

Article

Hypernetwork Representation Learning Based on Hyperedge Modeling

Yu Zhu ¹ , Haixing Zhao ^{2,*}, Xiaoying Wang ¹  and Jianqiang Huang ¹¹ Department of Computer Technology and Applications, Qinghai University, Xining 810000, China² State Key Laboratory of Tibetan Intelligent Information Processing and Application, Qinghai Normal University, Xining 810000, China

* Correspondence: h.x.zhao@163.com

Abstract: Most network representation learning approaches only consider the pairwise relationships between the nodes in ordinary networks but do not consider the tuple relationships, namely the hyperedges, among the nodes in the hypernetworks. Therefore, to solve the above issue, a hypernetwork representation learning approach based on hyperedge modeling, abbreviated as HRHM, is proposed, which fully considers the hyperedges to obtain ideal node representation vectors that are applied to downstream machine learning tasks such as node classification, link prediction, community detection, and so on. Experimental results on the hypernetwork datasets show that with regard to the node classification task, the mean node classification accuracy of HRHM approach goes beyond other best baseline approach by about 1% on the MovieLens and wordnet, and with regard to the link prediction task, except for HPHG approach, the mean AUC value of HRHM approach surpasses that of other baseline approaches by about 17%, 18%, and 6%, respectively, on the GPS, drug, and wordnet. The mean AUC value of HRHM approach is very close to that of other best baseline approach on the MovieLens.

Keywords: representation learning; pairwise relation; tuple relationships; hyperedge modeling



check for updates

Citation: Zhu, Y.; Zhao, H.; Wang, X.; Huang, J. Hypernetwork Representation Learning Based on Hyperedge Modeling. *Symmetry* **2022**, *14*, 2584. <https://doi.org/10.3390/sym14122584>

Academic Editors: João Ruivo Paulo, Cristina P. Santos and Gabriel Pires

Received: 9 November 2022

Accepted: 4 December 2022

Published: 7 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of artificial intelligence, hypernetwork representation learning has gradually become a research hotspot in the field of machine learning. Hypernetwork representation learning maps the nodes in the hypernetwork to a low-dimensional vector representation space. The learned node representation vectors can be applied to node classification [1], link prediction [2], community detection [3], and so on.

According to the types of the hypernetwork, the hypernetwork representation learning can be divided into homogeneous hypernetwork representation learning and heterogeneous hypernetwork representation learning. Homogeneous hypernetwork representation learning aims to map the nodes of a single type in the homogeneous hypernetwork to a low-dimensional vector representation space. For example, Zhou [4] proposed the hypergraph embedding approach on the basis of the spectral hypergraph clustering [5], but the high computational complexity of this approach limits the wide application. HGNN [6] extended the convolution operation to hypergraph embedding, but the datasets used in this approach are not really hypernetwork datasets. In a word, although the above homogeneous hypernetwork representation learning approaches have good representation learning abilities, they do not consider the heterogeneity of the hypernetwork. Therefore, the researchers propose heterogeneous hypernetwork representation learning approaches, which aim to learn distinguishing representation vectors for different types of nodes in the heterogeneous hypernetwork. For example, HHNE [7] is designed as a fully connected graph convolution layer to project different types of nodes into a common low-dimensional vector representation space, but the computational complexity of this approach is too high

to be suitable for large-scale heterogeneous networks. DHNE [8] realizes the local and global proximity of nonlinear tuple similar functions in the embedding space, but because the multi-layer perceptron is used in the DHNE approach, the approach is limited to heterogeneous hyperedges with fixed size, and the relationships among multi-type instances with unfixed size cannot be considered.

To sum up, although the above hypernetwork representation learning methods can obtain nice node representation vectors, there are various issues, especially high computational complexity and the limitation of the hyperedges with a fixed size. Therefore, to solve the above issues, a hypernetwork representation learning approach based on hyperedge modeling to effectively capture complex tuple relationships (i.e., hyperedges) among the nodes is proposed, which is suitable for the hypernetwork with the hyperedges with unfixed size and improves the computational efficiency.

The following two aspects are the features of this paper:

- A hypernetwork representation learning approach based on hyperedge modeling is proposed to map the nodes in the hypernetwork to a low-dimensional vector representation space, where the main components of the learned node representation vectors are the hypernetwork structure and the hyperedges;
- The advantage of HRHM approach is that the hyperedges with unfixed size are encoded in the learned node representation vectors. The disadvantage of HRHM approach is that the partial information of the hypernetwork structure is lost because the hypernetwork abstracted as the hypergraph is transformed into the ordinary network abstracted as two-section graph.

2. Related Works

Nowadays, researchers have proposed some hypernetwork representation learning approaches to obtain node representation vectors that are rich in hypernetwork structure information. The existing hypernetwork representation learning approaches can be divided into homogeneous hypernetwork representation learning approaches and heterogeneous hypernetwork representation learning approaches. With regard to homogeneous hypernetwork representation learning approaches, Zhou [4] proposed the hypergraph embedding approach on the basis of the spectral hypergraph clustering. HyperGCN [9] approximates each hyperedge of a hypergraph by a set of pairwise edges connecting the nodes in the hyperedge and treats the hypergraph learning as graph learning. HGNN [6] extends the convolution operation to hypergraph embedding, which is convolved by the hypergraph Laplacian function in the spectral domain and further approximated by truncated Chebyshev polynomials. LHCN [10] maps the hypergraph to a weighted attributed line graph and learns graph convolutions on this line graph. With regard to heterogeneous hypernetwork representation learning approaches, HHNE [7] is designed as a fully connected graph convolutional layer to project different types of nodes into a common low-dimensional space and uses a tuple similarity function to protect the network structure, and a rank-based loss function is used to improve the similarity scores of hyperedges in the embedding space. DHNE [8] is a new deep model to realize the local and global proximity of the nonlinear tuple similarity function in the embedding space. HPHG [11] is a deep model called Hyper-gram to capture pairwise and tuple relationships in the node sequences. Hyper-SAGNN [12] utilizes a self-attention mechanism [13] to aggregate hypergraph information.

3. Problem Definition of HRHM Approach

The hypernetwork $H = (V, E)$ abstracted as the hypergraph consists of the node set $V = \{v_i\}_{i=1}^{|V|}$ and the hyperedge set $E = \{e_i = \{v_1, v_2, \dots, v_\tau\}\}_{i=1}^{|E|}$ ($\tau \geq 2$). The HRHM approach aims to learn a low-dimensional representation vector $r_n \in \mathbf{R}^k$ for each node n in the hypernetwork, where k is much smaller than $|V|$.

In order to understand the process of hypernetwork representation learning well, take the drug hypernetwork as an example, where the triplet $\langle \text{user}, \text{drug}, \text{adverse reaction} \rangle$ is

the hyperedge. Since each drug has several specific adverse reactions, there is a semantic relevance between the drug and the adverse reaction. How to assess the above semantic relevance is the real-life problem, which is solved by hypernetwork representation learning, which obtains a node representation vector to calculate the similarity between the drug and the adverse reaction to assess the semantic relevance.

4. Preliminaries

4.1. Two-Section Graph Transformed from Hypergraph

According to the literature [14], the two-section graph structure is closer to the hypergraph structure than the line graph and the incidence graph. Therefore, the two-section graph transformed from the hypergraph is used in this paper to carry out the research of hypernetwork representation learning. The hypergraph and its corresponding two-section graph are shown in Figure 1.

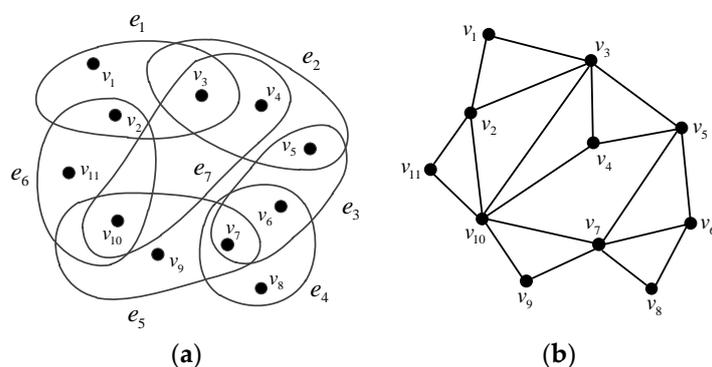


Figure 1. Hypergraph and two-section graph. (a) Hypergraph; (b) two-section graph.

The two-section graph $S = (V', E')$ transformed from the hypergraph $H = (V, E)$ is an ordinary graph to meet the following conditions:

- The node set V' of two-section graph S is identical with the node set V of the hypergraph H ;
- Any two different nodes are associated with one edge if and only if these two nodes belong to at least one hyperedge simultaneously.

4.2. TransE Model

TransE [15] is a knowledge representation model with the translation mechanism, which thinks that if the head entity h and the tail entity t are with the relationship r , the triplet (h, r, t) holds. Moreover, the head entity vector h plus the relationship vector r is almost identical with the tail entity vector t , namely $h + r \approx t$, when the triplet (h, r, t) holds; otherwise, $h + r \neq t$. The TransE framework is shown in Figure 2.

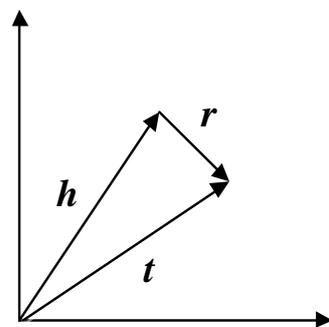


Figure 2. TransE framework, where $h + r \approx t$.

5. HRHM Approach

This section introduces hypernetwork representation learning based on hyperedge modeling in more detail. Firstly, the cognitive topology model is introduced in Section 5.1.

Secondly, the cognitive hyperedge model is introduced in Section 5.2. Finally, the joint optimization of the above two models is introduced in more detail in Section 5.3.

5.1. Cognitive Topology Model

In order to improve the computational efficiency, a cognitive topology model with the negative sampling [16] is introduced to capture the hypernetwork structure. Under the condition of the node sequences C , we try to maximize the following target function of the cognitive topology model to obtain the representation vectors rich in the hypernetwork structure.

$$D_1 = \prod_{n \in C} \prod_{u \in \{n\} \cup NEG_1(n)} p(u|context(n)) \quad (1)$$

where $NEG_1(n)$ is the subset of negative samples of the center node n , regarded as the positive sample. $context(n)$ is the contextual nodes of the center node n . $p(u|context(n))$ is defined as follows.

$$p(u|context(n)) = \begin{cases} \sigma(X_n^T \theta_u), & L^n(u) = 1 \\ 1 - \sigma(X_n^T \theta_u), & L^n(u) = 0 \end{cases} \quad (2)$$

where $\sigma(X_n^T \theta_u) = 1/(1 + e^{-X_n^T \theta_u})$ is a sigmoid function. X_n is the sum vector of all nodes representation vectors corresponding to $context(n)$. θ_n is the parameter vector. For $\forall u \in V$, the node label $L^n(u)$ is defined as follows.

$$L^n(u) = \begin{cases} 1, & u \in \{n\} \\ 0, & u \in NEG_1(n) \end{cases} \quad (3)$$

By means of Equation (3), Equation (2) can also be written as an integral expression.

$$p(u|context(n)) = [\sigma(X_n^T \theta_u)]^{L^n(u)} \cdot [1 - \sigma(X_n^T \theta_u)]^{1-L^n(u)} \quad (4)$$

By substituting Equation (4) into Equation (1), Equation (1) can be rewritten as follows:

$$D_1 = \prod_{n \in C} \prod_{u \in \{n\} \cup NEG_1(n)} \left\{ [\sigma(X_n^T \theta_u)]^{L^n(u)} \cdot [1 - \sigma(X_n^T \theta_u)]^{1-L^n(u)} \right\} \quad (5)$$

Formally, maximizing the target function D_1 makes the learned node representation vectors rich in hypernetwork structure.

5.2. Cognitive Hyperedge Model

Because the qualities of the node representation vectors from the above cognitive topology model do not consider that the hyperedges are not high, a novel cognitive hyperedge model with the negative sampling is proposed to consider the hyperedges to learn node representation vectors of high quality, where the hyperedges are deemed as the interaction relationships among the nodes and regarded as the translation operations in the representation vector space in the TransE model.

Under the condition of the hyperedge constraint, we try to maximize the following target function of the cognitive hyperedge model to obtain the representation vectors rich in the hyperedges.

$$D_2 = \prod_{n \in C} \prod_{r \in R_n} \prod_{h \in H_r} \prod_{\xi \in \{n\} \cup NEG_2(n)} p(\xi|h+r) = \prod_{n \in C} \prod_{r \in R_n} \prod_{h \in H_r} \prod_{\xi \in \{n\} \cup NEG_2(n)} \left\{ \sigma_{(h+r)}^T \theta_{\xi}^{\delta^n(\xi)} \cdot [1 - \sigma_{(h+r)}^T \theta_{\xi}]^{1-\delta^n(\xi)} \right\} \quad (6)$$

where R_n is the set of the hyperedges, namely relationships associated with the center node n ; H_r is the set of the nodes with the relation r with the center node n ; $NEG_2(n)$ is the subset of negative samples of the center node n ; θ_{ξ} is the parameter vector; and the parameter vector e_{h+r} is the sum vector of the parameter vectors e_h and e_r , namely $e_{h+r} = e_h + e_r$.

For $\forall \xi \in V$, the node label $\delta^n(\xi)$ in the Equation (6) is defined as follows:

$$\delta^n(\xi) = \begin{cases} 1, & \xi \in \{n\} \\ 0, & \xi \in NEG_2(n) \end{cases} \tag{7}$$

Formally, maximizing the target function D_2 makes the learned node representation vectors rich in hyperedges.

5.3. Joint Optimization

In this subsection, the hypernetwork representation learning approach based on hyperedge modeling, abbreviated as HRHM, is proposed. The HRHM approach can jointly optimize the cognitive topology model and cognitive hyperedge model. Figure 3 shows the HRHM framework.

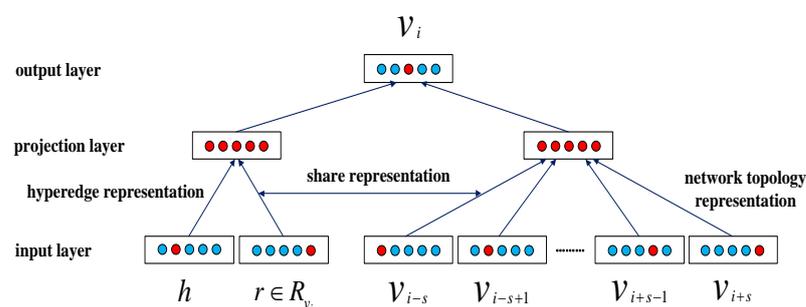


Figure 3. HRHM framework, where v_i is the center node, and other nodes v_{i-s} , v_{i-s+1} , v_{i+s-1} , v_{i+s} , etc. are contextual nodes of the center node v_i , namely $context(v_i)$, where the vectors corresponding to v_{i-s} , v_{i-s+1} , v_i , v_{i+s-1} and v_{i+s} are denoted as the first red dot from the left, the second red dot from the left, the red dot in the middle, the second red dot from the right, the first red dot from the right. r is the interaction relation namely the hyperedge; h is a node with the relationship r with the center node v_i ; and R_{v_i} is the hyperedge set associated with the center node v_i , where the vectors corresponding to r and h are denoted as the first red dot from the right and the second red dot from the left. The left-hand rectangle denoted as all red is the projection layer representation derived from r and h ; The right-hand rectangle denoted as all red is the projection layer representation derived from v_{i-s} , v_{i-s+1} , v_{i+s-1} and v_{i+s} .

From Figure 3, the network topology representation and hyperedge representation from the cognitive topology model and cognitive hyperedge model, respectively, share the same representation.

For ease of calculation, take the logarithm of D_1 and D_2 to maximize the following joint optimization target function to make the hyperedges fully incorporated into the node representation vectors.

$$L = \sum_{n \in C} \left\{ \begin{aligned} & \sum_{u \in \{n\} \cup NEG_1(n)} \{L^n(u) \cdot \log[\sigma(X_n^T \theta_u)] + [1 - L^n(u)] \cdot \log[1 - \sigma(X_n^T \theta_u)]\} + \\ & \beta \cdot \sum_{r \in R_n} \sum_{h \in H_r} \sum_{\xi \in \{n\} \cup NEG_2(n)} \{ \delta^n(\xi) \cdot \log[\sigma(e_{h+r}^T \theta_\xi)] + [1 - \delta^n(\xi)] \cdot \log[1 - \sigma(e_{h+r}^T \theta_\xi)] \} \end{aligned} \right\} \tag{8}$$

$$= \sum_{n \in C} \left\{ \begin{aligned} & \sum_{u \in \{n\} \cup NEG_1(n)} \{L^n(u) \cdot \log[\sigma(X_n^T \theta_u)] + [1 - L^n(u)] \cdot \log[1 - \sigma(X_n^T \theta_u)]\} + \\ & \sum_{r \in R_n} \sum_{h \in H_r} \sum_{\xi \in \{n\} \cup NEG_2(n)} \beta \cdot \{ \delta^n(\xi) \cdot \log[\sigma(e_{h+r}^T \theta_\xi)] + [1 - \delta^n(\xi)] \cdot \log[1 - \sigma(e_{h+r}^T \theta_\xi)] \} \end{aligned} \right\}$$

where the harmonic factor β is used to equilibrate the contribution rate between the cognitive topology model and cognitive hyperedge model.

For ease of derivation, $L(n, u, r, h, \xi)$ is denoted as follows:

$$L(n, u, r, h, \xi) = \{L^n(u) \cdot \log[\sigma(X_n^T \theta_u)] + [1 - L^n(u)] \cdot \log[1 - \sigma(X_n^T \theta_u)]\} + \beta \cdot \left\{ \delta^n(\xi) \cdot \log[\sigma(e_{h+r}^T \theta_\xi)] + [1 - \delta^n(\xi)] \cdot \log[1 - \sigma(e_{h+r}^T \theta_\xi)] \right\} \tag{9}$$

The stochastic gradient ascent approach is used to optimize the target function L . The acquisition of four kinds of gradients of L is the key to this paper.

Firstly, the gradient on the parameter vector θ_u is calculated as follows:

$$\begin{aligned} \frac{\partial L(n, u, r, h, \xi)}{\partial \theta_u} &= L^n(u) \cdot [1 - \sigma(X_n^T \theta_u)] \cdot X_n - [1 - L^n(u)] \cdot \sigma(X_n^T \theta_u) \cdot X_n \\ &= \{L^n(u) \cdot [1 - \sigma(X_n^T \theta_u)] - [1 - L^n(u)] \cdot \sigma(X_n^T \theta_u)\} \cdot X_n \\ &= [L^n(u) - \sigma(X_n^T \theta_u)] \cdot X_n \end{aligned} \tag{10}$$

Therefore, the parameter vector θ_u is updated as follows:

$$\theta_u = \theta_u + \alpha \cdot [L^n(u) - \sigma(X_n^T \theta_u)] \cdot X_n \tag{11}$$

where α is the learning rate.

Secondly, the gradient on the sum vector X_n is calculated as follows via the symmetry property between θ_u and X_n :

$$\frac{\partial L(n, u, r, h, \xi)}{\partial X_n} = [L^n(u) - \sigma(X_n^T \theta_u)] \cdot \theta_u \tag{12}$$

Therefore, the representation vector $v_{v'}$ is updated as follows, where $v' \in context(n)$:

$$\begin{aligned} v_{v'} &= v_{v'} + \alpha \cdot \sum_{u \in \{n\} \cup NEG_1(n)} \frac{\partial L(n, u, r, h, \xi)}{\partial X_n} \\ &= v_{v'} + \alpha \cdot \sum_{u \in \{n\} \cup NEG_1(n)} [L^n(u) - \sigma(X_n^T \theta_u)] \cdot \theta_u \end{aligned} \tag{13}$$

Thirdly, the gradient on the parameter vector θ_ξ is calculated as follows:

$$\begin{aligned} \frac{\partial L(n, u, r, h, \xi)}{\partial \theta_\xi} &= \beta \cdot \left\{ \frac{\partial}{\partial \theta_\xi} \left\{ \delta^n(\xi) \cdot \log[\sigma(e_{h+r}^T \theta_\xi)] + [1 - \delta^n(\xi)] \cdot \log[1 - \sigma(e_{h+r}^T \theta_\xi)] \right\} \right\} \\ &= \beta \cdot \left\{ \delta^n(\xi) \cdot [1 - \sigma(e_{h+r}^T \theta_\xi)] \cdot e_{h+r} - [1 - \delta^n(\xi)] \cdot \sigma(e_{h+r}^T \theta_\xi) \cdot e_{h+r} \right\} \\ &= \beta \cdot \left\{ \left\{ \delta^n(\xi) \cdot [1 - \sigma(e_{h+r}^T \theta_\xi)] - [1 - \delta^n(\xi)] \cdot \sigma(e_{h+r}^T \theta_\xi) \right\} \cdot e_{h+r} \right\} \\ &= \beta \cdot [\delta^n(\xi) - \sigma(e_{h+r}^T \theta_\xi)] \cdot e_{h+r} \end{aligned} \tag{14}$$

Therefore, the parameter vector θ_ξ is updated as follows:

$$\theta_\xi = \theta_\xi + \alpha \cdot \beta \cdot [\delta^n(\xi) - \sigma(e_{h+r}^T \theta_\xi)] \cdot e_{h+r} \tag{15}$$

Finally, the gradient on the parameter vector e_{h+r} is calculated as follows via the symmetry property between θ_ξ and e_{h+r} :

$$\frac{\partial L(n, u, r, h, \xi)}{\partial e_{h+r}} = \beta \cdot [\delta^n(\xi) - \sigma(e_{h+r}^T \theta_\xi)] \cdot \theta_\xi \tag{16}$$

Specially, $e_{h+r} = e_h + e_r$, and the gradient $\frac{\partial L(n, u, r, h, \xi)}{\partial e_{h+r}}$ is used to update the parameter vectors e_h and e_r , respectively, as follows:

$$e_h = e_h + \alpha \cdot \beta \cdot \sum_{\xi \in \{n\} \cup NEG_2(n)} [\delta^n(\xi) - \sigma(e_{h+r}^T \theta_\xi)] \cdot \theta_\xi \tag{17}$$

$$e_r = e_r + \alpha \cdot \beta \cdot \sum_{\xi \in \{\{n\} \cup \text{NEG}_2(n)\}} [\delta^n(\xi) - \sigma(e_{h+r}^T \theta_\xi)] \cdot \theta_\xi \quad (18)$$

The stochastic gradient ascent approach is used for optimization. More details are shown in Algorithm 1.

Algorithm 1: HRHM

```

1  Input:
2      hypernetwork  $H = (V, E)$ 
3      vector dimension size  $d$ 
4  Output:
5      node representation matrix  $Y \in \mathbf{R}^{|V| \times d}$ 
6  for node  $n$  in  $V$  do
7      initializing the representation vector  $v_n \in \mathbf{R}^{1 \times d}$ 
8      initializing the parameter vector  $\theta_n \in \mathbf{R}^{1 \times d}$ 
9      for hyperedge  $r$  in  $R_n$  do
10         for node  $h$  in  $H_r$  do
11             initializing the parameter vector  $e_{h+r} \in \mathbf{R}^{1 \times d}$ 
12         end for
13     end for
14 end for
15 node sequences  $C = \text{randomwalk}()$ 
16 for  $(n, \text{context}(n))$  in  $C$  do
17     updating the parameter vector according to the Equation (11)
18     updating the representation vector according to the Equation (13)
19     updating the parameter vector according to the Equation (15)
20     for hyperedge  $r$  in  $R_n$  do
21         for node  $h$  in  $H_r$  do
22             updating the parameter vector according to the Equation (17)
23             updating the parameter vector according to the Equation (18)
24         end for
25     end for
26 end for
27 for  $i = 0; i < |V|; i++$  do
28      $Y_i = v_n$ 
29 end for
30 return  $Y$ 

```

6. Experiments and Result Analysis

6.1. Datasets

Four hypernetwork datasets are used to evaluate HRHM approach. Detailed dataset statistics are shown in Table 1.

Table 1. Dataset statistics.

Dataset		Node Type		#(V)		#(E)	
GPS	user	location	activity	146	70	5	1436
MovieLens	user	movie	tag	457	1688	1530	5965
drug	user	drug	reaction	4	132	221	1195
wordnet	head	relation	tail	1754	7	1549	2174

Four datasets are shown as follows:

- GPS [17] describes a situation where a user takes part in an activity in a location. The triplet $\langle \text{user}, \text{location}, \text{activity} \rangle$ is utilized to construct the hypernetwork;
- MovieLens [18] describes personal tag activities from MovieLens. The triplet $\langle \text{user}, \text{movie}, \text{tag} \rangle$ is utilized to construct the hypernetwork, where each movie has at least one genre;

- Drug (<http://www.fda.gov/Drugs/>, accessed on 27 January 2020) describes a situation where the user who takes some drugs has certain reactions that lead to adverse events. The triplet < user, drug, reaction> is utilized to construct the hypernetwork;
- Wordnet [15] is made up of a set of triplets <head, relation, tail> extracted from Wordnet3.0. The triplet < head, relation, tail> is utilized to construct the hypernetwork.

6.2. Baseline Approaches

DeepWalk: DeepWalk [19] is a popular approach for learning node representation vectors to encode social relationships.

Node2vec: Node2vec [20] maps the nodes in the network to a low-dimensional representation space to preserve network neighborhoods of the nodes.

LINE: LINE [21] embeds huge networks into low-dimensional vector spaces to preserve both local and global network structures.

GraRep: GraRep [22] integrates global graph structural information into the process of representation learning.

HOPE: HOPE [23] is a learning approach that preserves higher-order proximities of large scale graphs and captures the asymmetric transitivity.

SDNE: SDNE [24] is a semi-supervised learning approach with multiple layers of non-linear functions to capture the highly non-linear network structure.

HPGH: HPHG [11] proposes a deep model called Hyper-gram to capture pairwise and tuple relationships in the node sequences.

HRHM: HRHM regards the interaction relationships among the nodes as the translation operation in the representation space and incorporates the relationships among the nodes into node representation vectors.

6.3. Node Classification

Because the labels are only on the MovieLens and wordnet, our approach is assessed via node classification [1] on the two datasets. The node classification accuracies are calculated by means of SVM [25].

The observations from Tables 2 and 3 are shown as follows:

- With regard to the two datasets, the mean node classification accuracy of HRHM approach surpasses that of other baseline approaches, and in terms of the node classification accuracies with different training ratios, HRHM approach surpasses other baseline approaches. Furthermore, the node classification accuracies of HRHM approach is directly proportional to the training ratios, which shows that a large amount of training data is helpful for node classification. It is worth noting that the node classification accuracies of HRHM approach are not high. The reason is that the categorical attributes on the two datasets are less prominent;
- The mean node classification accuracy of DeepWalk ranks only second to that of HRHM approach because DeepWalk captures the hypernetwork structure to a certain extent in the node sequences generated by random walks.

Table 2. Node classification results on MovieLens (%).

Approaches	Training Ratios										Mean	Rank
	10%	20%	30%	40%	50%	60%	70%	80%	90%			
DeepWalk	48.01	50.35	51.41	52.60	52.59	53.47	53.57	54.23	54.09	52.26	2	
node2vec	46.93	49.28	50.77	51.51	52.62	52.58	53.04	53.44	52.71	51.43	4	
LINE	43.93	45.46	46.52	47.29	47.70	48.16	48.02	49.09	48.34	47.17	6	
GraRep	47.75	50.11	51.16	52.01	52.10	53.15	53.34	53.43	53.24	51.81	3	
HOPE	46.33	48.57	49.95	50.69	51.06	51.04	51.29	52.53	51.79	50.36	5	
SDNE	41.74	41.79	42.34	42.73	43.36	43.27	43.89	43.43	42.81	42.82	7	
HRHM	48.73	51.26	52.71	53.62	54.38	54.57	55.03	55.82	56.30	53.60	1	

Table 3. Node classification results on wordnet (%).

Approaches	Training Ratios									Mean	Rank
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
DeepWalk	29.91	33.44	34.53	35.05	35.70	36.80	37.93	36.71	39.00	35.45	2
node2vec	29.27	32.23	33.71	34.52	36.17	36.05	37.53	37.66	37.30	34.94	4
LINE	22.77	24.11	25.11	24.94	25.23	25.59	25.87	26.60	25.44	25.07	6
GraRep	32.59	34.74	34.63	35.21	35.38	36.05	35.10	36.63	37.79	35.35	3
HOPE	30.53	33.61	35.02	35.97	34.90	35.11	36.21	36.20	34.84	34.71	5
SDNE	21.96	21.57	22.05	22.37	23.26	22.59	23.63	23.60	25.31	22.93	7
HRHM	31.30	33.79	35.59	36.18	36.95	37.94	38.14	38.63	40.07	36.51	1

In short, it is found that the node representation vectors from the HRHM approach are better than other baseline approaches, which shows that HRHM approach is effective.

6.4. Link Prediction

The link prediction is assessed by the AUC [26]. The observations from Tables 4–7 are shown as follows.

Table 4. Link prediction results on GPS.

Approaches	Training Ratios							Mean	Rank
	60%	65%	70%	75%	80%	85%	90%		
DeepWalk	0.4308	0.4278	0.4205	0.4583	0.4418	0.4914	0.4831	0.4505	4
node2vec	0.3660	0.3614	0.3808	0.3939	0.3834	0.3958	0.3649	0.3780	7
LINE	0.4575	0.4829	0.4761	0.4562	0.4429	0.4663	0.4574	0.4628	3
GraRep	0.3873	0.3805	0.3882	0.3765	0.3820	0.3857	0.3874	0.3839	6
HOPE	0.3805	0.3676	0.3416	0.2971	0.2794	0.2518	0.2334	0.3073	8
SDNE	0.3262	0.4371	0.4319	0.3157	0.4379	0.3527	0.4540	0.3936	5
HPHG	0.9026	0.9158	0.9142	0.9269	0.9347	0.9326	0.9315	0.9226	1
HRHM	0.6845	0.6428	0.6483	0.6403	0.6216	0.6005	0.5856	0.6319	2

Table 5. Link prediction results on MovieLens.

Approaches	Training Ratios							Mean	Rank
	60%	65%	70%	75%	80%	85%	90%		
DeepWalk	0.7845	0.8129	0.8301	0.8440	0.8729	0.8800	0.9025	0.8467	2
node2vec	0.7078	0.7390	0.7418	0.7696	0.7939	0.8036	0.8296	0.7693	6
LINE	0.8282	0.8242	0.8253	0.8320	0.8365	0.8172	0.8231	0.8266	4
GraRep	0.7290	0.7833	0.7907	0.8121	0.8277	0.8481	0.8544	0.8065	5
HOPE	0.6895	0.7333	0.7203	0.7522	0.7787	0.7986	0.8049	0.7539	7
SDNE	0.4004	0.3511	0.3494	0.3406	0.3433	0.3598	0.4171	0.3660	8
HPHG	0.9356	0.9367	0.9388	0.9276	0.9138	0.9245	0.9301	0.9296	1
HRHM	0.8495	0.8497	0.8351	0.8325	0.8387	0.8281	0.8320	0.8379	3

Table 6. Link prediction results on drug.

Approaches	Training Ratios							Mean	Rank
	60%	65%	70%	75%	80%	85%	90%		
DeepWalk	0.4852	0.4954	0.4934	0.4580	0.4901	0.4638	0.4713	0.4796	5
node2vec	0.4500	0.4525	0.4490	0.4525	0.4329	0.4712	0.4345	0.4489	7
LINE	0.4750	0.4672	0.4636	0.4625	0.4741	0.4523	0.4768	0.4674	6
GraRep	0.5025	0.5089	0.4867	0.5051	0.5557	0.5835	0.5362	0.5255	3
HOPE	0.5055	0.5269	0.4933	0.4690	0.4941	0.4668	0.4271	0.4832	4
SDNE	0.2948	0.4310	0.4454	0.5050	0.5196	0.3536	0.3836	0.4190	8
HPHG	0.9451	0.9458	0.9467	0.9552	0.9583	0.9548	0.9489	0.9507	1
HRHM	0.7153	0.7071	0.7134	0.7134	0.6868	0.7108	0.7240	0.7101	2

Table 7. Link prediction results on wordnet.

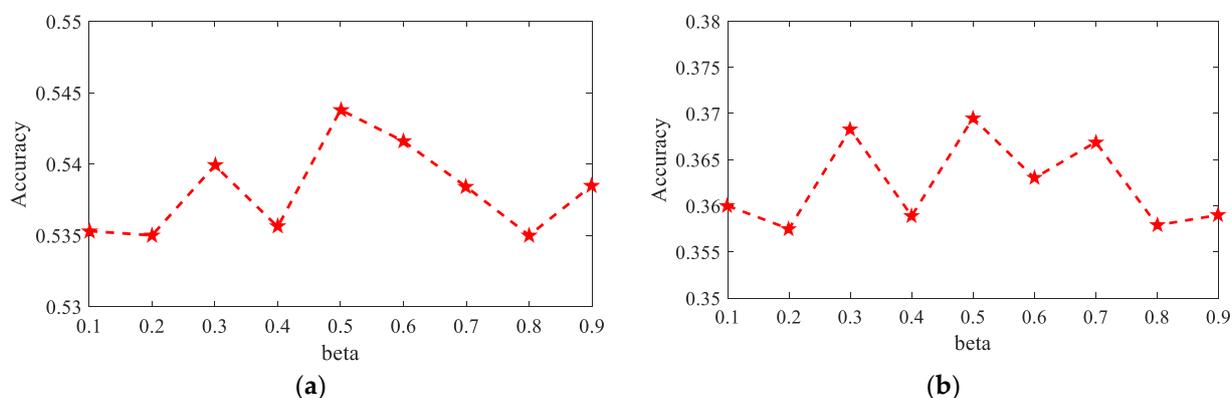
Approaches	Training Ratios							Mean	Rank
	60%	65%	70%	75%	80%	85%	90%		
DeepWalk	0.7780	0.8181	0.8305	0.8341	0.8708	0.8765	0.8880	0.8423	3
node2vec	0.7807	0.8242	0.8309	0.8285	0.8519	0.8503	0.8595	0.8323	4
LINE	0.8063	0.8184	0.8056	0.8091	0.8000	0.7938	0.7926	0.8037	5
GraRep	0.7685	0.7742	0.7888	0.7806	0.7958	0.7972	0.7756	0.7830	6
HOPE	0.6902	0.7314	0.7417	0.7403	0.7649	0.7763	0.7700	0.7450	7
SDNE	0.3712	0.5348	0.4784	0.4824	0.4254	0.6159	0.4850	0.4847	8
HPHG	0.9217	0.9378	0.9386	0.9488	0.9556	0.9495	0.9501	0.9432	1
HRHM	0.9030	0.9115	0.9050	0.9016	0.9098	0.9027	0.8912	0.9035	2

- The HRHM approach performs worse than HPHG approach. To be specific, the mean AUC value of HPHG approach goes beyond that of HRHM approach by about 29%, 9%, 24%, and 4%, respectively, for the GPS, MovieLens, drug, and wordnet. The reason is that HRHM approach transforms the hypergraph into a two-section graph, which leads to partial loss of the hypernetwork structure information, but the HPHG approach does not decompose the hyperedges, which leads to almost complete preservation of hypernetwork structure information;
- Except for HPHG approach, the mean AUC value of HRHM approach surpasses that of other baseline approaches on the GPS, drug, and wordnet. The mean AUC value of HRHM approach is very close to that of the other best baseline approach DeepWalk on the MovieLens. To sum up, HRHM approach surpasses most baseline approaches, which shows that the HRHM approach is effective;
- The HRHM approach performs consistently at different training ratios compared to other baseline approaches, which shows its feasibility and robustness;
- The HRHM approach almost surpasses other baseline approaches that do not consider the hyperedges, which verifies the assumption that the hyperedges are useful for link prediction.

In short, the above observations show that the HRHM approach can learn node representation vectors of high quality.

6.5. Parameter Sensitivity

The contribution rate between the cognitive topology model and the cognitive hyper-edge model is equilibrated by the harmonic factor β . We set the raining ratio to 50% and calculate node classification accuracies with different β within the ranges from 0.1 to 0.9 on MovieLens and wordnet. Figure 4 shows the comparisons of node classification accuracies.

**Figure 4.** Parameter sensitivity. (a) Sensitivity on MovieLens. (b) Sensitivity on wordnet.

From Figure 4, it is found that the variation ranges of node classification accuracies on the two datasets are both within 2%, which indicates that the HRHM approach is not sensitive to the parameter β and shows the robustness of the HRHM approach.

In short, the best node classification results on both MovieLens and wordnet datasets are achieved at $\beta = 0.5$.

7. Conclusions

This hypernetwork representation learning approach based on hyperedge modeling consists of the cognitive topology model and the cognitive hyperedge model, which incorporate the hypernetwork topology structure and the hyperedges into node representation vectors, respectively, where the learning process of node representation vectors is regarded as a joint optimization problem, which is resolved via the stochastic gradient ascend approach. The advantage of the HRHM approach is that the hyperedges with unfixed size are encoded in the learned node representation vectors. The experimental results show that the performance of the HRHM approach is almost all better than that of other baseline approaches, more or less, except for the HPHG approach. In future research, we will transform the hypergraph into the ordinary graph but regard the hyperedges as a whole to carry out the research of the hypernetwork representation learning.

Author Contributions: Conceptualization, Y.Z. and H.Z.; methodology, Y.Z. and H.Z.; software, Y.Z.; validation, Y.Z.; formal analysis, Y.Z.; investigation, Y.Z. and H.Z.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z. and H.Z.; visualization, Y.Z.; supervision, Y.Z. and H.Z.; project administration, Y.Z. and H.Z.; funding acquisition, Y.Z., X.W. and J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by National Natural Science Foundation of China, grant number 62166032, grant number 62162053, and grant number 62062059; by Natural Science Foundation of Qinghai Province, grant number 2022-ZJ-961Q; by the Project from Tsinghua University, grant number SKL-IOW-2020TC2004-01; and by the Open Project of State Key Laboratory of Plateau Ecology and Agriculture, Qinghai University, grant number 2020-ZZ-03.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pedroche, F.; Tortosa, L.; Vicent, J.F. An eigenvector centrality for multiplex networks with data. *Symmetry* **2019**, *11*, 763. [[CrossRef](#)]
2. Papageorgiou, I.; Bittner, D.; Psychogios, M.N.; Hadjidemetriou, S. Brain immunoinformatics: A symmetrical link between informatics, wet lab and the clinic. *Symmetry* **2021**, *13*, 2168. [[CrossRef](#)]
3. Guerrero, M.; Banos, R.; Gil, C.; Montoya, F.G.; Alcayde, A. Evolutionary algorithms for community detection in continental-scale high-voltage transmission grids. *Symmetry* **2019**, *11*, 1472. [[CrossRef](#)]
4. Zhou, D.Y.; Huang, J.Y.; Schölkopf, B. Learning with hypergraphs: Clustering, classification and embedding. In Proceedings of the 19th International Conference on Neural Information Processing Systems, Vancouver, Canada, 4–7 December 2006; pp. 1601–1608.
5. Sharma, K.K.; Seal, A.; Herrera-Viedma, E.; Krejcar, O. An enhanced spectral clustering algorithm with s-distance. *Symmetry* **2021**, *13*, 596. [[CrossRef](#)]
6. Feng, Y.F.; You, H.X.; Zhang, Z.Z.; Ji, R.R.; Gao, Y. Hypergraph neural networks. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3558–3565.
7. Baytas, I.M.; Xiao, C.; Wang, F.; Jain, A.K.; Zhou, J.Y. Heterogeneous hyper-network embedding. In Proceedings of the 18th IEEE International Conference on Data Mining, Singapore, 17–20 November 2018; pp. 875–880.
8. Tu, K.; Cui, P.; Wang, X.; Wang, F.; Zhu, W.W. Structural deep embedding for hyper-networks. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 426–433.
9. Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; Talukdar, P. HyperGCN: A new approach of training graph convolutional networks on hypergraphs. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
10. Bandyopadhyay, S.; Das, K.; Murty, M.N. Line hypergraph convolution network: Applying graph convolution for hypergraphs. *arXiv* **2020**, arXiv:2002.03392.

11. Huang, J.; Liu, X.; Song, Y.Q. Hyper-path-based representation learning for hyper-networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 449–458.
12. Zhang, R.C.; Zou, Y.S.; Ma, J. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. *arXiv* **2019**, arXiv:1911.02613.
13. Song, G.; Li, J.W.; Wang, Z. Occluded offline handwritten chinese character inpainting via generative adversarial network and self-attention mechanism. *Neurocomputing* **2020**, *415*, 146–156. [[CrossRef](#)]
14. Zhu, Y.; Zhao, H.X. Hypernetwork representation learning with the set constraint. *Appl. Sci.* **2022**, *12*, 2650. [[CrossRef](#)]
15. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
16. Zhu, Y.; Ye, Z.L.; Zhao, H.X.; Zhang, K. Text-enhanced network representation learning. *Front. Comput. Sci.* **2020**, *14*, 146322. [[CrossRef](#)]
17. Zheng, V.W.; Cao, B.; Zheng, Y.; Xie, X.; Yang, Q. Collaborative filtering meets mobile recommendation: A user-centered approach. In Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; pp. 236–241.
18. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 19. [[CrossRef](#)]
19. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
20. Grover, A.; Leskovec, J. Node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
21. Tang, J.; Qu, M.; Wang, M.Z.; Zhang, M.; Yan, J.; Mei, Q.Z. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
22. Cao, S.S.; Lu, W.; Xu, Q.K. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 891–900.
23. Ou, M.D.; Cui, P.; Pei, J.; Zhang, Z.W.; Zhu, W.W. Asymmetric transitivity preserving graph embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1105–1114.
24. Wang, D.X.; Cui, P.; Zhu, W.W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234.
25. Gholamnia, K.; Nachappa, T.G.; Ghorbanzadeh, O.; Blaschke, T. Comparisons of diverse machine learning approaches for wildfire susceptibility mapping. *Symmetry* **2020**, *12*, 604. [[CrossRef](#)]
26. Almardeny, Y.; Boujnah, N.; Cleary, F. A novel outlier detection method for multivariate data. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 4052–4062. [[CrossRef](#)]