Adaptive Dual Channel Convolution Hypergraph Representation Learning for Technological Intellectual Property

Yuxin Liu¹, Yawen Li²*, Yingxia Shao¹, Zeli Guan¹

¹Beijing Key Laboratory of Intelligent Communication Software and Multimedia, School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876
²School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876

Abstract: In the age of big data, the demand for hidden information mining in technological intellectual property (tech-IP) is increasing in discrete countries. Definitely, a considerable number of graph learning algorithms for technological intellectual property have been proposed. The goal is to model the technological intellectual property entities and their relationships through the graph structure and use the neural network algorithm to extract the hidden structure information in the graph. However, most of the existing graph learning algorithms merely focus on the information mining of binary relations in technological intellectual property, ignoring the higher-order information hidden in non-binary relations. Therefore, a hypergraph neural network model based on dual channel convolution is proposed. For the hypergraph constructed from technological intellectual property data, the hypergraph channel and the line expanded graph channel of the hypergraph are used to learn the hypergraph, and the attention mechanism is introduced to adaptively fuse the output representations of the two channels. The proposed model outperforms the existing approaches on a variety of datasets.

Keywords: Graph Convolution; Hypergraph; Line Expansion; Attention; Representation Fusion

1 Introduction

Technological intellectual property data is generally composed of complex information like creative entities, resource entities, inter-entity relationships, and entity attributes [1]. Mainstream models usually need to map data into a vector space that is convenient for calculation [2], filter and fuse data in the limited space [7], and cannot ignore the correlation information of the data itself. The purpose of this paper is to propose a hypergraph learning algorithm to represent the hypergraph modeled the complex information about tech-IP data for further analysis and mining.

In terms of the modeling tech-IP data, existing methods include a homogeneous [12] and a heterogeneous graph [15]. However, homogeneous and heterogeneous graphs merely model binary relationships, which are not able to model non-binary relationships in tech-IP data. Consequently, graph learning algorithms based on a homogenous graph and a heterogeneous graph are unable to represent nonlinear higher-order correlation among tech-IP entities. Some researchers have proposed to use multi-agent to learn multiple features and solve the above problems through feature fusion [18], but this method requires a lot of artificial construction work. Instead, hypergraph [22] has been used as a modeling method of non-binary relations due to its structural characteristics, such as the research [23] on natural language processing, the research [24][25] on recommendation systems, the research [26] on image classification, and the research on various social networks [27]. Accordingly, using hypergraph to model the entities and relationships of tech-IP data is able to fully express its complexity.

For hypergraph learning, traditional algorithms are typically separated into two categories: one is based on incidence matrix decomposition, and the other is based on hypergraph expansion [28], which converts hypergraphs into weighted simple graphs. Neither the performance nor the efficiency of traditional methods can be compared with neural networks, so this paper mainly discusses hypergraph neural network algorithms. The neural network algorithms include hypergraph neural networks based on hypergraph expansion and hypergraph neural networks based on non-hypergraph expansion. Methods [30] based on hypergraph expansion commonly use simple graph learning methods such as the Graph Convolution Network (GCN) to represent the converted simple weighted graph from the hypergraph. However, it would cause the loss of some information for the same reason that it simplifies the hypergraph into a simple weighted graph. The methods [33] based on non-hypergraph expansion are to carry out hypergraph learning through hypergraph convolution neural networks or hypergraph attention neural networks. Some of these methods are merely appropriate for learning k-uniform graphs with k hypernodes on each hyperedge but not for the tech-IP data that hypergraph structure is a k-uniform graph. The others [36] integrate the features of hypergraph and simple graph expanded from hypergraph. Nevertheless, the method of fusion is the average of the representation of two graphs.

In view of the problems with the above methods, we propose an adaptive dual-channel hypergraph node representation algorithm (ADHCN), based on hypergraph convolution and line expansion (LE) of the hypergraph. Firstly, we construct a hypergraph based on some tech-IP data by taking the author as the hyperedge and his
publications as the hypernodes. Secondly, using the method proposed by [32] expands the hypergraph to a weighted simple graph. Thirdly, the hypergraph and the LE graph pass through the hypergraph convolution channel and the graph convolution channel, respectively. Finally, we use the attention mechanism to adaptively fuse the two outputs of dual channels. Furthermore, we use the node classification task to determine the effect of node representation.

The main contributions of this paper are as follows:

1) We propose an adaptive hypergraph dual channel convolution node representation learning method (ADHCN). The hypergraph line expansion convolution channel and the hypergraph convolution channel are used to represent nodes, so as to avoid the information loss in the simple graph, which is line expanded from the hypergraph.

2) We introduce the attention mechanism to fuse the representations of the two channels to avoid the deviation caused by direct averaging.

3) The experimental results demonstrate that ADHCN exceeds the compared models in terms of accuracy, F1, and recall. Moreover, the experiment shows that the effect of the model is significantly improved after introducing the attention mechanism for adaptive fusion.

2 Related Work

A hypergraph $G = <V, E>$ is a generalized graph, where $V$ is the set of hypernodes, $E$ is the set of hyperedges. Especially, each hyperedge connects more than two hypernodes. Moreover, the incidence matrix of the hypergraph is $H$ which elements only take 0 or 1.

For the methods in hypergraph neural networks based on expansion, Yadati et al. [30] proposed the hyperGCN model, whose principle is to use the GCN training graph transformed from the hypergraph and introduce the mediator to prevent the loss of the information.

Bandyopadhyay [30] simplified the hypergraph into line graph by taking the hyperedge as the vertices. Yang et al. [31] proposed the line expansion (LE) of hypergraphs to transform the hypergraph into a weighted simple graph $G_t = (V_t, E_t)$. The $V_t$ is the node set which is composed of hyperedge-hypernode pairs from the original hypergraph. The edge in $E_t$ connects two pairs if both pairs have a common hypernode or hyperedge. The adjacency matrix of the graph is defined by the pairwise relation $u \in V_t$ and $v \in V_t$:

$$A_t(u, v) = \begin{cases} w_u u = (v_h, e_h), v = (v_h', e_h'), v = v_h' & 1 \\ w_u u = (v_h, e_h), v = (v_h', e_h'), e_h = e_h' & 0 \\ \end{cases}$$

The above algorithm simplifies tech-IP hypergraph learning by using simple graph learning, but they have the problem of losing the hypergraph structure information. For the methods in hypergraph neural networks based on non-expansion, Feng et al. [33] proposed a hypergraph convolution neural network by extending graph convolution to hypergraph. Owing to KNN being the method of constructing hypergraphs, this model is merely appropriate for k-uniform hypergraphs. In order to address the issue that the hypergraph structure in [33] changes as a result of the alteration in node representation. Jiang et al. [34] proposed a dynamic hypergraph neural network (DHGNN) that contains dynamic hypergraph reconstruction that reconstructs the hypergraph at each layer. However, the method is incapable of solving the k-uniform graph problem. Bai et al. [35] proposed a hypergraph convolution theory from another perspective, which is consistent with the hypergraph convolution proposed by [33]. Xia et al. [36] proposed a dual channel hypergraph convolution network to solve the problem of session-based recommendation. Especially, the model uses hypergraph convolution and line graph convolution to learn hypergraph and fuse the learning result. Nevertheless, there are still some problems in the model, such as information loss in the line graph channel and the simplicity of the fusion method.

![Figure 1 ADHCN model structure](image-url)
3 ADHCN: Adaptive Dual Channel Hypergraph Convolution Network

Figure 1 shows the model structure proposed in this paper. It mainly includes the hypergraph construction and the line expansion (HCLE) module, the dual channel convolution (DHC) module, and the adaptive representation fusion (ARF) module.

Firstly, the tech-IP dataset is constructed into a hypergraph \( H \) and a line-expanded simple graph \( G_t \) through the HCLE module. Secondly, DHC module learns two constructed graphs to obtain two channel outputs. Thirdly, the ARF module fuses the outputs of the two channels to obtain the final representation.

3.1 HCLE: Hypergraph Construction and Line Expansion

ADHCN constructs entities and relationships in tech-IP data as hypergraphs \( G_h = \langle V, E \rangle \), in which publications are viewed as hypernodes \( V \) and authors as hyperedges \( E \). The incidence matrix \( H \in \{0,1\}^{V \times |E|} \) with \( H_{ij} = 1 \) is defined by the relationship between an author and his publications. And the initial representations \( X_h \) of the publications are the text embeddings of their abstracts.

ADHCN adopts the line expansion of the hypergraph as the input of another convolution channel. The line expansion method of the hypergraph of tech-IP data is to treat the author-publication pair as nodes to obtain a weighted simple graph \( G_t \). And, there is one edge between the into two pairs if either two authors create the same publication or one author creates two publications.

The initial input of the hypergraph convolution channel \( X_h \) is converted to initial input \( X_t \) of the LE channel according to the hypernode projection matrix \( P_e \) proposed in [32]. The construction of the hypernode projection matrix is shown in Eq.2.

\[
P_e(v, v) = \begin{cases} 1 & v_1 = (v, e), \exists e \in E \\ 0 & \text{otherwise} \end{cases} \tag{2}
\]

3.2 DHC: Dual Channel Convolution Module

The dual channel convolution module is divided into two channels of convolution, one is line expansion (LE) convolution and the other is hypergraph convolution.

ADHCN adopts the convolution form proposed by GCN as the calculation method of the LE convolution channel. Accordingly, the output of the LE convolution channel is \( \bar{Z}_l \).

\[
\bar{Z}_l = \sigma (D^{-1/2}A_lD^{-1/2}X_l \theta_1) \tag{3}
\]

where \( \sigma(\cdot) \) is the nonlinear activation function, \( D^{-1/2} \in R^{[V \times |V|]} \) is the expanded graph node degree matrix, \( \bar{A}_l = A_l + 2I \in R^{[V \times |V|]} \) is the adjacency matrix with the adjustment by adding two-orders of self-loop, \( \theta_1 \in R^{d \times k} \) is the channel parameters to be learned.

To obtain the final output \( Z_l \) of the LE channel, the representation \( \bar{Z}_l \) needs to be mapped back to the hypergraph using the hypernode reflection matrix. The construction of hypernode reflection matrix is shown in Eq.4.

\[
P_v = \begin{cases} \frac{1}{\sigma(e)} & v_1 = (v, e), \exists e \in E \\ 0 & \text{otherwise} \end{cases} \tag{4}
\]

where \( \sigma(e) \) is the degree of the hyperedge \( e \).

ADHCN adopts the hypergraph convolution form proposed in HGNN [33] as the calculation rule of the hypergraph convolution channel. Accordingly, the output of the hypergraph convolution channel is \( Z_h \).

\[
Z_h = \sigma(D_h^{-1/2}HWD_h^{-1}H^T D_h^{-1/2}X_h \theta_h) \tag{5}
\]

where \( \sigma(\cdot) \) is the nonlinear activation function, \( H \in R^{[P \times |E|]} \) is the hypergraph incidence matrix, \( W \in R^{[E \times |E|]} \) is the hyperedge weight matrix, \( D_h^{-1/2} \in R^{[P \times |P|]} \) is the hypernode degree matrix, \( D_h^{-1} \in R^{[E \times |E|]} \) is the hyperedge degree matrix, \( \theta_h \in R^{d \times h} \) is the parameters to be learned.

3.3 ARF: Adaptive Representation Fusion Module

The ARF module is mainly used to fuse the node representations learned by the two channels. The main principle is adaptive fusion through attention. We use the attention mechanism to calculate the weight of the two channel outputs. The calculation process of the weights and the final output is shown in Eq.6 and Eq.7.

\[
W = [\omega_1, \omega_2, \omega_3] = \text{Attention} ([Z_l, Z_h, Z_c]) \tag{6}
\]

\[
Z_{out} = \omega_1 Z_l + \omega_2 Z_h + \omega_3 Z_c \tag{7}
\]

where \( \text{Attention}(\cdot) \) is the attention function, \( Z_l \) and \( Z_h \) are the outputs of the DHC module and \( Z_c \) is the output of the LE channel.

We use hypernode classification task to guide model training and verify model performance. The classification loss is computed using the cross-entropy loss function, as is typical for most models.

4 Experiments

4.1 Experimental Settings

We evaluate the quality of ADHCN representation on four tech-IP datasets through the node classification task. The four datasets are Dblp [37], Citeseer [38], Cora [39], and patentDB. The last built by this paper consists of Chinese patents and their applicants or inventors, in which the applicants are the hypernodes and the patents are the hyperedges. The applicant categories are companies and individuals. In order to evaluate the model, we use classification accuracy, macro-F1, and recall as the evaluation index. In this study, accuracy, macro-F1, and recall are denoted by the letters Acc, F1, and R, respectively.

4.2 ADHCN Validation

We utilized the methodology specified in the experimental design for comparison design to confirm the
The efficacy of ADHCN. The experimental results are shown in Table I.

| Model       | Cora (Acc) | Cora (R) | Cora (F1) | Citeeseer (Acc) | Citeeseer (R) | Citeeseer (F1) | Dbp (Acc) | Dbp (R) | Dbp (F1) | PatentDB (Acc) | PatentDB (R) | PatentDB (F1) |
|-------------|------------|----------|-----------|----------------|--------------|----------------|-----------|--------|---------|----------------|--------------|--------------|
| 1-hyperGCN  | 0.601      | 0.386    | 0.391     | 0.679          | 0.637        | 0.635          | 0.789     | 0.778  | 0.778   | 0.500          | 0.462        |               |
| fasthyperGCN| 0.626      | 0.423    | 0.449     | 0.675          | 0.636        | 0.635          | 0.815     | 0.806  | 0.805   | 0.803          | 0.498        | 0.465        |
| hyperGCN    | 0.635      | 0.442    | 0.471     | 0.681          | 0.638        | 0.637          | 0.822     | 0.811  | 0.813   | 0.815          | 0.501        | 0.465        |
| LE-GCN[32]  | 0.646      | 0.464    | 0.503     | 0.702          | 0.649        | 0.643          | 0.859     | 0.851  | 0.853   | 0.849          | 0.509        | 0.501        |
| HGNN[33]    | 0.639      | 0.451    | 0.494     | 0.680          | 0.639        | 0.634          | 0.830     | 0.820  | 0.823   | 0.845          | 0.506        | 0.501        |
| ADHCN(our)  | 0.646      | 0.443    | 0.487     | 0.706          | 0.650        | 0.647          | 0.873     | 0.862  | 0.867   | 0.852          | 0.513        | 0.503        |

4.3 Adaptive Representation Fusion Module Validation

In order to verify the effectiveness of the adaptive fusion module, we compare the performance of ADHCN and the model that obtains the fused feature by Z_f = Z_h + αZ_f (α is set to 0.1, 0.5, 1.0, 0.09, 0.09, 0.5, 0.7 and 0.9 respectively). Table II displays the experiment’s findings across four datasets.

We adopted the adaptive representation fusion performed better than the model with the fusion method of weighted summation. Consequently, the adaptive representation fusion module can adaptively adjust the fusion of the two channels to obtain better model performance.

5 Conclusions

In order to make full use of the high-order information in the tech-IP data, we used hypergraph to model the tech-IP entities and their relationships, and proposed an adaptive dual channel hypergraph convolution model to learn the constructed hypergraph, in which dual channels are used to prevent information loss caused by hypergraph expansion, and an attention mechanism is introduced to realize adaptive fusion between the two channels. Experimental results demonstrate that the suggested approach outperforms other comparable algorithms, and the proposed adaptive fusion mechanism improves the performance of the model.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.62192784, No.62172056).

References

[1] Ang Li, Junping Du, Feifei Kou, Zhe Xue, Xin Xu, Mingyong Xu, Yang Jiang. Scientific and Technological Information Oriented Semantics-adversarial and Media-adversarial Cross-media Retrieval. arXiv preprint arXiv:2203.08615, 2022.
[2] Wenling Li, Yingmin Jia, Junping Du. Distributed consensus extended Kalman filter: a variance-constrained approach. IET Control Theory & Applications, 11(3): 382-389, 2017.
[3] Yawen Li, Isabella Yunfei Zeng, Ziheng Niu, Jiahao Shi, Ziyang Wang and Zeli Guan. Predicting vehicle fuel consumption based on multi-view deep neural network, Neurocomputing, 502:140-147, 2022.
[4] Wenling Li, Yingmin Jia, and Junping Du. Distributed extended Kalman filter with time-correlated multiplicative measurement noise. IET Control Theory & Applications, 2016, 11(1): 122-128.
[5] Wenling Li, Jian Sun, Yingmin Jia, Junping Du, and Xiaoyan Fu. Variance-constrained state estimation for nonlinear complex networks with uncertain coupling strength. Digital Signal Processing, 2017, 67: 107-115.
[6] Feifei Kou, Junping Du, Congxian Yang, Yansong Shi, Wancu Cui, Meiyu Liang, and Yue Geng. Hashtag recommendation based on multi-features of microblogs. Journal of Computer Science and Technology, 2018, 33(4): 711-726.
[7] Xinlei Wei, Junping Du, Meiyu Liang, and Lingfei Ye. Boosting deep attribute learning via support vector regression for fast moving crowd counting. Pattern Recognition Letters, 2019, 119: 12-23.
[8] Qingping Li, Junping Du, Fuzhao Song, Chao Wang,
