Interest-Aware Message Passing Recommender Model based on Graph Convolutional Networks

merrick connelly (merrickconnelly91.laucs@hotmail.com)
Laurentian University
Lysandra Sinclair
Laurentian University
Solenne Dubois
Laurentian University
Galen Rawlins
Laurentian University

Research Article

Keywords: Recommender systems, Graph Convolutional Networks, Attention Mechanisms, Message Passing, User-item Representation, Preference

Posted Date: April 4th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2771529/v1

Interest-Aware Message Passing Recommender Model based on Graph Convolutional Networks

Merrick Connelly, Lysandra Sinclair, Solenne Dubois, Galen Rawlins,

Abstract—With the rapid development of deep learning, recommender systems have become an important tool to solve the problem of information overload. Collaborative filtering-based models have made effective progress in learning user and item representations. Graph Convolutional Networks (GCN) have been introduced to extract connection characteristics between users and items, but suffer from over-smoothing issues. This paper proposes an interest-aware message passing recommendation model that uses attention mechanisms and two methods of partitioning graphs to mitigate the over-smoothing problem. The model uses two methods to partition the graph into subgraphs, namely, partitioning the users and items into subgraphs separately. High-order graph convolution operations are performed in the subgraphs to learn user-item node representations based on weak user-item relationships between subgraphs, as well as user-item node representations based on weak item-item relationships between subgraphs, in order to better express users' interests and preferences. By using attention mechanisms, attention weights are added to each layer of the graph neural network to improve the representation of more important layers in the user feature representation.

Index Terms—Recommender systems, Graph Convolutional Networks, Attention Mechanisms, Message Passing, User-item Representation, Preference.

1 Introduction

The advent of the Internet era has led to an explosion in the amount of data people have access to, creating an information overload problem that makes it difficult for people to select the right item for their needs from a wide range of products. Recommendation systems have become an important tool to solve this problem. They analyze different preferences of users based on their historical behaviors and filter the information to give personalized recommendations. Recommender systems have become one of the important technologies for various online platforms whose purpose is to predict whether a user will interact with a project or not [1]. Among them, collaborative filtering-based models have made effective progress in learning user and item representations by modeling historical user-item interactions [2, 3].

In recent years, deep learning methods have developed rapidly in the fields of natural language, speech, image and video [4]. It is the mainstream practice of traditional recommendation algorithms to use deep learning methods to extract the hidden features of user items from user IDs, review texts, and other data, and to predict the user's rating of new items based on these features to give recommendations. Since the introduction of graph neural networks, researchers have found that graph neural networks can be used to extract the connection characteristics between users and items that cannot be extracted in traditional recommendation algorithms. Most of the recommendation algorithms based on graph neural networks currently use the ID information of user items to extract user item features, ignoring the user item features implied by other data such as review text in the data, or using auxiliary information to make the network training more complicated. Graph Convolution Neural Network [5, 6] based models have been effective in recommendation, and the main function of GCN models is to improve the embedding representation of users and items by iteratively aggregating feature information from neighbors using graph connectivity to extract additional information. NGCF [7] exploits the higher-order connectivity of graphs to alleviate the sparsity problem in recommender systems. However, GCN suffers from the over-smoothing problem because the graph convolution operation can actually be regarded as a special kind of graph Laplacian smoothing, which makes the node representation indistinguishable after multi-layer graph convolution. Chen et al. [8] proposed the LR-GCN model to mitigate the oversmoothing effect by adding residual operations. He et al. [9] further pointed out that the feature transformations and nonlinear activations in the original GCN model have little positive impact on the final performance and even more negative impact with training; the proposed Light-GCN model based on this model only retains the neighborhood aggregation operation and removes the The proposed Light-GCN model only retains the neighborhood aggregation operation and removes the "self-loop" from the aggregation operation to alleviate the over-smoothing problem. Wei et al. [10] designed a Multi- modal Graph Convolution Network (MMGCN) framework built upon the message-passing idea of graph neural networks, which can yield modal-specific representations of users and micro-videos to better capture user preferences. They proposed to exploit user-item interactions to guide the representation learning in each modality, and further personalized micro-video recommendation. To reduce the propagation of negative

*Merrick Connelly is the corresponding author.

- *Merrick Connelly, Lysandra Sinclair, Solenne Dubois, and Galen Rawlins are with Laurentian University. (e-mail: merrickconnelly91.laucs@hotmail.com).*

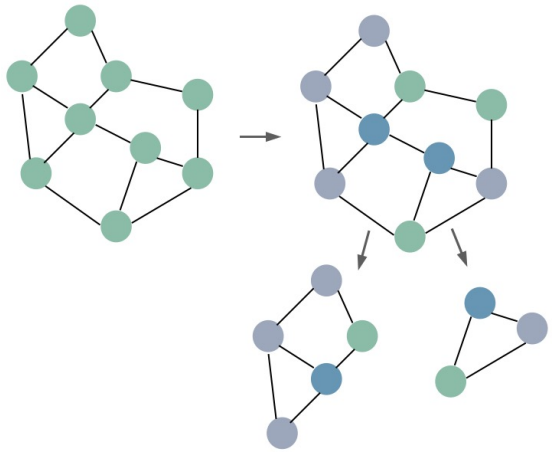

Fig. 1: Graph Convolution Neural Networks.

information, DropEdge [11] randomly removes a certain number of edges from the input graph in each training cycle, and theoretically demonstrates that DropEdge either reduces the convergence speed of oversmoothing or mitigates the information loss caused by oversmoothing. The IMPGCN [12] model assumes that not all information from higher-order neighbors is positive in reality and filters out the propagation of negative information in higher-order graph convolution operations by randomly dividing users into two subgraphs for learning and cutting off the associated edges of users between the two subgraphs. However, in IMP-GCN, dividing users into subgraphs with no intersection only leads to partial loss of information and ignores the potential association between different groups of users, and because only users are divided, the influence of product factors on the purchase outcome is ignored in the learning process. In addition, when aggregating the embedded nodes of multiple subgraphs, IMP-GCN uses a simple summation approach to learn the final representation of nodes without considering the actual weights of different parts. Based on the considerations above, this paper proposes an interest-aware message passing recommendation model. The model uses two methods to partition the graph into subgraphs, namely, partitioning the users and items into subgraphs separately. High-order graph convolution operations are performed in the subgraphs to learn user-item node representations based on weak user-item relationships between subgraphs, as well as user-item node representations based on weak item-item relationships between subgraphs, in order to better express users' interests and preferences. By using attention mechanisms, attention weights are added to each layer of the graph neural network to improve the representation of more important layers in the user feature representation.

2 Related Work

2.1 Graph neural network based recommendation model

In the year 2009, the graph neural network (GNN) model was proposed [13], and the emergence of the GCN (graph convolutional neural network) model in 2017 enabled the rapid development of graph neural networks [14], which have been widely used in the field of recommendation systems. The NGCF model introduced GCN into recommendation algorithms, modeling the high-order connectivity between users and items to provide recommendations [7]. The KGCN-PN model sorts and samples neighboring entities in the graph, and uses GCN to fuse information along relationship paths to obtain entity feature vectors. The KGAT algorithm integrates attention mechanisms into graph neural networks and achieves good results [15]. The BGANR algorithm uses attention mechanisms and adds biases to better capture the high-order connectivity between nodes [16]. The NIA-GCN model further considers the mutual interaction between neighboring nodes on the basis of GCN, and can effectively aggregate information from neighboring nodes at various depths. The GCN-ONCF model designs GCN as an encoder, uses outer product operations to transform encoding vectors into two-dimensional feature matrices, and achieves convolutional matrix decomposition through convolutional autoencoders. The LightGCN model simplifies the redundant parts of the bipartite GCN based on NGCF, thereby improving the efficiency and performance of the model. Most conventional approaches learn knowledge representation from structured triples only and ignore other information types that are widely available in the Internet, such as text type and image type data. Recently, researchers have proposed several approaches for multimodal knowledge graph representation [17, 18, 19], and these works show that the introduction of multimodal information can effectively enhance tasks such as Knowledge Graph Completion and Triple Classification metrics.

Feature-based approaches treat multimodal information as an auxiliary feature of an entity (Entity) [20]. These approaches extend the TransE model to take visual modal features or features of other modalities into account for knowledge graph representation learning. They start by extracting visual information through pre-trained visual models. In these methods, the energy of the triad is defined according to the structure of the knowledge graph and the visual representation of the entity, which means that each entity must contain graph image features. However, in real scenarios, some entities do not contain multimodal information, so this approach cannot be widely used [17]. Unlike feature-based approaches that treat multimodal information as an auxiliary feature of each entity, entity-based approaches treat information of different modalities as structured knowledge, i.e., multimodal information appears directly as entities in the knowledge graph at the same level as other structured knowledge entities. The entity-based approach introduces new relationships through which the entity-based approach associates multimodal entities with ordinary entities. MKBE [19] is a representative work of the entity-based approachThe entity-based approach does not require multimodal information for each entity in the knowledge graph, and thus can alleviate the problem of high data source requirements similar to the feature-based approach. However, the existing entity-based methods deal with each triad independently and ignore the multimodal information fusion, which is not friendly to the triads containing multimodal information.

2.2 Graph Attention Network

The attention mechanism allows a neural network to focus only on the information required for the task at hand by selecting specific inputs [21]. Introducing attention mechanisms in GNNs can help the neural network focus on nodes and edges that are more relevant to the task, improving the effectiveness of training and the accuracy of testing, leading to the development of Graph Attention Networks (GAT). There are different types of attention mechanisms, such as self-attention, multi-head attention, and hierarchical attention. In terms of self-attention, reference [22] was the first to apply this mechanism to GNNs. Based on spatial graph convolution, this method stacks the masked self-attention layer into the aggregation function of the spatial graph convolution. The calculation formula for the attention correlation coefficient is as follows:

$$e_{ij} = a(Wh_i, Wh_j) \tag{1}$$

where, $a(\cdot)$ denotes the attention calculation function, and a single-layer feedforward neural network is used with W representing the weight matrix. This formula expresses the importance of neighbor node j to node i. It implicitly assigns weight factors to the neighbor nodes of the central node, and nodes with different degrees are assigned different weight parameters. This indicates the different importance of neighbor nodes to the central node and can be understood as structural information in the spatial domain. The advantage of this method is that it does not require prior knowledge of the graph's topological structure and can calculate node-neighbor pairs in parallel, avoiding costly matrix operations such as inversion [23]. However, GAT matrix multiplication is only applicable to second-order tensors, which can limit the batch processing capability of the attention network layer in datasets with multiple graphs, potentially impacting GPU performance. Additionally, neighborhoods of some graphs may highly overlap, increasing the additional computational cost during parallelization.

Busbridge et al. [24] proposed a special type of relational graph attention network, the Relational Graph Attention Network (RGAT), based on the Relational Graph Convolutional Network (RGCN) [25]. The attention mechanism is extended to the relational graph domain, and two variants are developed: the Within-Relation Graph Attention (WIRGAT) and the Across-Relation Graph Attention (ARGAT). Zhang et al. [26] used the Multi-Head Attention Mechanism [27] to propose a Gated Attention Network (GaAN) with a sub-net convolution that constructs a Graph Gated Recurrent Unit (GGRU). They believe that while applying multiple attention mechanisms, overusing attention will aggregate less important relationship information in subspaces for a single node, ultimately leading to a decrease in prediction accuracy. Therefore, they designed a Soft Gate to control multiple attention ends and explored the importance of node features generated by different attention ends in the subspace, thus constructing a key-value attention mechanism. In terms of hierarchical attention mechanisms, Do et al. [28] proposed a Graph Attention Model for Multi-Label Learning (GAML). In multi-label classification problems, accuracy and interpretability are difficult, and there is a certain correlation between the undiscovered relationship between labels and subgraphs. Therefore, GAML treats all labels as individual nodes called label nodes. Ordinary graph nodes are called data nodes, and then label nodes and data nodes are used as the input of the neural network.

3 METHODOLOGY

3.1 Problem definition

Given a dataset D containing X samples, where each sample (u, i, r_{ui}) represents a user u writing a review r_{ui} for an item i. The core task of this paper is to train a model to learn the association between user u and item i based on all interactions between all users and all items (excluding the interaction between user u and item i). At the same time, the model learns the general preference representations of user u and item i based on the comments set of user u (excluding r_{ui}) and the comments set of item i (excluding r_{ui}). By combining these two feature representations, the model predicts the rating P_{ui} that user u would give to item i. Using the same method to obtain the rating set Pu for all products of user u, and based on this rating set, provide the top K recommended products for user u. The ultimate goal is to make the recommended product set more closely aligned with user u's future purchasing behavior.

3.2 Embedding layer

Assuming that there are N users and M products, the ID embedding vector d of the i-th user can be represented as $e^0_{ui} \in R^d$, and the ID embedding vector of the j-th product can be represented as $e^0_{ij} \in R^d$, where d is the dimension of the embedding vector, which can be considered as an adjustable hyperparameter. The ID embedding vectors of all users form the set $e^0_u \in R^{N \times d}$.

$$e^0_u = [e^0_{u1}, \ldots, e^0_{ui}, \ldots, e^0_{uN}] \tag{2}$$

Similarly, the ID embedding vectors of all products form the set $e^0_i \in R^{M \times d}$.

$$e^0_i = [e^0_{i1}, \ldots, e^0_{ij}, \ldots, e^0_{iM}] \tag{3}$$

Both the ID embedding vectors of products and users are in their initial state, and they are further refined by propagating in the forward propagation layer, so that the ID embedding vectors can better express their inherent association.

3.3 Propagation layer

By randomly constructing subgraphs, we hope that the information transmitted in the subgraph will help with the embedding learning of the node. The approach is to randomly divide users into n groups, and the items associated with those users also belong to this subgraph. Then, for the original input, the items are also randomly divided into n groups, and the users associated with those items also belong to this subgraph. Let G_{sU} denote a subgraph under user classification, and G_{sI} denote a subgraph under item classification. The message passing process will be analyzed below using

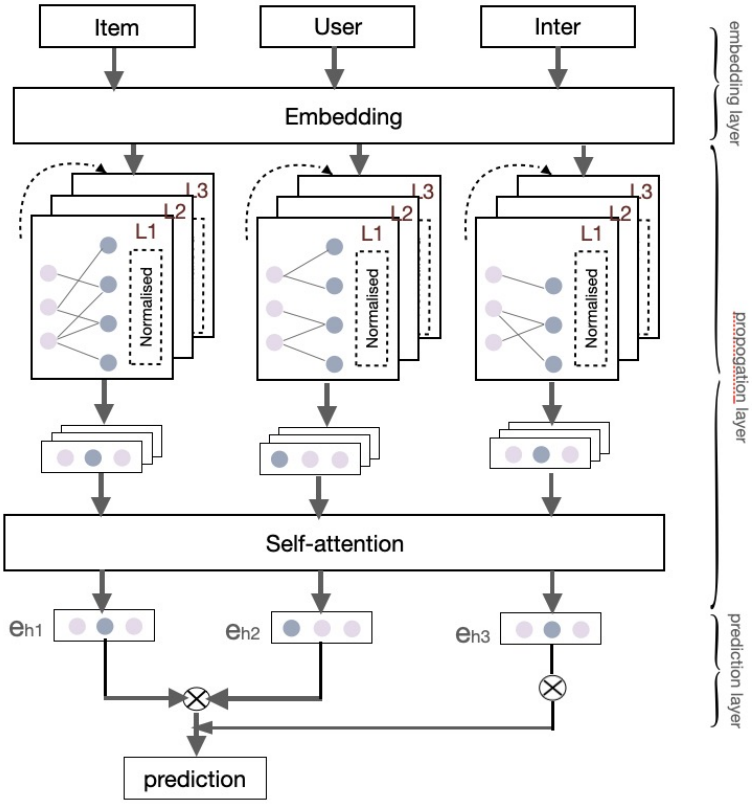

Fig. 2: Schematic diagram of our model.

the example of a user classification subgraph. Given that the direct neighbors between users and items provide the most interesting and important information to users, in particular, when conducting first-order propagation, all first-order neighboring nodes are used. Let e_u^0 represent the embedding of user u in the 0-th layer, and let e_i^0 represent the embedding of item i in the 0-th layer under user classification. The convolution operation is as follows.

$$e_u^1 = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_u^0 \tag{4}$$

$$e_i^1 = \sum_{i \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} e_i^0 \tag{5}$$

where e_u^1 represents the node representation of user u in the 1st layer under user classification, and e_i^1 represents the node embedding of item i in the 1st layer under user classification.

For high-order graph convolution, in order to reduce the influence of noise, nodes in the subgraph can only use information from nodes within that subgraph. Since all direct nodes corresponding to a user belong to the same subgraph during classification, the user node can receive information from all reachable nodes. However, all direct nodes corresponding to an item may be in different subgraphs, and the item node can only receive information from all reachable nodes within the subgraphs. Let e_{us}^k represent the embedding representation of item i in subgraph G_{sU} after k layers of convolution under user classification. Therefore, high-order propagation can be defined as follows:

$$e_u^k = \sum_{is \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_{us}^{k-1} \tag{6}$$

$$e_i^k = \sum_{u \in N_{is}} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} e_{is}^{k-1} \tag{7}$$

For the comment embedding vector $b_{ui} \in R^{RN \times RL \times c}$ of user u_i, assuming there are m convolution kernels, each convolution kernel $K_{ki} \in ks \times ks$, where ks is an adjustable convolution kernel width. For the general preference feature extraction of user u_i, it is first convolved as follows:

$$z_{ui} = LeakyReLU(reshape(b_{ui}) * K_{ki} + \mu_{ui}) \tag{8}$$

Among them, the reshape operation converts the dimension of b_{ui} into a second-order tensor, so that $b_{ui} \in R^{(RN \times RL) \times c}$; $*$represents the convolution operation, and μ_{ui} is the bias. After the convolution operation, the max-pooling operation is performed on each vector z_{ui} of the convolution layer output to further extract features. The specific approach is to find the maximum value o_{ui} of z_{ui}.

$$o_{ui} = max\{z_{u1}, z_{u2} \ldots, z_{u(RN \times RL)}\} \tag{9}$$

$$O = [o_1, o_2, \ldots, o_m] \tag{10}$$

After the pooling layer, the obtained features O are inputted to the fully connected layer. The propagation calculation formula for the fully connected layer is as follows:

$$b_{ui} = LeakyReLU(w_3 \otimes (p_{ui} \odot o) + g_{ui}) \tag{11}$$

where the weight matrix $w_3 \in R^{d \times m}$, d is the same dimension as the ID embedding of the user. g_{ui} is the bias.

$b_{ui} \in R^d$ is the final general preference feature of user u_i. Similarly, the final general preference feature of item i, b, can be calculated.

3.4 Attention mechanism

For both the user classification method and the item classification method, the learned embedding representations within each method are used to learn the weight values of the attention mechanism for the two sets of node embeddings, thus obtaining the final node representation. Each node has learned a representation for each type of edge. However, these two representations are not completely independent. The model utilizes the embedding learning method to concatenate the two types of node embedding representations obtained from the two classification methods into a matrix U for the k-th layer.

$$U_i = (E_{Ui}, E_{Ii}) \tag{12}$$

Where the i-th node embedding representation for the user classification and item classification methods is denoted as E_{Ui} and E_{Ii}, respectively. The weight of each node for each classification method is calculated using a self-attention mechanism.

$$a_{iU} = softmax(w_r^T tanh(W_r U_i))^T \tag{13}$$

$$a_{iI} = softmax(w_I^T tanh(W_I U_i))^T \tag{14}$$

where w_r and W_r are parameters trained by classifying users, while w_I and W_I are parameters trained by classifying projects. Next, the representation vector of node v_i under the relationship r is obtained:

$$E_{Ui} = U_i a_{iU} \tag{15}$$

$$E_{Ii} = U_i a_{iI} \tag{16}$$

$$E_I = E_{Ui} + E_{Ii} \tag{17}$$

3.5 Prediction layer

After passing through the forward propagation layer, the feature vector $e_{ui}, e_{ij} \in R^d$ representing the association relationship between user u_i and product i_j is obtained, and the general preference feature vector $b_{ui}, b_{ij} \in R^d$ is also obtained. In this paper, the concat method is used to fuse the two feature vectors to obtain the final feature representation U_i and I_j of user u_i and product i_j, respectively.

$$U_i = e_{ui} \oplus b_{ui} \tag{18}$$

$$I_i = e_{ij} \oplus b_{ij} \tag{19}$$

The final rating prediction of user u_i for product i_j is:

$$P_{ij} U_i \otimes I_j^T \tag{20}$$

TABLE 1: Statistical information about the datasets.

Dataset	users	iterms	interactions	sparsity
Amazon)	15280	8157	226477	99.98%
Yelp	31668	38048	1561406	99.87%
Gowalla	29858	40981	1027370	99.92%

TABLE 2: Model Performance Comparison on Amazon.

Model	recall@20	ndcg@20
LFM	5.31	4.17
NeuMF	6.45	4.76
NGCF	7.56	5.48
MMGCN	8.68	6.53
OURS	9.05	7.12

3.6 Optimization

The objective is to achieve top-n recommendations, i.e., to recommend a set of n items that match the target user preferences. Compared to rating prediction, this is a more practical task in real-world business systems, and it is optimized using pairwise learning methods.

$$Loss = \sum_{(u,a,b) \in O} -\ln \sigma(P_{ua} - P_{ub}) + \lambda ||\Theta||_2^2 \tag{21}$$

where, $O = \{||(u,a,b)|(u,a) \in R^+, (u,b) \in R^-||\}$ represents the pairwise training data, where R^+ represents the positive samples, i.e., the observed interaction items, and R^- represents the negative samples, i.e., the unobserved interaction items. σ represents the sigmoid function, and $\lambda||\Theta||_2^2$ is the regularization term.

4 EXPERIMENTS

4.1 Datasets

The experimental data consists of three publicly available datasets: Amazon, Yelp, and Gowalla. The metric statistics for each dataset are shown in Table 1.

4.2 Metrics

In this paper, the top-K recommendation method is used for recommendation, where $K = 20$. Two evaluation metrics, recall and normalized discounted cumulative gain (NDCG), are used to evaluate the performance of the model. Recall is used to measure the proportion of recommended items in the top-20 list that have been interacted with by the user in the test set, compared to all the items that the user has interacted with in the test set. A higher recall indicates

TABLE 3: Model Performance Comparison on Yelp.

Model	recall@20	ndcg@20
LFM	5.17	3.32
NeuMF	5.75	3.56
NGCF	6.37	4.08
MMGCN	7.14	4.92
OURS	8.22	5.32

TABLE 4: Model Performance Comparison on Gowalla.

Model	recall@20	ndcg@20
LFM	10.12	9.33
NeuMF	10.89	9.76
NGCF	11.54	10.35
MMGCN	12.03	11.12
OURS	12.76	11.56

better model performance. Assuming that the set of all users in the test set is U, and for any user $u \in U$, the top-20 recommendation list is L_u, and the true interaction list of user u in the test set is L_u^t, then the formula for calculating the model recall is as follows:

$$recall@20 = \sum_{u \in U} \frac{1}{|U|} \frac{|L_u \cap L_u^t|}{|L_u^t|} \tag{22}$$

NDCG measures the relevance scores of recommended items at different positions in the recommendation list. The higher the relevance score of the recommended item with respect to the user, the higher it should be ranked in the list, resulting in a higher score. NDCG evaluates the quality of the recommendation list for all users. Assuming that Li is the recommendation at the i-th position in the top-20 recommendation list, and $f(x)$ takes the value 1 when $x > 0$, otherwise takes 0. Assuming that the relevance score $rel \in \{0, 1\}$, the NDCG formula is as follows:

$$NDCG@20 = \sum_{u \in U} \frac{DCG@20}{IDCG@20} \tag{23}$$

4.3 Baselines

We compare our model with some relevant or current state-of-the-art methods.

- LFM: LFM is the most classic matrix factorization algorithm, which only uses user-item interactions to learn the latent features of users and items, and learns with item ratings as the prediction target. Its recommendation performance is affected by data sparsity.
- NeuMF [29]: NeuMF is the deep learning version of matrix factorization, which adds multiple hidden layers at the element level and connection mode to characterize the nonlinear interaction between users and items.
- NGCF [7]: NGCF follows the standard GCN model and successfully applies it to recommendation systems. It successfully utilizes bipartite graphs to extract the embedding interaction between users and items (i.e., their association) to obtain rating predictions.
- MMGCN [10]: Multimodal Graph Convolution Network is the latest multimodal-based recommendation system model that considers individual user-item interactions for each modality. Specifically, MMGCN constructs user-item bipartite graphs for each modality, then uses GCN to train each bipartite graph, and finally merges the node information of different modal node information of different modalities.

4.4 Result Analysis

Our model's comparison results with other models are shown in Table 2. All experiments were conducted under a three-layer neural network. By observing Table 2, it can be seen that the model improved the recall by 2.66%, 2.81%, and 1.49%, respectively, in the three different datasets. The NDCG was improved by 1.52%, 1.43%, and 4.15%, respectively. The performance of the MF model was poor in all four datasets, indicating that matrix factorization is not sufficient to better capture the characteristics of users and items in extremely sparse datasets, where the sparsity levels were in the order of one in a thousand. The NeuMF model, which added hidden layers to MF, showed better results than MF, with an average increase in experimental effect of 3.5%. The experiments showed that the addition of hidden layers to the model can further extract non-linear feature interactions between users and items, but it is essentially a type of MF, so the improvement in model performance is not particularly significant. The NGCF model introduced GCN into the recommendation algorithm, and the experimental results showed that its model performance improved by about 15% compared to the NeuMF model, with a significant improvement, indicating that graph neural networks have a relieving effect on data sparsity in extremely sparse datasets and contribute significantly to improving recommendation performance. It can be seen that the MMGCN model improves the performance of the NGCF model by about 19%, and the improvement effect is significant, indicating the effectiveness of the improvement. From the comparison between our model and other models, it can be seen that our model performs better than other models on three datasets, with at least a 0.21% improvement in recall and at least a 0.43% improvement in NDCG. This proves that the design of our model has a corresponding contribution to the improvement of model performance.

5 Conclusion and Future Work

Recommender systems have become one of the important technologies for various online platforms whose purpose is to predict whether a user will interact with a project or not. Among them, collaborative filtering-based models have made effective progress in learning user and item representations by modeling historical user-item interactions. In recent years, deep learning methods have developed rapidly in the fields of natural language, speech, image and video. It is the mainstream practice of traditional recommendation algorithms to use deep learning methods to extract the hidden features of user items from user IDs, review texts, and other data, and to predict the user's rating of new items based on these features to give recommendations. Since the introduction of graph neural networks, researchers have found that graph neural networks can be used to extract the connection characteristics between users and items that cannot be extracted in traditional recommendation algorithms. Most of the recommendation algorithms based on graph neural networks currently use the ID information of user items to extract user item features, ignoring the user item features implied by other data such as review text in the data, or using auxiliary information to make the network training more complicated. Graph Convolution Neural Network based models have been effective in recommendation, and the main function of GCN models is to improve the embedding representation of users and items by iteratively aggregating feature information from neighbors using graph connectivity to extract additional information. Based on the considerations above, this paper proposes an interest-aware message passing recommendation model. The model uses two methods to partition the graph into subgraphs, namely, partitioning the users and items into subgraphs separately. High-order graph convolution

operations are performed in the subgraphs to learn user-item node representations based on weak user-item relationships between subgraphs, as well as user-item node representations based on weak item-item relationships between subgraphs, in order to better express users' interests and preferences. By using attention mechanisms, attention weights are added to each layer of the graph neural network to improve the representation of more important layers in the user feature representation.

6 Conflict of interest statement

All authors have no conflict and declare that: (i) no support, financial or otherwise, has been received from any organization that may have an interest in the submitted work ; and (ii) there are no other relationships or activities that could appear to have influenced the submitted work.

References

[1] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 635–644.

[2] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[6] Y. Deldjoo, M. Schedl, B. Hidasi, Y. Wei, and X. He, "Multimedia recommender systems: Algorithms and challenges," in *Recommender systems handbook*. Springer, 2022, pp. 973–1014.

[7] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[8] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 27–34.

[9] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

[10] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1437–1445.

[11] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," *arXiv preprint arXiv:1907.10903*, 2019.

[12] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware message-passing gcn for recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 1296–1305.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

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

[16] J. Sun, Y. Zhang, W. Guo, H. Guo, R. Tang, X. He, C. Ma, and M. Coates, "Neighbor interaction aware graph convolution networks for recommendation," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1289–1298.

[17] R. Xie, Z. Liu, H. Luan, and M. Sun, "Image-embodied knowledge representation learning," *arXiv preprint arXiv:1609.07028*, 2016.

[18] M. Hao, Z. Li, Y. Zhao, and K. Zheng, "Mining high-quality fine-grained type information from chinese online encyclopedias," in *International Conference on Web Information Systems Engineering*. Springer, 2018, pp. 345–360.

[19] H. Mousselly-Sergieh, T. Botschen, I. Gurevych, and S. Roth, "A multimodal translation-based approach for knowledge graph representation learning," in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 2018, pp. 225–234.

[20] Q. Wang, Y. Wei, J. Yin, J. Wu, X. Song, and L. Nie, "Dualgnn: Dual graph neural network for multimedia recommendation," *IEEE Transactions on Multimedia*, 2021.

[21] Y. Wei, X. Wang, W. Guan, L. Nie, Z. Lin, and B. Chen, "Neural multimodal cooperative learning toward micro-video understanding," *IEEE Transactions on Image Processing*, vol. 29, pp. 1–14, 2019.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[23] Y. Wei, X. Wang, L. Nie, X. He, and T.-S. Chua, "Graph-refined convolutional network for multimedia recommendation with implicit feedback," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3541–3549.

[24] D. Busbridge, D. Sherburn, P. Cavallo, and N. Y. Hammerla, "Relational graph attention networks," *arXiv preprint arXiv:1904.05811*, 2019.

[25] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 2018, pp. 593–607.

[26] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] K. Do, T. Tran, T. Nguyen, and S. Venkatesh, "Attentional multilabel learning over graphs: a message passing approach," *Machine Learning*, vol. 108, no. 10, pp. 1757–1781, 2019.

[29] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.