Computer Science and Information Systems 19(2):573–593 https://doi.org/10.2298/CSIS210228064S

Enhancing Interactive Graph Representation Learning for Review-based Item Recommendation

Guojiang Shen, Jiajia Tan, Zhi Liu, and Xiangjie Kong *

College of Computer Science and Technology, Zhejiang University of Technology
Hangzhou 310023, China
xjkong@ieee.org

Abstract. Collaborative filtering has been successful in the recommendation systems of various scenarios, but it is also hampered by issues such as cold start and data sparsity. To alleviate the above problems, recent studies have attempted to integrate review information into models to improve accuracy of rating prediction. While most of the existing models respectively utilize independent module to extract the latent feature representation of user reviews and item reviews, ignoring the correlation between the latent features, which may fail to capture the similarity of user preferences and item attributes hidden in different review text. On the other hand, the graph neural network can realize the information interaction in high dimensional space through deep architecture, which has been extensively studied in many fields. Therefore, in order to explore the high dimensional relevance between users and items hidden in the review information, we propose a new recommendation model enhancing interactive graph representation learning for review-based item recommendation, named IGRec. Specifically, we construct the user-review-item graph with users/items as nodes and reviews as edges. We further add the connection of the user-user and the item-item to the graph by meta-path of user-item-user and item-user-item. Then we utilize the attention mechanism to fuse edges information into nodes and apply the multilayer graph convolutional network to learn the high-order interactive information of nodes. Finally, we obtain the final embedding of user/item and adopt the factorization machine to complete the rating prediction. Experiments on the five real-world datasets demonstrate that the proposed IGRec outperforms the state-of-the-art baselines.

Keywords: Collaborative Filtering, Recommendation, Graph Convolutional Network, Embedding.

1. Introduction

With the rise of e-commerce, personalized recommendation systems are designed to provide users with personalized information services and decision supports. Excellent recommendation systems can improve the operational efficiency of e-commerce platforms. Of course, there are some researchers who make interesting recommendations in other areas [31], [39], [36], [13]. Collaborative Filtering (CF) [16], [2], [14], [24] has been successful in the recommendation systems of various scenarios. Many of the successful CF techniques are based on Matrix Factorization (MF) [8], [23], [28] that decomposes the

* The corresponding author

user-item rating matrix into two low-dimensional matrices which represent the latent features of the user and the item respectively. However, the recommendation performance of MF methods will degrade significantly when the rating matrix is extremely sparse, and the rating only shows the overall feeling of the user, but do not explain why the user prefers to buy this item. Recently, user-generated reviews after purchasing have become a novel source of recommendation data and some works have introduced reviews to alleviate the above problems [29], [41], [1]. In these works, the convolutional neural network (CNN) architecture instead of topic models [19] is employed to extract the user and item latent features from the corresponding reviews respectively. Compared to a model that only uses rating information to make recommendations, the addition of reviews not only improve the performance of the model, but also increase its interpretability. Although these studies have improved the accuracy of the predictions, there are still some problems to be solved.

- Most of the existing works have learned the latent features of users and items in a static and independent way, which ignores the semantic relevance hidden in the review text.
- Some works have attempted to explore interactions between latent features [18], [2]. However, they hardly extract more complex and higher-order interactions information, which only utilize operation in low-dimensional space.

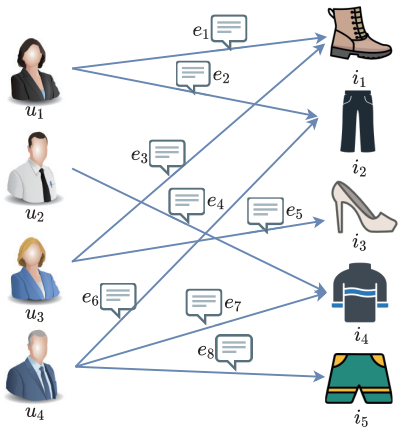

Fig. 1. The Example of User-Review-Item Graph, u_j, e_j, i_j represent user nodes, reviews, item nodes respectively.

To solve the above problems, we propose an IGRec model. The model utilizes the graph neural network to fuse edge (review) and node (user and item) information to achieve predictive rating. The contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to introduce graph neural network to learn review-based user/item representation, and we define graph convolution operator to aggregate edges and nodes features.

- We propose a novel idea that using graph to cover the users, items, and reviews information as shown in the Figure 1 and further add the connection of the user-user and the item-item to graph. To avoid the loss of information, we extract interactive information in the whole graph.
- The experiments are performed on five real-world datasets and the experimental results show that the proposed IGRec model achieves better rating prediction accuracy than the existing state-of-the-art methods.

The remainder of this paper is organized as follows: Section 2 reviews the related work in the recommender systems. Section 3 presents the preliminaries. Section 4 introduces the overall framework of IGRec model in detail. The experimental settings and results are presented in Section 5. Section 6 concludes the article.

2. Related Work

Our work is related to two lines of literatures, the review text used for recommendation and graph neural networks for recommendation. We review the recent advances in both areas.

2.1. Review Text for Recommendation

In order to alleviate the problem of sparse data and cold-start in the recommendation system, researchers began to try to introduce auxiliary information closely related to users and items when build the model, especially, review text of users has become a research hotspot to improve the performance of recommender systems. Some researchers [17] introduced the topic model into the framework and used the review text to improve prediction accuracy and some interpretable work on the model. McAuley. et al. in [19] utilized Latent Dirichlet Allocation (LDA) to extract the topic of the reviews and couple the latent topics and ratings, which accuracy significantly improved in the task of rating prediction. The EFM [40] model and the TriRank [6] model regarded the latent topics as aspects of user and item and provided explanations for recommendation. However, these methods all belong to the category of Bag-of-words models which ignore word order and local context information. Hence, a lot of specific information in the form of phrases and sentences has been lost. To alleviate this limitation, several methods introduce the deep neural network into the traditional MF framework. Kim. et al. in [8] utilized a CNN network to obtain semantic representation of reviews and take into account the word order and local context information. CNN and probability matrix decomposition (PMF) were combined to predict the rating. Zheng. et al. in [41] used two long documents formed by concatenating all reviews of users and items as datasets and learning the representation by two convolution structures vector of the document. Finally, the embedding is concatenated and input into FM for prediction rating. Despite these models having significant improvements in recommendation performance, these works have learned the latent features of users and items in a static and independent way which neglects the information-rich interactions between users and items. Recently, attention mechanisms are fused into the model to capture the importance of different latent features [1], [34], [25] and learn user-item interactions [18]. Seo, S. et al. in [25] introduced word-level attention on the DeepCoNN, different words

are of different importance to the modeling user and the item. Chen. et al. in [1] used the attention mechanism to compute the usefulness of reviews for recommendation. Tay. et al. in [29] employed review-level co-attention to pick out the important reviews, then selected the words in the important reviews for word-level co-attention. Wu. et al. in [34] derived a joint representation for a given user-item pair based on their individual latent features and latent feature interactions. The DAML model learned user-item interactions by a dual attention mechanism that the local attention layer focuses on the importance of different words in the sentences, while the mutual attention layer focuses on the learning of feature interactions [18]. Z Wang. et al. in [33] proposed a hybrid deep collaborative filtering model that two attention-based GRU networks attempt to learn context-aware representation as textural feature for users and items from reviews. The above approach explores the importance of words in sentences, the importance of individual review to the overall document, and the importance of reviews to users and items. However, few studies above focused on the interactive importance of user-item topology graph which contain a wealth of semantic information. In this paper, we regard users and items as nodes and reviews as edges, and construct the user-item-review graph. We further add the connection of the user-user and the item-item to the graph by meta-path of user-item-user and item-user-item.

2.2. Graph Learning for Recommendation

Another direction of research exploits the user-item interaction graph to infer user preference. Yang. et al. in [38] utilized a random walk to capture higher-order relationships between users and items, combining with the degree of vertex, the positive samples of different order are sampled with certain probability, and the attenuation coefficient is assigned to the positive samples of different order. Kong X. et al. in [12] built the weighted-citation graph and the random walks were used for top-K paper recommendation. However, in recent years, there has been increasing interest in developing graph algorithms based on deep learning. The most widely used algorithm is based on Graph Convolutional Network (GCN) [11]. GCN achieved significant improvements compared to previous graph-mining methods such as DeepWalk [21]. Xue, G. et al. in [37] made a comprehensive summary of network representation technology. After that, instead of transductive learning of GCN, Hamilton. et al. in [5] proposed an inductive framework that leverages node sampling and feature aggregation function to generate node embeddings for unseen data. The GAT [30] model introduced the attention mechanism into GCN. It doesn't depend on the full graph structure, only on the edges and it can handle the case of a directed graph. By learning attention coefficients among nodes, GAT assigns different weights to different adjacency nodes, which make decisions to focus on the most relevant neighbors. Naturally, some studies began to introduce GCN into the recommendation system to extract the information of the user-item interaction graph. The NGCF model extracted the high-dimensional connection relation of the user-item through multiple convolution operations [32]. Shen, G. et al. in [26] proposed an unsupervised commercial district recommendation framework via embedding space clustering on graph convolution networks. Ge. et al. in [4] constructed a bipartite graph with the historical user click information and recommend news through the graph attention network [7]. The NIRec model captured interaction patterns of node pairs by different meta-paths in the heterogeneous network, then the attention mechanism is used to fuse the information obtained from different meta-paths.

Although these methods improve performance, they just use the information of the nodes and ignore the information of the edges that contains a wealth of semantic information. Hence, in this paper, we regard reviews as edges and use word2vec [20] and TextCNN [9] to extract review text information, then in each iteration, the aggregation and combination operator are used to extract the information of nodes and edges.

3. Preliminaries

3.1. TextCNN

Although Bert [3] and GPT [22] models have been widely used in the field of NLP recently, in order to maintain the running efficiency of the model, we still choose the combination of word2vec+TextCNN to extract text information. Next we will briefly introduce TextCNN. TextCNN consists of two layers: embedding layer, convolutional layer. In the embedding layer, a word is converted to a d dimensional vector by pre-trained embedding, such as trained Wikipedia corpus using word2vec. Then a fixed length L is extracted for each sentence, which intercept for longer and pad for shorter. The sentence can then be represented as a matrix $M \in R^{L\times d}$. Following the embedding layer is the convolutional layer, we can view embedding matrix as an image and convolutional neural network is used to extract features. Text convolution differs from image convolution in that it is convolved only in one direction (vertical) of the text sequence. Different features are extracted by different filter $K \in R^{t\times d}$. t is the sliding window length, if convolutional layer have m filters, i^{th} filter produces features as:

$$c_i = ReLU(M * K_i + b_i) \tag{1}$$

where $ReLU$ is a nonlinear activation function. $*$ is the convolution operation, b_i is the bias. After that, $c_1, c_2, \cdots {c_i}^{(T-t+1)}$ produced by i^{th} filters. Then max-pooling operation to unify features arising from different filters. which is defined as:

$$o_i = \max(c_1, c_2, \cdots {c_i}^{(T-t+1)}) \tag{2}$$

The final output of the convolutional layer is the concatenation of the output from m filters,

$$O = [o_1, o_2, \cdots o_m] \tag{3}$$

In general, the output O are passed to a fully connected layer with weight matrix W and bias g, which is:

$$X = WO + g \tag{4}$$

3.2. GCN-based Models

The researchers created the GCN so as to use convolution operations on graph-structured data. Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node $v \in \mathcal{V}$, edge $(v, v') \in \mathcal{E}$. $H^0 \in R^{n\times d_0}$ represents the initialization node feature matrix, which n is the number of nodes and d_0 denotes the feature dimension of node. $H^l \in R^{n\times d_l}$ represents l-th layer hidden state of nodes, where d_l denotes the feature dimension of l-th layer node.

The original GCN [11] model following layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{5}$$

where $\tilde{A} = A + I_N$ is the added self-connections adjacency matrix of the graph $\mathcal{G}$. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ denotes degree matrix and $W^{(l)}$ is l-th layer trainable weight matrix. $\sigma(\cdot)$ denotes an activation function. At each propagation, the nodes are updated simultaneously. Recently, there have been many variants based on GCN. As summarized in [35], [15], a propagation layer can be separated into two sub-layers: aggregation and combination, which is defined as:

$$h_{agg}^{(l)} = \sigma\left(W^{(l)} \cdot \mathrm{AGG}\left(\left\{h_{v'}^{(l-1)}, \forall v' \in A(v)\right\}\right)\right) \tag{6}$$

$$h_v^{(l)} = \mathrm{COMBINE}\left(h_v^{(l-1)}, h_{agg}^{(l)}\right) \tag{7}$$

Here, $A(v)$ is a set of nodes adjacent to node v. $\mathrm{AGG}(\cdot)$ is a aggregation function that aggregate features from neighbors of node v. Some operators are selectable as aggregation function, such as mean-pooling, max-pooling [5] or attention mechanism [30]. $W^{(l)}$ is l-th layer trainable weight matrix. $\mathrm{AGG}(\cdot)$ denotes aggregated neighbors feature vector of node v at l-th layer. $\mathrm{COMBINE}(\cdot)$ is combination function that combine node v self feature and aggregated neighbors feature, which optional operators include element-wise product, concatenation [5] and so on. In original GCN, there is no explicit combination step, which is because the adjacency matrix in the original GCN has self-connections. Hence in aggregation step that node self feature has been combinated with those of its neighbors features.

4. The proposed model

In this section, we will introduce our IGRec model in detail. IGRec is devised to predict a rating for a new user-item pair by exploiting existing review data and user-item interaction information. As demonstrated in Figure 2, there are three components in the IGRec: (1) Eembedding layer that extract the text information for the review as edge embedding and offer ID embedding of user and item as node embedding. (2) Interaction layer that update the embedding by learning high-order connectivity relations. (3) Prediction layer that the factorization machine model is designed for final rating prediction.

4.1. Constructing Graph Model

We construct the user-item-review graph that regard users and items as nodes and reviews as edges, and further add the edges of the user-user and the item-item to the graph by meta-path of user-item-user and item-user-item.

4.2. Embedding Layer

The embedding layer initializes the embedding of nodes and edges. We model users and items via an embedding matrix in which the user and item embedding vectors have the

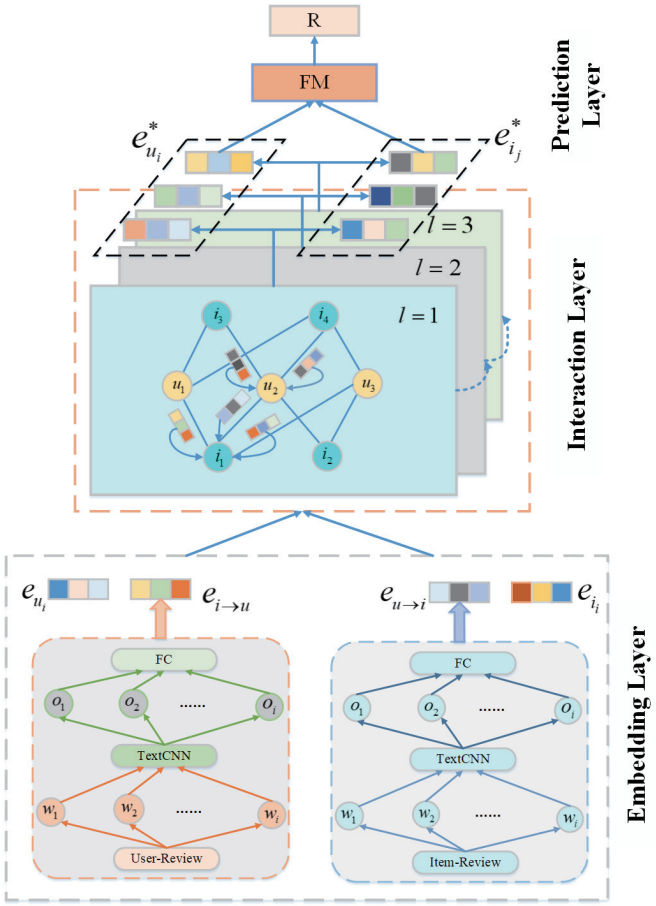

Fig. 2. The architecture of the IGRec. e_{u_i} and e_{i_j} denote initial user ID feature and item ID feature respectively. $e^*_{u_i}$ and $e^*_{i_j}$ denote final user and item feature respectively. $e_{u\to i}$ and $e_{i\to u}$ denote review feature of user and item.

same dimension d. In addition, we use word2vec to initialize the embedded representation of all reviews.

Node Embedding: Let $U = \{u_1, u_2, \cdots, u_M\}$, $I = \{i_1, i_2, \cdots, i_N\}$ denote the user set of M users and the item set of N items respectively. We design an user u (item i) with an embedding vector $e_{u_i} \in R^d$ ($e_{i_j} \in R^d$), of which user and item embedding dimension are both d. Hence, the users and items representation matrix can be defined as:

$$E_u = [e_{u_1}, e_{u_2}, \cdots, e_{u_M}] \tag{8}$$

$$E_i = [e_{i_1}, e_{i_2}, \cdots, e_{i_N}] \tag{9}$$

Since the user and item representation dimensions are equal, we can represent the nodes representation matrix that the stack of embedding matrix of users and items, which is:

$$E_n = \begin{pmatrix} E_u \\ E_i \end{pmatrix} \tag{10}$$

Here, $E_n \in R^{(M+N)\times d}$ is nodes initial representation matrix.

Edge Embedding: Let a review text $x \in X$, which contains fixed-length l words, we use word2vec model to construct the initial word vector and map each word x into the word vector, then concatenate these words vector denote review embedding matrix, which is:

$$D = [w_1, w_2, \cdots, w_l] \tag{11}$$

where $D \in R^{l\times p}$, l stands for the number of words in a single review, p is the embedding dimension of each word. Let user review text $x_u \in X_u$, item review text $x_i \in X_i$. Considering that reviews have different effects on users and items, we apply different TextCNN (cf Eq. (1,2,3)) to extract the information from the initial embedding matrix of the users and items, then we gain the embedding matrix of a single edge, which is:

$$e^*_{u\to i} = [o_{u_1}, o_{u_2}, \cdots, o_{u_d}] \tag{12}$$

$$e^*_{i\to u} = [o_{i_1}, o_{i_2}, \cdots, o_{i_d}] \tag{13}$$

where $e^*_{u\to i}$ is the edge from user to item, $e^*_{i\to u}$ is the edge from item to user. $o_{u_i} \in R^g$ and $o_{i_i} \in R^g$ respectively represent the features obtained from the convolution layer of different user and item TextCNN, g is the feature dimension after max-pooling layer. To facilitate subsequent operations, we use the dimension in which the ID embeding as the number of output channels for the convolution. Finally, we connect a full connection layer to convert the matrix to an one-dimensional vector to represent the embedding of a single edge, which is:

$$e_{u\to i} = W_u O_u + b_u \quad e_{i\to u} = W_i O_i + b_i \tag{14}$$

where $O_u \in R^{d\times g}$, $O_i \in R^{d\times g}$ respectively denote the concatenation of the $[o_{u_1}, o_{u_2}, \cdots, o_{u_d}]$, $[o_{i_1}, o_{i_2}, \cdots, o_{i_d}]$, $W_u, W_i \in R^{g\times d}$ are trainable weight matrices, b_u, b_i are the bias.

4.3. Interaction Layer

The interaction layer is the most important layer of the model. It mainly solves two problems. One is how to fuse edge information into the node, the other is how to transfer the node information with edge information to other nodes. For the first problem, the edge information fusion sub-layer uses the attention mechanism to allocate edge weight so as to provide greater weight values for important reviews. For the second problem, the message propagation sub-layer that is similar to GCN are introduced to fuse the features of adjacent nodes and the multi-hop node information can be obtained by repeated training of the sub-layer.

Edge Information Fusion: In order to realize the training of mini-batch, we fuse the edges information into the nodes and transform the problem into the information transmission between the nodes. Since a single node connects multiple edges, how to selectively

pick out the important edges is a problem we need to solve. In recent years, attention mechanism has been frequently introduced into the recommendation system model [1], [19], [18] and improved the performance of recommender systems. Hence, the model use a linear-layer network to compute the attention score of the edges connected to a single node. Let an user node $e_{u_i} \in R^d$, user edge $e_{u_i \to i_j} \in R^d$ which e_{u_i} is the user node embedding and $e_{u_i \to i_j}$ is the edge embedding that obtained through the embedding layer. $E_{u_i \to i} = [e_{u_i \to i_1}, e_{u_i \to i_2}, \cdots, e_{u_i \to i_m}]$ is the edge set of user node e_{u_i} and as the input of attention layer, the attention network is defined as:

$$\alpha^*_{uj} = ReLU(W_u E_{u_i \to i} + b_u) \tag{15}$$

where $W_u \in R^{d \times m}$, $b_u \in R^m$ are model parameters, $ReLU$ is a nonlinear activation function.

Softmax function was used to normalize the above attention score to obtain the final weight of the edges (reviews), which could be interpreted as the contribution of m edges to user u_i:

$$\alpha_{uj} = \frac{exp(\alpha^*_{uj})}{\sum_{j=1}^{m} exp(\alpha^*_{uj})} \tag{16}$$

After that, we obtain the attention weight of each edge, the feature vector of edges set for user node u_i is calculated as the following weighted sum:

$$e_{u_i \to i} = \sum_{j=1}^{m} \alpha_{uj} e_{u_i \to i_j} \tag{17}$$

where $e_{u_i \to i} \in R^d$ is the attention-weighted embedding sum of the connecting edges of node u_i.

For the item node, we do the same processing. Let an item embedding $e_{i_i} \in R^d$, through the attention layer, the adjacent edge information is aggregated, which is:

$$e_{i_i \to u} = \sum_{j=1}^{m} \alpha_{ij} e_{i_i \to u_j} \tag{18}$$

where $e_{i_i \to u} \in R^d$ is the attention-weighted embedding sum of the connecting edges for item node i_i.

After that, features of adjacent edges are fused into the embedding of the node, the model uses the element-wise product to combine these two kinds of embeddings, which is:

$$e^{\Delta}_{u_i} = e_{u_i} \odot e_{u_i \to i} \quad e^{\Delta}_{i_i} = e_{i_i} \odot e_{i_i \to u} \tag{19}$$

where $e^{\Delta}_{u_i} \in R^d$, $e^{\Delta}_{i_i} \in R^d$ respectively represent the user and the item embedding that have fused edges information. $\odot$ is element-wise product.

Message Propagation Layer: After the edge information fusion sub-layer, we get the node embedding that fused edge information. In this layer, the model uses aggregation and combination operators to extract interactive information of latent features in the graph.

Aggregation Operator: For a pair of connected user-item pair (u_i, i_j). $e_{u_i} \in R^d$, $e_{i_j} \in R^d$ respectively represent embedding of the user u_i and the item i_j. Since the user

and item have the same embedding dimension, we make an unified matrix representation of the nodes. $\mathcal{H}_n = [e^{\Delta}_{u_1}, \cdots e^{\Delta}_{u_M}, e^{\Delta}_{i_1}, \cdots e^{\Delta}_{u_M}]$, the aggregation process of l-th layer is defined as:

$$Agg_n^{(l)} = ReLU({W_1}^{(l)}\mathcal{L}\mathcal{H}_n^{(l-1)} + {b_1}^{(l)}) \tag{20}$$

where $\mathcal{L}$ represents the Laplacian matrix for the user-item graph, which is formulated as:

$$\mathcal{L} = I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \quad and \quad A = \begin{bmatrix} U^* & R \\ R^T & I^* \end{bmatrix} \tag{21}$$

Instead of the traditional adjacency matrix, we further add user-user and item-item connections into adjacency matrix A. $U^* \in R^{M\times M}$ is the connections of user-user by meta-path user-item-user. $I^* \in R^{N\times N}$ is the connections of item-item by meta-path item-user-item.

R is the the user-item interaction matrix. D is the degree matrix. ${W_1}^{(l)}$ is the l-th layer trainable parameter matrix, ${b_1}^{(l)}$ is the l-th layer bias. $ReLU$ is the nonlinear activation function.

Combination Operator: After the information of the adjacent node is collected, it is necessary to fuse it with the information of the node itself, which is formulated as:

$$Comb_n^{(l)} = {W_2}^{(l)}(\mathcal{H}_n^{(l-1)} \odot Agg_n^{(l)}) + {b_2}^{(l)} \tag{22}$$

where $\odot$ denotes the element-wise product that information can be transmitted through the correlation between nodes. ${W_2}^{(l)}$ denotes the importance of the adjacent node, and ${b_2}^{(l)}$ is the l-th layer bias.

4.4. Prediction Layer

After L-layer convolution (aggregation and combination) operations, for user node u_i, we obtain multiple representations $\{e^{\Delta^{(0)}}_{u_i}, \cdots, e^{\Delta^{(L)}}_{u_i}\}$. Just like the two dimensional convolution, different convolutions may acquire different latent features. We concatenate them to constitute the final embedding for the user. In addition, for an item i_j, we do the same operation to concatenate item embedding $\{e^{\Delta^{(0)}}_{i_j}, \cdots, e^{\Delta^{(L)}}_{i_j}\}$ and get the final item embedding:

$$e^*_{u_i} = e^{\Delta^{(0)}}_{u_i}||\cdots||e^{\Delta^{(L)}}_{u_i} \quad and \quad e^*_{i_j} = e^{\Delta^{(0)}}_{i_j}||\cdots||e^{\Delta^{(L)}}_{i_j} \tag{23}$$

The concatenation of $[e^*_{u_i}, e^*_{i_j}]$ is passed into a factorization machine (FM). The FM function is defined as follows:

$$t2F(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \langle v_i, v_j\rangle x_i x_j \tag{24}$$

where $x \in R^k$ is the input feature vector. $\langle ., .\rangle$ is the element-wise product. The parameters v_i are factorized parameters used to model pairwise interactions (x_i, x_j). w_0 is the bias, $\sum_{i=1}^{n} w_i x_i$ represents a linear regression. The output of FM is the final rating of user-item pair:

$$\widehat{R}_{u,i} = FM([e^*_u, e^*_i]) \tag{25}$$

where e^*_u, e^*_i respectively represent the final users and the items embedding.

4.5. Learning

Because the task of this paper is regression, we exploit squared loss as the objective function:

$$loss = \sum_{(u,i)\in\Gamma} (\widehat{R}_{u,i} - R_{u,i})^2 + \lambda_\Theta \|\Theta\|^2 \tag{26}$$

where Γ denotes the set of instances for training, $R_{u,i}$ is the ground truth rating assigned by the user u to the item i.

$\widehat{R}_{u,i}$ is the prediction rating, Θ denotes all the parameters of the model is used as regularization to prevent the model from overfitting. The entire framework can be effectively trained by using end-to-end paradigm reverse propagation.

To optimize the objective function, we adopt the Adaptive Moment Estimation (Adam) [10] as the optimizer. Additionally, to prevent overfitting, we adopt dropout strategy [27] to the linear layer of the model.

Table 1. Statistics of datasets used in this paper

Dataset	users	items	ratings	reviews per user	reviews per item	length of review	density
Office Products	4905	2420	53258	14	35	124	0.44%
Amazon Instant Video	5130	1685	37126	8	27	101	0.43%
Toys and Games	2555	2211	19925	8	11	117	0.35%
Digital Music	5541	3568	64706	13	24	202	0.32%
Beauty	15201	9680	154150	11	19	97	0.11%

5. Experiments

In this section, we present our experimental setup and empirical evaluation. Our experiments are designed to answer the following research questions (RQs):

(1) **RQ1**-How does IGRec perform as compared with state-of-the-art review-based recommendation methods?

(2) **RQ2**-How do different hyper-parameter settings, such as depth of 'GCN' layer and mode of information transmission of interaction layer, affect IGRec?

(3) **RQ3**-Does the model really take advantage of the review information, and what is the effect of removing the review information?

5.1. Datasets

In our experiments, we used five publicly accessible datasets to evaluate our model. The five datasets are from Amazon 5-core, which include reviews on Office Products, Amazon Instant Video, Toys and Games, Digital Music, and Beauty. Since the raw data is very large and sparse, we use data preprocessing to ensure that each user and item has at least one review. For each dataset, we selected a fixed number of reviews for users and items

respectively. For each review, we selected a fixed-length word for TextCNN to extract the text information, which intercept for longer and pad for shorter. The model selects value at three-quarters of the total review length value as the hyper-parameter. The characteristics of these datasets are shown in Table 1.

In the experiments, we randomly split each dataset into three parts: training set (80 %), validation set (10 %) and test set (10 %). The final performance comparison results derive from the test set.

5.2. Evaluation Metric

The Mean Square Error (MSE) is adopted for performance evaluation:

$$MSE = \frac{1}{T} \sum_{(u,i) \in T} (\widehat{R}_{u,i} - R_{u,i})^2 \tag{27}$$

where T is the set of the user-item pairs in the testing set.

5.3. Baselines

To verify the performance of the IGRec model proposed in this paper, we compared the model with the following state-of-art recommendation methods.

- **DeepCoNN**[41]: Deep collaborative neural network is based on two parallel CNNs to learn the latent feature vectors of user and item from user review documents and item review documents respectively, and FM is used for rating prediction.
- **D_ATTN**[25]: Dual attention mechanisms that local and global attention are used to achieve the interpretability of latent features of user and item.
- **NARRE**[1]: Neural attentional regression model exploits two parallel CNNs and attention mechanism to learn the latent features of reviews, and integrates the reviews and items to complete the rating prediction.
- **NGCF**[32]: Neural graph collaborative filtering model only uses interactive connection data to extract high-dimensional interactive information by using the graph neural network, and then makes recommendations based on the learned embedding vectors of users and items.
- **MPCN**[29]: Multi-pointer co-attention networks utilize review-level co-attention and word-level co-attention by multi-pointer-learning to gain latent features of reviews, the final predicted rating is obtained through FM.
- **DAML**[18]: Dual attention mutual learning model exploits the local attention and mutual attention to learn latent features and integrate the rating and review features into an unified neural network to predict rating.

5.4. Parameter Setting

We use grid search to tune the hyper-parameters for all the methods based on the setting strategies reported by their papers. The latent dimension size is optimized from [8, 16, 32, 64, 128]. The embedding dimension size of the word in all models is set to 300. We set the batchsize to 128 for all models. The learning rate is tuned from [0.01, 0.001].

The range of dropout ratio is searched in [0.1, 0.3, 0.5, 0.7]. For the CNN text training module, the number of convolution filters is set to 100, the size of the sliding window is 3. The regularization parameter λ_Θ is set to 0.001. FM is used as the prediction layer for all models, which v is set to 10. For the NGCF model, the number of GCN layers is fine-tuned in [1, 2, 3, 4].

For IGRec, the dimensions of user embedding and item embedding are set to 8, embedding dimension of the word is set to 300, the sliding window size is set to 3, the dropout ratio is searched in [0.1, 0.3, 0.5], the learning rate is set to 0.001, the regularization parameter λ_Θ is set to 0.001, the depth of the 'GCN' layer is set to 2. v is set to 10 for FM rating prediction.

Table 2. Performance comparison on five datasets for all methods. The best and the second best results are highlighted by boldface and underlined respectively. $\Delta\%$ denotes the improvement of IGRec over the best baseline performer.

Method	Office Products	Amazon Instant Video	Toys and Games	Digital Music	Beauty
DeepCoNN	0.7337	0.9634	0.9789	0.8129	1.2052
D_ATTN	0.7064	0.9663	0.8854	0.8135	1.2113
NARRE	0.6931	0.9737	0.9108	0.8098	1.2035
NGCF	0.7066	0.9930	0.8904	0.8065	1.2182
MPCN	0.7109	0.9645	<u>0.8781</u>	0.8655	1.2386
DAML	<u>0.6852</u>	<u>0.9583</u>	0.9085	<u>0.8037</u>	<u>1.1878</u>
IGRec	**0.6728**	**0.9428**	**0.8660**	**0.7798**	**1.1520**
$\Delta\%$	1.80	1.61	1.37	2.97	3.01

5.5. Performance Comparison

The overall performance of all methods is reported in Table 2. We can see that the IGRec outperforms the baselines on the five datasets. This ascertains the effectiveness of our proposed model and clearly answers **RQ1**. Here, we make the following observations.

First, we can see that DeepConn performs worst on all five datasets. However, it is not far behind the other two CNN-based methods in some datasets. Such as in the Digital Music dataset, DeepConn and D_ATTN have similar performances, in the Beauty dataset, DeepConn and NARRE gain similar MSE. This can be explained by the fact that although the model can capture reviews information to some extent, the lack of interaction information affects the effect of the model. Surprisingly, the NGCF model does not use review data, but only interactive connection data. The performance of the NGCF model was good in all five datasets, which also verified the importance of interactive connection data. Further improvement of the NGCF model was limited by the absence of review information.

Second, MPCN is not stable in the five datasets, especially in the Digital Music dataset, its performance lags far behind that of other models. However, it get the second best result in the Toys and Games dataset. One possible explanation is that MPCN can

get better extraction of reviews information through word-level co-attention and review-level co-attention. However, in the sparse data environment, the review information may be less important than the interaction information between users and items. On the other hand, the prediction performance is adversely affected by irrelevant information within the reviews. The performance of DAML can be used as evidence to explain the importance of interactive information. DAML uses dual attention mechanism to learn user-item interactions and get the second best result in the four datasets. This can be explained that DAML not only uses the local attention mechanism to learn the importance of words in sentences, but also uses mutual attention to extract the interactive information of latent features. However, the model uses an attention mechanism alone and may fail to capture the full high-order feature interaction. Third, IGRec consistently achieves the best MSE

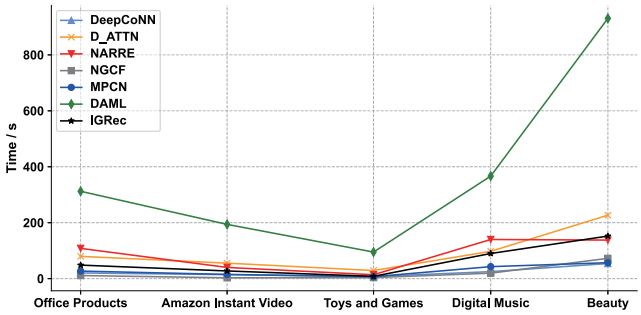

Fig. 3. The average consuming time of each epoch of different models in all five datasets.

scores across the five datasets. Surprisingly, as data sparsity increases, so does the performance of the model. Compared with the DAML model, IGRec does not use a more complex information extraction process for the review text information, but only extracts the review information through TextCNN, so the complexity of the model decreases dramatically. Compared with the review text information, we pay more attention to the interactive information extraction of latent features. The graph neural network for interactive information extraction can make up for the lack of review text information extraction, hence improving the model prediction performance.

As shown in Figure 3, we unified the batchsize of all models as 128, and calculated the average time spent for each epoch. The results show that the DAML model is the most time-consuming among all the models due to the fine-grained processing of the review text data. Because the NGCF model does not process review data, it can get training results more quickly when the number of users and items is small. However, when the number of users and items is large, for example in Beauty, it consumes more time than DeepCoNN and MPCN, which is caused by the increase of nodes and interaction data in the graph neural network. Although our model has a slight increase in time consumption compared with NGCF, DeepCoNN and MPCN, it consumes less time compared with other models. In addition, compared with NGCF, DeepCoNN and MPCN, the accuracy

of our model has been greatly improved, which shows that the efficiency of our model is the best.

5.6. Parameter Sensitivity Analysis

In order to answer **RQ2**, we mainly analyze: the dimension of latent feature, dropout ratio, depth of 'GCN' layer and mode of information transmission of interaction layer.

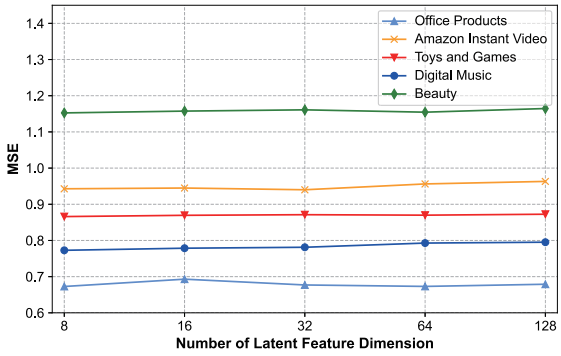

Fig. 4. The plots of the impact of latent feature dimension number.

The Dimension of Latent Feature: As shown in Figure 4, the dimension of latent feature is tuned from [8, 16, 32, 64, 128]. We can find that the performance of the IGRec model changes very little as the dimensions change. Even if the dimension number is much smaller, such as 8, the model can still achieve nearly optimal prediction accuracy in the five datasets. So we believe the latent feature dimension have little effect on the experimental performance. Moreover, the increase of latent feature dimension can increase the computational complexity of the model. Therefore, the dimension of the latent feature is set to 8.

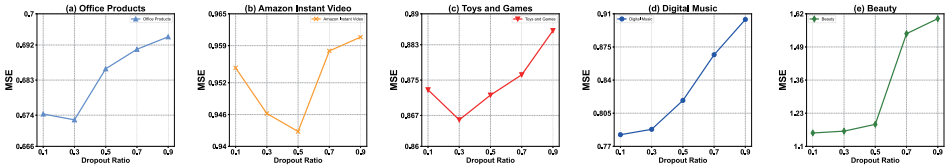

Fig. 5. The plots of the impact of dropout ratio.

The Impact of Dropout: As shown in Figure 5, a suitable dropout ratio greatly affects the performance of the model, which tends to perform better in smaller cases. The best dropout ratio is different for the five datasets. For Amazon Instant Video dataset, the best

dropout ratio is 0.5. While the best dropout for other datasets is 0.1 or 0.3. So the dropout ratio for the model is searched in [0.1, 0.3, 0.5].

Table 3. Performance comparison on five datasets for depth of the 'GCN' layer. The best results are highlighted by boldface.

	Office Products	Amazon Instant Video	Toys and Games	Digital Music	Beauty
l=0	0.6856	0.9612	0.8836	0.8392	1.2334
l=1	0.6812	0.9536	0.8748	0.8021	1.1845
l=2	0.6728	**0.9428**	**0.8660**	0.7798	**1.1525**
l=3	**0.6723**	0.9468	0.8717	**0.7732**	1.1532
l=4	0.6842	0.9568	0.8730	0.8045	1.1872

Table 4. Performance comparison on five datasets for mode of the 'GCN' layer. The best results are highlighted by boldface.

	Office Products	Amazon Instant Video	Toys and Games	Digital Music	Beauty
$Contact + Linear$	0.7023	0.9832	0.9089	0.8129	1.2052
$Dot + Attention$	0.6934	0.9726	0.8854	0.8135	1.1913
$Dot + Linear$	**0.6812**	**0.9536**	**0.8748**	**0.8021**	**1.1845**

Depth of 'GCN' Layer: To investigate whether IGRec can benefit from multiple 'GCN' layers in the interaction layer, we vary the model depth, which search the layer numbers in the range of [0, 1, 2, 3, 4]. Table 3 summarizes the experimental results, l is the number of 'GCN' layers. Through experiments, we have the following observations. As the number of layers increases, the performance of the model will improve at first. On some datasets (Amazon Instant Video, Toys and Games, Beauty), the two layers are optimal, while on other datasets (Office Products, Digital Music), the l=3 are optimal. Different from the NGCF model in the original paper, which the model experimental results show optimal number obtained when l=3 or l=4. In this paper, we found through experiments that the NGCF model and our model often reached the optimum at l=2. This is because we add the connection of user-user and the connection of item-item so that the user and item information can be included at one hop of the adjacency node. Therefore, the adjacency matrix that we designed to enable the model to aggregate information more quickly, and reduce the calculation of the model. When the depth of layer exceeds three, the performance of the model begins to degrade. This may be due to that deeper structure may cause noise for embedding. On the other hand, when l=0, it means that 'GCN' is not used for relation extraction, and the performance on all data sets is the worst, which verifies the effectiveness of 'GCN'.

Mode of Information Transmission: To investigate how the 'GCN' layer affects the performance, we transform the different message propagation functions in the interaction layer and set depth of 'GCN' layer l=1. We fixed the aggregation operator, only changed the combination operator. Table 4 summarizes the experimental results. $Linear$ denotes the full connection layer, Dot is the element-wise product, $Attention$ is the attention mechanism that defined as:

$$u_i = tanh(Wh_i + b) \tag{28}$$

$$a_i = \frac{exp(u_i^{\mathrm{T}} u_w)}{\sum_i exp(u_i^{\mathrm{T}} u_w)} \tag{29}$$

$$s = \sum_i a_i h_i \tag{30}$$

In Table 4, we can see that the combination of $Dot + Linear$ achieves the best performance in the all five datasets. The reason for the poor performance of $Contact + Linear$ may be that in each mini-batch training, the $Contact$ operation is to contact the information of all nodes. However, each mini-batch only contains the information of relevant nodes, the model must global updates in each mini-batch iteration, which lead to noise and affect the performance. The same limitation apply to attention mechanism, it is the calculation of global attention in each mini-batch iteration, all nodes are involved in the calculation, and some irrelevant nodes affect the calculation of attention. However, for the combination of $Dot + Linear$, in each batch of training, each embedding only interacts with the embedding at the corresponding position, which avoids unnecessary noise and improves model performance.

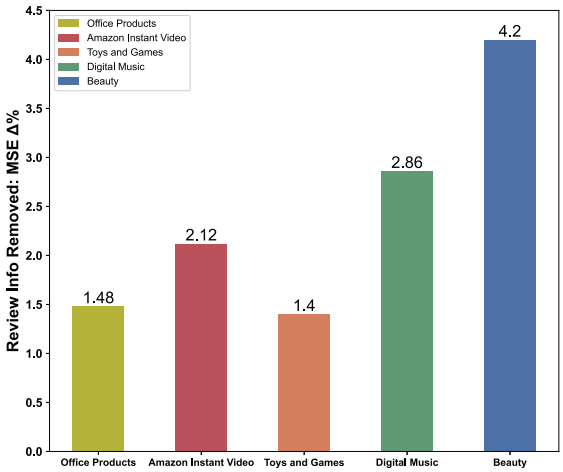

Fig. 6. The plots of the impact of review information. $\Delta\%$ denotes the percentage of performance degradation after the deletion of review information.

5.7. The Review Importance Analysis

To answer **RQ3**, we conducted an experiment in which review information was removed from five datasets. The experimental results are shown in Figure 6. We can see that in the five datasets, after the review information is removed, the performance of the model is significantly reduced, which indicates that the review information is an important resource for the recommendation system. Particularly in the Beauty dataset, the deletion of review information resulted in the greatest degradation of model performance. This may be because Beauty is the sparsest of the five datasets, the review information as additional information can greatly compensate for the lack of interaction information and thus improve the performance of the model.

6. Conclusion

In this work, we propose a novel enhancing interactive graph representation learning for review-based item recommendation. In this model, we construct the graph with users and items as nodes and reviews as edges, and further add the connection of the user-user and the item-item to the graph. TextCNN is used to extract review text information as edge embedding. The attention mechanism is developed to fuse edges information into node and the graph neural network is exploited for high-dimensional information interaction of nodes. Finally, The learned embedding of user-item pair is inputted in factorization machine to get the final rating. Experiments on five real-world datasets from Amazon show that our method consistently outperforms the existing state-of-the-art methods.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China (62073295, 62072409), Zhejiang Provincial Natural Science Foundation (LR21F020003), and Fundamental Research Funds for the Provincial Universities of Zhejiang (RF-B2020001). Zhejiang Province Basic Public Welfare Research Project under Grant No. LGG20F030008.

References

1. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 World Wide Web Conference. pp. 1583–1592 (2018)
2. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Ge, S., Wu, C., Wu, F., Qi, T., Huang, Y.: Graph enhanced representation learning for news recommendation. In: Proceedings of The Web Conference 2020. pp. 2863–2869 (2020)
5. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1024–1034 (2017)
6. He, X., Chen, T., Kan, M.Y., Chen, X.: Trirank: Review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1661–1670 (2015)
7. Jin, J., Qin, J., Fang, Y., Du, K., Zhang, W., Yu, Y., Zhang, Z., Smola, A.J.: An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 75–84 (2020)

8. Kim, D., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 233–240 (2016)
9. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Kong, X., Mao, M., Wang, W., Liu, J., Xu, B.: Voprec: Vector representation learning of papers with text information and structural identity for recommendation. IEEE Transactions on Emerging Topics in Computing (2018)
13. Kong, X., Zhang, J., Zhang, D., Bu, Y., Ding, Y., Xia, F.: The gene of scientific success. ACM Transactions on Knowledge Discovery from Data (TKDD) 14(4), 1–19 (2020)
14. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
15. Li, A., Qin, Z., Liu, R., Yang, Y., Li, D.: Spam review detection with graph convolutional networks. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2703–2711 (2019)
16. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
17. Ling, G., Lyu, M.R., King, I.: Ratings meet reviews, a combined approach to recommend. In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 105–112 (2014)
18. Liu, D., Li, J., Du, B., Chang, J., Gao, R.: Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 344–352 (2019)
19. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on Recommender Systems. pp. 165–172 (2013)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
21. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 701–710 (2014)
22. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog 1(8), 9 (2019)
23. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Recommender Systems Handbook, pp. 1–35. Springer (2011)
24. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning. pp. 791–798 (2007)
25. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of the 11th ACM Conference on Recommender Systems. pp. 297–305 (2017)
26. Shen, G., Zhao, Z., Kong, X.: Gcn2cdd: a commercial district discovery framework via embedding space clustering on graph convolution networks. IEEE Transactions on Industrial Informatics (2021)
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)

28. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009 (2009)
29. Tay, Y., Luu, A.T., Hui, S.C.: Multi-pointer co-attention networks for recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2309–2318 (2018)
30. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
31. Wang, W., Liu, J., Yang, Z., Kong, X., Xia, F.: Sustainable collaborator recommendation based on conference closure. IEEE Transactions on Computational Social Systems 6(2), 311–322 (2019)
32. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 165–174 (2019)
33. Wang, Z., Xia, H., Du, B., Chen, S., Chun, G.: Joint representation learning with ratings and reviews for recommendation. Neurocomputing (2020)
34. Wu, L., Quan, C., Li, C., Wang, Q., Zheng, B., Luo, X.: A context-aware user-item representation learning for item recommendation. ACM Transactions on Information Systems (TOIS) 37(2), 1–29 (2019)
35. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
36. Xu, Z., Jiang, H., Kong, X., Kang, J., Wang, W., Xia, F.: Cross-domain item recommendation based on user similarity. Computer Science and Information Systems 13(2), 359–373 (2016)
37. Xue, G., Zhong, M., Li, J., Chen, J., Zhai, C., Kong, R.: Dynamic network embedding survey. arXiv preprint arXiv:2103.15447 (2021)
38. Yang, J.H., Chen, C.M., Wang, C.J., Tsai, M.F.: Hop-rec: high-order proximity for implicit recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems. pp. 140–144 (2018)
39. Załuski, A., Ganzha, M., Paprzycki, M., Bădică, C., Bădică, A., Ivanović, M., Fidanova, S., Lirkov, I.: Experimenting with facilitating collaborative travel recommendations. In: 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC). pp. 260–265. IEEE (2019)
40. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 83–92 (2014)
41. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 425–434 (2017)

Guojiang Shen received the B.Sc. degree in control theory and control engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively. He is currently a Professor with College of Computer Science and Technology, Zhejiang University of Technology. His current research interests include artificial intelligence theory, big data analytics, and intelligent transportation systems.

Jiajia Tan received the B.S. degree in Mathematics and Applied Mathematics from Xiamen University of Technology, China, in 2018. He is currently pursuing the M.S. degree in Zhejiang University of Technology. His current research interests include graph neural network and recommendation system.

Zhi Liu received the B.S. degree in Automatic Control and the M.S. degree in System Engineering from Xi'an Jiaotong University, Xi'an, China, in 1991 and 1994, respectively, and the Ph.D. degree in Computer Science and Technology from Zhejiang University, Hangzhou, China, in 2001.She is currently a Professor in the College of Computer Science and Technology, Zhejiang University of Technology. She is a member of the China Computer Federation. Her current main research interests include intelligent transportation system and intelligent computing.

Xiangjie Kong received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Full Professor with College of Computer Science and Technology, Zhejiang University of Technology. Previously, he was an Associate Professor with the School of Software, Dalian University of Technology, China. He has published over 150 scientific papers in international journals and conferences (with over 130 indexed by ISI SCIE). His research interests include network science, mobile computing, and computational social science. He is a Distinguished Member of CCF, a Senior Member of the IEEE, and a Member of ACM.

Received: February 28, 2021; Accepted: November 15, 2021.