

Article

Hypernetwork Representation Learning with the Set Constraint

Yu Zhu ¹  and Haixing Zhao ^{2,*}

¹ Department of Computer Technology and Application, Qinghai University, Xining 810000, China; zhuyu@qhu.edu.cn

² State Key Laboratory of Tibetan Intelligent Information Processing and Application, Qinghai Normal University, Xining 810000, China

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

Abstract: There are lots of situations that cannot be described by traditional networks but can be described perfectly by the hypernetwork in the real world. Different from the traditional network, the hypernetwork structure is more complex and poses a great challenge to existing network representation learning methods. Therefore, in order to overcome the challenge of the hypernetwork structure faced by network representation learning, this paper proposes a hypernetwork representation learning method with the set constraint abbreviated as HRSC, which incorporates the hyperedge set associated with the nodes into the process of hypernetwork representation learning to obtain node representation vectors including the hypernetwork topology structure and hyperedge information. Our proposed method is extensively evaluated by the machine learning tasks on four hypernetwork datasets. Experimental results demonstrate that HRSC outperforms other best baseline methods by about 1% on the MovieLens and wordnet datasets in terms of node classification, and outperforms the other best baseline methods, respectively, on average by about 29.03%, 1.94%, 26.27% and 6.24% on the GPS, MovieLens, drug, and wordnet datasets in terms of link prediction.

Keywords: representation learning; hypernetwork structure; hyperedge information; set constraint



Citation: Zhu, Y.; Zhao, H.

Hypernetwork Representation Learning with the Set Constraint. *Appl. Sci.* **2022**, *12*, 2650. <https://doi.org/10.3390/app12052650>

Academic Editor: Melih Kandemir

Received: 31 January 2022

Accepted: 28 February 2022

Published: 4 March 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

Networks are ubiquitous in our daily life, and many real-life applications focus on mining valuable information from networks. A basic issue of data mining is how to learn ideal node representation vectors. To cope with this issue, network representation learning has been proposed to learn a low-dimensional representation vector for each node in the network, which can be applied to lots of machine learning tasks such as node classification [1], link prediction [2], and community detection [3].

Most of the related works learn node representation vectors only from the network topology structure, such as Deepwalk [4], node2vec [5], LINE [6], SDNE [7], and HARP [8]. The basic assumption of these topology-based representation learning methods is that the nodes with similar topological contexts should be tightly distributed in the low-dimensional vector representation space. However, in many real scenarios, the vectors learned only from the network topology structure are not desirable vectors. For example, in the social network, if two users share some common tags or topics, they are likely to be similar but topologically far apart. Thus, topology-based network representation learning methods cannot effectively capture their similarity. In such a case, other types of heterogeneous information should be incorporated to learn node representation vectors of better quality. Therefore, the researchers propose some representation learning methods that utilize both the network topology and heterogeneous information to learn node representation vectors, such as CANE [9], CNRL [10], and PPNE [11].

Nevertheless, most existing network representation learning methods are designed for traditional networks with pairwise relationships, wherein each edge only links one pair of nodes. However, the relationships among the data objects in the real world are much more

complex and often not necessarily pairwise. For example, it forms a high-order relation $\langle \text{James, movie, war} \rangle$ that James watches a war movie. The network that captures the high-order relationship is often referred to as the hypernetwork. In view of the above facts, the researchers extend the spectral clustering technique [12] to hypergraphs [13] to propose a series of methods [14–16]. However, these methods mainly aim at homogeneous hypernetworks, which construct the hyperedges through potential similarities such as common tags. As for the heterogeneous hypernetwork, tensor decomposition [17,18] can be directly applied to learn node representation vectors, but the time complexity of tensor decomposition is often too high to effectively scale to large-scale hypernetworks. HGNN [19] is a hypergraph neural network model for data representation learning. However, the datasets to evaluate this model is not real hypernetwork datasets, but traditional network datasets. HPHG [20] proposes a deep model named as hyper-gram to learn low-dimensional representation vectors, but the hyperedge is not sufficiently considered in this model so that the hypernetwork topology structure cannot be well preserved. DHNE [21] provides a solution for modeling hyperedges directly without hyperedge decomposition. However, due to the neural network structure used in this method, this method is limited to heterogeneous hyperedges of fixed type and fixed size, and the relationship between multiple types of objects of unfixed size cannot be considered.

In view of the above facts, this paper proposes a universal hypernetwork representation learning framework, which can effectively encode both the hypernetwork topology structure and hyperedge information (which is handled by solving the following challenge). It is not easy how to combine the hypernetwork topology structure and the hyperedges well into a unified representation learning process under a universal framework. Since there are complex interactions between the hypernetwork topology structure and the hyperedges, it is difficult to incorporate the hyperedges into the existing topology-based network representation learning methods.

To deal with the above challenge, this paper proposes a universal hypernetwork representation learning method with the set constraint HRSC to effectively incorporate the hyperedges into the process of hypernetwork representation learning, which is formulated as a joint optimization problem solved by the stochastic gradient ascent (SGA) algorithm. To be more specific, a negative sampling method is utilized to capture the hypernetwork topology structure, which aims to exploit the correlations between the nodes by maximizing the prediction probabilities of the center nodes given their contextual nodes in the random walk node sequences. Besides, the negative sampling method is still utilized to capture semantic correlations between the center nodes and their associated hyperedges by maximizing the prediction probabilities of the center nodes given the hyperedge set associated with the center nodes.

The contribution of this paper mainly includes the following three aspects:

- The hypernetwork is transformed into a traditional network abstracted as a two-section graph approximating the hypernetwork topology structure. Then, on the basis of this traditional network, a hypernetwork representation learning method HRSC is proposed to learn node representation vectors by comprehensively utilizing both the hypernetwork topology structure and the hyperedges.
- HRSC incorporates the set of the hyperedges associated with the nodes into the process of hypernetwork representation learning, regarding the learning process of node representation vectors as a joint optimization problem.
- Efficient and scalable learning algorithm for HRSC has been developed. Our learning algorithm is able to handle large-scale hypernetworks efficiently.

The rest of this paper is organized as follows. In Section 2, we review related works. Section 3 formally defines our studied problem. Section 4 introduces the preliminaries. In Section 5, we introduce the hypernetwork representation learning method with the set constraint in detail. The experimental results are given in Section 6. Finally, we conclude this paper in Section 7.

2. Related Works

Network representation learning aims to learn a low-dimensional vector for each node in the network, preserving the structure and inherent characteristics of the network. More and more researchers have focused on network representation learning in recent years. For example, Deepwalk [4] introduces Skip-Gram [22], a widely-used distributed word representation method, into the study of the network to learn node representation vectors. node2vec [5] maximally preserves the network neighborhoods of the nodes through a biased random walk strategy, and maps the network nodes to a low-dimensional feature representation space. LINE [6], which can be easily extended to a network of millions of nodes and billions of edges, carefully designs an objective function that preserves both first-order and second-order proximities. SDNE [7] captures highly nonlinear network structures and utilizes first-order and second-order proximities to characterize local and global network structures. HARP [8] is proposed to learn low-dimensional representations of a graph's nodes to preserve high-order structural features by compressing the input graph prior to graph embedding. However, the above-mentioned network representation learning methods only utilize the network structure to learn the node representation vectors, and do not consider heterogeneous information associated with the nodes, such as text, community or label information of the nodes. To cope with this issue, researchers have incorporated heterogeneous information associated with the nodes into the topology-based network representation learning methods. For example, CANE [9] learns the context-aware representation vectors of the nodes with mutual attention mechanism, and models semantic relations between the nodes. CNRL [10] utilizes the similarities between the text theme and the network community to incorporate community information, a key network analysis factor, into the process of network representation learning, which simultaneously detects the community distribution of each node and learns the representation vectors of the nodes and the communities. PPNE [11] effectively incorporates the network topology and attribute information of the nodes into the process of network representation learning.

Nevertheless, the network representation learning methods listed here only consider the pairwise relationships between the nodes in the traditional network. However, in the real world, the complex interaction relationships among no less than two nodes need to be considered sometimes, and the complex relationships can be modeled by the hypernetwork. Unfortunately, the above network representation learning methods are not appropriate for the hypernetwork, since these methods do not fully consider the hypernetwork topology structure.

So far, only a few hypernetwork representation learning methods have been proposed. For example, Zhou [14] extends the powerful method of spectral clustering that is originally run on undirected graphs to the hypergraph, and further develops the hypergraph embedding algorithm on the basis of the spectral hypergraph clustering method. Liu [15] extends the traditional spectral hashing to the hypergraph and accelerates the similarity search of social images by mapping semantically related nodes to similar binary codes within a short hamming distance. Wu [16] utilizes two hypergraph views to represent text-related and theme-related images and proposes a spectral multi-hypergraph clustering algorithm to classify the images. HGNN [19] a hypergraph neural network model for data representation learning. HPHG [20] proposes a deep model named as hyper-gram to learn low-dimensional representation vectors. DHNE [21] is proposed to learn low-dimensional representation vectors of the hypernetwork with non-decomposable hyperedges. HHNE [23] is a depth model based on graph convolutional network, which incorporates the features of the nodes into the process of network representation learning. However, although all of the above hypernetwork representation learning methods have good representation learning ability, there are various problems. In order to deal with these problems, this paper proposes a universal hypernetwork representation learning method with the set constraint to learn discriminative node representation vectors.

Table 1 shows the statistics of the pros and cons for different network representation learning methods.

Table 1. The statistics of the pros and cons.

Network Type	Methods	Pros	Cons
traditional network	Deepwalk	to introduce Skip-Gram into network representation learning firstly	random walk without constraints
	LINE	to preserve first-order and second-order proximities	direct concatenation of first-order vector and second-order vector
	SDNE	to capture highly nonlinear network structures	not applicable to some networks
	CANE	to learn context-aware representation vectors of the nodes with mutual attention mechanism	no inter-relation between structure-based vector and text-based vector
hypernetwork	HPHG	to preserve hypernetwork topology structure by a random walk based on hyper-path	not to consider hyperedges sufficiently
	DHNE	a solution to model hyperedges without hyperedge decomposition	limited to heterogeneous hyperedges of fixed type and fixed size
	HRSC	to consider hyperedges sufficiently	to lose partial hypernetwork structure information

3. Problem Definition

This section formally defines our studied problem. The hypernetwork is usually abstracted as the hypergraph $H = (V, E)$ with the set of nodes belonging to T types $V = \{v_i\}_{i=1}^{|V|} = \{V_t\}_{t=1}^T$ and the set of hyperedges which can be associated with no less than two nodes $E = \{e_i = (v_1, v_2, \dots, v_m)\}_{i=1}^{|E|}$ ($m \geq 2$). If the number of nodes is two for each hyperedge, the hypernetwork degenerates to a traditional network. If $T \geq 2$, the hypernetwork is defined as a heterogeneous hypernetwork. Here the problem of hypernetwork representation learning with the set constraint is formally defined as follows.

Definition 1. Given a hypernetwork $H = (V, E)$, the problem of hypernetwork representation learning with the set constraint aims to learn a low-dimensional vector $r_n \in R^k$ for each node n in the hypernetwork, where k is expected to be much smaller than $|V|$, making the learned node representation vectors explicitly preserve both the hypernetwork topology structure and the hyperedges.

4. Preliminaries

Since the study on traditional graphs is relatively mature, it can be regarded as a desirable way to study the hypergraph to deepen the understanding of the corresponding hypergraph by the study on the traditional graph (if the hypergraph can be transformed into the traditional graph). In the literature [13], it is proposed that the hypergraph can be transformed into three types of traditional graphs, namely a line graph, two-section graph, and incidence graph, where the line graph has only the nodes converted from the hyperedges and no nodes are associated with the hyperedges (which indicates that the relationship between the nodes associated with the hyperedges is lost, leading to a large loss of the hypernetwork topology structure). The incidence graph introduces the nodes converted from the hyperedges to weaken the strong correlation relationships between the nodes associated with the hyperedges, which leads to a certain amount of loss of hypernetwork topology structure. The two-section graph makes all of the nodes

associated with each hyperedge joined into a complete graph, which leads to less loss of hypernetwork topology structure than other two graphs. Therefore, a two-section graph that approximates the hypernetwork topology structure has the characteristic of preserving the hypergraph topology structure better than other transformation strategies. Hence, in this paper, the hypergraph is transformed into a two-section graph, and then the traditional network topology structure abstracted as two-section graph and the hyperedges in the hypernetwork abstracted as the hypergraph are comprehensively utilized to carry out the study of the hypernetwork representation learning. The detailed strategy for transforming the hypergraph into a two-section graph is as follows.

Given a hypergraph $H = (V, E)$, two-section graph $S = (V', E')$ corresponding to the hypergraph H is a traditional graph that meets the following conditions:

- $V' = V$, that is, the node set of 2-section graph S is the same as the node set of the hypergraph H .
- One edge is linked between any two different nodes if and only if they are simultaneously contained in at least one hyperedge (that is, all of the nodes in each hyperedge are joined into a complete graph).

A hypergraph and two-section graph of its transformation are shown in Figure 1.

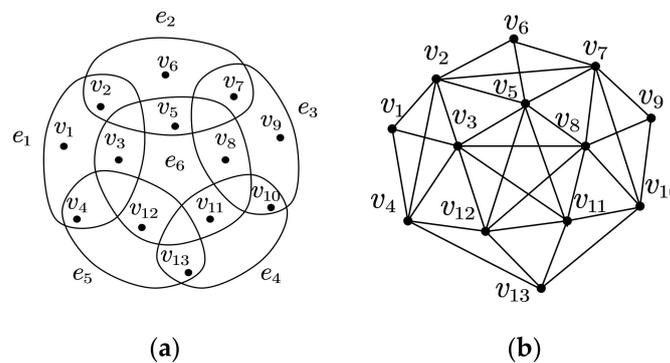


Figure 1. The hypergraph and two-section graph of its transformation. (a) Hypergraph and (b) two-section graph.

5. Our Method

In this section, we introduce the details of hypernetwork representation learning method with the set constraint. Firstly, Section 5.1 gives the introduction of topology-derived objective function. Secondly, Section 5.2 introduces the set constraint objective function. Thirdly, HRSC based on the joint optimization of the above two objective functions is introduced in details in Section 5.3. Finally, Section 5.4 introduces the complexity analysis of HRSC.

5.1. Topology-Derived Objective Function

Continuous Bag-of-Words [22], abbreviated as CBOW, is a popular distributed word representation method whose computational efficiency is higher than that of Skip-Gram. Hence, in order to improve the computational efficiency of node representation vectors, following the idea of CBOW, with such an assumption that the nodes with similar topology context tend to be similar, we aimed to maximize the prediction probabilities of the center node given its contextual nodes, which are defined as the previous nodes and after nodes of the center node with a fixed-size window in the node sequences generated by random walk. We propose a novel negative sampling-based topology-derived objective function in which the node representation vectors are regarded as the parameters. In the optimization procedure, the representation vectors are updated and the finally learned vectors preserve the hypernetwork topology structure.

A set of node sequences C is generated by the same random walk on the traditional network abstracted as 2-section graph as Deepwalk. For each node, a node sequence of

length t to start from this node is generated in the random walk iteration, which will repeat τ times to generate enough sequences.

Under the condition of a center node n and its contextual nodes $context(n)$, the node n is regarded as the positive sample, other nodes are the negative samples, and $NEG(n)$ is the subset of negative samples of the center node n with a predefined size ds . For $\forall u \in V$, the labels of the node are defined as follows:

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

$p(u|context(n))$ denotes the prediction probability of the node u under the condition of the contextual nodes $context(n)$. On the basis of the node sequences C , we try to maximize the following objective function.

$$D_1 = \prod_{n \in C} \prod_{u \in \{n\} \cup NEG(n)} p(u|context(n)) \quad (2)$$

As for each node n , this paper designs the embedding vector and the parameter vector, where the embedding vector v_n is the representation of the node n when it is treated as the contextual node, while the parameter vector θ_n is the representation of n when it is treated as the center node. $p(u|context(n))$ in Formula (2) is defined as follows.

$$p(u|context(n)) = \begin{cases} \sigma(X_n^T \theta_u), & L^n(u) = 1 \\ 1 - \sigma(X_n^T \theta_u), & L^n(u) = 0 \end{cases} \quad (3)$$

where $\sigma(X_n^T \theta_u) = \frac{1}{1 + e^{-X_n^T \theta_u}}$ is a sigmoid function, where X_n is the summing operation of the representation vectors corresponding to $context(n)$. Formula (3) can also be rewritten as an integral expression.

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

Accordingly, Formula (2) can be rewritten as follows.

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

Formally, maximizing D_1 means maximizing the prediction probabilities of positive samples and minimizing the prediction probabilities of negative samples simultaneously, by which the hypernetwork topology structure is encoded into the node representation vectors.

5.2. Set Constraint Objective Function

The model based on the above topology-derived objective function only takes the traditional network topology structure, abstracted as on a two-section graph approximating the hypernetwork topology structure as input to learn node representation vectors, but ignores the relations among the nodes (namely hyperedges). Hence, it cannot learn representation vectors of the nodes associated with the hyperedges well. In order to learn node representation vectors better, the set of the hyperedges associated with the nodes is incorporated into the process of hypernetwork representation learning.

Inspired by the above topology-derived objective function, this paper proposes a novel negative sampling-based set constraint objective function, where T_n denotes the set of the hyperedges associated with the center node $n \in V$, and also the set of the nodes associated with the center node $n \in V$ if the hyperedge is regarded as the node. The center node n is regarded as the positive sample, and other nodes that are not associated with the center node $n \in V$ are regarded as the negative samples. For $\forall v \in T_n$, $NEG(v)$ is the

subset of negative samples with a predefined size ds , and the labels of the node are defined as follows:

$$\delta(\vartheta|v) = \begin{cases} 1, & \vartheta \in \{n\} \\ 0, & \vartheta \in NEG(v) \end{cases} \tag{6}$$

On the basis of the set constraint and the node sequences C , we aim to maximize the following objective function to force the embedding vectors to satisfy the extracted constraints.

$$\begin{aligned} D_2 &= \prod_{n \in C} \prod_{v \in T_n} p(n|v) = \prod_{n \in C} \prod_{v \in T_n} \prod_{\vartheta \in \{\{n\} \cup NEG(v)\}} \left\{ \sigma(e_v^T \theta_\vartheta)^{\delta(\vartheta|v)} \cdot [1 - \sigma(e_v^T \theta_\vartheta)]^{1-\delta(\vartheta|v)} \right\} \\ &= \prod_{n \in C} \prod_{v \in T_n} \left\{ \sigma(e_v^T \theta_n) \cdot \prod_{\vartheta \in NEG(v)} [1 - \sigma(e_v^T \theta_\vartheta)] \right\} \end{aligned} \tag{7}$$

where e_v is the parameter vector corresponding to $v \in T_n$.

By maximizing D_2 , the hyperedges is encoded into node representation vectors.

5.3. Joint Optimization Objective Function

A hypernetwork representation learning method with the set constraint HRSC, which aims to jointly optimize the topology-derived objective function and set constraint objective function, is proposed in this subsection. Figure 2 indicates HRSC framework.

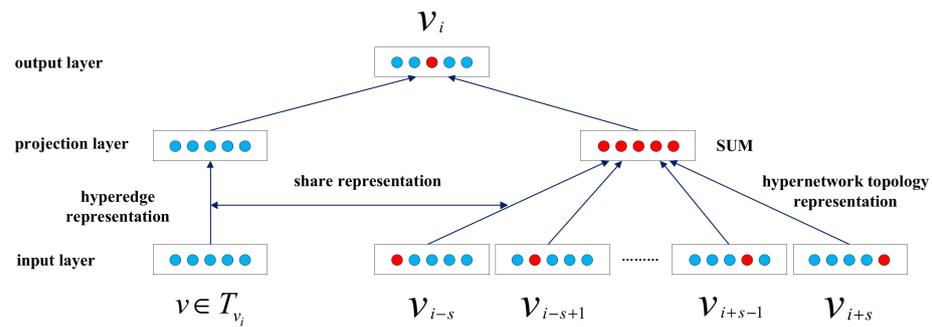


Figure 2. HRSC framework, where v_i is the center node, other nodes $v_{i-s}, v_{i-s+1}, v_{i+s-1}, v_{i+s}$, etc. are contextual nodes of the center node v_i , namely $context(v_i)$, where the vectors corresponding to $v_{i-s}, v_{i-s+1}, v_i, v_{i+s-1}, v_{i+s}$ are denoted as the first red dot from 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, SUM denoted as all red is the summation of all of the vectors corresponding to $context(v_i)$, and T_{v_i} is the set of hyperedges associated with the center node v_i , and also the set of the nodes associated with the center node v_i if the hyperedge is regarded as the node.

Compared to other hypernetwork representation learning methods, HRSC is improved at two levels: (I) at the hypernetwork topology structure level, HRSC exploits the correlations between the nodes by maximizing the prediction probabilities of the center nodes given their contextual nodes in random walk node sequences; (II) at the node-hyperedge level, HRSC captures semantic correlations between the center nodes and their associated hyperedges by maximizing the prediction probabilities of the center nodes given the set of the hyperedges associated with the center nodes. By means of the above improvements, HRSC effectively incorporates the hyperedges into the process of hypernetwork representation learning to get node representation vectors of better quality.

As shown in Figure 2, the hypernetwork topology representation and hyperedge representation learned by the model based on the topology-derived objective function and the model based on the set constraint objective function respectively share the same representation, which can comprehensively utilize the contexts and hyperedge information of each node to get the vectors of better quality.

For ease of calculation, take the logarithm of D_1 and D_2 , we aim to maximize the following joint objective function of HRSC.

$$\begin{aligned}
 L &= \sum_{n \in C} \left\{ \sum_{u \in \{n\} \cup NEG(n)} \left\{ L^n(u) \cdot \log \left[\sigma \left(X_n^T \theta_u \right) \right] + [1 - L^n(u)] \cdot \log \left[1 - \sigma \left(X_n^T \theta_u \right) \right] \right\} + \right. \\
 &\quad \left. \beta \cdot \sum_{v \in T_n} \sum_{\vartheta \in \{n\} \cup NEG(v)} \left\{ \delta(\vartheta|v) \cdot \log \left[\sigma \left(e_v^T \theta_\vartheta \right) \right] + [1 - \delta(\vartheta|v)] \cdot \log \left[1 - \sigma \left(e_v^T \theta_\vartheta \right) \right] \right\} \right\} \\
 &= \sum_{n \in C} \left\{ \sum_{u \in \{n\} \cup NEG(n)} \left\{ L^n(u) \cdot \log \left[\sigma \left(X_n^T \theta_u \right) \right] + [1 - L^n(u)] \cdot \log \left[1 - \sigma \left(X_n^T \theta_u \right) \right] \right\} + \right. \\
 &\quad \left. \sum_{v \in T_n} \sum_{\vartheta \in \{n\} \cup NEG(v)} \beta \cdot \left\{ \delta(\vartheta|v) \cdot \log \left[\sigma \left(e_v^T \theta_\vartheta \right) \right] + [1 - \delta(\vartheta|v)] \cdot \log \left[1 - \sigma \left(e_v^T \theta_\vartheta \right) \right] \right\} \right\} \tag{8}
 \end{aligned}$$

where β is the harmonic factors to balance the model based on the topology-derived objective function and the model based on the set constraint objective function.

For ease of derivation, $L(n, u, v, \vartheta)$ is defined as follows.

$$\begin{aligned}
 L(n, u, v, \vartheta) &= \left\{ L^n(u) \cdot \log \left[\sigma \left(X_n^T \theta_u \right) \right] + [1 - L^n(u)] \cdot \log \left[1 - \sigma \left(X_n^T \theta_u \right) \right] \right\} + \\
 &\quad \beta \cdot \left\{ \delta(\vartheta|v) \cdot \log \left[\sigma \left(e_v^T \theta_\vartheta \right) \right] + [1 - \delta(\vartheta|v)] \cdot \log \left[1 - \sigma \left(e_v^T \theta_\vartheta \right) \right] \right\} \tag{9}
 \end{aligned}$$

And then the joint objective function L is optimized by means of the stochastic gradient ascent algorithm. The key is to give four kinds of gradients of L .

Firstly, the gradient on θ_u of $L(n, u, v, \vartheta)$ is calculated as follows.

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

Accordingly, the updating formula of θ_u is expressed 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 of HRSC.

Secondly, the gradient on X_n of $L(n, u, v, \vartheta)$ is calculated. The symmetry property between θ_u and X_n is utilized to get the gradient on X_n .

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

Accordingly, the updating formula of $v_{v'}$ is expressed as follows, where $v' \in context(n)$.

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

Thirdly, the gradient on θ_ϑ of $L(n, u, v, \vartheta)$ is calculated as follows.

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

Accordingly, the updating formula of θ_ϑ is expressed as follows.

$$\theta_\vartheta = \theta_\vartheta + \alpha \cdot \beta \cdot [\delta(\vartheta|v) - \sigma(e_v^T \theta_\vartheta)] \cdot e_v \tag{15}$$

Fourthly, the gradient on e_v of $L(n, u, v, \vartheta)$ is calculated. The symmetry property between θ_ϑ and e_v is utilized to get the gradient on e_v .

$$\frac{\partial L(n, u, v, \vartheta)}{\partial e_v} = \beta \cdot [\delta(\vartheta|v) - \sigma(e_v^T \theta_\vartheta)] \cdot \theta_\vartheta \tag{16}$$

Accordingly, the updating formula of e_v is expressed as follows, where $v \in T_n$.

$$e_v = e_v + \alpha \cdot \beta \cdot [\delta(\vartheta|v) - \sigma(e_v^T \theta_\vartheta)] \cdot \theta_\vartheta \tag{17}$$

The stochastic gradient ascent (SGA) algorithm is used for optimization. In our implementation, the effect of β is approximated through instance sampling (node—node and node—hyperedge) in each training epoch. More details are shown in Algorithm 1.

Algorithm 1: HRSC

Input:

Hypernetwork $H = (V, E)$
 Embedding size d

Output:

Embedding matrix $X \in R^{|V| \times d}$

for node n in V **do**

 initialize embedding vector $v_n \in R^{1 \times d}$

 initialize parameter vector $\theta_n \in R^{1 \times d}$

for node v in T_n **do**

 initialize parameter vector $e_v \in R^{1 \times d}$

end for

end for

node sequences $C = \text{RandomWalk}()$

for $(n, \text{context}(n))$ in C **do**

 update parameter vectors following Formula (11)

 update embedding vectors following Formula (13)

 update parameter vectors following Formula (15)

for node v in T_n **do**

 update parameter vectors following Formula (17)

end for

end for

for $i = 0; i < |V|; i++$ **do**

$X_i = v_{v_i}$

end for

return X

5.4. Complexity Analysis

The time complexity of Deepwalk is $O(|C| \cdot 2b \cdot \log |V|)$, where b is the window size of the contextual nodes. In Formula (8), the time complexity of HRSC based on joint objective function is further reduced to $O(|C| \cdot (ds + 1) \cdot (\beta \cdot M + 1))$, where the time complexities of the models based on the topology-derived objective function and the set constraint objective function are $O(|C| \cdot (ds + 1))$ and $O(|C| \cdot (ds + 1) \cdot M)$, respectively, where ds is the predefined size of the negative sample set and also a constant number irrelevant to the size of the network, and $M = \max\{|T_{v_1}|, |T_{v_2}|, \dots, |T_{v_{|V|}}|\}$ is the maximum number of the set of the hyperedges associated with the node v_i . Compared to Deepwalk, the computational efficiency of HRSC is improved, since the computational efficiency of the models in HRSC derived from CBOW is higher than that of Skip-Gram in Deepwalk.

Meanwhile, the time complexity of the negative sampling adopted in HRSC is lower than that of the hierarchical SoftMax adopted in Deepwalk.

6. Experiments

6.1. Dataset

Four hypernetwork datasets, including a GPS network, an online social network, a drug network and a semantic network are utilized to evaluate the effectiveness of our proposed method. Detailed dataset statistics is shown in Table 2.

Table 2. 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 real-world hypernetwork datasets are as follows:

- GPS [24] describes a situation in which a user participates in an activity in a location. The relationships <user, location, activity> are regarded as hyperedges to construct a hypernetwork.
- MovieLens [25] describes personal tag activities from MovieLens. The relationships <user, movie, tag> are regarded as hyperedges to construct a hypernetwork, where each movie has at least one genre.
- Drug (<http://www.fda.gov/Drugs/>, 27 January 2020) is from FDA adverse event reporting system (FAERS), including information on adverse events and medication error reports submitted to the FDA. The relationships <user, drug, reaction> are regarded as hyperedges to construct a hypernetwork, that is, the users who take some drugs have certain reactions to lead to adverse events.
- wordnet [26] is composed of a set of triplets <head, relation, tail> extracted from WordNet3.0. The relationships <head, relation, tail> are regarded as hyperedges to construct a hypernetwork.

6.2. Baseline Methods

Deepwalk. Deepwalk is a popular topology-derived network representation learning method, and utilizes local information obtained from truncated random walk node sequences to learn node representation vectors.

node2vec. node2vec is an algorithmic framework for learning node representation vectors, which designs a biased random walk strategy to maximally preserve the network neighborhoods of the nodes.

LINE. LINE is proposed to learn node representation vectors for large scale networks, which designs an objective function that preserves both first-order proximity and second-order proximity.

GraRep. GraRep [27] captures global structure properties of a graph using k-step loss functions defined on the graph that integrates rich local structure information.

HOPE. HOPE [28] preserves higher-order proximity of a graph and captures the asymmetric transitivity.

SDNE. SDNE captures highly nonlinear network structure and utilizes first-order proximity and second-order proximity to characterize local and global network structures.

HRSC. HRSC incorporates the set of the hyperedges associated with the nodes into the process of hypernetwork representation learning, regards the learning process of node representation vectors as a joint optimization problem, and obtains node representation vectors including the hypernetwork topology structure and the hyperedges by means of the stochastic gradient ascent (SGA) algorithm to solve this joint optimization problem.

6.3. Experiment Setup

The experiments are conducted on four hypernetwork datasets. The node classification and link prediction tasks are adopted to verify the feasibility of our proposed method. As for all four datasets, the vector dimension is set as 100, the number of random walks to start at each node as 10, and the length of random walks to start at each node as 40. A portion of datasets are randomly selected as training set, and the rest is testing set.

6.4. Node Classification

The multi-label classification tasks are carried out on MovieLens and wordnet datasets, since only the two datasets have label information. In addition, we remove the nodes that did not have labels in the two datasets. The node representation vectors are taken as input to train SVM [29] classifier to calculate node classification accuracies.

Tables 3 and 4 show node classification accuracies on the MovieLens and wordnet datasets. From the two tables, we have obtained the following observations:

- On the MovieLens and wordnet datasets, the node classification performance of HRSC is superior to that of other baseline methods. For example, for these two datasets, HRSC outperforms other best baseline method (i.e., Deepwalk) by about 1%. Meanwhile, the node classification performance of HRSC exceeds the remaining baseline methods to a certain extent. Experimental results demonstrate that HRSC is effective and robust, since HRSC incorporates the hyperedges into the process of hypernetwork representation learning to learn node representation vectors of better quality than Deepwalk and other baseline methods. Meanwhile, from the above results, we can find that the node classification performance of HRSC exceeds other baseline methods very little, since the categorical attributes on the MovieLens and wordnet datasets are not obvious.
- The node classification performance of GraRep is second only to that of HRSC and DeepWalk, and its node classification performance is nearly the same as that of DeepWalk. The reason for this is that GraRep is an algorithm framework that integrates the global structure information of a graph into the representation learning process, and considers the hyperedges to a certain extent in the process of network representation learning.
- Although SDNE preserves local and global structure information in the process of network representation learning, it does not perform well in the hypernetwork dataset, which indicates that SDNE is not universal.

Table 3. Node classification accuracies on MovieLens (%).

Methods	Training Ratios								
	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
node2vec	46.93	49.28	50.77	51.51	52.62	52.58	53.04	53.44	52.71
LINE	43.93	45.46	46.52	47.29	47.70	48.16	48.02	49.09	48.34
GraRep	47.75	50.11	51.16	52.01	52.10	53.15	53.34	53.43	53.24
HOPE	46.33	48.57	49.95	50.69	51.06	51.04	51.29	52.53	51.79
SDNE	41.74	41.79	42.34	42.73	43.36	43.27	43.89	43.43	42.81
HRSC	48.60	50.81	52.02	53.19	53.73	54.12	54.83	54.95	56.14

In summary, we find that HRSC obtains high-quality node representation vectors by comprehensively considering the hypernetwork topology structure and the hyperedges.

6.5. Link Prediction

Link prediction is widely used in real life, especially in recommendation systems. In this subsection, we carry out link prediction tasks on all four datasets. As for the link prediction, the evaluation measure AUC [30] is adopted. As shown in Tables 5–8, the edges

with different ratios are randomly removed from the four datasets respectively to evaluate the AUC.

Table 4. Node classification accuracies on wordnet (%).

Methods	Training Ratios								
	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
node2vec	29.27	32.23	33.71	34.52	36.17	36.05	37.53	37.66	37.30
LINE	22.77	24.11	25.11	24.94	25.23	25.59	25.87	26.60	25.44
GraRep	32.59	34.74	34.63	35.21	35.38	36.05	35.10	36.63	37.79
HOPE	30.53	33.61	35.02	35.97	34.90	35.11	36.21	36.20	34.84
SDNE	21.96	21.57	22.05	22.37	23.26	22.59	23.63	23.60	25.31
HRSC	31.54	33.94	34.98	36.84	37.35	38.02	38.78	40.22	41.10

Table 5. AUC values on GPS.

Methods	Training Ratios						
	60%	65%	70%	75%	80%	85%	90%
Deepwalk	0.4308	0.4278	0.4205	0.4583	0.4418	0.4914	0.4831
node2vec	0.3660	0.3614	0.3808	0.3939	0.3834	0.3958	0.3649
LINE	0.4575	0.4829	0.4761	0.4562	0.4429	0.4663	0.4574
GraRep	0.3873	0.3805	0.3882	0.3765	0.3820	0.3857	0.3874
HOPE	0.3805	0.3676	0.3416	0.2971	0.2794	0.2518	0.2334
SDNE	0.3262	0.4371	0.4319	0.3157	0.4379	0.3527	0.4540
HRSC	0.7516	0.7562	0.7488	0.7449	0.7236	0.7325	0.7279

Table 6. AUC values on MovieLens.

Methods	Training Ratios						
	60%	65%	70%	75%	80%	85%	90%
Deepwalk	0.7845	0.8129	0.8301	0.8440	0.8729	0.8800	0.9025
node2vec	0.7078	0.7390	0.7418	0.7696	0.7939	0.8036	0.8296
LINE	0.8282	0.8242	0.8253	0.8320	0.8365	0.8172	0.8231
GraRep	0.7290	0.7833	0.7907	0.8121	0.8277	0.8481	0.8544
HOPE	0.6895	0.7333	0.7203	0.7522	0.7787	0.7986	0.8049
SDNE	0.4004	0.3511	0.3494	0.3406	0.3433	0.3598	0.4171
HRSC	0.8714	0.8706	0.8681	0.8644	0.8706	0.8651	0.8528

Table 7. AUC values on drug.

Methods	Training Ratios						
	60%	65%	70%	75%	80%	85%	90%
Deepwalk	0.4852	0.4954	0.4934	0.4580	0.4901	0.4638	0.4713
node2vec	0.4500	0.4525	0.4490	0.4525	0.4329	0.4712	0.4345
LINE	0.4750	0.4672	0.4636	0.4625	0.4741	0.4523	0.4768
GraRep	0.5025	0.5089	0.4867	0.5051	0.5557	0.5835	0.5362
HOPE	0.5055	0.5269	0.4933	0.4690	0.4941	0.4668	0.4271
SDNE	0.2948	0.4310	0.4454	0.5050	0.5196	0.3536	0.3836
HRSC	0.7871	0.7774	0.7822	0.7920	0.7747	0.7983	0.8056

From Tables 5–8, we have following observations:

- On the GPS, drug and wordnet datasets, the link prediction performance of HRSC is superior to that of other baseline methods. On the MovieLens dataset, the link prediction performance of HRSC is better than that of most of other baseline methods, specially, HRSC performs better than DeepWalk and GraRep at the smaller training

ratios, and worse than DeepWalk and GraRep at larger training ratios, which indicates the effectiveness of HRSC, and verifies that HRSC has the ability to accurately estimate the relations among the nodes.

- The link prediction performance of DeepWalk is second only to that of HRSC, which indicates that node representation vectors obtained by directly training DeepWalk on the hypernetwork datasets contains the hyperedge information to a certain extent. It is worth noting that SDNE is the worst performing baseline method overall and has unstable performance at different training ratios, which verifies that SDNE is not universal. In contrast, HRSC performs consistently at different training ratios, which demonstrates its feasibility and robustness.
- By introducing the hyperedges, the node representation vectors learned by HRSC are better than that of other baseline methods without considering the hyperedges, which verifies our assumption that it is beneficial to link prediction task to incorporate the hyperedges into the process of hypernetwork representation learning.

Table 8. AUC values on wordnet.

Methods	Training Ratios						
	60%	65%	70%	75%	80%	85%	90%
Deepwalk	0.7780	0.8181	0.8305	0.8341	0.8708	0.8765	0.8880
node2vec	0.7807	0.8242	0.8309	0.8285	0.8519	0.8503	0.8595
LINE	0.8063	0.8184	0.8056	0.8091	0.8000	0.7938	0.7926
GraRep	0.7685	0.7742	0.7888	0.7806	0.7958	0.7972	0.7756
HOPE	0.6902	0.7314	0.7417	0.7403	0.7649	0.7763	0.7700
SDNE	0.3712	0.5348	0.4784	0.4824	0.4254	0.6159	0.4850
HRSC	0.8953	0.9079	0.9086	0.9036	0.9093	0.9045	0.9034

In short, the above observations show that HRSC can learn high-quality node representation vectors, which help to accurately estimate the relations among the nodes. In addition, the experimental results of the link prediction task demonstrate the effectiveness and robustness of HRSC.

6.6. Parameter Sensitivity

HRSC has a harmonic factor β to balance the model based on the topology-derived objective function and the model based on the set constraint objective function. We fix the training ratio to 50%, and test node classification accuracies of HRSC with different β . We let β range from 0.1 to 0.9 on the MovieLens and wordnet datasets. Figure 3 shows the comparisons of node classification accuracies of HRSC with different β .

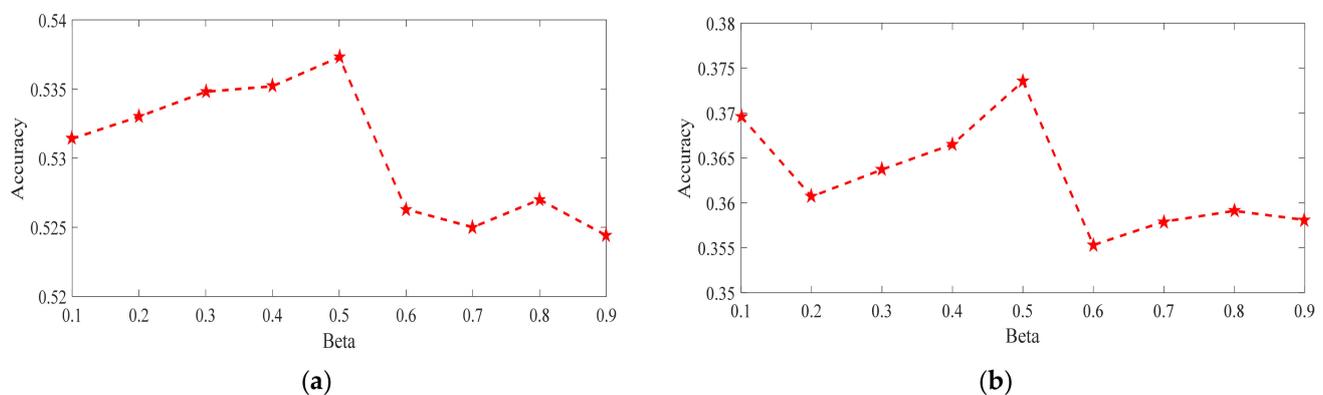


Figure 3. Parameter sensitivity on the two datasets. (a) Sensitivity on MovieLens; (b) sensitivity on wordnet.

From Figure 3, the variation trend of node classification accuracies is different with the increase of the parameter β on the two datasets, but all of the variation ranges are within 2%, which indicates that the overall performance of HRSC is not sensitive to the parameter β , demonstrating the robustness of HRSC.

For the MovieLens and wordnet datasets, the best evaluated results in terms of node classifications are achieved at $\beta = 0.5$. Although the hyperedges are fully incorporated into the process of hypernetwork representation learning to learn node representation vectors, partial hypernetwork topology structure information is still lost, which indicates that the quality of node representation vectors is optimal.

7. Conclusions

This paper proposes a hypernetwork representation learning method with the set constraint, which effectively incorporates the hyperedges into the process of hypernetwork representation learning and regards the learning process of node representation vectors as a joint optimization problem. By using an effective stochastic gradient ascend algorithm to solve the joint optimization problem, the node representation vectors including the hypernetwork topology structure and the hyperedges can be obtained. We conducted extensive experiments with four hypernetwork datasets. The experimental results demonstrate that HRSC outperforms other best baseline methods by about 1% on the MovieLens and wordnet datasets in terms of node classification, and outperforms other best baseline methods respectively on average by about 29.03%, 1.94%, 26.27%, and 6.24% on the GPS, MovieLens, drug and wordnet datasets in terms of link prediction.

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. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 62166032 and grant number 62101299.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained in the article.

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

References

1. Li, C.T.; Lin, H.Y. Structural hierarchy-enhanced network representation learning. *Appl. Sci.* **2020**, *10*, 7214. [[CrossRef](#)]
2. Hou, X.J.; Liu, Y.S.; Li, Z.F. Convolutional adaptive network for link prediction in knowledge bases. *Appl. Sci.* **2021**, *11*, 4270. [[CrossRef](#)]
3. Lyu, D.S.; Wang, B.; Zhang, W.Z. Large-scale complex network community detection combined with local search and genetic algorithm. *Appl. Sci.* **2020**, *10*, 3126. [[CrossRef](#)]
4. 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.
5. 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.
6. 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.
7. 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.
8. Chen, H.C.; Perozzi, B.; Hu, Y.F.; Skiena, S. HARP: Hierarchical representation learning for networks. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 2127–2134.
9. Tu, C.C.; Liu, H.; Liu, Z.Y.; Sun, M.S. CANE: Context-aware network embedding for relation modeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, MD, Canada, 30 July–4 August 2017; pp. 1722–1731.

10. Tu, C.C.; Zeng, X.K.; Wang, H.; Zhang, Z.Y.; Liu, Z.Y.; Sun, M.S.; Zhang, B.; Lin, L.Y. A unified framework for community detection and network representation learning. *IEEE TKDE* **2019**, *31*, 1051–1065. [[CrossRef](#)]
11. Li, C.Z.; Wang, S.Z.; Yang, D.J.; Li, Z.J.; Yang, Y.; Zhang, X.M.; Zhou, J.S. PPNE: Property preserving network embedding. In Proceedings of the 22nd International Conference on Database Systems for Advanced Applications, Suzhou, China, 27–30 March 2017; pp. 163–179.
12. Shen, C.X.; Qian, L.P.; Yu, N.N. Adaptive Facial imagery clustering via spectral clustering and reinforcement learning. *Appl. Sci.* **2021**, *11*, 8051. [[CrossRef](#)]
13. Bretto, A. *Hypergraph Theory: An Introduction*; Springer Press: Berlin, Germany, 2013; pp. 24–27.
14. 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, MD, Canada, 4–7 December 2006; pp. 1601–1608.
15. Liu, Y.; Shao, J.; Xiao, J.; Wu, F. Hypergraph spectral hashing for image retrieval with heterogeneous social contexts. *Neurocomputing* **2013**, *119*, 49–58. [[CrossRef](#)]
16. Wu, F.; Han, Y.H.; Zhuang, Y.T. Multiple hypergraph clustering of web images by mining word2image correlations. *JCST* **2010**, *25*, 750–760.
17. Liang, L.; Wen, H.B.; Liu, F.; Li, G. Feature extraction of impulse faults for vibration signals based on sparse non-negative tensor factorization. *Appl. Sci.* **2019**, *9*, 3642. [[CrossRef](#)]
18. Li, L.; Bai, R.; Lu, J.F.; Zhang, S.Q.; Chang, C.C. A watermarking scheme for color image using quaternion discrete fourier transform and tensor decomposition. *Appl. Sci.* **2021**, *11*, 5006. [[CrossRef](#)]
19. 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, Hawaii, HI, USA, 27 January–1 February 2019; pp. 3558–3565.
20. 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.
21. 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.
22. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, CA, USA, 5–10 December 2013; pp. 3111–3119.
23. 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.
24. 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.
25. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *ACM Trans. Inter. Intell. Syst.* **2015**, *5*, 19. [[CrossRef](#)]
26. 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, CA, USA, 5–10 December 2013; pp. 2787–2795.
27. 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.
28. 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.
29. Gu, Y.; Zhang, S.Y.; Qiu, L.M.; Wang, Z.L.; Zhang, L.C. A layered KNN-SVM approach to predict missing values of functional requirements in product customization. *Appl. Sci.* **2021**, *11*, 2420. [[CrossRef](#)]
30. Venkatesan, N.J.; Dong, R.S.; Choon, S.N. Nodule detection with convolutional neural network using apache spark and GPU frameworks. *Appl. Sci.* **2021**, *11*, 2838. [[CrossRef](#)]