Enhanced Collaborative Filtering Recommendation Model for Graph Neural Networks Based on Meta-Paths

Benjamin Fischer (✉ BenjaminFischer1991.wms@hotmail.com)
Western Michigan University

Ethan Brown
Western Michigan University

Maria Perez
Western Michigan University

Andrew Sanchez
Western Michigan University

Research Article

Keywords: Recommendation systems, Collaborative Filtering, Graph Convolutional Networks, Meta-Paths, Feature

Posted Date: March 28th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2744708/v1

Enhanced Collaborative Filtering Recommendation Model for Graph Neural Networks Based on Meta-Paths

Benjamin Fischer, Ethan Brown, Maria Perez, Andrew Sanchez,

Abstract—With the development of web information technology, recommendation systems are widely used in social media networks, news delivery, and shopping platforms. Collaborative filtering (CF) is one of the main methods in recommendation systems. In recent years, the graph convolutional neural network (GCN) has emerged as the hottest research method in the current recommendation system due to its powerful ability to learn embedding representations on graph structures. Although graph convolutional neural networks have been recognized as having significant advantages in recommendation tasks, the proposed GCN-based recommendation models have generally neglected two aspects. The graph convolution layer of most models does not take full advantage of the covariance signals of higher order neighbors. For example, GCMC only uses the signal update node inlay of the 1st order neighborhood. The graph convolution layer of existing models pays little attention to the impact of different user opinions on the recommendation results when propagating and aggregating synergistic signals. In the recommendation scenario, the rating represents the user's opinion. To address these problems, this paper proposes an enhanced collaborative filtering algorithm for graph neural networks based on meta-paths (MPECFG). The study exploits the synergistic signals of 2nd order neighbors and user's view to improve the accuracy of graph convolutional neural networks in learning node embedding. In particular, we first construct bipartite graphs of users and items, embed their historical interactions into feature vectors, and then obtain higher-order representations of users and items by multi-layer propagation of the neural network. We experimented on public datasets and compared them with traditional algorithms to verify the effectiveness of our model.

Index Terms—Recommendation systems, Collaborative Filtering, Graph Convolutional Networks, Meta-Paths, Feature.

1 Introduction

With the development of web information technology, recommendation systems are widely used in social media networks, news delivery, and shopping platforms. Colaborative filtering (CF) is one of the main methods in recommendation systems [1]. Depending on whether a machine learning model is used, it can be divided into: memory-based CF [2, 3] and model-based CF [4]. Compared with the former, which relies on similarity functions, has high memory consumption, and is difficult to discover non-linear and implicit connections among entities, the model-based approach learns the embedding representation of entities using matrix-implicit space decomposition [5, 6], M arkov chain [7], and deep neural network [8, 9], which can capture the complex connections among entities more effectively from entity interactions and auxiliary information. It is the most widely studied and recommended method.

In recent years, the graph convolutional nerve network (GCN) has been emerging [10, 11] as the hottest research method in the current recommendation system due to its powerful ability to learn embedding representations on graph structures [12, 13, 14, 15, 16]. GCN uses spectral graph theory to define graph convolution operations to propagate and aggregate messages from neighboring nodes on topological graphs, and has the advantage of learning entity embedding representations using both entity features and graph structure features. In essence, it is an extension of the excellent feature learning and representation capabilities of traditional convolutional nerve networks (CNNs) to non-regular graph data [17, 18]. Heterogeneous information networks [19] are widely used in various fields because they contain many types of nodes and edges [20, 21]. Since heterogeneous information networks can characterize rich auxiliary information, recommendation algorithms based on heterogeneous networks have been widely studied. Sun et al. [22] proposed PathSim to make recommendations based on the similarity of meta-paths. Feng et al. [23] proposed OptRank to solve the cold start problem of social recommendation system based on heterogeneous information. Yu et al. [24] further optimized the recommendation algorithm based on the relationship between heterogeneous web entities. Shi et al. [25] fused the heterogeneous information obtained from random walks with the traditional matrix decomposition model. These methods obtain the potential characteristics of users and products through the auxiliary information in the heterogeneous information network, but may ignore the information of user and product history interaction. Wang et al. [14] proposed an NGCF model, which uses GCN's message transmission mechanism to stack multiple graph convolution layers to achieve the implicit transmission of cooperative signals of higher-order neighbors on the

Benjamin Fischer is the corresponding author.

- *Benjamin Fischer, Ethan Brown, Maria Perez, and Andrew Sanchsez are with Western Michigan University). (e-mail: BenjaminFischer1991.wms@hotmail.com).*

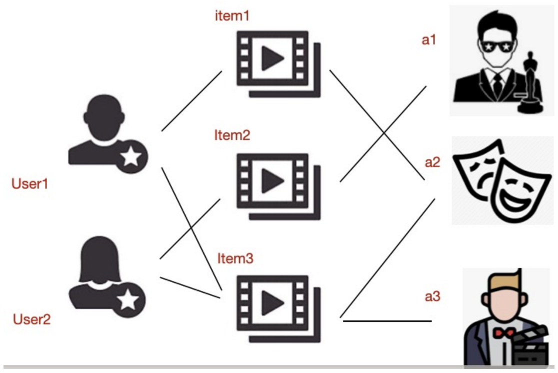

Fig. 1: Heterogeneous Knowledge Graphs.

bipartite graph, and finally merges the node embedding of all layers for recommendation, which can compensate the weakness that the GCMC model can only use the first-order cooperative signals to some extent. Wei et al. [26] 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. In addition, some scholars have improved the GCN-based recommendation model by improving the efficiency of graph convolution operation, utilizing the attention mechanism [27, 28], introducing auxiliary information [29], improving scalability, and overcoming the cold start [30, 31]. These studies have shown that GCN-based recommendation models can better capture the implicit, nonlinear covariance signals on entity interaction graphs due to the powerful feature representation learning capability, and usually achieve better accuracy than traditional collaborative recommendation models.

Although graph convolutional neural networks have been recognized as having significant advantages in recommendation tasks, the proposed GCN-based recommendation models have generally neglected two aspects. The graph convolution layer of most models does not take full advantage of the covariance signals of higher order neighbors. For example, GCMC only uses the signal update node inlay of the 1st order neighborhood. The graph convolution layer of existing models pays little attention to the impact of different user opinions on the recommendation results when propagating and aggregating synergistic signals. In the recommendation scenario, the rating represents the user's opinion. To address these problems, this paper proposes an enhanced collaborative filtering algorithm for graph neural networks based on meta-paths(MPECFG). The study exploits the synergistic signals of 2nd order neighbors and user's view to improve the accuracy of graph convolutional neural networks in learning node embedding. In particular, we first construct bipartite graphs of users and items, embed their historical interactions into feature vectors, and then obtain higher-order representations of users and items by multi-layer propagation of the neural network. We experimented on public datasets and compared them with traditional algorithms to verify the effectiveness of the our model.

2 Related Work

2.1 Heterogeneous graphical neural network

Heterogeneous graph neural networks have been proposed as an effective approach for modeling complex information networks. A heterogeneous graph comprises various nodes and edges, where the nodes can represent different entities such as users and items, and the edges capture the interactions among them. Due to their ability to model heterogeneous data flexibly [32], heterogeneous graph neural networks have been widely used to incorporate rich auxiliary information in recommendation systems. Using the representation of heterogeneous graph neural networks, a recommendation system can be seen as a similarity search of meta-paths on heterogeneous graph neural networks. Several approaches have been proposed to measure the similarity between entities based on meta-paths. For instance, Ni et al. [33] proposed a method that employs random walking on meta-paths to calculate the similarity between entities. Yu et al. [34] incorporated meta-path similarity into the regularization term of a matrix decomposition model. Luo et al. [35] proposed a recommendation algorithm that combines heterogeneous attributes and collaborative filtering. Liang et al. [36] proposed a collaborative filtering recommendation algorithm based on meta-paths, taking into account the influence of meta-paths in heterogeneous graphs.

2.2 GCN-based recommendation model

Recommendation models based on graph convolutional neural networks use the idea of message passing to propagate and aggregate collaborative signals over a bipartite graph, learning nodes of embedded representations and using them for recommendation prediction [37]. Their model structure usually consists of three parts: the input layer, the graph convolution layer, and the prediction layer. The entity nodes on graph $\mathcal{G}$ are encoded and their initial low-dimensional embedding vector e^0 is obtained.

$$e^0 = Enc(x, \theta^{in}), \tag{1}$$

where x represents the information of the entity node, either by the entity characteristics of the node or by the unique hot encoding vector based on the node serial number. e^0 is the initial embedding vector after coding, which can be subdivided into user embedding e_u^0 and project embedding e_v^0 . θ^{in} is the set of parameters to be learned for the coding function $Enc(\cdot)$.

The graph convolution layer uses the structural features of the bipartite graph to propagate and aggregate the cooperative signals of neighboring nodes on the graph to achieve the node embedding update. The main operations include co-signal construction and node embedding update. Take the example of a rating of item v by user u. This action reflects a certain degree of preference of the user for the item. Therefore, the first-order cooperative signal propagated by this action can be constructed as follows.

$$S_{(u,v)} = f(e_u, e_v, p_{u,v}, \theta^f), \tag{2}$$

where $p_{u,v} = \frac{1}{\sqrt{|N_u|\cdot|N_v|}}$ is the signal strength coefficient. The set of first-order neighbors directly adjacent to the users u and v on the bipartite graph is denoted by Nu and Nv, respectively, and $|N_v|$ and $|N_u|$ are their nodal degrees. $f(\cdot)$ is the concordant signal coding function. θ^f is the set of parameters to be learned for this function, and different models have different forms of definitions for this function. The embedding update operation brings together the co-signals of all 1st order neighbors on the bipartite graph and is used to update the embedding representation of user u.

$$e_u^p = g(e_u, \{s_{u,v} | v \in N_v\}), \tag{3}$$

Among them, the aggregation function $g(\cdot)$ can be implemented by means of mean, maximum pooling, LSTM network, etc.

Similarly, when item v receives a rating from user u, it indicates that the quality of the item matches the user's preference to some extent. Therefore, the embedding of the item entities can be updated using the users' collaborative signals.

$$S_{(v,u)} = f(e_u, e_v, p_{u,v}, \theta^f), \tag{4}$$

$$e_u^p = g(e_u, \{s_{v,u} | v \in N_v\}), \tag{5}$$

In this way, the graph convolution layer implements a single update of the embedding representation of entity nodes on the bipartite graph using the 1st-order cooperative signal.

The prediction layer uses the final nodes embedded in e_u^l and e_v^l , for any prediction of unknown scores between any user u and item v.

$$\hat{r}_{(v,u)} = f(e_u^l, e_v^l, \theta^h), \tag{6}$$

In particular, the prediction function $h(\cdot)$ is usually implemented using a bilinear decoding function or a multilayer perceptron network [5,20]. θ^h represents the set of its parameters.

3 Methodology

3.1 Embedding layer

The embedding layer transforms the input sparse vectors into dense vectors, and randomly initializes the vector matrix $E \in R^{(M+N) \times d}$ of users and items.

$$E = [e_{u1}, \ldots, e_{um}, e_{i1}, \ldots, e_{in}] \tag{7}$$

where the number of users is M, the number of items N, and d is the size of the representation dimension.

3.2 Input Layer

The input layer is responsible for encoding the user's or project's The resulting low-dimensional embedding vector is used as the input to the graph convolution layer.

$$\begin{cases} e_u^0 = W_u^{in} \cdot x_u, \\ e_v^0 = W_v^{in} \cdot x_v, \end{cases} \tag{8}$$

where $W_u^{in} \in R^{h \times d_u}$ and $W_v^{in} \in R^{h \times d_v}$ are the encoding matrices to be learned for the user and the project, respectively. This method uses a direct encoding of the original entity features of the user or project to obtain their initial embedding vectors. This method allows to encode more semantically rich entity features or auxiliary information into the graph convolution layer, which is more conducive to learning the embedding representation of nodes accurately; moreover, since only the encoding matrix needs to be optimized, the model has fewer optimization parameters. The details of the encoding of entity features are described in the experimental section.

3.3 Propagation layer

Based on the adjacency matrix of users and items, the feature vectors are fused with the vectors of their neighboring nodes, and then the final feature vectors are obtained by passing information through a multilayer neural network. For any node pair (u, i) of a user's item, the information transfer function from product i to user u is defined as shown in the following equation.

$$m_{u,i} = \frac{1}{\sqrt{|N_i||N_u|}} (W_1 e_i + W_2 (e_u \odot e_i)) \tag{9}$$

where e_i is the feature vector of the commodity, e_u is the feature vector of the user, $\frac{1}{\sqrt{|N_i||N_u|}}$ is the weight coefficient between user u and commodity i, N_u and N_i represent the number of neighbor nodes of user and commodity respectively, $W_1, W_2 \in R^{d \times d}$ are the parameters of the weight matrix, $\odot$ is the product of the corresponding elements of the vectors.

Since the user may have more than one neighbor node, the final vector representation of the user is the fusion of all its neighbor nodes. Through the information transfer at layer l, the user can obtain the information of its neighbors of order l. The feature vector representation at layer l is shown in following equation. It can be seen that the user not only fuses the features of its neighboring nodes It can be seen that the user not only incorporates the features of its neighboring nodes, but also retains the features of the user itself.

$$e_u^{(l)} = LeakyReLU(m_{u,u}^{(l)} + \sum_{i \in N_u} m_{u,i}^{(l)}) \tag{10}$$

$$m_{u,i}^{(l)} = p_{ui}(W_1^{(l)} e_i^{(l-1)} + W_2^{(l)} (e_u^{(l-1)} \odot e_i^{(l-1)})) \tag{11}$$

$$m_{u,u}^{(l)} = W_1^{(l)} e_u^{(l-1)} \tag{12}$$

where p_{ui} is the weight coefficient, W_1 and W_2 are matrix parameters, $e_i^{(l-1)}$ is the eigenvector of the items in layer $l-1$, $e_u^{(l-1)}$ is the feature u of the users in layer $l-1$ vector. The feature vectors of users and items are finally represented as follows.

$$e_u^* = e_u^{(l)} || \ldots || e_u^{(l)} \tag{13}$$

where || denotes the concat between vectors.

3.4 Enhanced graph convolution layer

This model achieves layer-by-layer refinement of the node embedding representation by stacking L graph convolutional layers. Let $u_i \rightarrow u_k \rightarrow u_j \rightarrow \ldots$ be any connected path from user u_i on the bipartite graph G. The subscripts i, k, and j denote the node numbers. We call v_k the first-order neighbor of u_i, u_j the second-order neighbor of u_i, and so on. Let u_j be the 2-th order neighbor of the target user u_i on the graph G, and $u_i \rightarrow u_k \rightarrow u_j$ be an arbitrary connected path of length 2 between them.

$$s_{i,j}^l = \sum p_{i,j,k} (e_{u,j}^{l-1} + e_{u,i}^{l-1} \odot e_{u,j}^{l-1}) \tag{14}$$

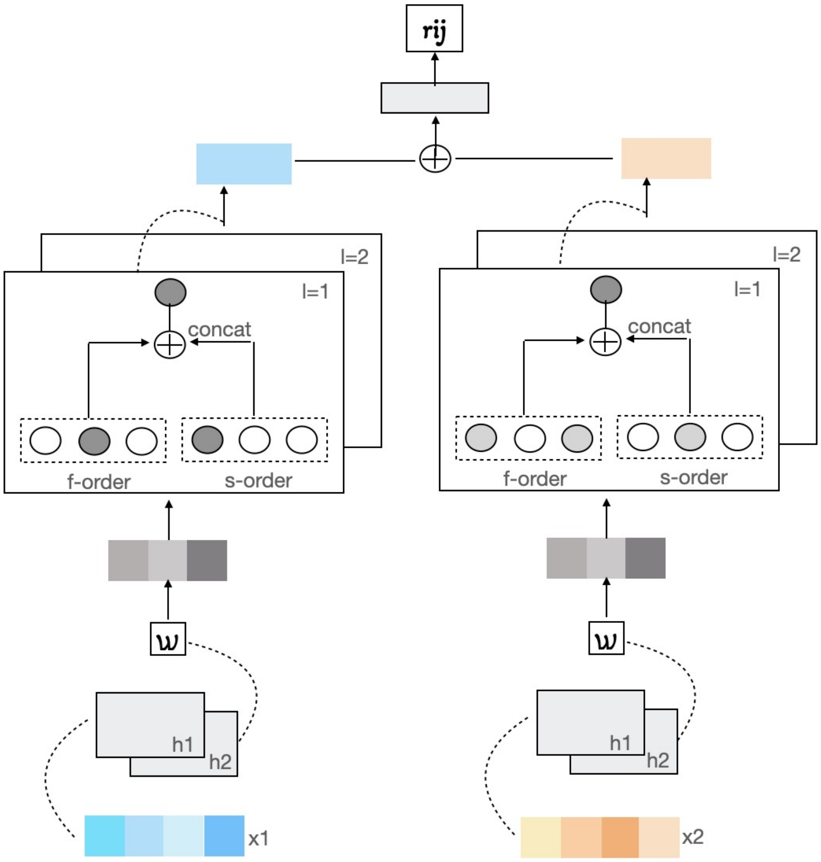

Fig. 2: Schematic diagram of our model.

where $e_{u,j}^{l-1}$ is the embedding of user u_j at layer $l-1$. $\odot$ is the Hadamard product by element. $p_{i,j,k}$ is the 2-th order covariance signal strength coefficient of the graph convolutional nerve network, which is related to the node degree on the path.

We propose a method for constructing a first-order cooperative signal with a user's point of view, based on the message passing mechanism of GCN. It defines the first-order cooperative signal converged to user u_i as follows.

$$s_{u,i}^{l(1)} = \sum_{j\in N_i} c_{i,j} p_{i,j} W_1^l e_{v,j}^{(l-1)} \tag{15}$$

where the opinion coefficient $c_{i,j}$ represents the user u_i opinion on item v_j, and the value taken in this paper is the normalized score. The constants $p_{i,j}$ are the 1-th order signal strengths. $W_1^l \in R^{h\times h}$ is the weight matrix to be learned in the lth layer.

3.5 Nonlinear prediction layer

The nodes of the final learning of the graphical convolutional layer are embedded in e_u^L and e_v^L and fed into a 2-layer MLP network to predict the unknown ratings $\hat{r}_{u,v}$ between user u and item v.

$$\hat{r}_{u,v} = W_4(ReLU(W_3(e_u^L \oplus e_v^L))) \tag{16}$$

4 EXPERIMENTS

4.1 Datasets

To verify the effectiveness of our method, we conducted extensive experiments on three real datasets,including Movielens-100k, Movielens-1M, and Yelp. The statistical properties of the datasets are summarized in Table 1. In the preprocessing stage, we encode the original entity features of the user and the project directly on the bipartite graph using auxiliary information and graph structure features. In this paper, we set it as a 256-dimensional vector generated on the user or item adjacency matrix using the Node2vec model. For the MovieLens dataset. The entity features of items consist of node degree F1, rating feature F2, and semantic feature F3. The entity features of the items are composed of three parts: node degree F1, rating feature F2, and semantic feature F3.

4.2 Metrics

In this paper, the recommended model is evaluated using the root mean square error (RMSE), which is the arithmetic square root of the mean squared error. The RMSE is the expected value of the squared difference between the estimated and true values of the parameters. The smaller the RMSE value, the better the recommended model and the more accurate the prediction.

$$RSME = \sqrt{\frac{1}{|D_t|}\sum_{(i,j)\in D_t} (r_{u,i} - \hat{r}_{u,i})^2}, \tag{17}$$

where D_t denotes the test set data, $r_{u,i}$ and $\hat{r}_{u,i}$ denote the actual scores of the node pair (u,i) and the predicted scores of the model, respectively.

TABLE 1: Statistical information of datasets.

dataset	users	items	u-i interactions	u-u interactions
Movielens-100k	943	1682	100000	47150
Movielens-1M	6040	3706	1000209	18802
Yelp	16284	14234	198397	158590

TABLE 2: Model Performance Comparison.

Model	Movielens-100k	Movielens-100k	Yelp
NeuCF	0.917	0.863	0.825
GCMC	0.902	0.844	0.801
GraphRec	0.886	0.821	0.795
NGCF	0.867	0.797	0.774
Ours	0.842	0.768	0.752

4.3 Experiments result analysis

4.3.1 Training set ratio experiments

By varying the ratio of the training set to the test set, we test the RMSE of the model at different ratios. For Yelp data, the ratio of training set data is set to (90%, 80%, 70%, 60%) due to sparsity, while for Movielens-100k, Movielens-1M, the ratio of training set data is set to (80%, 60%, 40%, 20%), the experimental results are shown in Figure 3, from left to right. The results of Yelp, Movielens-100k, and Movielens-1M are shown in Figure 3. From Fig. 2, we can see that Yelp data achieves the best RMSE results with 90% of the training set data. Movielens-100k, Movielens-1M data achieve the best RMSE results when 80% of the training set data are used. Therefore, we set the ratio of training set to test set for Y elp to 9:1, and the ratio of training set to test set for Movielens-100k and Movielens-1M to 8:2.

4.3.2 Comparative experiments

We compare the performance of the MPECFG model with the comparison algorithm on each dataset. The RSME of their predictions on the test set is shown in Table 2.

The prediction error of the four recommendation models based on graph convolutional neural networks is significantly better than that of the NeuCF model based on the moment matrix decomposition. This suggests that the former can learn the embedded representation of nodes more accurately and obtain better prediction results due to the integration of entity features of nodes and graph structure features on the topological graph. GCMC aggregates the cooperative signals of 1st order neighbors and has insufficient learning ability for node embedding, so it has the worst performance among the four GCN-based recommended models. Both GraphRec and NGCF models utilize higher-order covariance signals and both achieve better prediction results than GCMC. The difference is that GraphRec directly aggregates social friends' synergy signals, which are equivalent to similar higher-order neighbors in a bipartite graph. NGCF implicitly propagates and aggregates higher-order synergy signals by stacking multiple graph convolutional layers. This shows that aggregating higher-order collaborative signals can help improve the performance of the recommendation model. The MPECFG model in this paper achieves the lowest prediction error on most of the data sets. The superior performance of this model can be attributed to its increased utilization of higher-order synergistic signals in the graph convolution layer and the introduction of user's views into the signal aggregation process. In contrast, GraphRec only obtains the higher-order synergistic signals of users directly from the social relationships without considering the higher-order signals of items; NGCF model does not explicitly utilize the higher-order synergistic signals inside the graph convolution layer and does not consider the influence of users' opinions.

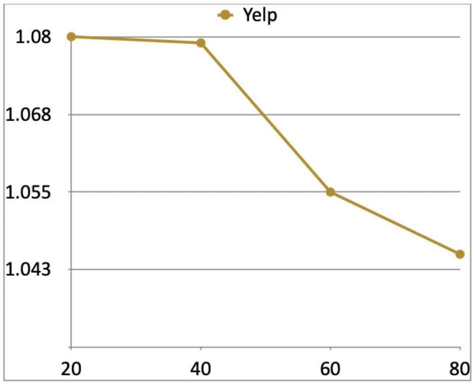

Fig. 3: RMSE values at different training set ratios on Yelp.

4.3.3 Convolutional layers experiments

Considering that our model and NGCF are stacked with multiple graph convolution layers to achieve layer-by-layer refinement of the node embedding. We experimentally investigate the prediction error when the model takes different numbers of graph convolution layers.

The prediction errors of both models are significantly affected by the number of graph convolution layers and show similar trends in the two data sets: the prediction error first decreases significantly with the number of layers, but after 3 layers, it no longer decreases significantly and shows a small increase. This indicates that stacking too many convolutional layers is prone to overfitting problems. In all cases, the prediction error of this model is significantly better than that of NGCF. The advantage is especially obvious when the number of convolutional layers is small. This is due to the advantage of the augmented convolutional layers used in our model, which improves the accuracy of node embedding learning by aggregating the second-order synergistic signals and incorporating the user's point of view within the layers to more rationally aggregate the synergistic signals.

5 Conclusion and Future Work

In recent years, the graph convolutional nerve network (GCN) has been emerging as the hottest research method in the current recommendation system due to its powerful ability to learn embedding representations on graph structures. GCN uses spectral graph theory to define graph convolution operations to propagate and aggregate

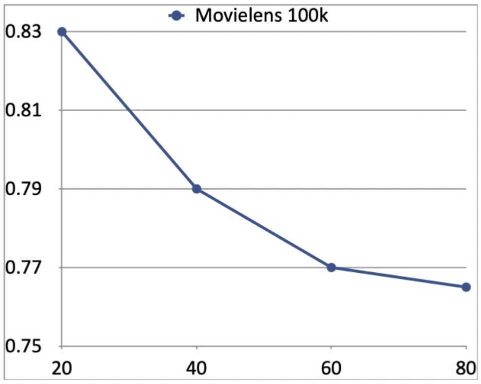

Fig. 4: RMSE values at different training set ratios on Movielens-100k.

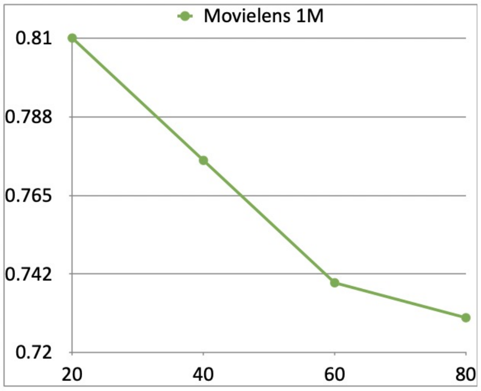

Fig. 5: RMSE values at different training set ratios on Movielens-1M.

messages from neighboring nodes on topological graphs, and has the advantage of learning entity embedding representations using both entity features and graph structure features. In essence, it is an extension of the excellent feature learning and representation capabilities of traditional convolutional nerve networks (CNNs) to non-regular graph data. Heterogeneous information networks are widely used in various fields because they contain many types of nodes and edges. Since heterogeneous information networks can characterize rich auxiliary information, recommendation algorithms based on heterogeneous networks have been widely studied. Sun et al. proposed PathSim to make recommendations based on the similarity of meta-paths. Feng et al. proposed OptRank to solve the cold start problem of social recommendation system based on heterogeneous information. Yu et al. further optimized the recommendation algorithm based on the relationship between heterogeneous web entities. Shi et al. fused the heterogeneous information obtained from random walks with the traditional matrix decomposition model. These methods obtain the potential characteristics of users and products through the auxiliary information in the heterogeneous information network, but may ignore the information of user and product history interaction. Wang et al. proposed an NGCF model, which uses GCN's message transmission mechanism to stack multiple graph convolution layers to achieve the implicit transmission of cooperative signals of higher-order neighbors on the bipartite graph, and finally merges the node embedding of all layers for recommendation, which can compensate the weakness that the GCMC model can only use the first-order cooperative signals to some extent. Although graph convolutional neural networks have been recognized as having significant advantages in recommendation tasks, the proposed GCN-based recommendation models have generally neglected two aspects. The graph convolution layer of most models does not take full advantage of the covariance signals of higher order neighbors. For example, GCMC only uses the signal update node inlay of the 1st order neighborhood. The graph convolution layer of existing models pays little attention to the impact of different user opinions on the recommendation results when propagating and aggregating synergistic signals. In the recommendation scenario, the rating represents the user's opinion. To address these problems, this paper proposes an enhanced collaborative filtering algorithm for graph neural networks based on meta-paths(MPECFG). The study exploits the synergistic signals of 2nd order neighbors and user's view to improve the accuracy of graph convolutional neural networks in learning node embedding. In particular, we first construct bipartite graphs of users and items, embed their historical interactions into feature vectors, and then obtain higher-order representations of users and items by multi-layer propagation of the neural network. We experimented on public datasets and compared them with traditional algorithms to verify the effectiveness of the our model.

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] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," *The adaptive web: methods and strategies of web personalization*, pp. 291–324, 2007.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[3] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: Two sides of the same coin?" *Communications of the ACM*, vol. 35, no. 12, pp. 29–38, 1992.

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

[5] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[6] 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.

[7] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier, "An mdp-based recommender system." *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[8] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A neural autoregressive approach to collaborative filtering," in *International Conference on Machine Learning*. PMLR, 2016, pp. 764–773.

[9] L. Zheng, V. Noroozi, and P. S. Yu, "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*, 2017, pp. 425–434.

[10] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] 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.

[18] 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.

[19] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han, "Text classification with heterogeneous information network kernels," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[20] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "Shine: Signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 592–600.

[21] Z. Jiang, J. Wang, L. Zhao, C. Sun, Y. Lu, and X. Liu, "Cross-domain aspect category transfer and detection via traceable heterogeneous graph representation learning," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 289–298.

[22] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[23] W. Feng and J. Wang, "Incorporating heterogeneous information for personalized tag recommendation in social tagging systems," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1276–1284.

[24] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han, "Collaborative filtering with entity similarity regularization in heterogeneous information networks," *IJCAI HINA*, vol. 27, 2013.

[25] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2018.

[26] 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.

[27] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *Proceedings of the Twelfth ACM international conference on web search and data mining*, 2019, pp. 555–563.

[28] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[29] 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.

[30] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5382–5390.

[31] J. Zhang, X. Shi, S. Zhao, and I. King, "Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems," *arXiv preprint arXiv:1905.13129*, 2019.

[32] 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.

[33] N. Lao and W. W. Cohen, "Fast query execution for retrieval models based on path-constrained random walks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 881–888.

[34] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, pp. 283–292.

[35] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 917–922.

[36] T. Liang, X. Sheng, L. Zhou, Y. Li, H. Gao, Y. Yin, and L. Chen, "Mobile app recommendation via heterogeneous graph neural network in edge computing," *Applied Soft Computing*, vol. 103, p. 107162, 2021.

[37] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.