EasyChair Preprint

№ 8541

Hypergraph-based Academic Paper Recommendation

Yu Jie, He Junchen and Xu Lingyu

July 25, 2022

Hypergraph-based Academic Paper Recommendation

Jie Yu, Junchen He, and Lingyu Xu*

School of Computer Engineering and Science, Shanghai University,
Shanghai, 200444, P. R. China
{jieyu,junchenhe,*xly}@shu.edu.cn

Abstract. Academic paper recommendation aims to provide personalized recommendation services for scholars from massive academic papers. Deep Learning-based Collaborative Filtering plays an important role in it, and most of existing method are based on bipartite graph, which causes it fail to realize multi-features fusion, and the over-smooth property of GCN limits the generation of embedding with high-order similarity, resulting in the decline of recommendation quality. In this paper, we propose a hypergraph-based academic paper recommendation method. Based on hypergraph, APRHG (Academic Paper Relation HyperGraph) is constructed to not only model the complex academic relationship between users and papers, but also realize the multi-features fusion. In addition, the L-HGCF (Light HyperGraph based Collaborative Filtering) algorithm, which could mine high-order similarity between papers, is proposed to provide trusted recommendations. We conduct experiments on the public dataset, and compare the performance with several deep learning based Collaborative Filtering to confirm the superiority of our method.

Keywords: Hypergraph, Collaborative Filtering, Academic paper Recommendation.

1 Introduction

Academic papers are considered to be important indicators of advances in a field. They are also important mediums for researchers to communicate with each other. Due to the rapid emergence of big academic data recently, the cognitive burden of scholars to search for the academic knowledge they want has increased. In the construction of digital library, Information Retrieval (IR) technology[13] are used to alleviate this problem to some extent, but sometimes junior researchers may have no idea on how to choose appropriate queries fed into the search box. In such an environment, there is an urgent requirement for an effective academic paper recommendation technology, by which researchers can be freed from tedious and time-consuming paper screening.

* Corresponding author.

Collaborative Filtering (CF) has become one of the most popular and widely used algorithms for academic paper recommendation, which is based on the assumption that users with similar behaviors have similar preferences for papers. However, except for the users' historical preference for papers, the co-citation and co-keyword relationship between papers are two other key features by which academic paper recommendation differentiates from general recommendation tasks. When the emerging CF algorithms, especially some GCN-based CF algorithms[2, 6], are applied to academic paper recommendation, the absence of these two key features will lead to unsatisfactory recommendation results. These GCN-based CF algorithms are generally based on bipartite graph which uses two kinds of vertices to represent users and papers respectively, and use edges represent users' preferences for papers. However, bipartite graph has insufficient ability of fusing co-citation and co-keyword relationship between papers. In addition, the preferences of the current user and similar users are represented in form of third-order neighbors. And at least three layers of GCN are needed to be stacked to complete embedded propagation between them. The over-smooth property of GCN[9] limits the generation of embedding with high-order similarity, resulting in the decline of recommendation quality.

To tackle this challenging issue, in this paper, we propose a hypergraph-based academic paper recommendation method. According to users' historical preference and academic relation between papers, we present APRHG (Academic Paper Relation HyperGraph) which meets the requirements complex relationship description and multi-features fusion. Academic relations including co-citation and co-keyword relationship between papers, they are fused by extending the semantic of hyperedges (i.e., regard co-citation relationship as a hyperedge). In addition, we propose the L-HGCF (Light HyperGraph based Collaborative Filtering) algorithm based on the simplified hypergraph convolution network (HGCN) to provide trusted recommendations. Compared with GCN, L-HGCF algorithm has a clearer logic and has advantages in embedding propagation, and extensive experiments show the whole recommendation method achieves great performance in the task of academic paper recommendation.

The main contributions of our work are summarized as follows:

(1) Based on hypergraph, we proposed the APRHG which takes full advantage of co-citation and co-keyword relationship between papers, and because user vertices are not introduced into the graph, we avoid stacking multi-layers GCN for mining high-order similarity, as with bipartite graph.
(2) Based on the constructed APRHG, we proposed an L-HGCF algorithm to learn latent vectors which represent relationship between papers and then provide trusted recommendations.
(3) We conducted experiments compared with four deep learning algorithms on the public dataset CiteUlike-A[15]. At the same time, we conducted detailed ablation experiments to verify the rationality of the components. Experimental results demonstrate the effectiveness of our method.

The rest of this paper is organized as follows. In the next section we review related work. Section 3 illustrates the overall framework of our recommendation

method. Section 4 evaluates the experimental results compared with baseline models and section 5 presents ablation studies. Finally in section 6, we provide a conclusion and discuss prominent future research directions.

2 Related Work

Collaborative Filtering (CF) is a prevalent recommendation technique and has been successfully applied in multiple domains including e-commerce[3], movies[8] and so on. The early CF methods [8] worked by establishing a database of users' preferences for items.

With the development of intelligent technology[10, 4], especially the application of graph convolution neural network (GCN)[11], researchers began to explore GCN-based CF problems, they applied deep learning to CF problems and made good progress. GCMC [2] represented interaction data such as movie ratings as bipartite user-item graph with labeled edges, and then applied a graph-based auto-encoder on it. In order to learn embeddings such as e-commerce item embedding, NGCF [17] propagated embedding of users and items on the bipartite graph, which effectively inject the collaborative signal into the embedding process. Based on NGCF, LightGCN [6] simplified the GCN operation for collaborative filtering, so that the model only contains the most important components in GCN, neighborhood aggregation.

The traditional CF algorithms have been widely used in academic paper recommendation system. Yang *et al.* [19] presented an academic paper recommendation system which uses a ranking-oriented CF method based on users' access logs. Sugiyama *et al.* [14] developed an academic paper recommendation system, which incorporate the associated papers information when building the user profile according to the citation network. Xia *et al.* [18] proposed to establish the additional relationship between papers according to the co-author information. They constructed the tripartite graph which contains user, papers and authors to combine the co-author relationship between papers with users' historical preferences for papers.

However, limited by structure of bipartite graph, GCN-based CF deep learning algorithms cannot fuse multi-features provided by the academic papers. Meanwhile, the over-smooth property of GCN[9] limits the generation of embedding with high-order similarity, resulting in the decline of recommendation quality. Based on these problems, we propose to build our academic paper recommendation method based on hypergraph, which will be described in detail later.

3 HyperGraph-based Academic Paper Recommendation

3.1 Architecture of recommendation method

Figure 1 sketches the overall framework of our proposed academic paper recommendation method. The method is divided into two stages: APRHG construction and L-HGCF algorithm. APRHG is used to describe the complex academic

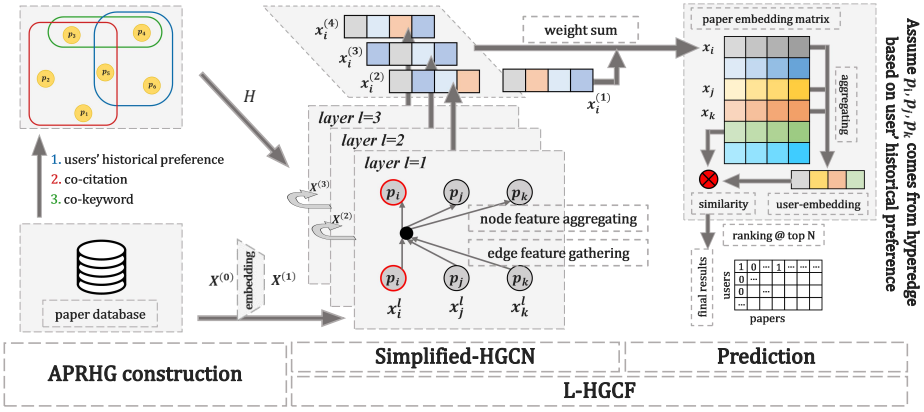

Fig. 1. The overall framework of our proposed academic paper recommendation method.

relationship between users and papers, and L-HGCF algorithm is proposed to provide trusted recommendations based on the constructed hypergraph.

Section 3.2 and section 3.3 will introduce the detail of APRHG construction and L-HGCF algorithm, respectively.

3.2 Academic Paper Relation Hypergraph

Hypergraph is a special graph whose edge connect two or more vertices [1]. Based on Hypergraph, we propose an APRHG (Academic Paper Relation Hypergraph) which is defined as follows:

Definition 1. *APRHG (Academic Paper Relation HyperGraph) is defined by* $G = (P, E, W)$ *, where*

- $P = \{p_1, p_2, ..., p_n\}$ *is the vertex set,* p_i *represent academic paper* i*;*
- $E = \{e_i^r = (p_{i1}^r, p_{i2}^r, ..., p_{ik}^r) | r \in R\}$ *is the hyperedge set, where* R *denote different kinds of hyperedge. Each hyperedge* e_i^r *is the subset of the paper set, they have academic relation* r *with each other;*
- W *is a diagonal matrix assign weight to each hyperedge.*

From the definition of APRHG, it can be seen that academic papers that have academic relation can be gathered by hyperedge. If some vertices are surrounded by more same hyperedges, it indicates that they have higher correlation and the corresponding papers have more similarity.

An APRHG can be denoted by a $|P| \times |E|$ incidence matrix H, with entries defined as Eq. (1)

$$h(p, e) = \begin{cases} 1, p \in e \\ 0, p \notin e \end{cases} \tag{1}$$

Considering the academic relationship including relation between users and papers, as well as papers, we define three types of hyperedges:

1): Hyperedges based on users' historical preference for papers. Each user in the record is regarded as a hyperedge, by which the user's preferred papers are connected with each other.

2): Hyperedges based on co-citation relationship between papers. Given the original citation network, each time one vertex in the citation network is selected as the center point and its connected vertices including the center point itself are used to generate one hyperedge.

3): Hyperedges based on co-keyword relationship between papers. Considering that the keywords of title has rich semantics, the semantic information of the title will be diluted if the title and abstract are processed into bag of word together, so we treat keywords in the title as hyperedges, and papers whose title contain this keyword are treated as vertices which are connected by these hyperedges.

In this paper, we adopt the stack strategy to obtain the complete hypergraph structure, which can be formulated as $H = H^{(0)} \| H^{(1)} \| H^{(2)} \| ... \| H^{(n)}$, where $\|$ denotes concatenation operation, $H^{(0)}$ to $H^{(n)}$ denote hyperedges constructed by different features, respectively.

3.3 Light Hypergraph Algorithm based Collaborative Filtering

Inspired by the work of Feng *et al.*[5] and Jiang *et al.*[7] , we propose L-HGCF algorithm in this paper.

Firstly, The title, abstract, and keywords of papers are processed into the bag of words as papers attribution $X^{(0)}$. Based on $X^{(0)}$, the embedding matrix $X^{(1)}$ is obtained through an embedding component. The paper embedding matrix $X^{(1)}$, together with the constructed incidence matrix H of ARPHG, are then transferred into the simplified HGCN (HyperGraph Convolution Network) layer[5].

For the target paper p_i, messaging happens between p_j connected by the common hyperedges, and then follows an aggregation process which is weighted according to weight of hyperedges. It can be represented in a matrix form as Eq. (2)

$$X^{(l+1)} = \sigma(D_p^{-\frac{1}{2}} H W D_e^{-1} H^T D_p^{-\frac{1}{2}} X^{(l)} \theta^{(l)}) \tag{2}$$

where $X^{(l)}$ is the input of (l)-th layer. D_p and D_e are the diagonal matrices of the paper degree defined as $d(p) = \sum_{e \in E} W(e) h(p, e)$ and hyperedge degree defined as $\delta(e) = \sum_{p \in P} h(p, e)$, respectively. W is a diagonal matrix store all the positive weight of hyperedges. $\theta^{(l)}$ is the learnable weight matrix and $\sigma(.)$ is non-linear activation function like eLU and LeakyReLU.

In order to light the burden of model, we abandon the use of feature transformation $\theta^{(l)}$ and nonlinear activation $\sigma(.)$, and retain only the most important component in HGCN, neighborhood aggregation, in our model. As shown in Eq. (3).

$$X^{(l+1)} = D_p^{-\frac{1}{2}} H W D_e^{-1} H^T D_p^{-\frac{1}{2}} X^{(l)} \tag{3}$$

The input of the first layer HGCN $X^{(1)}$, is the embedding from the attribute values of the original papers $X^{(0)}$. Once $X^{(1)}$ are given, the embeddings at higher layers are computed via our Simplified HGCN layer which is defined in Eq. (3). After K-layer calculation, we further combine the embeddings which are obtained at each layer to form the final representation of a paper, show in Eq. (4):

$$X = \sum_{k=1}^{K+1} \alpha_k X^{(k)} \tag{4}$$

where $\alpha_k > 0$ denotes the importance of the k-th layer embedding in constituting the final embedding. In our experiment, we set α_k uniformly as $1/(1+k)$.

In our proposed academic paper recommendation method we regard only papers as vertices. Therefore, a strategy is needed to aggregate user embedding. In this paper, user embedding is obtained by averaging embedding of historical preferred papers, show as Eq. (5).

$$x_u = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{5}$$

And then the similarity between users and papers is defined as the inner product of their final representations, show as Eq. (6):

$$y_{ui} = x_u^T x_i \tag{6}$$

The final recommendation results to user u are decided according to the descending order of similarity y_{ui}.

The only trainable parameters in our algorithm come from the embedding component which outputs $X^{(1)}$, i.e.,$\theta = \{X^{(1)}\}$. We design a tuple-wise loss function which satisfies the hyperedge property as show in Eq. (7).

$$L = -\sum_{u=1}^{M} \sum_{i \in u} (\sum_{j \in u, j \neq i} ln\sigma(x_i^T x_j) - \sum_{k \notin u} ln\sigma(x_i^T x_k)) + \lambda \|X^{(1)}\|^2 \tag{7}$$

where λ controls the L2 regularization strength. $\sigma(.)$ is the sigmoid function. M includes only hyperedges which are based on users' historical preference for papers. We see papers connected by these hyperedges as positive examples, and sample the negative samples to be 10 times the amount of the positive samples. This loss function encourages the prediction of positive entry to be higher than its negative counterparts. We employ the Adam optimizer and use it in a mini-batch manner.

The computational cost of our approach to generate paper recommendations for users mainly comes from three steps: paper similarity computation - equation Eq. (3)(4). User embedding computation - Eq. (5) and recommendation

list generation - Eq. (6). In order to avoid repeat computing, we calculate the matrix of $D_p^{-\frac{1}{2}} H W D_e^{-1} H^T D_p^{-\frac{1}{2}}$ and store it beforehand, so that time complexity of the first step is $O(n^2 d)$, the second step is $O(mn)$, and last one is $O(mn)$. Where we assume the number of user and paper is m and n, the dimension of paper paper attribution is d.

4 Experimental Analysis

4.1 Experimental Settings

Table 1. Statistics of the preprocessed experimented data.

Dataset	#User	#Paper	#Interaction	#Citation	#Key word
CiteUlike-A	4880	11845	172267	31517	582

We select dataset collected on the famous academic literature sharing website CiteULike[1], named CiteUlike-A[15]. The dataset contains users' historical preference for papers, the citation network of papers and raw data of title and abstract, for which the statistics are shown in Table 1. For this dataset, we filter out the papers and their records with damaged title or abstract, and filter out users less than 10 records and papers less than 5 records. And then we randomly divide the dataset into two subsets, in which 80% of the historical preference records constitute the training set and the other 20% of the preference records constitute the test set.

For performance evaluation, we adopt three widely-used metrics, including precision@K, recall@K, and normalized discounted cumulative gain (ndcg@K). The metrics are computed by the Eq. (8), Eq. (9), and Eq. (10), respectively. In our experiments, K is set as 20.

$$precision@k = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|} \tag{8}$$

$$recall@k = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|} \tag{9}$$

where $R(u)$ is the list of recommendations to user u and $T(u)$ is the list of true selections of user u.

$$ndcg@k = Z_K \sum_{i=1}^{K} \frac{2^{r_i} - 1}{log_2(1+i)} \tag{10}$$

[1] https://citeulike.org/

where Z_K is the normalization factor, the value of r_i is 0 or 1, indicating the predictive correlation of paper i, and $1/log_2(1+i)$ represents the importance coefficient at i position.

The proposed L-HGCF algorithm is implemented in PyTorch. The hidden dimension, i.e., the embedding size for papers, are fixed as 64 and the embedding parameters are initialized with the Xavier method. The batch size is set as 32. For all compared methods, we opt for the Adam optimizer and use the default learning rate of 0.001 and the random seed of all models is set to 1024. The L2 regularization coefficient λ is searched in the range of $\{0, 1e{-}5, 1e{-}4, 1e{-}3, 1e{-}2, 1e{-}1\}$. The layer combination coefficient α_k is uniformly set to $1/(1{+}k)$, where K is the number of layers. We test K in the range of 1 to 4, and satisfactory performance can be achieved when K equals to 2.

4.2 Performance Comparison with Baseline Models

Four recent competitive methods are selected for performance comparison. They are representatives of three graph structures, namely graph free structure (BPR-MF[12]), simple graph structure (SNDE[16]) and Bipartite graph structure (LightGCN[6], MCMC[2],), respectively.

(1) BPR-MF[12]: a classic matrix factorization method using the BPR as loss function, which optimizes the embedding of users and items through pairwise ranking between the positive instances and sampled negative items.

(2) SDNE[16]: a graph representation learning frame-work which jointly exploits the first-order and second-order proximity to preserve the network structure.

(3) GCMC[2]: a graph auto-encoder framework for recommendation system from the perspective of matrix completion, in which a graph convolution layer is introduced to generate user and item embeddings through message passing.

(4) LightGCN[6]: a simplified GCN-based recommendation framework which integrates the user-item interactions into the embedding process. For this method, we construct two bipartite graphs. One is the general model namely LightGCN-1 as Eq. (11). In the other one namely LightGCN-2, extra citation network is introduced to build the graph as Eq. (12).

$$A = \begin{pmatrix} 0 & R \\ R^T & 0 \end{pmatrix} \tag{11}$$

$$A = \begin{pmatrix} 0 & R \\ R^T & C \end{pmatrix} \tag{12}$$

where R is the Rating matrix, C is Citation network between papers.

Meanwhile, according to the different types of hyperedge. Four hypergraphs for L-HGCF algorithm are designed. They are:

1): APRHG-1 contains hyperedges based on users' historical preference for papers;

2): APRHG-2 adds additional hyperedges based on co-citation relationship;

3): APRHG-3 adds additional hyperedges based on co-keyword relationship;

4): APRHG-4 combines all the three hyperedges.

And corresponding algorithms are named as L-HGCF-1, L-HGCF-2, L-HGCF-3 and L-HGCF-4, respectively.

Table 2. Overall performance comparison of all baselines and L-HGCF algorithm.

Method	recall	precision	ndcg	historical preference	co-citation	co-keyword
BPR-MF	0.02248	0.00768	0.06669	✓		
SDNE	0.00629	0.00215	0.02145	✓		
GCMC	0.02413	0.00824	0.07120	✓		
LightGCN-1	0.03446	0.01177	0.10485	✓		
LightGCN-2	0.01949	0.00665	0.05835	✓	✓	
L-HGCF-1	0.03491	0.01192	0.10776	✓		
L-HGCF-2	0.03507	0.01198	0.11197	✓	✓	
L-HGCF-3	0.03491	0.01192	0.11136	✓		✓
L-HGCF-4	0.03507	0.01198	0.11197	✓	✓	✓

The experimental results are shown in Table 2. From these results we can have the following observations.

1): BPR-MF: BPR-MF is the only model that does not rely on graph structure for information transmission in our experiment. It's simple and effective design brings good performance. The recall@20 of BPR-MF achieves 0.02248.

2): SDNE: The performance of SDNE is poor, and its recall@20 is only 0.00629. The reason is that in SDNE every two papers marked by the same user are connected as neighbors, and the large number of generation of edges blur the relation between papers, which result in the decline of the quality of recommendation.

3): GCMC: GCMC achieved similar performance with BPR-MF(the recall@20 of GCMC is 0.02413). From the perspective of features used, both of them use only users' historical preference for papers. Structurally, GCMC uses only one-layer GCN, which failed to mine the high-order relationships between papers.

4): LightGCN-1 and LightGCN-2: The bipartite graph structure and three-layers GCN operation of LightGCN-1 lead to good results. The recall@20 of LightGCN-1 achieves 0.03446. However, when it comes to LightGCN-2, in which extra citation network is introduced to build the graph, the recall@20 achieves only 0.01949. Integrating extra citation network results in poor performance. It can be seen that the effect of bipartite graph structure in multi-feature fusion is not satisfactory.

5): L-HGCF: Due to the flexible modeling of APRHG and well-crafted L-HGCF algorithm, our method of attribute combination for hyperedge all have achieved excellent performance. APRHG modeling meets the requirements complex relationship description and multi-features fusion. The L-HGCF algorithm has a clear logic and an advantage in embedding propagation compared with GCN.

5 Ablation and Effectiveness Analysis

5.1 On the Number of Layers

Simplified-HGCN plays an important role in our L-HGCF algorithm. To explore how the number of layers of Simplified-HGCN affects the performance of our L-HGCF algorithm, we vary the depth of Simplified-HGCN in the range of $\{1, 2, 3, 4\}$. At the same time, we design two variant components to verify the beneficial of Simplified-HGCN and layer combination in our L-HGCF algorithm:

1): Complete-HGCN that use complete HGCN defined by Eq. (2) instead of simplified HGCN as Eq. (3), and we applied layer combination in this variant component.

2): Single-HGCN that use only the output of the last layer of simplified HGCN and does not use layer combination.

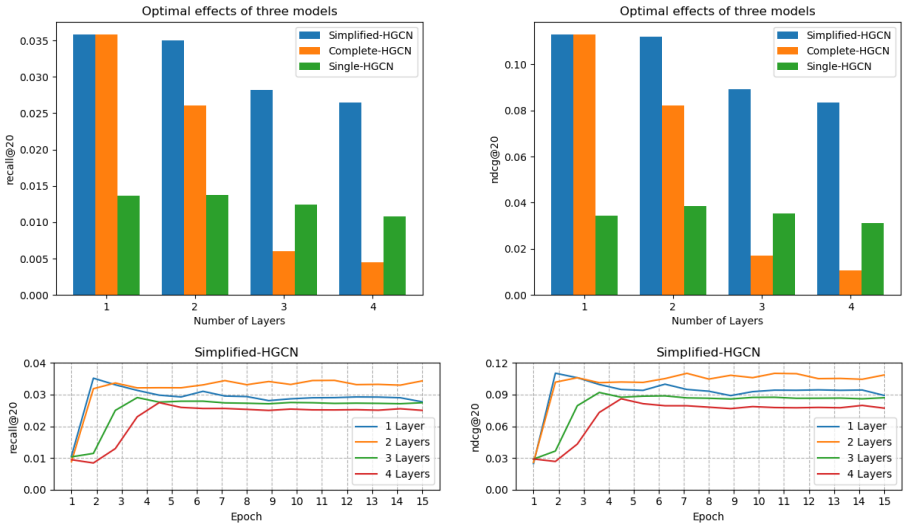

Fig. 2. (up)Optimal performance of recall@20 and ndcg@20 at different layers of Simplified-HGCN, the variant Complete-HGCN and Single-HGCN. (down) Convergence performance of recall@20 and ndcg@20 w.r.t. epoch of Simplified-HGCN at different layers

Figure 2(up) shows optimal performance of recall@20 and ndcg@20 at different layers of Simplified-HGCN, Complete-HGCN and Single-HGCN, respectively.

Focusing on Complete-HGCN, whose recall@20 declines from 0.03581 of layer 1 to 0.00450 of layer 4. With the increase of network layers, its performance continues to decline. The reason is that the increasing number of learnable parameters causes the over-fitting problem in the model.

Focusing on Single-HGCN, we find that when the number of layers increases from 1 to 4, its performance improves at the beginning, the peak point is located at layer 2 and the recall@20 is 0.01379. And after that, it decreases and drops to the worst point of layer 4, where the recall@20 is only 0.01078. This shows that the embedding of smoothing vertices using first-order and second-order neighbors is useful for recommendation, but it will suffer from over-smoothing problems when using higher-order neighbors.

Focusing on Simplified-HGCN whose recall@20 varies from 0.03581 of layer 1 to 0.02652 of layer 4, its performance is significantly better than Single-HGCN. Compared with quickly drop of Complete-HGCN, Simplified-HGCN has a much more stable and nice performance. This shows the effectiveness of layer combination for addressing over-smoothing and simplified HGCN for addressing overfitting problem.

Figure 2(down) shows the convergence curve of Simplified-HGCN at different layers. Although 1-layer Simplified-HGCN obtains an optimal performance with recall@20 is 0.03581 at the second epoch, it couldn't maintain this advantage later. In comparison, 2-layer Simplified HGCN has a much more robust and stable performance whose best recall@20 is 0.03507 at the 11th epoch. Therefore, the final number of layers of Simplified-HGCN is determined as 2.

5.2 On the aggregation schemes of user embedding

In our proposed academic paper recommendation method, we regard only papers as vertices, and an extra strategy is needed to generate user embedding. To explore how the aggregation schemes of user embedding affects the performance, we design three strategies to generate user embedding.

1): Mean-strategy: All the papers are assigned by the same weights, which is applied in our method;

2): Norm-strategy: Different papers are assigned by different weights. Suppose the weight matrix is denoted by M, paper weight can be computed by Eq. (13):

$$M = D^{-\frac{1}{2}} R B^{-\frac{1}{2}} \tag{13}$$

where R is the rating matrix, D and B are the diagonal matrices indicate degree of the users and papers, respectively.

3): Net-strategy: We take users as vertices and construct the other hypergraph model. The generated users embedding then be transferred to the L-HGCF algorithm for training together with papers embedding.

Figure 3 shows the convergence curves of recall@20 and ndcg@20 of three strategies. since the high similarity of curves of mean-strategy and norm-strategy, the details of them at epoch 4 to 15 are show in the figure in the right. It can be seen that compared with Norm-strategy, Mean-strategy has subtle advantages, and they both achieve convergence from the 3th epoch. The net-strategy has the slowest convergence speed and insufficient performance of recommendation. The convergence curves show that mean-strategy is effective, therefore we adopted the mean-strategy in our L-HGCF algorithm.

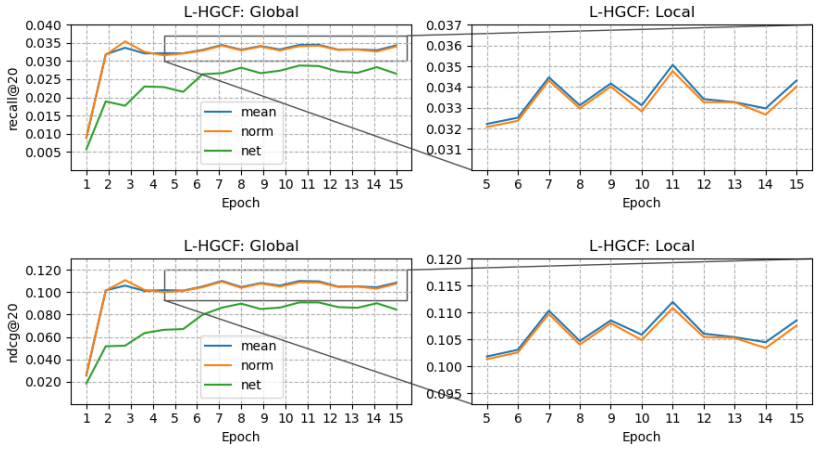

Fig. 3. Convergence performance of recall@20 and ndcg@20 w.r.t. epoch of the 2-layer L-HGCF with different choices of user aggregation schemes. the details of the mean and norm at epoch 5 to 15 are show in the figure right

6 Conclusions

In this work, we proposed a hypergraph-based academic paper recommendation method. The method is divided into two stages: APRHG construction and L-HGCF algorithm. Specifically, it allows explicitly modeling complex academic relationship and realizing multi-features fusion, and thus can yield more informative embeddings using the proposed L-HGCF algorithm, which can provide trusted recommendations. Extensive experiments on public dataset demonstrate significant improvements over competitive baselines. As shown in the experimental results, we can conclude that the hypergraph modeling and multi-features information are useful for papers representation.

Digital libraries are on the rise, and online research social platforms such as ScholarMate[2] are more and more popular, we believe that academic paper recommendation model is instructive for the future development of them. Hypergraph naturally has the ability of modeling complex relationships and multi-features fusion. Future work will focus on the construction of user portrait that can mine high-order similarity between users and improve the recommendation ability of our academic paper recommendation method.

References

1. Bai, S., Zhang, F., Torr, P.H.: Hypergraph convolution and hypergraph attention. Pattern Recognition **110**, 107637 (2021)

[2] www.scholarmate.com

2. Berg, R.v.d., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017)
3. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 7–10 (2016)
4. Fei, H., Lakdawala, S., Qi, H., Qiu, M.: Low-power, intelligent sensor hardware interface for medical data preprocessing **13**(4), 656–663 (2009)
5. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3558–3565 (2019)
6. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: 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. pp. 639–648 (2020)
7. Jiang, J., Wei, Y., Feng, Y., Cao, J., Gao, Y.: Dynamic hypergraph neural networks. In: IJCAI. pp. 2635–2641 (2019)
8. Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 659–667 (2013)
9. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI conference on artificial intelligence (2018)
10. Li, Y., Song, Y., Jia, L., Gao, S., Li, Q., Qiu, M.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE Transactions on Industrial Informatics **17**(4), 2833–2841 (2020)
11. Qiu, H., Zheng, Q., Msahli, M., Memmi, G., Qiu, M., Lu, J.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE transactions on intelligent transportation systems **22**(7), 4560–4569 (2020)
12. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
13. Samatha, B.: A novel ontological domain model for information retrieval within digital library
14. Sugiyama, K., Kan, M.Y.: Scholarly paper recommendation via user's recent research interests. In: Proceedings of the 10th annual joint conference on Digital libraries. pp. 29–38 (2010)
15. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 448–456 (2011)
16. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1225–1234 (2016)
17. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 165–174 (2019)
18. Xia, F., Liu, H., Lee, I., Cao, L.: Scientific article recommendation: Exploiting common author relations and historical preferences. IEEE Transactions on Big Data **2**(2), 101–112 (2016)
19. Yang, C., Wei, B., Wu, J., Zhang, Y., Zhang, L.: Cares: a ranking-oriented cadal recommender system. In: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries. pp. 203–212 (2009)