory by interpolating the external neighborhood memory using attention weights from the input as

$$O_{pi} = \sum_{q\in N(i)} \varphi_{piq} e_v, \tag{4}$$

where e_v is another embedding vector for user v, which is called external memory.

In the memory networks model, our objective is to learn attention embedding from users' implicit feedback, aiming to provide a better solution and help the network to better memorize inputs. As shown in Fig. 2, we choose the pairwise of user and item and the candidate list of negative items for each user, where a given user liked the observed item $[u_o, i_m{}^+]$ and disliked the list of negative items $(u_o, [i_n{}^-])$. We denote our loss function as

$$\begin{aligned} \hat{r}_{ui^+} &= u_o * i_m{}^+ \\ \hat{r}_{ui^-} &= u_o * i_n{}^- \\ Loss &= -\sum_{u,i^+,i^-} \sigma(\hat{r}_{ui^+} - \hat{r}_{ui^-}), \end{aligned} \tag{5}$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the logistic sigmoid function. Since the entire model architecture is differentiable, the memory networks model can be efficiently trained with the backpropagation algorithm. Subsequently, we apply the optimization layer to optimize the model. Finally, the output of the model is the users' embedding vector, the items' embedding vector, and the intermediate embedding vector.

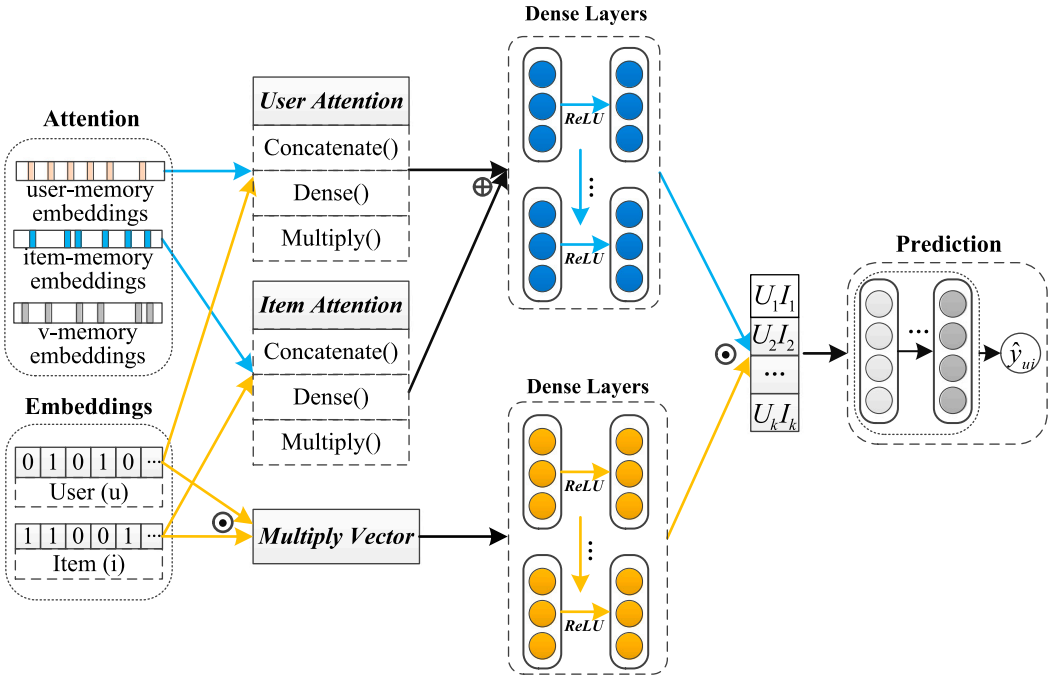

Fig. 3. The architecture of the recommendation model.

3.3. The unified recommendation module

Attention based recommendation model aims to learn the relative importance of different users and items of the interaction sequences. By applying the user and item attention weights in recommender systems, we can filter out uninformative content and select the most representative users and items. In the attention model, we integrate the attention weights learned from user–item interactions in a nonlinear fashion. The architecture of our proposed recommendation model is depicted in Fig. 3.

In Section 3.2, we got the embeddings of each user E_{au} and each item E_{ai}. Also, we have the one-hot user embedding E_{ou} and item embedding E_{oi}, which are transformed from the original user–item rating matrix. We combine the two embedding vectors into a unified representation of the current user and item as

$$\begin{aligned} \tilde{E}_u &= E_{au} \oplus E_{ou} \\ \tilde{E}_i &= E_{ai} \oplus E_{oi}, \end{aligned} \tag{6}$$

where $\oplus$ denotes the concatenation operation of vectors. Subsequently, we feed the unified $\tilde{E}_u$ and $\tilde{E}_i$ into user attention and item attention, respectively, in order to implement a nonlinear function for modeling complicated interactions as

$$\begin{aligned} \hat{r}_u &= User\_Attention(\tilde{E}_u) \\ \hat{r}_i &= Item\_Attention(\tilde{E}_i), \end{aligned} \tag{7}$$

where the user-attention and item-attention components have two hidden layers with $ReLU$ as the activation function, and the output layer is implemented with the softmax function. With the premise that neural network model can learn more abstractive features of data via using a small number of hidden units for higher layers, we empirically add a dense layer to implement a tower structure for the neural network model, halving the layer size for each successive higher layer (i.e., the dimension of hidden layer is set to [40,20, ..., 10]). Afterwards, different fusion methods are applied to integrate the output data, such as concatenation, element-wise product, followed by some dense layers in the recommendation model. In summation, we elaborately design a unified neural collaborative recommendation model incorporating users' and items' attention, where attention weights provide supporting evidence for deeper recommendation architectures to capture higher order complex user–item interactions.

3.4. The prediction module

The prediction module is implemented with a classification function $\hat{y}_{ui} = W\sigma_L(F)+b$, where W_{L-1}, b_{L-1} and σ_{L-1} denote the weight matrix, bias vector, and activation function for the Lth layer, respectively. In the prediction layer, the vector F is fed into the fully connected layers. Meanwhile, we stack some hidden layers to improve the recommendation performance based on the theory that the degree of nonlinearity could be enhanced by stacking more nonlinear layers (He et al., 2017). Let $U_p \in R^K$ and $I_q \in R^K$ be the representation vectors of user p and item q transformed from the original user–item rating matrix, and K is the dimension of latent vectors. $F = [f_1, f_2, \ldots, f_K]$ denotes the output representation of the user–item pair. The training process is defined as

$$\begin{aligned} F &= w_{p,q} \odot (U_p \odot I_q) \\ \sigma_2(F) &= \sigma_1(W_1F + b_1) \\ &\cdots\cdots \\ \sigma_L(F) &= \sigma_{L-1}(W_{L-1}\sigma_{L-1}(F) + b_{L-1}), \end{aligned} \tag{8}$$

where $\odot$ denotes the element-wise product, and $w_{p,q} \in R^K$ is the attention vector of the user p for the item q. From Eq. (5), we can get $f_K = w_{p,q,K} \odot U_{p,K} \odot I_{q,K}$, where f_K denotes the Kth attention factor in F. It reflects the recommendation strategy based attention mechanism that for the interaction of each pair of user U_p and item I_q, the attention weight $w_{p,q,K}$ is aim to capture the importance weight factor K of item I_q with respect to the user U_p.

Different from the rating prediction tasks which utilize squared error loss as the objective function, we adopt the cross-entropy loss in our optimization function, and our goal is to output the recommendation list. Since we have the implicit feedback available, we learn the model parameters with negative sampling, and the objective for an interaction $[U, I]$ can be formulated as

$$Loss_{u,i} = -\log \hat{r}_{u,i} - E_{j \in I_{Neg}}[\log(1 - \hat{r}_{u,j})], \tag{9}$$

Table 2
Statistics of the datasets.

Datasets	Users#	Items#	Interactions#	Sparsity
Amazon(Baby)	21 263	746	5193	99.45%
Amazon(Beauty)	42 270	1248	8942	99.62%
Yelp	442 476	1266	93 270	99.63%

where the first term models the observed interaction, and the second term models the feedback drawn from the negative recommendation list I_{Neg}.

4. Experiments

In this section, we conduct extensive experiments on three real world datasets. Our experiments are designed to answer the three research questions: (RQ1) Can the attention weight obtained by the memory network effectively improve the performance of the recommendation model? (RQ2) Does our proposed NCRAE algorithm outperform the state-of-the-art methods? (RQ3) Can the NCRAE algorithm solve the long-term and short-term problem?

4.1. Experimental settings

4.1.1. Evaluation datasets

The evaluation datasets are obtained from different domains, i.e., Amazon product datasets[1] (McAuley & Leskovec, 2013) and Yelp dataset.[2] Amazon datasets contain user reviews and metadata information of various products. The datasets cover approximately 35 million comments, which span a period of 18 years from July 1995 to March 2013. We adopt Baby dataset and Beauty dataset from the 24 product categories as the first two datasets. Meanwhile, Yelp dataset records user ratings on local businesses, and also contains social relations and attribute information of businesses, which is adopted as the third dataset. In order to reduce the sparseness of the data, we selected the user with at least 20 interaction ratings for Amazon dataset, and we set the threshold of the number of user–item interaction as 200 for Yelp dataset. The statistics of the selected datasets are presented in Table 2.

4.1.2. Evaluation metrics

To evaluate the recommendation performance, we randomly split each dataset into training sets and test sets. For each user we randomly hold out one item the user has interacted with, and sample 100 unobserved or negative items to form the test sets. We adopt two common ranking evaluation metrics, i.e., Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). HR is a way of calculating how many "hits" a user has in an n-sized list of ranked items, which is defined as

$$HR@N = \frac{Numberof Hits@N}{|GT|}, \tag{10}$$

where $|GT|$ is collections of test dataset. NDCG accumulated at a particular rank position N, and it places stronger emphasis on retrieving relevant documents, which is defined as

$$NDCG@N = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{11}$$

where rel_i is the graded relevance of the recommendation result at position i.

4.1.3. Baseline methods

We implement our NCRAE algorithm using the python library of Keras,[3] and compare it with the following baseline methods:

- ItemKNN (Sarwar et al., 2001): This is a standard item-based collaborative filtering method by recommending similar items based on the adopted items previously. We adapt it for implicit feedback data by following the setting in Hu et al. (2008).
- MF (Matrix Factorization) (Koren et al., 2009): This method is a feature decomposition algorithm that decomposes the "user-item" scoring matrix to obtain a user hidden vector matrix and an item hidden vector matrix, thereby predicting missing scores in the original scoring matrix.
- HFT (Hidden Factors and Hidden Topics) (McAuley & Leskovec, 2013): It models ratings and review texts using MF and LDA respectively. Then combine latent rating dimensions (obtained highly interpretable textual labels) with latent review topics which are learned by topic models like LDA.
- NeuMF (He et al., 2017): It is a neural network collaborative filtering method for top-N recommendation using only implicit feedback, which consists of a generalized MF component and an MLP component. It is generic and can express and generalize matrix factorization under its framework.
- CMN (Ebesu et al., 2018): It is a deep architecture to unify the neighborhood-based and memory-based collaborative filtering methods, which capitalizes on the strengths of the global structure of latent factor model and local neighborhood-based structure in a nonlinear fashion.
- CL-LSTM (Perera & Zimmermann, 2018): It is a multi-layer long-term short-term memory network, which uses attention gating mechanism to capture long-term user preference, uses a higher-order interaction layer to reduce data sparsity, and uses time-conscious LSTM cell gates to capture irregular time intervals between user interactions.
- AttRec (Zhang, Tay et al., 2019): It utilizes self-attention mechanism to estimate the relative weights of each item in user interaction trajectories to learn better representations for user's transient interests for recommendation.

We deliberately selected the above seven methods to cover various recommendation strategies: ItemKNN algorithm is a representative of item-based collaborative filter methods, and it can be used to verify the utility to learn attention weights of our methods; MF and HFT algorithms are collaborative filtering methods based on matrix factorization; NeuMF is the state-of-the-art collaborative filtering ensemble method fusing MF and MLP in the latent space for recommendation; CMN is a deep architecture fusing a memory component and neural attention mechanism; CL-LSTM is a multi-layer long-term short-term memory network for recommendation; AttRec is a sequence recommendation model based on self-attention mechanism.

4.2. Experimental results

For our model, we randomly initialize model parameters with adaptive moment estimation (Adam), and set the batch size to 256, the learning rate to 0.00005. We first train it without regularization, then tune the learning rate in the range of $[10^{-6}, 10^{-5}, \ldots, 1]$ when overfitting is observed (i.e., training loss keeps decreasing while the performance becomes worse). The validation set is consisted of a randomly drew interaction for each user. For the embedding size k, we test the values of $[1, 5, 10, 20, 50]$, and set the attention factor to be same as the output dimensions of the user embedding and the item embedding.

[1] http://jmcauley.ucsd.edu/data/amazon/links.html.
[2] https://www.yelp.com/dataset/challenge.
[3] https://github.com/haomiaocqut/ReSys_NCRAE.

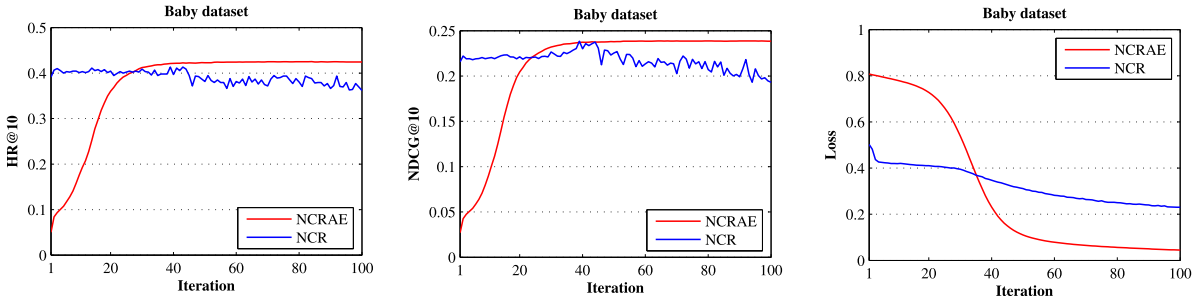

Fig. 4. Testing performance of NCRAE algorithm with embedding size 5 in each epoch.

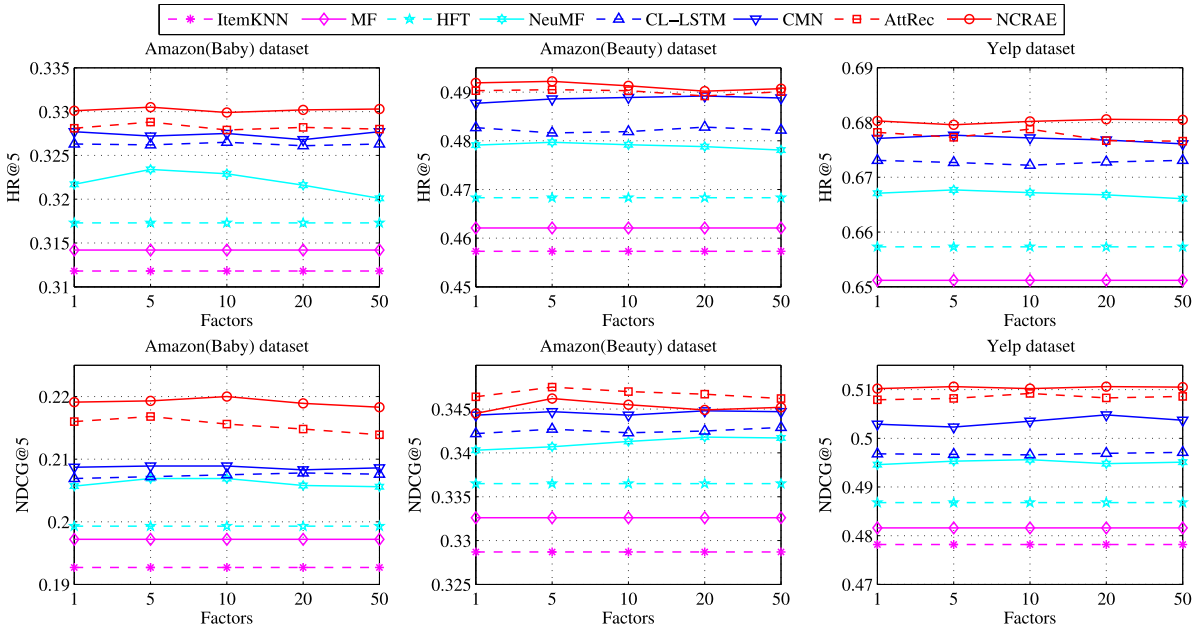

Fig. 5. Testing performance of NCRAE algorithm where factors range from 1 to 50 on three datasets.

4.2.1. Effectiveness experimental of the attention (RQ1)

In this section, we conduct an ablation study, aiming to further understand the performance of our model to answer RQ1.

In order to verify the impact of the attention weights obtained through pre-training on user–item interaction data, we also added a recommendation algorithm (i.e., NCR). The only difference between the NCRAE algorithm and NCR algorithm is that the learned attention weights is not considered in the latter. Fig. 4 shows the training loss and recommendation performance (HR@10 and NDCG@10) of NCRAE algorithm for each iteration on Amazon product dataset (Baby). Almost the same trend was found for the other two datasets, and their results are omitted due to space limitations. It can be seen from Fig. 4 that as the iteration number increases, the training loss of the NCRAE algorithm gradually decreases, and the recommendation performance is improved. However,for the NCR algorithm, its performance fluctuates up and down with the increase of iteration. Although the training loss of NCR keeps decreasing, its recommendation performance actually degrades. It can be seen from the first two sub-graphs of Fig. 4 that the fluctuation of the performance is relatively larger, and the performance (HR@10 and NDCG@10) is very unstable. The red line indicates the performance of our proposed NCRAE algorithm. Compared with the NCR algorithm, it converges more slowly. However, with the increase of the number of iterations, the performance of NCRAE algorithm obviously exceeds the NCR algorithm. The smoothness of the red curve shows that the performance of NCRAE algorithm is very stable, without large fluctuations. The experimental results also validate our motivation and prove that the performance can be effectively improved by adding attention weights to the model.

4.2.2. Experimental results and analysis (RQ2)

Fig. 5 shows the performance (HR@5 and NDCG@5) of all the competitive algorithms with respect to different numbers of predictive factors. Our proposed NCRAE algorithm has two additional hyper factors: the attention factor and the embedding size, which must be set to the same value. We test the values of factors as $[1, 5, 10, 20, 50]$, respectively. From Fig. 5, we can see that regardless of the setting of attention factor, the NCRAE algorithm outperforms all other algorithms on Baby dataset and Yelp dataset. As the attention factor changes, the performance also fluctuates within a small range. In general, the model achieved the best performances when $factor = 5, 10$. On Beauty dataset, The AttRec algorithm outperforms our NCRAE algorithm in NDCG@5 metrics. The possible reason is that the dataset is relatively sparse, and our memory network model has not learned effective attention weights from user–item interactions. Comparing our algorithm to the deep learning recommendation algorithm, i.e., CMN and AttRec algorithms on Baby dataset, the relative improvement is 0.85% and 0.52%, respectively, in HR@5 metrics, and the improvement is 5.31% and 1.48%, respectively, in NDCG@5 metrics. On Beauty dataset, the

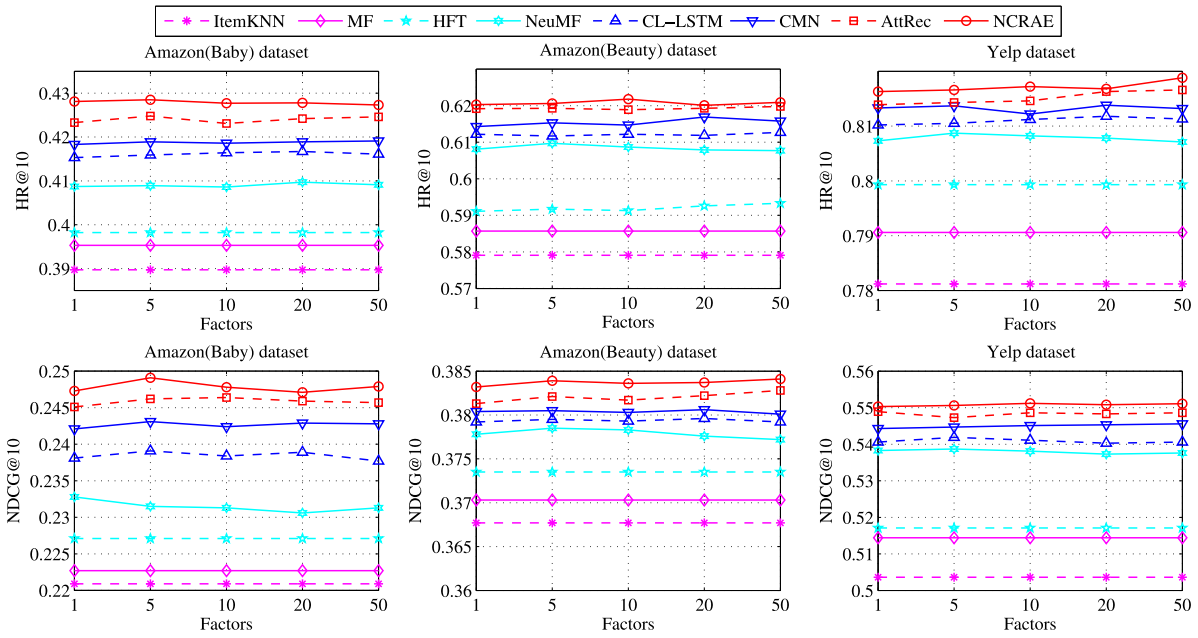

Fig. 6. Testing performance of NCRAE algorithm where factors range from 1 to 50 on three datasets.

relative improvement is 0.61% and 0.35%, respectively, in HR@5 metrics, and the improvement is 0.41% and −0.37%, respectively, in NDCG@5 metrics. On Yelp dataset, the relative improvement is 0.43% and 0.27%, respectively, in HR@5 metrics, and the improvement is 1.15% and 0.28%, respectively, in NDCG@5 metrics. The above experimental analysis demonstrates the benefits of considering attention weights in preference modeling. Compared to the state-of-the-art baseline algorithms, our model achieves promising performance through integrating attention embedding.

Fig. 6 shows the performance (HR@10 and NDCG@10) of all the competitive algorithms with respect to different numbers of predictive factors K. We can see that our NCRAE algorithm is better than all the competitive algorithms on three datasets with all evaluation metrics. On Baby dataset, the relative improvement of our algorithm is 2.24% and 0.87%, respectively, in HR@10 metrics, and the improvement is 2.47% and 1.10%, respectively, in NDCG@10 metrics. On Beauty dataset, the relative improvement is 0.79% and 0.32%, respectively, in HR@10 metrics, and the improvement is 0.92% and 0.34%, respectively, in NDCG@10 metrics. On Yelp dataset, the relative improvement is 0.61% and 0.27%, respectively, in HR@10 metrics, and the improvement is 1.03% and 0.42%, respectively, in NDCG@10 metrics. From the analysis of the experimental results, we can see that our NCRAE algorithm outperform state-of-the-art baseline algorithms on three datasets with all evaluation metrics. In particular, on the Baby dataset, the improvement effect of our algorithm is quite obvious. The improved performance exceeds 2.2% with the HR@10 metrics, and the improved performance exceeds 5.3% with the NDCG @5 metrics. Combining the experimental results of Figs. 5 and 6, we also see that the performance (HR and NDCG) of our algorithm changes relatively little with the factors.

The experimental results of Figs. 5 and 6 indicate the effectiveness of our NCRAE algorithm on the task of Top-N recommendation, which leverages attention weights and memory networks to improve the recommendation performance. The attention embeddings learned from the user–item interactions has a potential influence on the learned representations of users and items. Ignoring such influence may fail to achieve the satisfactory performance in utilizing the long-term sequential dependencies information. The substantial improvement of our model over the baselines could be credited to two aspects: (1) Our model capitalizes on memory networks for learning attention embedding from users' implicit feedback. The attention is capable of learning the relative importance of different users and items of the interaction sequences, and makes it easy to memorize very long-term sequential dependencies and concentrate on important parts of inputs. (2) Our model incorporates users' and items' attention, which can capture higher order complex user–item interactions. It is augmented with an external memory and neural attention, which learns an adaptive nonlinear weighting of the users' and items' to further improve the recommendation performance.

4.2.3. Short-term/long-term dependencies (RQ3)

In this section, we conduct an average ranking study, aiming to further understand the performance of our model to answer RQ3. Each recommendation method generates a ranking of all items in the list. Our method is to measure the average ranking of N next items in the user sequence. With the increase of N, the average ranking changes smoothly from the short-term predictions to the long-term predictions. For example, the user interacts with $2m$ items in the ranking list: $x_1, x_2, \ldots, x_{2m}$, and then we use the adopted algorithms to generate the ranking r over all the items. According to the first half of the sequence (from x_1 to x_m), we compute the average rank at N as

$$Ave\_r@N = \sum_{i=1}^{N} r_{x_{m+i}}/N,\ N \in 1, \ldots, m, \tag{12}$$

where r_i is the ranking of item i (smaller rank being better), and $Ave\_r@N$ is the average ranking of all users in the test set. It provides a useful representation of the quality of an algorithm on the short-term/long-term metrics. The performance of $Ave\_r@N$ of all the adopted methods are illustrated in Fig. 7.

From the experimental results of Fig. 7, we can see that the traditional recommendation algorithm (i.e., ItemKNN, MF, HFT) are much better on short-term prediction than on long-term ones. For the ItemKNN algorithm, this trend is the most obvious. The possible reason is that it is easier to capture short-term preferences of users through similarity calculations. The experiment results also show that some algorithms can be good on short-term prediction but bad on long-term predictions. When N is small, the $Ave\_r@N$ obtained by these three neural network algorithms (i.e., CL-LSTM, CMN, AttRec) are similar, indicating that they are good at capturing short-term preferences. When N becomes larger, the $Ave\_r@N$ also increases rapidly, indicating that

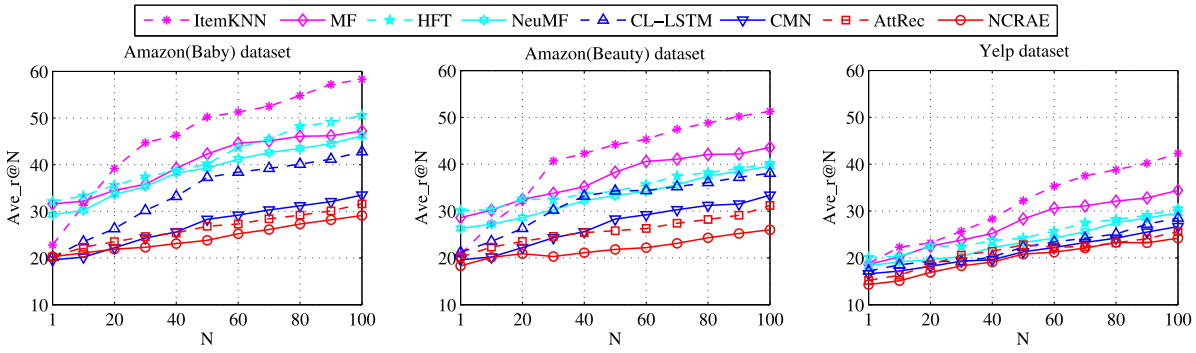

Fig. 7. Average rank of all users in the test set.

these algorithms are not effective in capturing long-term dependencies. With the increase of N, the $Ave\_r@N$ of NCRAE algorithm increases more slowly, which shows that our algorithm can model long-term and short-term preferences better. In particular, compared with other algorithms, our algorithm is more effective when modeling very long dependencies, which is well reflected on the Baby and Beauty datasets of Amazon. On the Yelp dataset, although our algorithm has not obvious advantages in short-term preference modeling compared with other algorithms, it has also achieved the smallest $Ave\_r@N$ when modeling long-term preference. Summarizing the experimental results shows that our algorithm can capture short-term and long-term dependence behavior well, which provides a better solution for concentrating on inputs and helps to better memorize long-term sequential dependencies.

5. Conclusion

In this paper, we present a novel unified neural collaborative recommendation algorithm that capitalizes on memory networks for learning attention embedding from users' implicit interaction (named NCRAE). Our key argument is that existing deep learning techniques in recommender system treat user–item interactions independently, thus may fail to cover the complex and hidden information that is inherently implicit in the local neighborhood surrounding an interaction sample. The proposed model can not only understand the relative importance of different users interacting with items in the interaction sequences, but also effectively integrate an attention mechanism to help the network to better memorize inputs. Specially, the attention mechanism makes it easy to memorize very long-term sequential dependencies in neural networks for concentrating on important parts of inputs. Extensive experiments on three real-world datasets show significant advantages of our proposed NCRAE algorithm over the competitive methods. Empirical evidence shows that using memory networks to learn attention weights of users' implicit interaction offers better recommendation performance.

In future, we are particularly interested in leveraging co-attention for modeling deep recommendation algorithm. Currently, our design of NCRAE algorithm considers the attention weights learning from the user–item interactions only. However, there indeed exist many heterogeneous information in recommendation system, which can be exploited to further improve the performance. Hence, we attempt to identify the co-attention weights from the rich meta-path based contexts (Hu et al., 2008), and explicitly incorporate them in heterogeneous information network.

CRediT authorship contribution statement

Yihao Zhang: Conceptualization, Methodology, Software, Validation, Data curation, Writing - original draft, Writing - review & editing. **Xiaoyang Liu:** Methodology, Formal analysis, Data curation, Writing - review & editing.

Declaration of competing interest

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

Acknowledgments

The work is supported by the National Natural Science Foundation of China (No. 61702063), the Natural Science Foundation of Chongqing (No. cstc2019jcyj-msxmX0544), the Science and Technology Research Program of Chongqing Municipal Education Commission, China (No. KJQN202001136).

References

Agarwal, D., & Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 19–28).

Cao, J., Hu, H., Luo, T., Wang, J., Huang, M., Wang, K., Wu, Z., & Zhang, X. (2015). Distributed design and implementation of svd++ algorithm for e-commerce personalized recommender system. In *Proceedings of the 13th national conference on embedded system technology* (pp. 30–44). Springer.

Chen, W., Cai, F., Chen, H., & Rijke, M. D. (2019). Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems (TOIS)*, *37*(4), 1–30.

Chen, J., Zhang, H., He, X., Nie, L., Liu, W., & Chua, T.-S. (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).

Cheng, Z., Ding, Y., He, X., Zhu, L., Song, X., & Kankanhalli, M. S. (2018). Aˆ 3NCF: An adaptive aspect attention model for rating prediction. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 3748–3754).

Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM conference on recommender systems* (pp. 101–109).

Ebesu, T., Shen, B., & Fang, Y. (2018). Collaborative memory network for recommendation systems. In *Proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 515–524).

Fu, W., Peng, Z., Wang, S., Xu, Y., & Li, J. (2019). Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *Proceedings of the 33rd AAAI conference on artificial intelligence* (pp. 94–101).

Gao, M., Zhang, J., Yu, J., Li, J., Wen, J., & Xiong, Q. (2021). Recommender systems based on generative adversarial networks: A problem-driven perspective. *Information Sciences*, *546*, 1166–1185.

Guo, Y., Ling, Y., & Chen, H. (2020). A neighbor-guided memory-based neural network for session-aware recommendation. *IEEE Access*, *8*, 120668–120678.

Guo, G., Yang, E., Shen, L., Yang, X., & He, X. (2019). Discrete trust-aware matrix factorization for fast recommendation. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 1380–1386).

He, X., He, Z., Song, J., Liu, Z., Jiang, Y.-G., & Chua, T.-S. (2018). NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *30*(12), 2354–2366.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE international conference on data mining* (pp. 263–272). IEEE.

Hu, B., Shi, C., Zhao, W. X., & Yu, P. S. (2018). Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1531–1540).

Ko, Y.-J., Maystre, L., & Grossglauser, M. (2016). Collaborative recurrent neural networks for dynamic recommender systems. In *Asian conference on machine learning* (pp. 366–381). PMLR.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., & Chen, E. (2018). Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1734–1743).

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1831–1839).

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 165–172).

Perera, D., & Zimmermann, R. (2018). Lstm networks for online cross-network recommendations. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 3825–3833).

Pi, Q., Bian, W., Zhou, G., Zhu, X., & Gai, K. (2019). Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2671–2679).

Pu, L., & Faltings, B. (2013). Understanding and improving relational matrix factorization in recommender systems. In *Proceedings the 7th ACM conference on recommender systems* (pp. 41–48).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on world wide web* (pp. 285–295).

Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 357–370.

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *Advances in Neural Information Processing Systems*, 2440–2448.

Tang, J., Belletti, F., Jain, S., Chen, M., Beutel, A., Xu, C., & H. Chi, E. (2019). Towards neural mixture recommender for long range dependent user sequences. In *Proceedings of the 2019 world wide web conference* (pp. 1782–1793).

Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM international conference on web search & data mining* (pp. 565–573).

Tay, Y., Anh Tuan, L., & Hui, S. C. (2018). Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 world wide web conference* (pp. 729–739).

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T.-S. (2019). Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 950–958).

Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., & Liu, W. (2018). Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the 32nd AAAI conference on artificial intelligence*.

Wang, M., Ren, P., Mei, L., Chen, Z., Ma, J., & de Rijke, M. (2019). A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 345–354).

Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, *69*, 29–39.

Weston, J., Chopra, S., & Bordes, A. (2014). Memory networks. In *Proceedings of the 3rd international conference on learning representations* (pp. 1–15).

Wu, C., Wu, F., An, M., Huang, J., Huang, Y., & Xie, X. (2019). NPA: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2576–2584).

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., & Chua, T.-S. (2017). Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 3119–3125).

Yang, C., Bai, L., Zhang, C., Yuan, Q., & Han, J. (2017). Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1245–1254).

Zhang, Q., Cao, L., Zhu, C., Li, Z., & Sun, J. (2018). Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. In *Proceedings of the 27th international joint conference on artificial intelligence*.

Zhang, S., Tay, Y., Yao, L., Sun, A., & An, J. (2019). Next item recommendation with self-attentive metric learning. In *Proceedings of the 33rd AAAI conference on artificial intelligence, Vol. 9*.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, *52*(1), 1–38.

Zhao, W. X., Li, S., He, Y., Chang, E. Y., Wen, J.-R., & Li, X. (2015). Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering*, *28*(5), 1147–1159.

Yihao Zhang is currently an Associate Professor in School of Artificial Intelligence, Chongqing University of Technology. He received the B.S. degree and the M.S. degree in computer science from Xinyang Normal University and Kunming University of Science and Technology, in 2006 and 2010 respectively, the Ph.D. degree in computer science from the Chongqing University in 2014. After that, he worked as a visiting scholar at National University of Singapore. His research interests include Recommender System, Natural Language Processing and Machine Learning.

Xiaoyang Liu is currently a Professor in School of Computer Science and Engineering, Chongqing University of Technology. He received the B.S. degree and the M.S. degree from Huainan Normal University and Kunming University of Science and Technology, in 2006 and 2010 respectively, the Ph.D. degree in communication and information systems from Northwestern Polytechnical University in 2013. From 2015 to 2018, he has completed Post-doctoral fellow in Chongqing University and the University of Alabama, USA. His main research interests include Online Social Network, Information Diffusion and Complex Network.