Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-03626-7

ORIGINAL RESEARCH

Integrating reviews and ratings into graph neural networks for rating prediction

Yijia Zhang[1,2] · Wanli Zuo[1,2] · Zhenkun Shi[3] · Binod Kumar Adhikari[4]

Received: 22 March 2021 / Accepted: 30 November 2021

Abstract
In the area of recommendation systems, one of the fundamental tasks is rating prediction. Most existing neural network methods independently extract user's and item's review features utilizing a parallel convolutional neural network(CNN) and use them as the representation of users and items to predict rating scores. There are two main drawbacks of these methods: (1) They typically only leverage user or item reviews but ignore the latent information provided by user-item interactions. (2) The historical rating scores are not integrated into the representation of users and items, they are simply used as labels to train models. Thus the rating information is not adequately utilized, leading to the prediction performance of these methods is not superior. To remedy these drawbacks mentioned above, in this paper, we build a unified graph convolutional network(GCN) to capture the interaction information between user and item, also obtain additional information provided by reviews and rating scores. As both reviews and ratings carry interactive messages among users and items, they would magnify the learning performance of user-item features. Specifically, we first construct a multi-attributed bipartite graph(MA-bipartite graph) to represent users, items, and their interactions through reviews and ratings. Then we divide the MA-bipartite graph into two sub-graphs according to the attributes of the edge types which represent the user-item interaction in review domain and item domain respectively. Next, an attributed GCN model is explicitly designed to learn latent features of users and items by incorporating review embeddings and rating score weights. Finally, the attention mechanism is carried to fuse user and item features dynamically to conduct the rating prediction. We conduct our experiments on two real-world datasets. The results demonstrate that the proposed model achieved the state-of-the-art performance, which increases the prediction accuracy by more than 3%, compared with baseline methods.

Keywords Reviews · Recommendation · Graph convolution network · Rating prediction

1 Introduction

Recommendation system(RS), as an information filtering tool, has been widely adopted in online E-commerce and social websites such as Amazon and Epinions. On these platforms, the rating score shows his/her tastes or satisfaction towards the item (Jin et al. 2016; Gojali and Khodra 2016; Xing et al. 2019). Accordingly, it is essential to estimate the user's unknown rating for the item, this will not only evaluate the user's preferences but also increase the revenue of the websites. Conventionally, collaborative filtering (CF) based methods(Sarwar et al. 2001) have been widely discussed and studied for the rating prediction. In these works, they usually learned the latent features of users and items based on matrix factorization(MF) with user-item historical records(Koren 2008; Koren et al. 2009; Mnih and Salakhutdinov 2007). However, CF-based methods easily suffer from the data sparsity and cold-start problems, thus reduces the accuracy of the recommendation, which are all inherent obstacles for latent factor learning-based solutions.

To tackle these limitations, some related work usually takes both user's historical records and various types of side information into consideration, such as social relations, textual reviews, contextual information, et al. The side information contains important complementary information which helps capture user preference and item characteristics in

Contributing author:Yijia Zhang, Binod Kumar Adhikari.

✉ Wanli Zuo
wanli@jlu.edu.cn

✉ Zhenkun Shi
zhenkun.shi@tib.cas.cn

Extended author information available on the last page of the article

Published online: 24 February 2022

sparse and cold-start scenarios. Among them, review information, as an important auxiliary information which provides rich semantic signals, is readily available and common in many e-commerce and social websites. The reviews written by the user for items accompanying explicit rating scores in the website can reveal both user preference and item characteristics, they can also provide complementary information to infer and explain the underlying dimensions for the rating prediction(Seo et al. 2017). Towards this end, Many techniques have been developed to exploit reviews for more accurate rating prediction. Previously, they typically utilized sentiment analysis(Pero and Horváth 2013) and topic technologies(Wang and Blei 2011; Tan et al. 2016; McAuley and Leskovec 2013) to regularize user and item features learned through MF. Although these methods have achieved some progress and alleviated the sparseness problem of traditional CF methods, the abilities to rely on these technologies to extract features from reviews are still limited.

In order to derive effective features from reviews, more and more researchers attempt to utilize deep learning methods such as convolutional neural network (CNN), recurrent neural network (RNN), and attention mechanism to learn review features(Zheng et al. 2017; Chen et al. 2018; Kim et al. 2016; Wu et al. 2019a; Liu et al. 2019; Cheng et al. 2019). These deep learning-based methods are powerful in processing unstructured multimedia data, thus generate better review features than conventional methods. Generally, these methods are mainly divided into two classes as follows. First, CNN-based methods such as (Zheng et al. 2017; Kim et al. 2016): CNN has been a popular method to extract review features in the task of rating prediction since the TextCNN was proposed by (Kim 2014), these CNN-based methods utilize the TextCNN to obtain semantic understandings of reviews, leading to significant improvements than traditional MF based methods. Second, attention mechanism-based methods such as (Chen et al. 2018; Liu et al. 2019): These methods combine CNN and attention mechanism to produce fine-grained review features. They utilize the attention mechanism to focus on important sentences, or they identify the important words from textual auxiliary information. Thus they can estimate dynamic review influence and provide a way to offer semantic interpretations for the recommendation(Cheng et al. 2019; Guan et al. 2019). Generally, these methods have better accuracy of rating prediction than methods that use CNN alone.

Despite these deep learning-based methods facilitate the rating prediction with reviews, there are still several limitations: (1) They typically encode user and item reviews to represent their features independently and fuse them by factorization machine (FM) to predict rating scores, which usually makes the model overconfident and over-fitting(Sachdeva and McAuley 2020). Moreover, they ignore the effect of the review as interactive information on the user and item. Reviews can not only be encoded as the feature representation of the user and the item, the relationship data it reflects can also enhance the learning of the user and product features. The experiments of previous review-based methods such as (Zheng et al. 2017; Catherine and Cohen 2017; Chen et al. 2018) have shown that the review corresponding to each user-item pair leads to more accurate rating predictions. Therefore, we argue that modeling the interaction through the corresponding review for each user-item pair can further improve the accuracy of the rating prediction. (2) Most methods ignore the explicit rating scores when encoding user and item features, they only utilize rating scores as labels by matching the predicted scores. However, rating scores also provide important information which indicates explicit interactions between users and items, they will complement the reviews to learn better user and item features. Therefore, rely on reviews alone to model the interaction between the user and the item is not effective enough, it is more appropriate to take both reviews and rating scores into considerations. (3) Despite some works have considered to learn user and item features in the review domain and item domain(Wu et al. 2019a), they still utilize a static strategy to combine them linearly, thus can not learn the contributions of the two domains adaptively.

To address the first two problems and inspired by the research of recent graph neural networks(GNN), we aim to build an attributed graph to model user-item interactions through both reviews and rating scores. Recently, some studies have introduced graph convolutional networks (GCN) into the recommendation to model interactions between users or items, which achieves better recommendation performance than other deep learning methods. These methods mine the hidden user-item interaction information utilizing user-item bipartite graphs and gain user and item features by high-order graph convolutional operations. Typical work such as (Berg et al. 2017) generally classifies user-item links according to different rating scores and employs one convolutional layer to exploit the direct connections between users and items, which achieves more accurate rating predictions than other deep learning based methods. Based on these GCN based methods,we construct an attributed user-item bipartite graph to represent heterogeneous user-item interactions. Different from the previous GCN based methods which only consider unattributed bipartite graphs to model direct connections between users and items, or encode the user and item embedding with the metadata, we utilize reviews and ratings both as edge attributes of the graph, instead of the unstructured bipartite graph using the user's click and purchase record. Based on this graph, we design the attributed graph convolutional network (AGCN) to aggregate both neighbor nodes and interaction attributions(i.e., reviews, rating scores) to learn better user and item features.

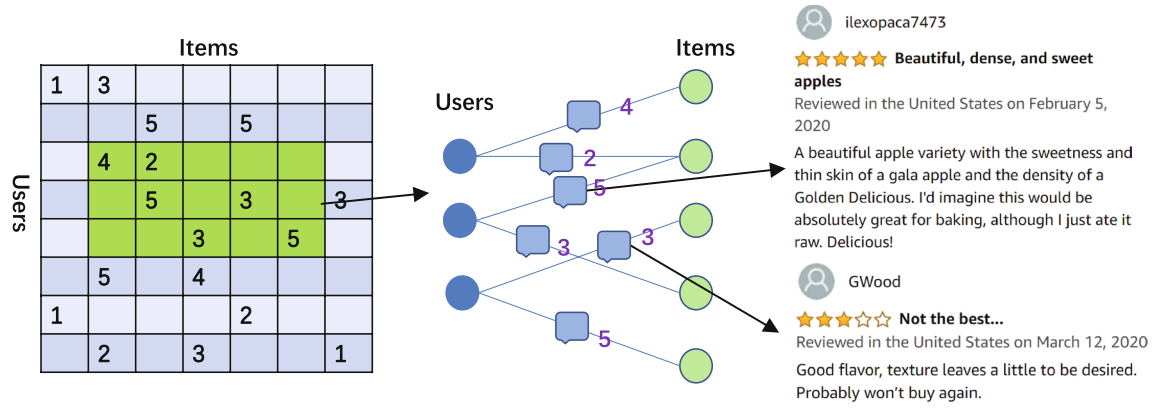


Fig. 1 Attributed user-item interactions graph. On the left side of the figure, there is the original user-item matrix, the values on the matrix represent the rating scores. On the right side of the figure, there is the attributed graph we constructed according to the user-item matrix, both rating scores and reviews are as edges on the graph

To adddress the last problem, we aim to utilize a dynamic method to fuse user and item features. Motivated by recent works based on the attention mechanism(Veličković et al. 2017; Chaudhari et al. 2019; Zhou et al. 2018; Tay et al. 2018; Wang et al. 2017; Zhu et al. 2019), we introduce the attention mechanism in this work to fuse the user and item feature flexibly. Attention mechanism has been widely used in many other tasks like natural language understanding(NLP)(Cui et al. 2016), recommendation (Zhang et al. 2018), computer vision (Xu et al. 2015), etc. It learns to pay attention to only the most important parts of the target and provide more accurate alignment for each feature. Specifically, self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence(Vaswani et al. 2017), which is an effective method for combining features. Therefore, in our work, we consider to utilize self-attention mechanism to learn the contributions of different features automatically. By utilizing self-attention mechanism, our model can combine user and item features from different domains more effectively compared to the static strategy adopted in the previous method, thus enhance the learning of user and item features.

Based on the above concerns, in this work, we propose our model named AGCR(Attributed GCN for Rating Prediction). Our model mainly consists of two modules: First, we use an attribute graph to represent the interactions between users and items through reviews and rating scores as Figure 1 shows, and learn user and item features by (attributed-GCN) AGCN method. Second, we design a dynamic strategy to fuse user and item features from the review and item domain utilizing an attention mechanism. The main contributions are as follows:

(1) We construct an attributed graph to model heterogeneous interactions between users and items called multi-attributed bipartite graph (MA-bipartite graph), then we propose a novel model via graph convolutional networks (GCN) for rating prediction working on this graph. To the best of our knowledge, our model is the first work that derives review to model semantic interactions between users and items based on GCN methods in the task of the recommendation.

(2) We design an attributed GCN method (AGCN) to learn user and item features in the review domain and item domain. In our AGCN method, reviews and rating scores are integrated into the graph convolutional operations to learn user and item features. We found that the combination of rating scores and review information can further improve the accuracy of rating prediction through the experimental results.

(3) We utilize an attention mechanism to fuse user and item features from the review and item domain dynamically, which can balance the contributions on the final features of two types of interactions (user-review-item and user-rating-item) adaptively. Our experimental results demonstrate that the fusion strategy of attention mechanism can fuse features from two domains effectively. Extensive experiments also demonstrate that utilize attention mechanism can achieve better the performance than the FM method.

(4) We conduct extensive experiments on real-world data sets, the experimental results demonstrate the superiority of our model over strong and several state-of-art baselines. The experimental results of the ablation study show that the AGCN module and the attention mechanism are both useful, and the experimental results of the parameter analysis show that our model can achieve the best performance with different parameters than all the baselines on the two datasets.

The remainder of this article is organized as follow: Related work is reviewed in Section 2. Section 3 introduces our model AGCR in detail. The experimental settings and results are presented in Section 4. Finally, Section 5 concludes the article and discusses future works.

2 Related work

Our work is related to the studies of review-based recommendation and graph convolutional network(GCN) based Recommendation. Therefore, the relevant literature will be reviewed briefly in this section.

2.1 Review-based recommendation

Although traditional CF-based methods (Koren 2008; Koren et al. 2009; Mnih and Salakhutdinov 2007) have gained significant success in the past decades, they have two main limitations: sparsity and cold-start. To tackle these limitations, reviews as the widely used auxiliary information have been utilized, and exploiting reviews for rating prediction has become a hot research topic in the recommendation domain. We mainly discuss these review-based methods from the following two aspects.

Topic-based methods: Previously, most methods employ topic modeling techniques such as Latent Dirichlet Allocation(LDA) to exploit latent topics from review text and incorporate it with matrix factorization(MF) methods (Bao et al. 2014; Wang and Blei 2011; McAuley and Leskovec 2013; Ling et al. 2014; Tan et al. 2016; Zhang and Wang 2016). Among them, HFT (McAuley and Leskovec 2013) and CTR (Wang and Blei 2011) are two typical methods that adopt similar topic modeling techniques, they employ LDA to model the reviews' likelihood and combine topic vectors with latent factors learned through MF to improve rating prediction accuracy. RBLT (Tan et al. 2016) also employs similar techniques to derive topic features from rating-boost reviews, the authors assume that the latent topics and latent factors are in a shared topic space, thus they linearly combine them into an MF framework to derive item characteristics. TopicMF (Bao et al. 2014) utilizes a biased matrix factorization model for rating prediction by jointly considering user ratings and review text, the authors transform the latent topic from the unstructured reviews in their transform function, then these latent topics are linked with the latent factors. RMR (Ling et al. 2014) learns item's features using topic models from reviews by a similar technique, but it models ratings using a mixture of Gaussian instead of MF methods. Despite these methods outperform traditional CF-based methods that solely rely on user-item interaction data, the techniques they rely on are linear text processing strategies. Hence, these methods are still not sufficient enough for rating prediction for ignoring to capture the nonlinear and complex structure of phrases and sentences in the unstructured reviews.

Deep learning-based methods. To tackle the above limitations, there is a trend to employ deep learning to deal with unstructured reviews recently. Several methods have been proposed, that apply deep textual modeling techniques on reviews for recommendations(Wang et al. 2015; Kim et al. 2016; Zheng et al. 2017; Seo et al. 2017; Catherine and Cohen 2017; Chen et al. 2018; Liu et al. 2019; Ahmed and Ghabayen 2020). CDL(Wang et al. 2015) utilizes SADE to learn the deep feature representations of reviews and integrates them into a probabilistic matrix factorization (PMF) (Mnih and Salakhutdinov 2007) to predict ratings. This work still employs bag-of-words (Collobert et al. 2011) representation to learn the latent topic, which limits the efficiency of review feature representations. Later, more researchers attempt to apply Convolutional Neural Network(CNN) and attention mechanism to further improve the review representations learning. For example, ConvMF(Kim et al. 2016) integrates CNN into PMF to capture contextual information in description documents for the rating prediction. DeepCoNN(Zheng et al. 2017) concatenates all reviews of a user or an item as an input to a CNN to learn the representation of the user or the item, then representations of users and items are concatenated and passed into a regression layer for rating prediction, that achieves better performance than previous review-based methods. However, DeepCoNN only achieves the best performance when the reviews for the target predicted user and item are available at train and test time. Based on this limitation, TransNets(Catherine and Cohen 2017) extends DeepCoNN by adding an additional layer (target network) to learn the representation of a target user-target item review at training time and then uses the learned representations to regularize the output of the source network which gains improvement in rating prediction against DeepCoNN.

There are also several methods for utilizing both CNN and attention mechanism to improve recommendation performance such as (Chen et al. 2018; Seo et al. 2017; Wu et al. 2019a; Liu et al. 2019), these methods can better characterize user's preference and provide interpretations for recommendations in review-level or aspect-level. For example, D-Attn(Seo et al. 2017) utilizes a dual attention-based CNN model to combine review text for rating prediction, the authors in this work apply both local and global attention layers and combine them in one network training, which makes the model more robust to eliminate noise and inconsistency in the review and rating data. NARRE(Chen et al. 2018) also utilizes attention-based CNNs to predict rating scores that provides a review-level explanation for rating prediction. CARL(Wu et al. 2019a) derives pair-dependent latent representations on the basis of user-item pairs instead

of learning a static user/item latent representation for rating prediction. It learns context-aware representations for each user-item pair based on their individual characteristics and their interactions together by exploiting both textual reviews and user-item interaction data. In this work, the author proposes a dynamic linear fusion strategy to aggregate the evidence from the two components for final rating prediction. These works only use user-item interactions to define the objective function for model training, that can't model complex and non-linear interactions between users and items. Different from this work, we utilize a graph structure-based method to encode user, item, and the interaction between them. In addition, these methods learn user and item review features as user and item features to predict ratings, which will lead to the over-fitting problem, thus the effectiveness of these methods is dependent on the bias(Sachdeva and McAuley 2020).

Based on the above discussion, there are three main differences between our model and recent work in the area of review-based recommendation. First, we consider not only review information to learn user and item features, but also rating scores, the two types of information are simultaneously used in our work to improve the accuracy of the rating prediction. Second, we model the direct interactions between the user and item, rather than learn user and item features dependently, which can learn better user and item features through heterogeneous interactions. Third, we adopt an attention mechanism to fuse user or item features that learned according to the two types of interactions (rating scores and review information), instead of using the FM framework to fuse these features as the previous works, which has been proved to have better performance to fuse features than the FM method.

2.2 Convolutional networks graph based recommendation

More recently, there has been a surge of methods that rely on graph structure or Graph Convolutional Network (GCN) for the recommendation task. Early, Bruna et al. (Bruna et al. 2013) developed a version of graph convolutions based on spectral graph theory. Following this work, a number of extension methods are proposed to make it adaptive to the recommendation(Hamilton et al. 2017; Ying et al. 2018; Berg et al. 2017; Wang et al. 2019a; He et al. 2020). Among them, GraphSAGE(Hamilton et al. 2017) is an inductive variant of GCN modified to avoid operating on the entire graph Laplacian. Later, many works adopt this method for large-scale recommendations. PinSage(Ying et al. 2018) is also a typical variant, it is a random-walk graph convolutional network utilizing a localized convolution operation, which is capable of learning embeddings for nodes in web-scale graphs containing billions. Some researchers tried to utilize GCN based methods to produce latent features of user and item nodes through a form of message passing on the graph, which incorporates collaborative signal into the GCN, such as (Berg et al. 2017; Wang et al. 2019a; He et al. 2020). Berg et al. propose GC-MC(Berg et al. 2017), a graph auto-encoder framework for the matrix completion task in recommender systems. In this work, they consider rating prediction as predicting labeled links in the bipartite user-item graph, combined with a bilinear decoder, new ratings are predicted in the form of labeled edges. He et al.(Wang et al. 2019a; He et al. 2020) utilize GCN to build recommendation frameworks dealing with implicit feedback recommendations, they first propose NGCF(Wang et al. 2019a) to explicitly encodes the collaborative signal in the form of high-order connectivities by performing embedding propagation of objects. Later, they improve NGCF by largely simplifies the model design by including only the most essential components in GCN for recommendation(He et al. 2020). There are also some works utilizing GCN to integrate various types of side information into the recommendation. Some methods consider structured auxiliary information such as social relations and utilize GCN to learn user preferences, such as (Wu et al. 2019c; Song et al. 2019). Another typical method utilize graph structure to represent sequential orders to deal with the sequential recommendation or the session-based recommendation, such as (Wu et al. 2019b). Different from these methods, we consider unstructured review text through GCN rather than simple structured side information. Moreover, we represent review text as attributes of edges instead of encoding them as node representations, semantic information contained in reviews could both reflect the user preference and user-item interactions.

Despite these methods achieve good performance, they still have two main problems as follows. First, Most methods only consider unattributed bipartite graphs, thus can't model fine-grained interactions between users and items. Despite some research classify user-item links according to different rating scores, they still ignore semantic interactions contained by textual reviews. In fact, reviews between users and items manifest how and why they are related, which reflects fine-grained user preferences towards items. Second, Most existing works integrate structured side information such as social relations and sequential orders through graphs, lacking exploiting complex unstructured side information which contains more useful information for the recommendation. Therefore, integrate review information into the graph can further capture fine-grained user-item interactions and improve the representations learning of user and item features.

3 The proposed model

In this section, we introduce our proposed model, AGCR. We will first describe the mathematical notations and the problem setting. Then the details of our proposed model will be presented.

3.1 Problem setting

Let $(u, i, r_{u,i}, rev_{u,i})$ be the tuple in the training set $\mathcal{T}$, which denotes a review $rev_{u,i}$ written by user u for item i with rating $r_{u,i}$. There are N tuples in the training set, noting that each review $s_{u,i}$ comes with an overall rating $r_{u,i}$, both express the satisfaction of the user on the item. According to these tuples, we build an attributed graph $G = (\mathcal{V}, \mathcal{E})$ as Figure 1 shows, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges between users and items. Nodes $\mathcal{V} = \{(u, i) \mid \forall u \in \mathcal{U}, \forall i \in \mathcal{I}\}$ include a set of users and items, where $\mathcal{U}$ and $\mathcal{I}$ denotes user and item sets. Each edge is associated with the rating score and review between user u and item i. In this work, our goal is to learn both user and item features according to the attributed graph and predict user's unknown ratings for items that users have not rated yet. The mathematical notations used in this paper are summarized in Table 1.

Definition 1 Multi-attributed bipartite graph(MA-bipartite graph). A multi-attributed bipartite graph $G = (\mathcal{V}, \mathcal{E})$ is a user-item bipartite graph with multiple attributes on the edges. $\forall \mathcal{E} \in E_t$, where E_t denotes the set of multiple types of edges. The edges denote heterogeneous types of corresponding relations between users and items(i.e., reviews, rating scores). In this paper, we only adopt reviews and rating scores as two types of interactions. The attributes of the edges can be extended to multiple types of interactions between users and items.

Definition 2 Attributed subgraph. An attributed subgraph $G = (\mathcal{V}', \mathcal{E}')$ is extracted from the multi-attributed bipartite graph(MA-bipartite graph), there is only single type of user-item interaction for the edge attribute in this subgraph.

3.2 Rating prediction architecture

We devise the architecture of our model based on the MA-bipartite graph, which is capable to model higher-order user-item latent features through their two types of interactions(i.e., reviews, rating scores). As Figure 2 shows, all the layers are classified into two parts, one is for encoding text review, the other is for learning features through the graph. To this end, we first derive a CNN-based module to encode text reviews, then design a GCN-based module to learn user and item features. After that, we calculate the unknown corresponding rating score as follows:

$$\hat{y}_{ui} = q_u q_i + \mu + \beta_u + \beta_i \tag{1}$$

Where β_u and β_i are the corresponding bias for user u and item i respectively and μ is the global biased term. q_u and q_i are learned user and item features.

Previous works usually design an independent architecture to learn user and item reviews, or an interactive architecture through shared weights, then sent them into an FM layer to model the interactions between different kinds of features. Different from them, our architecture is interactive through real user-item interactions. In addition, we adopt the attention mechanism rather than FM to combine user and item features.

Table 1 Notations

Symbol	Description
G	the input network
$\mathcal{U}$	the user node set
$\mathcal{I}$	the item node set
$\mathcal{N}(u)$	the neighbor set of user u in G
$\mathcal{N}(i)$	the neighbor set of item i in G
$V_{1:n}$	word vectors of each review
z_j	the j_{th} feature map in the convolutional layer
K_j	the j_{th} kernel in the convolutional layer
o_j	the output of j_{th} neuron in the convolutional layer
b_j	the bias of j_{th} convolutional kernel
t	the window size of convolutional kernel
W	the weight matrix of the fully connected layer
b	the bias of the fully connected layer
$\mathbf{h_s}$	the representation of review s in G
$\mathbf{h_u}$	the representation of user u in G
$\mathbf{h_i}$	the representation of item i in G
$\mathbf{p_u}$	the final features for rating prediction of user u
$\mathbf{q_i}$	the final features for rating prediction of item i
y_{ui}	the ground truth rating of user u towards item i
$\hat{y}_{ui}$	the predicted rating of user u towards item i
α	the learning rate
λ	the regularization parameter
M	the number of users
N	the number of items
e	the dimension of word embeddings
d	the dimension of latent factors

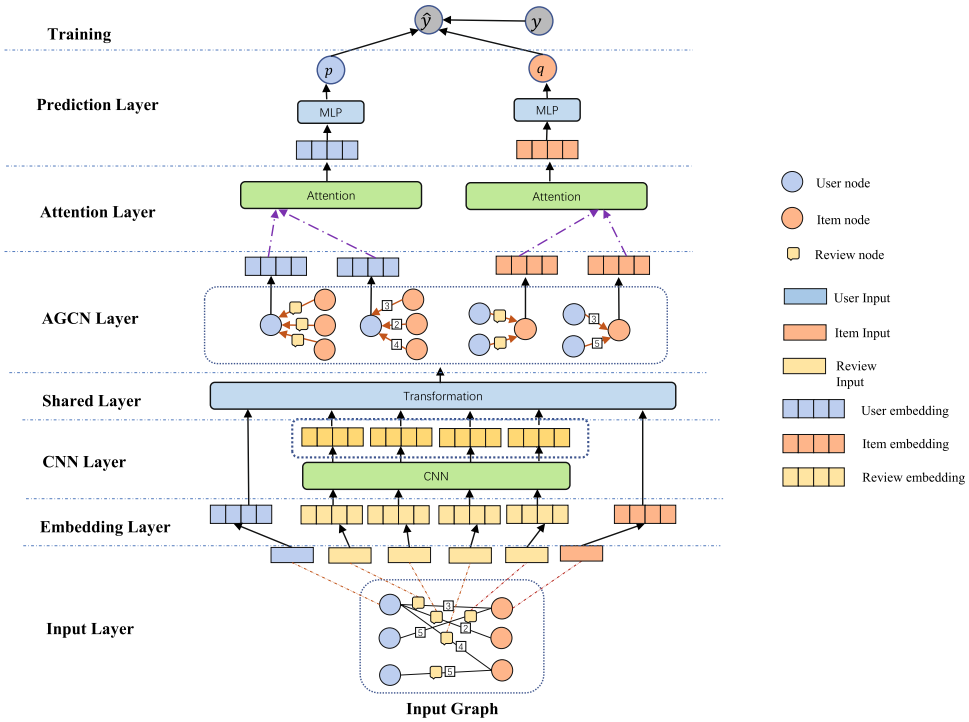


Fig. 2 The architecture of our model

3.3 Encode review features

To deal with unstructured reviews as edge attributes, our model first utilizes CNN to encode review features. Some previous work (Kim et al. 2016; Zheng et al. 2017) usually concentrate all the reviews of users and items to represent user preference and item characteristics. In addition, some attention mechanism-based methods consider each review related to the user-item pairs, which can predict rating scores more accurately. Inspired by this, we believe that concentrating all the reviews only indicates the approximate user preference and item attribute, they can not indicate the user's fine-grained preference for a particular item. Therefore, we first utilize CNN techniques to achieve the embedding of each review corresponded with the user and item. We design the following layers based on the work TextCNN(Kim 2014) that includes convolution layer, max pooling, and fully connected layer which are explained as follows:

Input Layer. Since we model interactions between users and items through heterogeneous information, we construct the MA-bipartite graph as the input of our model. In this layer, user and item nodes are represented by their identifications, reviews are represented by a set of words, rating scores are represented by the number range from 1 to 5.

Embedding Layer. We apply the one-hot embedding to represent users and items in this layer. As for the review, we adopt word embedding techniques (Collobert et al. 2011; Kim 2014) to exploit their semantic representations. Each review written by user u to item i are denoted as $S_{1:n} \in \mathbb{R}^{n \times e}$, composed of n words. Each word is mapped into a word embedding w_i following the work in (Mikolov et al. 2013) and then the review embedding matrix is represented as follows:

$$S_{1:n} = (w_1, ..., w_{i-1}, w_i, w_{i+1}, ..., w_n)^T \tag{2}$$

Where n is the number of words in the review, e is the embedded dimension of each word, w_i is the $i-th$ word in the document $S_{1:n}$.

Convolution Layer: In this layer, we perform a convolutional operation regarding to each filter f_j as:

$$z_j = ReLU(S_{1:n} * W_j + b_j) \tag{3}$$

Where $W_j \in \mathbb{R}^{t \times e}$ is the convolutional weight matrix for filter f_j, t is the filter size, b_j is a bias term, $*$ symbol is the convolutional operation, z_j is the document feature extracted by filter f_j over the sliding window. Specifically, we use Rectified Linear Units (ReLUs)(Vinod and Hinton 2010) as the activation function, which is defined as:

$$ReLU(X) = max(x, 0) \tag{4}$$

Max-pooling Layer: Later, we perform a max-pooling operation to reserve the most valuable contextual feature of each filter:

$$o_j = max\{z_1, z_2, z_3, ..., z_{(n-t+1)}\} \tag{5}$$

After the max-pooling operation, all reserving features are concatenated as the output of the max-pooling layer:

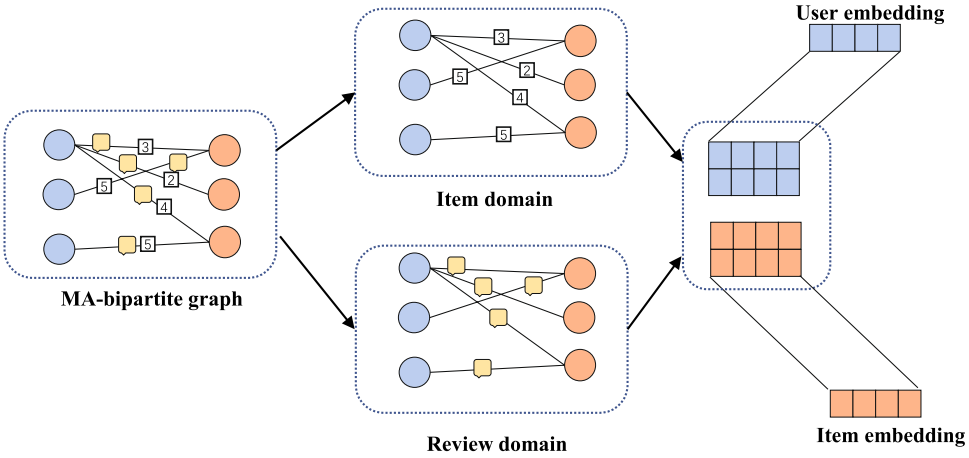


Fig. 3 Learning user and item representations through AGCN. We first extract two subgraphs from the attributed graph as shown on the left side of the figure. Then we use AGCN method to learn user and item representations on the two subgraphs. Finally, the representations are fused through attention mechanism as shown on the right side of the figure

$$O = \left\{o_1, o_2, o_3..., o_{n_1}\right\} \tag{6}$$

Fully Connected Layer: Finally the output of the max-pooling layer is passed to a fully connected layer to get the latent representation of each review as:

$$\mathrm{h}_s = f(W \times O + b) \tag{7}$$

Where W is the weight matrix and b is the bias term. f indicates the activation function, ReLUs is used as the function in this work.

3.4 Learn latent features by AGCN

After achieving the review features produced by the CNN module, we next integrate the review features into the graph convolutional network to learn user and item latent features. As there are two types of edges between the user and item, we learn user and item features respectively according to different interactions and fuse the two types of features to get the final user and item features. To achieve this, the whole graph is divided into two subgraphs as Figure 3 shows, one is the user-rating-item subgraph that represents user-item interactions in the item domain, where edges between user and item nodes indicate rating scores, the other is the user-review-item subgraph that represents user-item interactions in the review domain, where reviews are as edge attributes. Based on the two subgraphs, We design the attributed graph convolutional networks(AGCN) to learn user and item latent features in two domains.

As the heterogeneity of the user, item, and review features, we first design a transformation layer to transform different types of embeddings(i.e., user, item, and review) into a shared space. The further explanations are as follows:

Transformation Layer: In this layer, we project features of different types of nodes including users, items, and reviews into the same feature space. We utilize a transformation matrix M_t inspired by the work in (Wang et al. 2019b):

$$\mathrm{h}_v{}' = M_t \cdot \mathrm{h}_v, \mathrm{h}_v \in \{\mathrm{h}_u, \mathrm{h}_i, \mathrm{h}_r\} \tag{8}$$

Where h_v denotes the node embeddings in the graph. h_u and h_i indicate user and item one-hot embeddings as discussed in the input layer, h_r denotes review embeddings produced by CNN. After the transformation operations, we utilize these outputs to further learn user and item latent features through the proposed AGCN.

AGCN Layer: To learn user and item features in item domain and review domain respectively, we design the AGCN method in this paper, which integrates both review features and rating scores into the graph convolutional network for more accurate rating prediction. The algorithm(AGCN) is divided into two parts, which are responsible for integrating rating scores and reviews in the item domain and review domain respectively. Note that the embeddings of users and items are the same in two domains at the beginning, but they are independently trained on different attributed subgraphs. The initial representations of the embeddings of users and items in the AGCN layer are as follows:

$$\begin{aligned} \mathrm{h}_{u-ra}{}^{(0)} &= \mathrm{h}_{u-re}{}^{(0)} = \mathrm{h}_u{}' \\ \mathrm{h}_{i-ra}{}^{(0)} &= \mathrm{h}_{i-re}{}^{(0)} = \mathrm{h}_i{}' \end{aligned} \tag{9}$$

Where $\mathrm{h}_{u-ra}{}^{(0)}$ and $\mathrm{h}_{i-ra}{}^{(0)}$ are initialized user and item embeddings in the item domain, $\mathrm{h}_{u-re}{}^{(0)}$ and $\mathrm{h}_{i-re}{}^{(0)}$ are

initialized user and item embeddings in the review domain. These embeddings are then used in the AGCN Layer for further learning of the user and item latent features.

Item domain: In the item domain, the edges are the corresponding rating scores that the user towards for the item. Considering the model accuracy and computing costs of our model, we build our method based on the work(He et al. 2020). In this work, they proposed a light GCN method to deal with implicit feedback recommendations(He et al. 2020). As mentioned before, they found that the complex feature transformation and nonlinear activation of the basic GCNs are not essential in the task of the recommendation. Inspired by this work, we also reduce these complex parts in our model to learn nodes features more effectively. To integrate rating scores into the AGCN, we first normalize the rating score as follows:

$$r_{(u,i)} = R_{(u,i)}/max(R) \tag{10}$$

Where R is the rating scores sets. $R_{(u,i)}$ denotes the rating user u towards the item i. $r_{(u,i)}$ is the normalized rating that user u towards the item i. We believe that the rating score that the user towards an item can be seen as an important signal when aggregating different neighbors, because the rating scores show the user's direct and obvious preference for the item. Therefore we utilize the normalized rating scores as weights when aggregating the user and item node embeddings as follows:

$$\begin{aligned} \mathrm{h}_{u-ra}{}^{(k+1)} &= \sum_{i \in \mathcal{N}(u)} r_{(u,i)} \mathrm{h}_{i-ra}{}^{(k)} \\ \mathrm{h}_{i-ra}{}^{(k+1)} &= \sum_{u \in \mathcal{N}(i)} r_{(u,i)} \mathrm{h}_{u-ra}{}^{(k)} \end{aligned} \tag{11}$$

Where k denotes the k-th layer for GCN. We utilize ℓ_1 norm to avoid the scale of embeddings increasing with the increase of graph convolutional operations. By integrating rating scores into the aggregating process, we can model the explicit degree of interactions between users and items. Next, we will introduce how we integrate review features into the aggregate process to model the implicit degree of interactions between users and items.

Review domain: In the review domain, edge attributes are represented as unstructured review features rather than weight signals as expressed by rating scores. To address the review features, we utilize both node embeddings and edge embeddings for the aggregating process in our AGCN method. Specifically, we first utilize the review embedding as the edge embedding, and aggregate not only embeddings of neighbor nodes but also corresponding edge embeddings linked to the nodes. This is to say, for each user, we aggregate his/her neighbor item features and features of reviews that the user towards the item. Inspired by the work of (Hamilton et al. 2017), we randomly sample the neighbors for each node to alleviate large-scale convolutional costs. The aggregation functions for user u and item i are as follows:

$$\begin{aligned} \mathrm{h}_{\mathcal{N}(u)}{}^{(k+1)} &\leftarrow AGGREGATE(\{\mathrm{H}_{(i,r)}{}^{(k)}, \forall i \in \mathcal{N}(u), \forall r \in E(u,i)\}) \\ \mathrm{h}_{\mathcal{N}(i)}{}^{(k+1)} &\leftarrow AGGREGATE(\{\mathrm{H}_{(u,r)}{}^{(k)}, \forall u \in \mathcal{N}(i), \forall r \in E(u,i)\}) \end{aligned} \tag{12}$$

Where k denotes the $k-th$ layer for AGCN. For each user u and item i, $i \in \mathcal{N}(u)$ and $u \in \mathcal{N}(i)$ indicate items and users connected with them in the user-item bipartite graph respectively, $E(u, i)$ denotes the corresponding review. *AGGREGATE* indicates the aggregating function, we utilize mean function as the aggregating function in this work. $\mathrm{H}_{(u,r)}$ and $\mathrm{H}_{(i,r)}$ are defined as follows:

$$\begin{aligned} \mathrm{H}_{(i,r)}{}^{(k)} &= CONCAT(h_{i-re}{}^{(k)}, h_r{}') \\ \mathrm{H}_{(u,r)}{}^{(k)} &= CONCAT(h_{u-re}{}^{(k)}, h_r{}') \end{aligned} \tag{13}$$

Where h_r demonstrates the corresponding review embedding, $h_{i-re}{}^{(k)}$ and $h_{u-re}{}^{(k)}$ are the representations of the user and item in layer k, they are the concentrations of the node embedding and the edge embedding. *CONCAT* indicates the concentration operation. Note that the review embeddings are not updated during the training process. After aggregating the neighbors for user and item nodes, we follow a combination strategy as in (Hamilton et al. 2017) for the user and item nodes as:

$$\begin{aligned} \mathrm{h}_{u-re}{}^{(k+1)} &\leftarrow \sigma(W^k \cdot CONCAT(\mathrm{h}_{u-re}{}^{(k)}, \mathrm{h}_{\mathcal{N}(u)}{}^{(k+1)})) \\ \mathrm{h}_{i-re}{}^{(k+1)} &\leftarrow \sigma(W^k \cdot CONCAT(\mathrm{h}_{i-re}{}^{(k)}, \mathrm{h}_{\mathcal{N}(i)}{}^{(k+1)})) \end{aligned} \tag{14}$$

Where W^k denotes the weight matrix for user node and item node, *AGGREGATE* indicates the mean function. *CONCAT* indicates the concentration operation, σ is the activate function, we utilize ReLU in this work. We also utilize ℓ_1 norm to avoid the scale of embeddings increasing with the increase of graph convolutional operations. It is noteworthy that embeddings that include neighbor nodes and corresponding edge attributes are both aggregated in our model. Hence, our model can not only capture explicit user-item interactions but also semantic user-item interactions through this graph convolutional operation.

Our work differs a lot from the existing work about integrating reviews into rating prediction. Previous review-based

works only consider user-review, item-review, or user-item interactions. We learn user and item features through user-review-item and user-rating-item interactions, which can learn better features by leveraging multiple interactions between users and items. The experiments of our model also indicate that modeling these interactions can improve the accuracy of rating prediction.

Attention Layer: After learning user and item features in two domains based on our proposed AGCN, utilize a self-attention attention mechanism to fuse them. Self-attention has been used successfully in a variety of tasks to achieve more effective models(Vaswani et al. 2017). To achieve the attention scores, we first concentrate user and item features from two domains as follows:

$$\begin{aligned} H_u &= CONCAT(\mathrm{h}_{u-re}{}^{(K)}, \mathrm{h}_{u-ra}{}^{(K)}) \\ H_i &= CONCAT(\mathrm{h}_{i-re}{}^{(K)}, \mathrm{h}_{i-ra}{}^{(K)}) \end{aligned} \tag{15}$$

Where K denotes the last layer of the AGCR layer. We then apply the self-attention mechanism on H_u and H_i respectively. Following the work of (Lin et al. 2017), the coefficients $a_{u-ra} \in \mathbb{R}$ and $a_{u-re} \in \mathbb{R}$ for user node are calculated as:

$$\begin{aligned} e_{u-ra} &= w_{u-ra}{}' tanh(w_{u-ra} H_u) \\ e_{u-re} &= w_{u-re}{}' tanh(w_{u-re} H_u) \\ a_{u-ra} &= \frac{\exp(e_{u-ra})}{\exp(e_{u-ra}) + \exp(e_{u-re})} \\ a_{u-re} &= \frac{\exp(e_{u-re})}{\exp(e_{u-ra}) + \exp(e_{u-re})} \end{aligned} \tag{16}$$

Where $w_{u-ra} \in \mathbb{R}^{d \times d_a}$, $w_{u-ra}{}' \in \mathbb{R}^{d_a}$, $w_{u-re} \in \mathbb{R}^{d \times d_a}$, $w_{u-re}{}' \in \mathbb{R}^{d_a}$. d_a is the attention dimension. The coefficients $a_{i-ra} \in \mathbb{R}$ and $a_{i-re} \in \mathbb{R}$ for item node are the same as the user's:

$$\begin{aligned} e_{i-ra} &= w_{i-ra}{}' tanh(w_{i-ra} H_i) \\ e_{i-re} &= w_{i-re}{}' tanh(w_{i-re} H_i) \\ a_{i-ra} &= \frac{\exp(e_{i-ra})}{\exp(e_{i-ra}) + \exp(e_{i-re})} \\ a_{i-re} &= \frac{\exp(e_{i-re})}{\exp(e_{i-ra}) + \exp(e_{i-re})} \end{aligned} \tag{17}$$

Then the fused representations of user features and item features are as follows:

$$\begin{aligned} h_u &= a_{u-ra}\mathrm{h}_{u-ra} + a_{u-re}\mathrm{h}_{u-re} \\ h_i &= a_{i-ra}\mathrm{h}_{i-ra} + a_{i-re}\mathrm{h}_{i-re} \end{aligned} \tag{18}$$

Prediction layer: After fusing the user and item features, we derive two independent multilayer perceptrons(MLPs) to get the final user and item representations as follows.

$$\begin{aligned} q_u &= \sigma(W_{(u,L)}(...(W_{(u,2)}{}^T(W_{(u,1)}{}^T h_u + b_{(u,1)}) + b_{(u,2)})...) + b_{(u,L)}) \\ q_i &= \sigma(W_{(i,L)}(...(W_{(i,2)}{}^T(W_{(i,1)}{}^T h_i + b_{(i,1)}) + b_{(i,2)})...) + b_{(i,L)}) \end{aligned} \tag{19}$$

Where L is the number of layers of MLPs, W and b are weight matrix and bias term, we utilize ReLU function as the activation function σ. Then the predicted rating on item i by user u is equal to the inner product of the user and item feature vectors as:

$$\hat{y}_{ui} = q_u \odot q_i + \mu + \beta_u + \beta_i \tag{20}$$

Where β_u and β_i are biased terms regarding to user i and item j, and μ is the global biased term.

3.5 Model optimization

The objective function of our model is defined as follows:

$$\mathcal{L} = \sum_{(u,i) \in R} (\hat{y}_{ui} - y_{ui}) + \lambda_\Theta \|\Theta\|_2^2 \tag{21}$$

Where R is the user-item pairs, $\hat{y}_{ui}$ is the predicted rating of user u for item i, y_{ui} is the real rating. Θ denotes all the parameters. $\|\Theta\|_2^2$ denotes ℓ_2 norm for preventing overfitting model.

We use stochastic gradient descent (SGD) to learn the parameters by optimizing the objective function in Eq. 21. Additionally, we also adopt a dropout strategy(Srivastava et al. 2014) for the MLP layers to prevent the over-fitting.

The complete algorithm of our model is summarized in Algorithm 1. The inputs include the MA-bipartite graph we construct and some parameters as shown in the Algorithm 1, the outputs are user and item features that we learn. At the first line, we initialize all the model parameters of our model, including user/item embeddings, word embeddings, and other hyperparameters. In lines 2-4, we learn review features utilizing the TextCNN method that we have introduced in Section 3.3. In line 5, we generate the user and item pairs as the training samples. In line 6, we initialize the user, item, and review feature representations for the AGCN process, the initialized representations are calculated by Eq. 8. In lines 7-8, we initialize the user and item feature representations for the graph convolutional operations in the item domain and review domain. In lines 10-16, we learn the user feature by the AGCN method proposed in Section 3.4. Line 13 shows we learn the user feature representation in the item domain using Eq. 11, line 15-17 shows we learn the user feature representation in the review domain using Eq. 12, 13, and 14. In lines 19-25, we learn the item feature representation that has the same process as lines 11-18. Lines 28-31 show we utilize the attention mechanism to fusion the user and item

feature representations. Lines 32-33 show the optimization process.

4 Experiments

In this section, we conduct experiments on two real-world datasets for evaluating the performance of our model. We

Algorithm 1 The learning algorithm of our model

Input: network $G = (\mathcal{V}, \mathcal{E})$, number of layers L, user/item dimension d, review embedding e, learning rate λ

Output: user feature q_u, item feature q_i

1: Initialize all the model parameters Θ
2: **for** $re \in \mathcal{E}$ **do**
3: $\mathbf{h}_r = TextCNN(re)$ ▷ Encode review features by the TextCNN method
4: **end for**
5: Generate training samples $(\mathcal{U}, \mathcal{V})$
6: $\mathrm{h}_v{}' = M_t \cdot \mathrm{h}_v, \mathrm{h}_v \in \{\mathrm{h}_u, \mathrm{h}_i, \mathrm{h}_r\}$
7: $\mathrm{h}_{u-ra}{}^{(1)} = \mathrm{h}_{u-re}{}^{(1)} = \mathrm{h}_{u'}$ ▷ Initialize the user representation
8: $\mathrm{h}_{i-ra}{}^{(1)} = \mathrm{h}_{i-re}{}^{(1)} = \mathrm{h}_{i'}$ ▷ Initialize the item representation
9: **while** *not converged* **do**
10: **for** $k = 1, 2, 3...K$ **do**
11: **for** $u \in \mathcal{U}$ **do**
12: /* Learn the user feature representation in the item domain */
13: $\mathrm{h}_{u-ra}{}^{(k+1)} = \sum_{i \in \mathcal{N}(u)} r_{nor}\, \mathrm{h}_{i-ra}{}^{(k)}$
14: /* Learn the user feature representation in the review domain */
15: $\mathrm{H}_{(u,r)}{}^{(k)} = CONCAT(h_{u-re}{}^{(k)}, h_r{}')$
16: $\mathrm{h}_{\mathcal{N}(u)}{}^{(k+1)} \leftarrow AGGREGATE(\{\mathrm{H}_{(i,r)}{}^{(k)}, \forall i \in \mathcal{N}(u), \forall r \in E(u,i)\})$
17: $\mathrm{h}_{u-re}{}^{(k+1)} \leftarrow \sigma(W^k \cdot CONCAT(\mathrm{h}_{u-re}{}^{(k)}, \mathrm{h}_{\mathcal{N}(u)}{}^{(k+1)}))$
18: **end for**
19: **for** $i \in \mathcal{I}$ **do**
20: /* Learn the item feature representation in the item domain */
21: $\mathrm{h}_{i-ra}{}^{(k+1)} = \sum_{u \in \mathcal{N}(i)} r_{nor}\, \mathrm{h}_{u-ra}{}^{(k)}$
22: /* Learn the item feature representation in the review domain*/
23: $\mathrm{H}_{(i,r)}{}^{(k)} = CONCAT(h_{i-re}{}^{(k)}, h_r{}')$
24: $\mathrm{h}_{\mathcal{N}(i)}{}^{(k+1)} \leftarrow AGGREGATE(\{\mathrm{H}_{(u,r)}{}^{(k)}, \forall u \in \mathcal{N}(i), \forall r \in E(u,i)\})$
25: $\mathrm{h}_{i-re}{}^{(k+1)} \leftarrow \sigma(W^k \cdot CONCAT(\mathrm{h}_{i-re}{}^{(k)}, \mathrm{h}_{\mathcal{N}(i)}{}^{(k+1)}))$
26: **end for**
27: **end for**
28: $H_u = (\mathrm{h}_{u-re}{}^{(K)}, \mathrm{h}_{u-ra}{}^{(K)})$ ▷ Concentrate user feature representations
29: $H_i = (\mathrm{h}_{i-re}{}^{(K)}, \mathrm{h}_{i-ra}{}^{(K)})$ ▷ Concentrate item feature representations
30: Calculate the attention using Equation (16) and (17)
31: Calculate q_u and q_i using Equation (18) and (19)
32: Calculate the objective function $\mathcal{L}$ using Equation (21)
33: Update model parameters Θ by $\frac{\partial \mathcal{L}}{\Theta}$ with the learning rate α
34: **end while**

Table 2 Statistics of Datasets

Datasets	#Users	#Items	#Reviews	#Reviews per User	#Reviews per Item	Density
Automotive	2928	1835	20473	6.99	11.15	0.381%
Instant Video	5130	1685	37126	7.23	22.33	0.429%
Digital Music	5541	3586	64706	11.68	18.14	0.327%
Toys and Games	19412	11924	167597	8.63	14.05	0.073%
Kindle Store	68223	51934	982619	14.40	15.86	0.023%
Movies and TV	123960	50052	1697533	13.69	33.91	0.027%
Epinions	116256	41268	188473	1.62	4.56	0.004%

also investigate the contributions of different components of our model and the impacts of different parameter settings. We also present a detailed analysis of these experimental results.

4.1 Experimental setup

Dataset. We have used two publicly available datasets in our experiments that provide user reviews and rating scores, Amazon 5-core[1] and Epinions[2]. The two datasets are popular in E-commerce and social websites respectively. We utilize these two datasets to investigate the effectiveness of our model on different types of online websites.

Amazon dataset: We use the 'five-core' subsets from the publicly accessible Amazon product dataset released by (McAuley et al. 2015), where the 'five-core' means that each user and item in the subset has at least five reviews. This dataset contains user interactions (review, rating, votes etc.) on items as well as the item metadata (e.g., description, price, brand, image URL, etc.) from Amazon[3]. We utilize six datasets from this dataset including 'Automotive', 'Instant Video', 'Digital Music', 'Toys and Games', 'Kindle Store', 'Movies and TV'. These datasets have different sparseness and are usually used in review-based recommendations.

Epinions dataset: Epinions is a popular online consumer review website[4]. The dataset used in this work is collected and released by (Cai et al. 2017) that includes user interactions on items(ratings, reviews) as well as social (or trust) relationships between users. We only use interactions in this dataset ignoring social relations due to we aim at the review-based recommendation in this paper.

The statistics of these datasets are shown in Table 2. From the table, we can examine these datasets have different scales and sparseness. Our goal is to prove the effectiveness of our model by compared with other baselines for datasets with different sparseness. For all the review-based models, we remove the review written by the target user for the target item at test time. In the experiments, We randomly select 80% of each dataset as the training set, 10% as the validation set, and the remaining 10% as the testing set.

Baselines: We compare our model with eight state-of-the-art methods including (1) collaborative filtering based models, PMF; (2) review based models, CDL, ConvMF, DeepCoNN, D-Attn, NARRE, and CARL; (3) graph neural network-based models, GCMC.

(1) Probabilistic Matrix Factorization **(PMF)**(Mnih and Salakhutdinov 2007): PMF is a standard matrix factorization model which models latent factors of users and items by Gaussian distributions, this method only utilize the user's history record for rating prediction.
(2) Collaborative Deep Learning **(CDL)**(Wang et al. 2015): CDL is the first hierarchical bayesian model to build the connection between the deep learning technique (SDAE) and the MF model, which realizes the integration of ratings and reviews for rating prediction.
(3) Convolutional Matrix Factorization **(ConvMF)**(Kim et al. 2016): ConvMF extracts latent item features using CNN over the item documents and integrates CNN into PMF for rating prediction.
(4) Deep Cooperative Neural Networks **(DeepCoNN)**(Zheng et al. 2017): DeepCoNN uses two parallel CNN networks to extract latent feature vectors from both user reviews and item reviews, then they use a Factorization Machine (FM)(Rendle 2010) to concatenate user and item latent factors for rating prediction.
(5) Dual Attention-based Model **(DATTN)**(Seo et al. 2017): D-Attn leverages global and local attention to enable an interpretable embedding of users and items. Finally, the rating can be estimated by the dot product of the user and item embeddings.
(6) Neural attentional regression model **(NARRE)**(Chen et al. 2018): NARRE exploits two parallel CNNs to learn the latent features of reviews, and derives atten-

[1] https://jmcauley.ucsd.edu/data/amazon/ 5-core.
[2] https://cseweb.ucsd.edu/ jmcauley/datasets.html.
[3] https://www.amazon.com/.
[4] https://shopping.com/.

Table 3 Methods Comparison

Dataset	Technique			Information		Architecture	
Models	MF	CNN	GNN	Rating scores	Reviews	Interactive	Dependent
PMF	√				√		√
CDL	√				√		√
ConvMF	√	√			√		√
DeepCoNN		√			√		√
DATTN		√			√		√
NARRE		√			√		√
CARL		√			√		√
GCMC			√			√	
AGCR		√	√	√	√	√	

Table 4 Parameter values

Dataset	Amazon			Epinions		
Models	α	λ	*dropout*	α	λ	*dropout*
PMF	0.01	$\lambda_u = 1.0, \lambda_i = 10.0$	\	0.01	$\lambda_u = 0.1, \lambda_i = 0.1$	\
CDL	0.001	$\lambda_u = 0.1, \lambda_i = 0.1$	0.1	0.001	$\lambda_u = 0.1, \lambda_i = 0.1$	0.1
ConvMF	0.001	$\lambda_u = 1.0, \lambda_i = 10.0$	0.2	0.001	$\lambda_u = 1.0, \lambda_i = 10.0$	0.2
DeepCoNN	0.001	0.001	0.5	0.005	0.001	0.5
DATTN	0.005	0.001	0.5	0.005	0.001	0.5
NARRE	0.01	0.001	0.5	0.0001	0.001	0.5
CARL	0.005	0.001	0.5	0.0001	0.001	0.5
GCMC	0.001	\	0.3	0.01	\	0.3
AGCR	0.001	0.01	0.2	0.001	0.001	0.2

tion mechanism to explore the usefulness of reviews, which provides review-level interpretability while predicting rating scores.

(7) Context-aware user-item representation learning model **(CARL)**(Wu et al. 2019a): CARL exploits CNNs to learn the relevant features of user-item pairs and utilizes a dynamic linear fusion mechanism for the final rating prediction.

(8) Graph Convolutional Matrix Completion **(GCMC)**(Berg et al. 2017): GCMC constructs user and item embeddings through message passing on the bipartite user-item interaction graph, and new ratings are predicted in the form of labeled edges.

We summarized the differences between our model and these baselines which are shown in Table 3. We summarized the differences between our model and these baselines which are shown in Table 3. First, we compare them for different techniques they adopt including MF, CNN, and GNN techniques. Then, we compare these methods for different types of information they use including reviews and rating scores. Finally, we also compare them according to the architecture (interactive or dependent). Interactive architecture means the model considers the user and item interaction when learning user and item features, while dependently architecture means the model learns user and item features using two dependently modules. From the table, we can see all the baselines that utilize review information are not interactive, which is also one of the main differences between our model and theirs.

Evaluation Metric: We adopt two well-known metrics for performance evaluation: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

Table 5 RMSE and MAE Comparisons with baselines. Improvements of our model over the best baseline are shown in the last row

Method	Dataset / Metric	Automotive	Instant Video	Digital Music	Toys and Games	Kindle Store	Movies and TV	Epinions	Average
PMF	RMSE	1.3484	1.1320	0.9713	1.0926	0.9506	1.1007	4.0946	1.5271
	MAE	1.0945	0.8480	0.7198	0.8260	0.6733	0.8129	3.7766	1.2502
CDL	RMSE	0.9846	1.1211	1.1305	1.7689	0.9530	1.2514	1.8618	1.1552
	MAE	0.7142	0.8932	0.8931	1.3065	0.6912	0.9594	1.4908	0.9926
ConvMF	RMSE	0.9641	1.0135	1.0474	0.9521	0.9276	1.2249	1.7520	1.1259
	MAE	0.7200	0.7554	0.8001	0.7042	0.6786	0.9327	1.3231	0.8448
GC-MC	RMSE	1.0019	1.0916	0.9859	0.9629	0.8939	1.1170	1.4834	1.0766
	MAE	0.6708	0.7816	0.7134	0.6992	0.5538	0.7505	1.0275	0.7424
DATTN	RMSE	0.9140	0.9714	0.9230	0.9118	0.8386	1.0567	1.2891	0.9863
	MAE	0.6260	0.7042	0.6673	0.6410	0.5867	0.7859	0.9950	0.7151
DeepCoNN	RMSE	0.9102	0.9717	0.9221	0.9189	0.8267	1.0503	1.2731	0.9819
	MAE	0.6316	0.7257	0.6819	0.6626	0.5990	0.7823	0.9900	0.7247
CARL	RMSE	0.9135	0.9729	0.9364	0.9015	0.8140	1.0231	1.2830	0.9777
	MAE	0.6137	0.7415	0.7176	0.6404	0.5484	0.7751	0.9902	0.7181
NARRE	RMSE	0.9193	0.9796	0.9167	0.9000	0.8137	1.0317	1.2475	0.9726
	MAE	0.6507	0.7243	0.6627	0.6361	0.5795	0.7635	0.9522	0.7098
AGCR	RMSE	0.8817	0.9541	0.8978	0.8842	0.7958	1.0088	1.2070	0.9466
	MAE	0.5858	0.6790	0.6432	0.6158	0.5354	0.7191	0.9098	0.6697
Imp	RMSE	3.13%	1.78%	2.06%	1.76%	2.20%	1.40%	3.25%	2.67%
	MAE	4.55%	3.58%	2.94%	3.19%	2.37%	5.81%	4.45%	5.64%

$$RMSE = \sqrt{\frac{1}{\mid O_t \mid} \sum_{(u,i)\in O_t} (y_{ui} - \hat{y}_{ui})^2}$$
$$MAE = \frac{1}{\mid O_t \mid} \sum_{(u,i)\in O_t} \mid y_{ui} - \hat{y}_{ui} \mid \tag{22}$$

Where O_t is the set of the user-item pairs in the testing set.

Parameter Settings: We use grid search to tune the hyper-parameters for our model and all the baselines, the hyper-parameters for all the baselines are also based on the parameter settings in their work. The latent dimension size is optimized from [8, 16, 32, 64, 128], the word embedding dimension size for all models is set as 300. The learning rate is tuned from [0.0001, 0.0005, 0.001, 0.005, 0.01]. The regularization parameter is tuned from [0.05, 0.01, 0.005, 0.001]. The best performing values of common parameters (learning rates α,regularization parameter λ, dropout) of each model are shown in Table 4. For MF based models, there are regularization parameters for the user and item respectively. \ indicates the parameter is not provided in the paper. The batch size is set as 100. For all the CNN-based models, the number of convolution filters is set as 100 and the window size is set as 3. The length of the review is set as 300 on Amazon datasets and 50 on the Epinions dataset respectively for all the review-based models.

4.2 Experimental results

4.2.1 Performance evaluation

We first conduct Experiments for our model and all the baselines on all the datasets and results are presented in Table 5, where latent dimension size is set as 32 for all the baselines, the number of layers for our model is set as 1, and the number of neighbors for our model is set as 10. Other common parameters are shown in Table 4 and introduced in the Parameter Settings. From the results in Table 5, we can see that our model performs best on all the datasets for two metrics. This demonstrates that our model can learn better user and item latent features and actually improve the accuracy of rating prediction. On all the categories of Amazon dataset, the MAE of our model improves much bette than all the baselines, and the RMSE also improves from 1.40% and 3.13%. On Epinions dataset, the RMSE and MAE of our model both improve a lot than all the baselines for 3.25% and 4.45%.

We also calculate the average RMSE and MAE values of all the benchmark methods on the two datasets, the results are shown in the last column in Table 5. We rank all the baselines according to the RMSE values as shown in the first column in this table. Generally, a lower RMSE value usually along with a lower MAE value. If the comparison of these two metrics is inconsistent, we prefer to use the RMSE

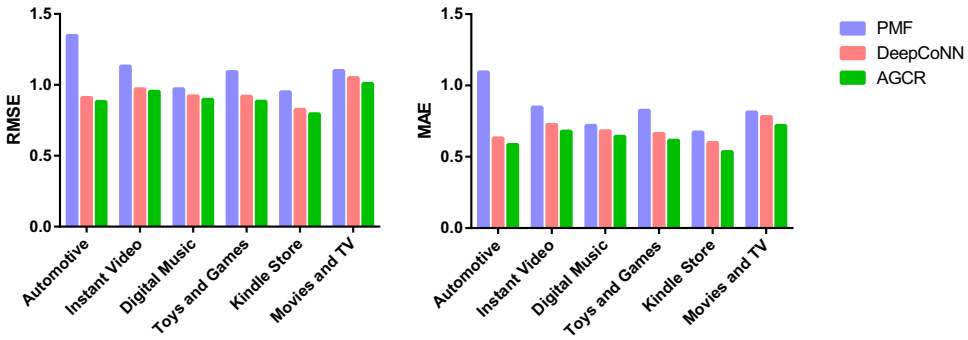


Fig. 4 Performance comparison on Amazon dataset

value to evaluate the performance of the method, due to most methods adopt the RMSE value to evaluate the performance. There are more observations for the experimental results as follows:

(1) Our model has significant improvements than PMF, CDL, and ConvMF, which indicates that our model can improve the CF-based methods a lot by utilizing advanced neural network methods.

(2) Our model also performs better than DeepCoNN, DATTN, CARL, and NARRE, demonstrating that our model is superior to the state-of-art CNN-based baselines. As Figure 5 and 6 shows, these CNN based methods achieves similar performance with different learning rates and dropouts, while our model has significant improvements than all the baselines. We find that these baselines simply encode user/item reviews as their representations rely on CNN or attention mechanism, lacking complex nonlinear interactions modeling between them, which is probably the main reason that limits the effectiveness of these CNN-based methods. Different from them, we utilize CNN as the first step to achieve review features, then integrate the features into the graph convolutional network(GCN) to further learn user and item features, which can further enhance the learning of user and item features than these baselines.

(3) Our model performs better than GCMC. This method builds the unattributed graph to learn user and item latent features without any side information integrated into the model. Therefore, we can find that reviews as side information can help improve the accuracy of rating prediction, and our model can learn better user and item latent features compared with GCMC.

(4) Our model has significant improvements than all the baselines on the Epinions dataset. Epinions dataset is much sparser than the Amazon dataset, there are few reviews for each user and item. From the results on Epinions dataset, we can see that all the baselines get poor performance when the review is much sparser. On the contrary, our model has better performance even in the sparser scenario. This demonstrates our model can better alleviate the sparsity problem because it can capture more information by modeling multiple interactions between users and items.

(5) PMF(Mnih and Salakhutdinov 2007) and DeepCoNN(Zheng et al. 2017) are two typical methods in the recommendation domain, many researchers utilize the two models as their basic baselines to make the comparison. Therefore, we further compare with the two methods on the Amazon dataset to show the effectiveness of our model as shown in Figure 4. From the figure, we can see both DeepCoNN and our model AGCR perform much better than the PMF method. Moreover, our model has significant improvements than the PMF method and is also superior to the DeepCoNN method on all the categories of the Amazon dataset. The results prove that deep learning-based methods can achieve better performances than traditional MF-based methods, also demonstrate our model can have better improvements than the two basic baselines.

In our experiment, we adopt baselines from different domains, the results of these baselines provide some important information. From the results of all the baselines as Table 5 shows, we have noticed that :

(1) PMF and CDL perform worse than other baselines on all the datasets. This is mainly due to the two methods only consider user's history records or simple word information. Especially, PMF performs the worst on the Epinions dataset because the Epinions Dataset is sparser than the Amazon dataset, thus we can conclude that only relying on traditional collaborative filtering can't predict rating accurately due to the sparseness.

(2) ConvMF, DeepCoNN, DATTN, NARRE, and CARL have better performance than PMF and CDL on all datasets, which demonstrates reviews is beneficial for alleviating sparseness, and extracting review features through CNN and attention mechanism is more effective for obtaining review features than simple word

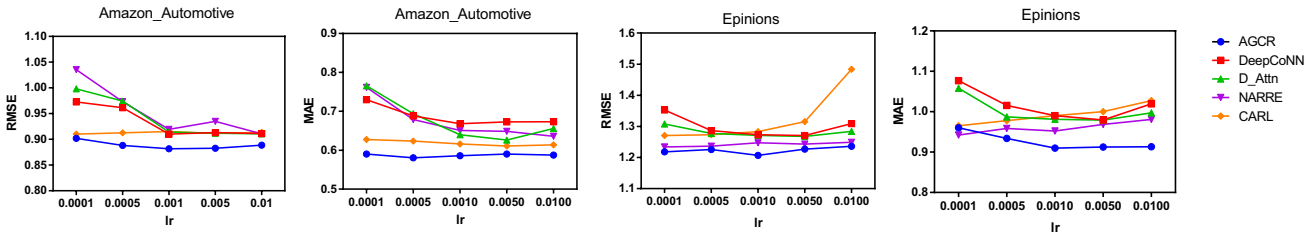


Fig. 5 Performance comparison varying different learning rates

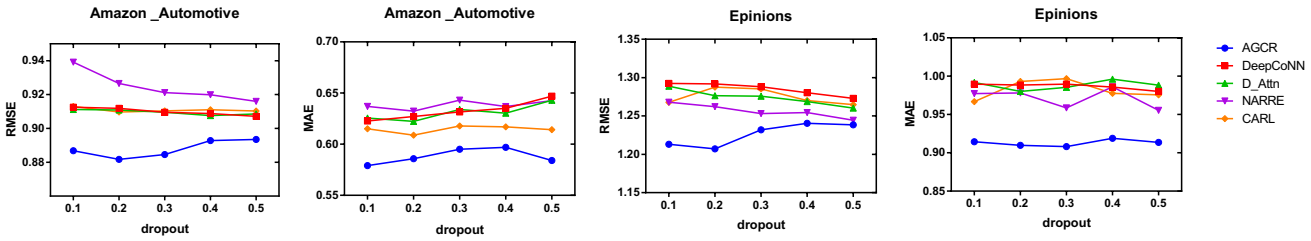
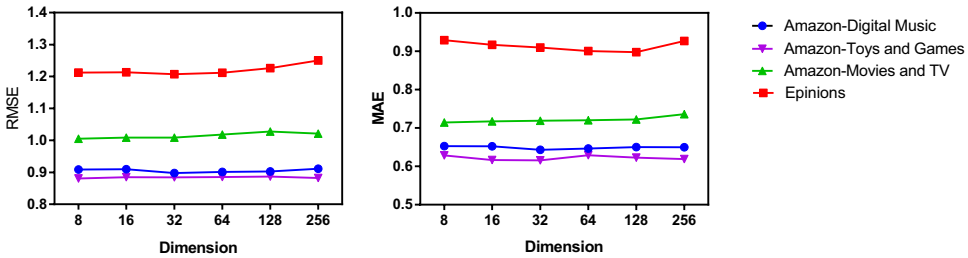


Fig. 6 Performance comparison varying different dropouts

Fig. 7 Performance comparison varying different the number of dimensions



processing techniques(CDL). In addition, we also find NARRE and DATTN perform better than DeepCoNN and ConvMF on most datasets. NARRE and DATTN use specific reviews rather than concentrating on all the reviews, indicating fine-grained review features modeling can also improve the accuracy of rating prediction.

(3) GCMC performs better than PMF on most datasets. The two models only consider the user's historical record for rating prediction, GCMC utilizes GCN to learn user and item features by modeling their interactions instead of using MF, thus it achieves better performance. This also demonstrates that GCN is more effective than traditional MF when learning the user and item latent features.

To investigate the performance of CNN based methods, we also conduct experiments on 'Automotive' category and 'Epinions' dataset for DeepCoNN, DATTN, NARRE, CARL and our model, varying different learning rates and dropout rates, fixing the latent dimension size as 32, the number of layers of our model as 1, the number of neighbors of our model as 10. The results are presented in Figure 5 and Figure 6.From the figures we can see: DATTN and CARL have better performance on 'Automotive' category. On 'Epinions' dataset, NARRE and CARL have better performances than other baselines. We also notice that our model achieves the best performance with different learning rates and dropouts among all the methods. Specifically, CARL fuses the dot product of user and item embeddings from the item and review domain by a hyper-parameter. The results of the two models demonstrate that both item domain and review domain are essential for better rating prediction. Our model still performs much better than CARL, which demonstrates our AGCN method is effective, and the fuse strategy of attention mechanism in

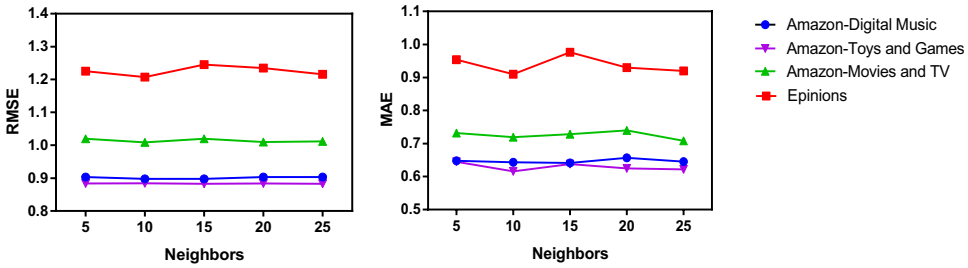


Fig. 8 Performance comparison varying different the number of neighbors

Table 6 RMSE and MAE Comparisons with different layer numbers K

Method	Datasets / Metric	Automotive	Instant Video	Digital Music	Toys and Games	Kindle Store	Movies and TV	Epinions
Layer-1	RMSE	0.8817	0.9541	0.8978	0.8933	0.7969	1.0088	1.207
	MAE	0.5858	0.6790	0.6432	0.6432	0.5354	0.7191	0.9098
Layer-2	RMSE	0.8826	0.9434	0.9072	0.8863	0.7955	1.0110	1.2284
	MAE	0.5898	0.6821	0.6477	0.6223	0.5365	0.7205	0.9042
Layer-3	RMSE	0.8805	0.9418	0.9097	0.8861	0.7948	1.0067	1.2254
	MAE	0.5901	0.6837	0.6576	0.6226	0.5362	0.7206	0.9096

our model is also useful by combining the review domain and the item domain non-linearly and dynamically.

4.2.2 Parameter analysis and ablation study

We first conduct experiments to analyze the influences of different parameters on our model. To achieve this goal, we investigate the impact of the latent factor the number of dimensions d, the number of neighbors $\mathcal{N}$ and the number of layers K. In the experiment, we fix other parameters and study the different performances of our model under the current parameter.

The impact of the number of latent factor dimensions d: To investigate the impact of he number of latent factor dimensions d on our model, we conduct experiments with different d values from [8, 16, 32, 64, 128, 256] on 'Digital Music', 'Toys and games', 'Movies and TV' categories and 'Epinions' dataset. These datasets have clear differences in the sparseness. The results are shown in Figure 7. From the results, we can see:

(1) In general, the performance of our model is not significantly affected by the dimension d, especially on Amazon dataset. On Epinions dataset, the increase of the d has more impact on the MAE of the model.
(2) On Amazon dataset, we can see the curves of RMSE and MAE are smooth on 'Digital Music' category, and our model achieves the best performance when d is set as 32. On 'Toys and games' category, the curve of RMSE is stable with different d values, and the curve of MAE shows that our model performs the best when $d = 32$. On 'Movies and TV' category, both RMSE and MAE have little differences with d from 8 to 32, but they increase faster when $d > 32$.
(3) On Epinions dataset, the RMSE decreases when d values varies from 8 to 32, and the two metrics begin to increase after $d = 32$. The MAE also decreases when the d values varies from 8 to 32. When $d > 32$, the decreasing trend becomes smoother.

The results show that our model is not sensitive to the the number of latent factor dimensions, which is probably due to the use of graph convolution operations reduces the influence of the the number of latent factor dimensions on our model. The embeddings can obtain more feature information with the increase of the the number of dimensions, but a high dimension size will lead to an over-fitting problem, this is why our model doesn't perform better when d value is increasing on some datasets. In addition, higher d values will also bring higher complexity. The results show our model achieves the best performance on most datasets when $d = 32$. Therefore we set the d value as 32 in this paper to achieve the best performance and avoid the high computational complexity and over-fitting problem.

The impact of the the number of neighbors $\mathcal{N}$. In this work, we randomly select a certain number of neighbors when aggregating neighbor nodes for the target node. To investigate the impact of the the number of neighbors $\mathcal{N}$ for the user and the item when sampling their neighbors for convolutional operations, we conduct experiments on 'Digital

Table 7 RMSE and MAE Comparisons with AGCR-rating, AGCR-review and our AGCR

Method	Dataset Metric	Automotive	Instant Video	Digital Music	Toys and Games	Kindle Store	Movies and TV	Epinions
AGCR-rating	RMSE	0.8951	0.9584	0.9088	0.8852	0.7871	1.0086	1.1994
	MAE	0.6001	0.6991	0.6602	0.6317	0.5604	0.7507	0.9686
AGCR-review	RMSE	0.8794	0.9665	0.9132	0.8904	0.7956	1.0224	1.2383
	MAE	0.5573	0.6712	0.6461	0.6132	0.5450	0.7354	0.9380
AGCR	RMSE	0.8817	0.9541	0.8978	0.8933	0.7969	1.0088	1.2070
	MAE	0.5858	0.6790	0.6432	0.6432	0.5354	0.7191	0.9098

Table 8 RMSE and MAE comparisons with AGCR-attn and AGCR-FM

Method	Dataset Metric	Automotive	Instant Video	Digital Music	Toys and Games	Kindle Store	Movies and TV	Epinions
AGCR-attn	RMSE	0.8817	0.9541	0.8978	0.8933	0.7969	1.0088	1.2070
	MAE	0.5858	0.6790	0.6432	0.6432	0.5354	0.7191	0.9098
AGCR-FM	RMSE	0.8826	0.9552	0.9115	0.8869	0.7988	1.0172	1.2231
	MAE	0.5708	0.6949	0.6639	0.6262	0.5577	0.7456	0.9219

Music', 'Toys and games', 'Movies and TV' categories from Amazon dataset and 'Epinions' dataset vary the $\mathcal{N}$ value from [5, 10, 15, 20, 25, 30]. Results are presented in Figure 8. From the figure we have the following observations:

(1) On Amazon dataset, the curves of RMSE on all the categories are stable. As for the MAE, On 'Digital Music' category, our model has worse MAE when the $\mathcal{N}$ is set as 20. On 'Toys and games' category, our model achieves better MAEs when the $\mathcal{N}$ is set as 10, 20, and 25. On 'Movies and TV' category, our model has better MAE when $\mathcal{N}$ is set as 10 and 25. Therefore, our model has better RMSE and MAE when the number of the $\mathcal{N}$ is set as 10, 25 in most cases on Amazon dataset.
(2) On 'Epinions' dataset, the curves decrease when $\mathcal{N}$ value ranges from 5 to 10, then increase when $\mathcal{N} > 10$, and begin to decrease when $\mathcal{N} > 15$.

Based on the above analysis, we can see that our model has better performance when the $\mathcal{N}$ value is set as 10 and 25. A lower value of $\mathcal{N}$ can't aggregate enough neighbor node features, thus it will decrease the accuracy of the model. However, a higher value will introduce more padding words and cause higher complexity of the model. Considering both model performance and model complexity, we set $\mathcal{N}$ value as 10 in our paper.

The impact of the number of layers K. To investigate the impact of the number of layers in AGCN, we conduct experiments on all the datasets with different K values from [1, 2, 3] , the results are shown in Table 6. From the results we can see:

(1) On Amazon dataset, our model achieves the best RMSEs when the number of layers is set as 3 on all the categories. On these two categories, our model has the best RMSEs when the number of layers is set as 1. Our model achieves the best MAEs when the number of layers is 1 on most datasets, and the number of layers is 2 on 'Toys and Games' category.
(2) On 'Epinions' dataset, RMSE is increasing with the increase of the number of layers, and MAE is stable with different layer numbers. This is probably due to the Epinions dataset is sparser and high layer numbers are not useful for alleviating the sparsity problem.

Despite a higher K value lead to better RMSEs and MAEs on most datasets, the improvements are not significant. Furthermore, it will bring the over-smoothing problem and be time-consuming due to more neighbors aggregated and complex convolution operations. In addition, a high K value will have an advantage when compared with the baselines. Therefore, we set the K value as 1 for basic experiments to avoid complexities.

Impact of reviews and rating scores. In this paper, we consider two types of interactions between users and items. To investigate the impact of the rating scores and reviews on our model, we conduct extensive experiments on all the datasets. AGCR-rating/review means we only use rating/review information as the attribute of the edge.

Table 7 shows the results of the experiments on all datasets. From the results, we can see AGCR-rating achieves the best RMSEs on most datasets, but it has the worst MAEs on all the datasets. This demonstrates that integrating history rating information into our model can lead to better RMSEs. AGCR-review achieves the best MAEs on most Amazon categories including 'Automotive', 'Instant Video', 'Toys and Games'. However, it doesn't achieve better RMSEs on all the categories of Amazon dataset, it only achieves the best RMSEs on the 'Automotive' category. This is probably due to the fact that utilization of review information can achieve better MAEs. On Epinions dataset, our model AGCR has the best RMSE and MAE which integrating both reviews and rating scores into the graph convolutional network.

In conclusion, our model AGCR achieves better RMSEs and MAEs by combing both review and rating information, and the RMSEs and MAEs are close to the best results between AGCR-rating and AGCR-review. This experimental results shows both rating and review information are essential for improving the effectiveness of the rating prediction, and utilizing an attention mechanism can fully incorporate reviews and rating scores, thus to achieve better user and item latent features.

Impact of the attention mechanism: In this work, we utilize an attention mechanism to fuse user and item latent features from two domains, then utilize a dot product to get the predicted rating score, instead of using the factorization machine(FM) as the previous models. To investigate the impact of the attention mechanism on our model, we also conduct extensive experiments on all the datasets to compare the different performances of our model using the attention mechanism and the FM. The results are shown in Table8. AGCR-attn in this table means our model, AGC-FM means we utilize FM to fuse user and item latent features from domain and achieve the predicted rating scores. From the results, we can see that AGC-FM achieves worse MAEs on both Amazon and Epinions datasets. In contrast, our model performs best on most categories of Amazon dataset and Epinions dataset, which demonstrates the attention mechanism can lead to better model performance than FM. Moreover, AGC-FM still performs better than all the baselines on the two datasets, which confirms the truth that the AGCN method we proposed in our model is effective for incorporating reviews in the task of the rating prediction.

5 Conclusion

In this paper, we utilize both rating scores and review information for the rating prediction. To achieve this goal, we first design a multi-attributed graph to model two types of interactions between users and items through reviews and rating scores respectively. To learn user and item features based on this graph, we then propose an attributed GCN method(AGCN) in the review domain and item domain to incorporate attributed edge weights(rating scores) and edge embeddings(reviews). In the end, we utilize an attention mechanism to fuse user and item features from two domains. Our model is the first work to represent side information as edge attributes in the rating prediction task, and we consider heterogeneous attributes of edges. Moreover, we can capture fine-grained user-item interactions thus to learn user and item features better. In the future, we will focus on integrating more heterogeneous information into the model to further improve the performance of recommendation, we will also design more effective graph convolutional networks methods to make them adaptive to large-scale recommendations.

Acknowledgements This work is sponsored by the National Natural Science Foundation of China (61976103, 61872161), the Scientific and Technological Development Program of Jilin Province (20190302029GX, 20180101330JC, 20180101328JC) and Tianjin Synthetic Biotechnology Innovation Capability Improvement Program (no. TSBICIP-CXRC-018).

References

Ahmed BH, Ghabayen AS (2020) Review rating prediction framework using deep learning. J Am Intel Hum Comput pp 1–10

Bao Y, Fang H, Zhang J (2014) Topicmf: simultaneously exploiting ratings and reviews for recommendation. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence 14:2–8

Berg R, Kipf TN, Welling M (2017) Graph convolutional matrix completion. Preprint at https://arxiv.org/abs/1706.02263

Bruna J, Zaremba W, Szlam A, et al (2013) Spectral networks and locally connected networks on graphs. Preprint at https://arxiv.org/abs/1312.6203

Cai C, He R, McAuley J (2017) Spmc: socially-aware personalized markov chains for sparse sequential recommendation. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 1476–1482

Catherine R, Cohen W (2017) Transnets: learning to transform for recommendation. In: proceedings of the eleventh ACM conference on recommender systems, 288–296

Chaudhari S, Polatkan G, Ramanath R, et al (2019) An attentive survey of attention models. Preprint at https://arxiv.org/abs/1904.02874

Chen C, Zhang M, Liu Y, et al (2018) Neural attentional rating regression with review-level explanations. In: proceedings of the 2018 world wide web Conference, 1583–1592

Cheng Z, Chang X, Zhu L et al (2019) Mmalfm: Explainable recommendation by leveraging reviews and images. ACM Trans Inform Syst (TOIS) 37(2):1–28

Collobert R, Weston J, Bottou L et al (2011) Natural language processing (almost) from scratch. J Machine Learn Res 12(2011):2493–2537

Cui Y, Chen Z, Wei S, et al (2016) Attention-over-attention neural networks for reading comprehension. Preprint at https://arxiv.org/abs/1607.04423

Gojali S, Khodra ML (2016) Aspect based sentiment analysis for review rating prediction. In:2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), 1-6

Guan X, Cheng Z, He X et al (2019) Attentive aspect modeling for review-aware recommendation. ACM Trans Inform Syst (TOIS) 37(3):1–27

Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: proceedings of the 31st International Conference on Neural Information Processing Systems, 1024–1034

He X, Deng K, Wang X, et al (2020) Lightgcn: simplifying and powering graph convolution network for recommendation. Preprint at https://arxiv.org/abs/2002.02126

Jin Z, Li Q, Zeng DD, et al (2016) Jointly modeling review content and aspect ratings for review rating prediction. In: proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval, 893–896

Kim D, Park C, Oh J, et al (2016) Convolutional matrix factorization for document context-aware recommendation. In: proceedings of the 10th ACM conference on recommender systems, 233–240

Kim Y (2014) Convolutional neural networks for sentence classification. Preprint at https://arxiv.org/abs/1408.5882

Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In:Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 426–434

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

Lin Z, Feng M, Santos CNd, et al (2017) A structured self-attentive sentence embedding. Preprint at https://arxiv.org/abs/1703.03130

Ling G, Lyu MR, King I (2014) Ratings meet reviews, a combined approach to recommend. In: proceedings of the 8th ACM conference on recommender systems, 105–112

Liu D, Li J, Du B, et al (2019) Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In:Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 344–352

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, 165–172

McAuley J, Targett C, Shi Q, et al (2015) Image-based recommendations on styles and substitutes. In:Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 43–52

Mikolov T, Chen K, Corrado G, et al (2013) Efficient estimation of word representations in vector space. Preprint at https://arxiv.org/abs/1301.3781

Mnih A, Salakhutdinov RR (2007) Probabilistic matrix factorization. Adv Neural Inform Process syst 20:1257–1264

Pero Š, Horváth T (2013) Opinion-driven matrix factorization for rating prediction. In: International conference on user modeling, adaptation, and personalization, 1–13

Rendle S (2010) Factorization machines. In:2010 IEEE International conference on data mining, 995–1000

Sachdeva N, McAuley J (2020) How useful are reviews for recommendation a critical review and potential improvements. In: proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 1845–1848

Sarwar B, Karypis G, Konstan J, et al (2001) Item-based collaborative filtering recommendation algorithms. In:Proceedings of the 10th international conference on World Wide Web, 285–295

Seo S, Huang J, Yang H, et al (2017) Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In:Proceedings of the Eleventh ACM Conference on Recommender Systems, 297–305

Song W, Xiao Z, Wang Y, et al (2019) Session-based social recommendation via dynamic graph attention networks. In: proceedings of the Twelfth ACM International conference on web search and data mining, 555–563

Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Machine Learn Res 15(1):1929–1958

Tan Y, Zhang M, Liu Y, et al (2016) Rating-boosted latent topics: Understanding users and items with ratings and reviews. In:Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 2640–2646

Tay Y, Luu AT, Hui SC (2018) Multi-pointer co-attention networks for recommendation. In:Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, 2309–2318

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In:Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000-6010

Veličković P, Cucurull G, Casanova A, et al (2017) Graph attention networks. Preprint at https://arxiv.org/abs/1710.10903

Vinod N, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In:Proceedings of the 27th International Conference on Machine Learning (ICML-10), 807-814

Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In:Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 448–456

Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In:Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 1235–1244

Wang X, Yu L, Ren K, et al (2017) Dynamic attention deep model for article recommendation by learning human editors' demonstration. In:Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining, 2051–2059

Wang X, He X, Wang M, et al (2019a) Neural graph collaborative filtering. In:Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 165–174

Wang X, Ji H, Shi C, et al (2019b) Heterogeneous graph attention network. In:The 2019 World Wide Web Conference, 2022–2032

Wu L, Quan C, Li C et al (2019) A context-aware user-item representation learning for item recommendation. ACM Trans Inform Syst (TOIS) 37(2):1–29

Wu S, Tang Y, Zhu Y et al (2019) Session-based recommendation with graph neural networks. Proc Conf Artif Intel 33:346–353

Wu Y, Lian D, Jin S, et al (2019c) Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference. In:Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence Main track, 3898-3904

Xing S, Wang Q, Zhao X et al (2019) A hierarchical attention model for rating prediction by leveraging user and product reviews. Neurocomputing 332:417–427

Xu K, Ba J, Kiros R, et al (2015) Show, attend and tell: Neural image caption generation with visual attention. In:International conference on machine learning, 2048–2057

Ying R, He R, Chen K, et al (2018) Graph convolutional neural networks for web-scale recommender systems. In:Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 974–983

Zhang S, Tay Y, Yao L, et al (2018) Next item recommendation with self-attention. Preprint at https://arxivorg/abs/180806414

Zhang W, Wang J (2016) Integrating topic and latent factors for scalable personalized review-based rating prediction. IEEE Trans Knowl Data Eng 28(11):3013–3027

Zheng L, Noroozi V, Yu PS (2017) 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, 425–434

Zhou C, Bai J, Song J, et al (2018) Atrank: An attention-based user behavior modeling framework for recommendation. In:Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 4564–4571

Zhu Q, Zhou X, Song Z et al (2019) Dan: deep attention neural network for news recommendation. Proc AAAI Conf Artif Intel 33:5973–5980

Authors and Affiliations

Yijia Zhang[1,2] · Wanli Zuo[1,2] · Zhenkun Shi[3] · Binod Kumar Adhikari[4]

Yijia Zhang
yijia18@mails.jlu.edu.cn

Binod Kumar Adhikari
b_k_adhikari@hotmail.com

[1] Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China

[2] College of Computer Science and Technology, Jilin University, Changchun 130012, China

[3] Tianjin Institute of Industrial Biotechnology, Chinese Academy of Sciences, Changchun 130012, China

[4] Amrit Campus, Tribhuvan University, Kathmandu 44600, Nepal