

Link Prediction with Hypergraphs via Network Embedding

Zijuan Zhao ^{1,†} , Kai Yang ^{2,*,†}  and Jinli Guo ^{1,*}¹ Business School, University of Shanghai for Science and Technology, Shanghai 200093, China² College of Information Engineering, Yangzhou University, Yangzhou 225127, China

* Correspondence: yangk@fudan.edu.cn (K.Y.); phd5816@163.com (J.G.)

† These authors contributed equally to this work.

Abstract: Network embedding is a promising field and is important for various network analysis tasks, such as link prediction, node classification, community detection and others. Most research studies on link prediction focus on simple networks and pay little attention to hypergraphs that provide a natural way to represent complex higher-order relationships. In this paper, we propose a link prediction method with hypergraphs using network embedding (HNE). HNE adapts a traditional network embedding method, Deepwalk, to link prediction in hypergraphs. Firstly, the hypergraph model is constructed based on heterogeneous library loan records of seven universities. With a network embedding method, the low-dimensional vectors are obtained to extract network structure features for the hypergraphs. Then, the link prediction is implemented on the hypergraphs as the classification task with machine learning. The experimental results on seven real networks show our approach has good performance for link prediction in hypergraphs. Our method will be helpful for human behavior dynamics.

Keywords: link prediction; hypergraph; network embedding; machine learning; heterogeneous network; library loan records; human behavior dynamics



Citation: Zhao, Z.; Yang, K.; Guo, J. Link Prediction with Hypergraphs via Network Embedding. *Appl. Sci.* **2023**, *13*, 523. <https://doi.org/10.3390/app13010523>

Academic Editors: Jing Zhang, Jipeng Qiang and Cangqi Zhou

Received: 9 October 2022

Revised: 21 December 2022

Accepted: 26 December 2022

Published: 30 December 2022



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

Link prediction [1–3] has been widely applied in many fields with extensive research studies, especially in society networks, such as community detection [4] and recommendation [5]. It aims to predict the potential links between nodes based on existing links, and has a wide range of applications in many fields, from bioinformatics [6,7] and social science [8] to computer science [9]. Existing traditional methods for link prediction [10–13] focus on simple graphs mostly and less on the interactions between pairs of nodes present in real-world systems, while research on high-order interactions is of great significance for modeling complex systems. For instance, in scientific collaboration networks, several researchers work together on a research project; in the brain network, a human behavior usually involves multiple neurons. Link prediction on high-order interactions leads to some challenges, while a hypergraph [14–16] provides a useful way to modeling such interactions. A hypergraph can reflect multiple nodes' relations with hyperlinks, and can be used in evaluating vital nodes [17], describing protein interaction [18] and so on. Hyperlink prediction on hypergraph has been investigated to predict higher-order links such as a user releasing a tweet containing a hashtag [19]. Hyperlink prediction [20] has also been helpful to predict multiactor collaborations [21]. By formulating various kinds of nodes and associations into a hypergraph, link prediction on heterogeneous networks has developed increasingly. Li Dong [19] modeled various types of objects and relations of networks as hypergraphs and used link proximities to construct a cost function to predict users' links. Maria [22] constructed relations between pairs of drugs into a hypergraph to predict multidrug interactions. Liu et al. [23] proposed a Metapath-aware HyperGraph Transformer (Meta-HGT) for node embedding to capture the high-order relations. Kang et al. [24] proposed dynamic hypergraph neural networks based on key hyperedges (DHKH) to consider

a dynamic hypergraph structure. Fan et al. [25] presented a method named heterogeneous hypergraph variational autoencoder (HeteHG-VAE) for link prediction in heterogeneous information networks (HINs) mapped to a heterogeneous hypergraph with a certain kind of semantics to capture both the high-order semantics and complex relations among nodes, while preserving the low-order pairwise topology information of the original HIN.

Network embedding [26,27] combining machine learning or deep learning with network science has made it possible to automatically learn and preserve network properties by representing nodes in a low-dimensional space. It is usually assumed that the distance between the representation vectors of nodes reflects the similarity of the nodes in networks [28]. Network embedding typically realizes a network representation through matrix factorization, random walk and neural network methods. The matrix factorization methods select an adjacency matrix, an incident matrix, a Laplacian matrix and their variant forms to factorize and obtain the embeddings, such as M-NMF [29] and Laplacian eigenmaps [30]. The random walk methods generate embeddings through a random walk of nodes on graphs and training node sequences in models; representative methods include the Deepwalk [31], Node2vec [32] and Graphwave models [33]. The methods based on a neural network realize an embedding by the nonlinear function of deep models to map the networks in a vector space, such as HeGan [34], VERSE [35] and SiNE models [36]. Furthermore, deep-learning-based link prediction methods on hypergraphs have achieved rapid development. Yadati et al. [37] proposed a neural hyperlink predictor (NHP) adapting graph convolutional networks (GCNs) [38] for link prediction in hypergraphs. Node2vec [32] with a single-layer perceptron (Node2vec-SLP) was an improved version of Node2vec for hyperlink prediction, which employed a one-layer neural network to compute hyperlink scores [39].

Considering that hypergraphs can represent higher-order systems more conveniently, the interaction information of nodes is characterized into vectors with network embedding, so that the link prediction on hypergraphs can be converted into a classification problem. Therefore, we provide a novel idea of link prediction with hypergraphs with network embedding (HNE) in this paper. Our motivation is to predict the relationships of students based on the library loan records of universities, instead of higher-order relationships of students. Thus, we investigate the link prediction with hypergraphs. We use a hypergraph to model all types of objects and relations of the library loan record networks. Firstly, we construct different kinds of nodes associations in a heterogeneous network with a hypergraph according to the library loan records of seven universities. Secondly, a network embedding method, Deepwalk, is utilized to extract structural information and represent nodes by vectors. Thirdly, a machine learning model, a random forest [40], is applied as a classifier for the link prediction. The experiments are conducted on seven sizes of heterogeneous networks and compare several typical link prediction methods to verify the performance of the proposed approach and achieve the promising results on the seven datasets.

The innovations in this paper are as follows: We propose a link prediction method using hypergraphs based on network embedding. The representation of the features of library loan record associations are novel in the process of our overall algorithm for link prediction of the relationship of students, which means that learning technology is applied to human behavior dynamics networks, that is, network embedding technology is introduced into human behavior dynamics networks. Then, a vector of each student for library loan records is constructed as a training set. Our method achieves promising results on the seven different datasets.

2. Materials and Methods

Figure 1 shows the complete flow chart for HNE, the link prediction approach we propose based on hypergraphs with network embedding. First, the heterogeneous networks constructed from library loan records of seven universities are explored, which consists of two types of nodes (*Node I* represents students, *Node II* indicates the books borrowed by the

students from libraries) and their interactions. The hypergraph is constructed according to these interactions; the hyperlinks represent *Node II* linked with *Node I*. The *Node I* network is constructed based on hypergraph properties. The incidence matrix denotes the relationships between *Node I* and hyperlinks. The adjacency matrix describes the links between *Node I*. Second, the embedding vectors of *Node I* are generated by the network embedding model. Then, the embedding vectors of links are generated by concatenating the vectors of pairwise nodes. Finally, the links vectors are divided into training data and testing data. The training data are put into the random forest classifier to train the model, then the testing data are used to predict potential links.

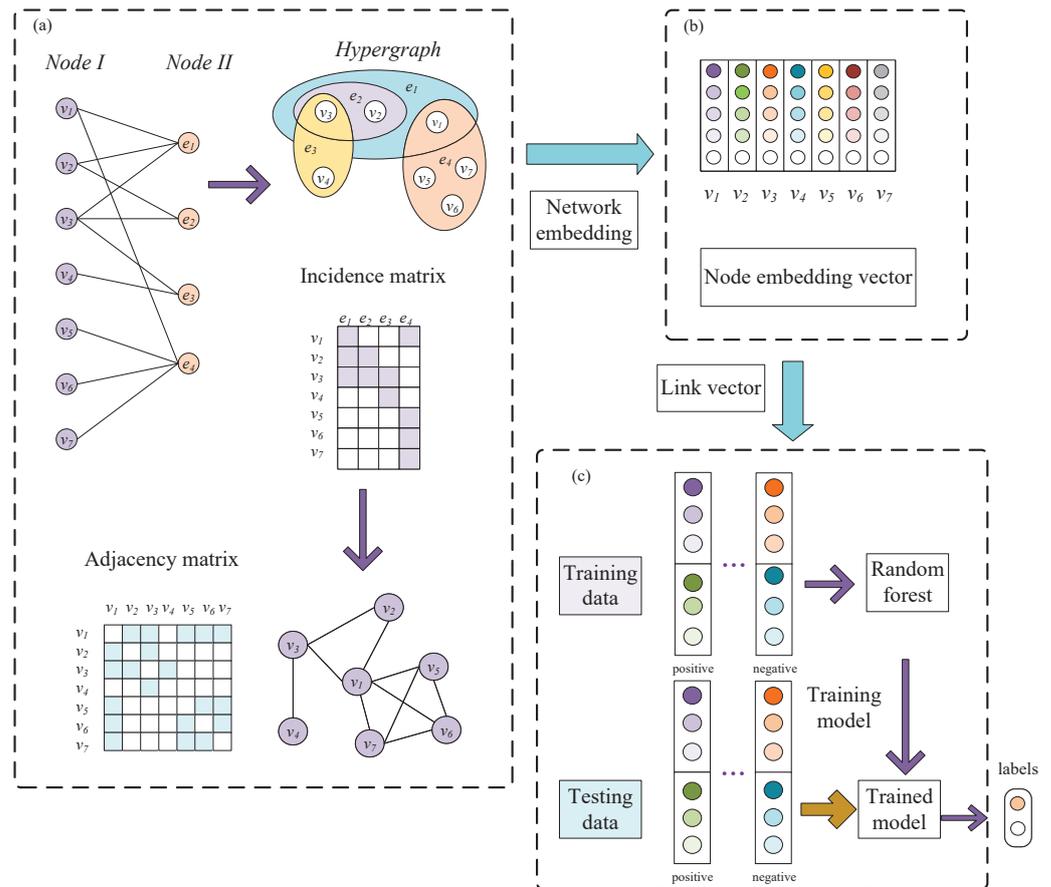


Figure 1. The framework of link prediction for hypergraphs via network embedding (HNE). (a) The heterogeneous network contains two types of nodes, *Nodes I* and *II*, with their interactions; it can be constructed by a hypergraph model. The incidence matrix represents the node–hyperlink interactions and the adjacency matrix describes node–node associations. (b) The Deepwalk model is applied to learn the node embedding vectors. (c) The random forest classifier is trained to predict link labels.

2.1. Hypergraph Construction

A hypergraph is defined as $H = (V, E)$ where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{E_1, E_2, \dots, E_m\}$ [41]. V is a set of n hypernodes and E is a set of m hyperlinks. The hyperlink $E_i = \{v_{i1}, v_{i2}, \dots, v_{ij}\}$, ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) contains j nodes, that is, the size of E_i is j . The $|V| \times |E|$ incidence matrix can be represented by H .

$$H(v, e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases} \tag{1}$$

Based on H , the node degree $d(v)$ of each node v meaning the number of neighbor nodes of node v is represented as

$$d(v) = \sum_{e \in E} H(v, e). \tag{2}$$

The hyperdegree $d_H(v)$ of node v denotes the number of hyperlinks which the node v participates in. The degree $\delta(e)$ of hyperlink e is the total number of neighbor hyperlinks of hyperlink e as follows,

$$\delta(e) = \sum_{v \in V} H(v, e). \tag{3}$$

The hyperdegree $\delta_H(e)$ of hyperlink e denotes the number of nodes of hyperlink e [42].

2.2. Learning Representations with Network Embedding

With the adjacency matrix from a hypergraph model, the representation learning vectors of nodes are obtained by a network embedding model. In this paper, we introduce the Deepwalk network embedding method which consists of two parts, that is, a random walk and Skip-gram. Firstly, some sequences of nodes with the same length t can be obtained by a random walk. Each node is the root of a walk sampling W_{v_i} ; the root node v_i randomly selects one of the links connected to it and moves to the neighbor node to start the next walk until the walk length reaches t ; the maximum length t denotes the size for a sequence of nodes. Secondly, a window of a specific length slides to sample the context for target node v_i in the sequence of nodes. Three layers are involved in the Skip-gram model: input, hidden and output layer. The initial representation of target node v_i is the input, the model parameters are trained and updated to maximize the probability of the neighbors of the target node v_i .

$$Pr(\{v_{i-w}, \dots, v_{i+w}\} \setminus v_i \mid \Phi(v_i)) = \prod_{j=i-w, j \neq i}^{i+w} Pr(v_j \mid \Phi(v_i)) \tag{4}$$

where $\Phi(v_i)$ denotes the current representation vector of node v_i , w is the size of the window in Skip-gram, $\{v_{i-w}, \dots, v_{i+w}\} \setminus v_i$ is the context of node v_i , and the hierarchical softmax adopts a binary tree to reduce the complexity of calculating $Pr(v_j \mid \Phi(v_i))$. The problem turns into maximizing the probability of paths from the root node to the tree nodes.

2.3. Loss Function

Finally, the node embedding output from this model is applied to the specific node classification task of semi-supervised learning, and the loss function is calculated to minimize the cross-entropy loss value between the true label and the predicted value in the training set. The calculation process is shown in Equation (5):

$$\iota = - \sum_{l \in L} Y^l \ln(C \cdot Z^l) \tag{5}$$

where C is the parameter of the classifier, L is the set of training set nodes, Y^l and Z^l represent the true labels corresponding to the training set data and the predicted values generated by the model, respectively. Based on the training set data, in this paper, we used the backpropagation method to train the parameters of the model for learning more accurate node embedding representations.

2.4. Datasets

In this paper, the real library loan records of seven universities in Shanghai, which were Shanghai University of Electric Power (SUEP), Shanghai Ocean University(SHOU), Shanghai University of Finance and Economics(SUFE), University of Shanghai for Science and Technology(USST), Shanghai International Studies University(SISU), Shanghai Normal University(SHNU) and Tongji University(TJU), were used to validate the performance

of our approach. The datasets were collected from Huiyuan sharing [43]. We organized the data from 2017 to 2018 and took two columns of data, ISBN and PATRON_ID, as the different types of nodes to construct the hypergraphs. PATRON_ID represented *NodeI* and ISBN denoted the hyperlinks in Figure 1. The structural properties of the hypergraphs are analyzed in the Table 1. As shown in Table 1, n denotes the number of nodes, m_0 refers to the total number of links between nodes, $\langle k \rangle$ means the average degree of nodes, m is the number of hyperlinks, $\langle d_H(v) \rangle$ refers to the average hyperdegree of a node, $\langle \delta(e) \rangle$ means the average degree of hyperlink, and $\langle \delta_H(e) \rangle$ is the average hyperdegree of a hyperlink.

Table 1. The structural properties of the seven hypergraphs.

Datasets	n	m_0	m	Density	$\langle k \rangle$	$\langle d_H(v) \rangle$	$\langle \delta(e) \rangle$	$\langle \delta_H(e) \rangle$
SUEP	906	24,362	19,530	0.0297	27	29	47	1.3
SHOU	2680	222,126	64,958	0.0309	81	41	108	1.7
SUFE	1720	148,188	35,727	0.0501	86	33	62	1.6
USST	2733	230,597	54,437	0.0308	84	36	93	1.8
SISU	3089	478,953	72,100	0.0502	155	46	142	2
SHNU	3557	263,305	93,996	0.0208	74	43	120	1.6
TJU	6150	988,516	131,199	0.0261	161	42	134	1.9

3. Experiments

To evaluate the performance of HNE, we conducted experiments on the seven datasets. Firstly, to train the model, we took the existing links as positive samples and then obtained random negative samples according to the number of positive samples. Given a test ratio (set as 30%) as input, the positive and negative samples were divided into a training set and a test set. Secondly, the embedding vectors of links were represented by concatenating the embedding vectors of the corresponding node pairs in the training set and test set for the unsupervised link prediction. After that, we input the embedding vectors of the samples in the training set into the random forest to learn the potential relationships among links and then input the embedding vectors of the samples in the test set into the trained random forest to predict possible links. Finally, the results of the link prediction were assessed with the AUC metric.

3.1. Compared Methods

In this paper, we compared the proposed HNE with three categories of baselines: similarity-based methods—CN [10], Jaccard coefficient [11], random-walk-based methods—Katz [12] and RWR [13], and deep-learning-based methods—Node2vec [32], GCN [38]. The existence probabilities of links were evaluated by the similarity between two nodes. Common neighbors (CN) is a link prediction method that is based on evaluating the overlap or similarity of two nodes by obtaining the number of common neighbors in a graph. The Jaccard coefficient is defined as the ratio of the common neighbor size of node i and node j to the size of all their neighbors. Katz centrality is an approach for summing all paths of nodes i and j , where the weight of paths decays exponentially according to their length, to evaluate how closely two nodes are related in the graph. Random walk with restart (RWR) provides a kind of random walk where node i moves to its neighbor with probability c or it jumps to the original node with probability $1 - c$. We set $c = 0.2$ in this paper. Node2vec learns a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes. GCN is a classical graph neural network to learn the representation of nodes in graphs by convolutional networks. For the deep-learning-based methods, we set the embedding dimension as 64, and for all methods, we randomly ran them 10 times and reported the average results.

The training set data selected in this experiment were obtained by random sampling. In order to more comprehensively evaluate the accuracy and validity of the experimental results, in this paper, we used a weighted average processing to consider a training sample, the n classification problem was decomposed into two classification problems, and then the prediction results of the model were evaluated. Four evaluation indicators, AUC, precision, recall and F1-score were evaluated in the experimental results of the model to ensure the reliability and validity of the HNE method.

3.2. Results

To evaluate the performance of the four methods of link prediction, the experiment was implemented 10 times to compute the average AUC score and the results are shown in Figure 2. We observe that the AUC scores of HNE were 0.8247, 0.9077, 0.844, 0.8433, 0.8418, 0.8693 and 0.8120, respectively, on the seven datasets, which were better than those of the other methods on the seven datasets. The AUC scores improved by 26.9%, 16%, 19%, 32%, 1.67% and 7.38% at most compared with the scores of the CN, the Jaccard coefficient, the Katz centrality, the RWR, Node2vec and GCN, respectively. Based on the above analysis, a promising performance was achieved for the HNE method. Moreover, the performance of HNE was very stable on the seven datasets with different sizes.

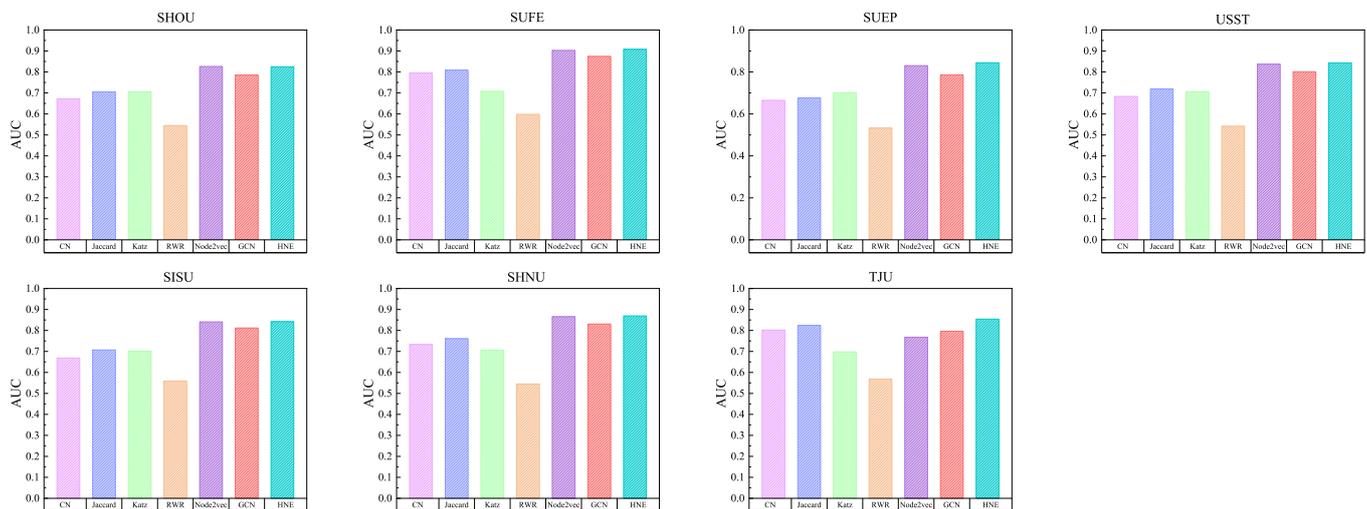


Figure 2. The AUC of CN, Jaccard, Katz, RWR, Node2vec, GCN and HNE on seven datasets.

We further evaluate the performance of our method with the precision, recall and F1-score on the seven datasets. As shown in Tables 2 and 3, the precision, recall and F1-score of our method achieved the best results on the seven datasets. Specifically, the precision of HNE was 0.9424 on the SUFE dataset, which was better than the other algorithms. For the recall and F1-score, our method improved by 2.2% and 28.7% and 7.7% and 19.2% compared to Node2vec and GCN, respectively. The F1-scores of HNE were superior to the other methods except on the TJU dataset. The experiment results show that our proposed method outperformed the CN, the Jaccard coefficient, the Katz centrality, the RWR, Node2vec and GCN on all datasets except the TJU dataset. Therefore, our algorithm showed a better performance and effectiveness for link prediction than traditional methods.

From the experiments, we can see that in the seven datasets, our method still maintained a relatively stable overall performance.

Table 2. The experimental results for the precision, recall and F1-Score on the SHOU, SUFE, SUEP.

	SHOU			SUFE			SUEP		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
CN	0.6971	0.6091	0.6499	0.8996	0.6654	0.7650	0.7063	0.5645	0.6275
Jaccard	0.7569	0.6034	0.6715	0.9034	0.6916	0.7834	0.7412	0.5424	0.6263
Katz	0.6705	0.8001	0.7333	0.6722	0.8121	0.7356	0.6663	0.8061	0.7296
RWR	0.5446	0.5456	0.5449	0.5929	0.6250	0.6083	0.5328	0.5366	0.5337
Node2vec	0.8317	0.8037	0.8223	0.9416	0.8566	0.898	0.8401	0.8154	0.8275
GCN	0.7959	0.7675	0.7814	0.9046	0.8372	0.8696	0.7934	0.7734	0.7832
HNE	0.8379	0.8052	0.8212	0.9424	0.8685	0.9040	0.8516	0.8331	0.8422

Table 3. The experimental results for the precision, recall and F1-score on the USST, SISU, SHNU and TJU.

	USST			SISU			SHNU			TJU		
	Precision	Recall	F1-Score									
CN	0.6902	0.6663	0.678	0.6885	0.6161	0.8354	0.7594	0.6821	0.7187	0.8289	0.7467	0.7856
Jaccard	0.7685	0.6271	0.6906	0.7481	0.6246	0.6808	0.8107	0.6800	0.7396	0.8726	0.7596	0.8122
Katz	0.6711	0.8093	0.7337	0.6682	0.8022	0.7291	0.6706	0.8085	0.7331	0.6666	0.7925	0.7241
RWR	0.5438	0.5191	0.5306	0.5603	0.5455	0.5526	0.5457	0.5368	0.5411	0.5718	0.547	0.5588
Node2vec	0.8603	0.805	0.8317	0.866	0.8068	0.6502	0.8706	0.8582	0.8644	0.8668	0.8355	0.8512
GCN	0.8185	0.7738	0.7955	0.8328	0.7798	0.8054	0.8358	0.8229	0.7293	0.7848	0.7375	0.7604
HNE	0.8632	0.8160	0.8389	0.8657	0.8092	0.8365	0.8706	0.8674	0.8691	0.8365	0.7789	0.7972

4. Conclusions and Discussions

In this paper, a link prediction approach with network embedding was proposed for hypergraphs. The proposed HNE method applied the Deepwalk model to extract features of nodes according to the hypergraphs constructed from library loan records, then a classifier was trained to predict the potential links between nodes. The experiment results on seven datasets showed that our approach outperformed typical link prediction methods. The comparison of AUC, precision, recall and F1-score with six methods demonstrated the effectiveness of the proposed approach.

In the future, the idea of combining hypergraphs and network embedding can not only be applied to link prediction, but also implement more tasks, such as node importance, community detection and node classification. In addition, our proposed algorithm has wide practical applications, such as recommendations for online social networks, knowledge reasoning for knowledge hypergraph construction, drug-target prediction or drug-disease prediction in the field of bioinformatics and so on. In addition, as more graph neural network methods [44–47] are proposed, we can explore hyperlink prediction algorithms and other graph neural network models for preserving more structural and semantic information of hypergraphs to solve the fundamental problems in hypergraph analysis.

Author Contributions: Conceptualization, Z.Z. and K.Y.; methodology, K.Y.; software, Z.Z.; validation, Z.Z., K.Y. and J.G.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z., K.Y. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 71571119), and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 22KJD120002).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Our datasets could be downloaded in <http://hdl.handle.net/20.500.12291/10022> (accessed on 8 October 2022).

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

References

1. Lü, L.Y.; Zhou, T. Link prediction in complex networks: A survey. *Phys. A* **2011**, *6*, 1150–1170. [\[CrossRef\]](#)
2. Zhang, M.H.; Chen, Y.X. Link prediction based on graph neural networks. In Proceedings of the Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; pp. 5171–5181.
3. Wang, H.W.; Zhang, F.Z.; Hou, M.; Xie, X.; Guo, M.Y.; Liu, Q. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina del Rey, CA, USA, 5–9 February 2018; pp. 592–600.
4. Pulipati, S.; Somula, R.; Parvathala, B.R. Nature inspired link prediction and community detection algorithms for social networks: A survey. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–18. [\[CrossRef\]](#)
5. Talasu, N.; Jonnalagadda, A.; Pillai, S.S.A.; Rahul, J. A link prediction based approach for recommendation systems. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Manipal, India, 13–16 September 2017; pp. 2059–2062.
6. Patel, R.; Guo, Y.H.; Alhudhaif, A.; Alenezi, F.; Althubiti, S.A.; Polat, K. Graph-Based Link Prediction between Human Phenotypes and Genes. *Math. Probl. Eng.* **2021**, *2022*, 8. [\[CrossRef\]](#)
7. Yang, K.; Zhao, X.Z.; Waxman, D.; Zhao, X.M. Predicting drug-disease associations with heterogeneous network embedding. *Chaos* **2019**, *12*, 123109. [\[CrossRef\]](#)
8. Kushwah, A.K.S.; Manjhar, A.K. A review on link prediction in social network. *Int. J. Grid Distrib. Comput.* **2016**, *2*, 43–50. [\[CrossRef\]](#)
9. Passino, F.S.; Turcotte, M.J.M.; Heard, N.A. Graph link prediction in computer networks using poisson matrix factorisation. *Ann. Appl. Stat.* **2022**, *3*, 1313–1332.
10. Zhou, T.; Kuscsik, Z.; Liu, J.-G.; Medo, M.; Wakeling, J.R.; Zhang, Y.-C. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 4511–4515. [\[CrossRef\]](#)
11. Real, R.; Vargas, J.M. The probabilistic basis of Jaccard's index of similarity. *Syst. Biol.* **1996**, *3*, 380–385. [\[CrossRef\]](#)
12. Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **1953**, *1*, 39–43.
13. Tong, H.h.; Faloutsos, C.; Pan, J.Y. Fast random walk with restart and its applications. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 613–622.
14. Gao, Y.; Zhang, Z.Z.; Lin, H.J.; Zhao, X.B.; Du, S.Y.; Zou, C.Q. Hypergraph learning: Methods and practices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2548–2566. [\[CrossRef\]](#)
15. Feng, Y.F.; You, H.X.; Zhang, Z.Z.; Ji, R.R.; Gao, Y. Hypergraph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3558–3565.
16. Bai, S.; Zhang, F.h.; Torr, P.H.S. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* **2021**, *110*, 107637. [\[CrossRef\]](#)
17. Xiao, Q. Node importance measure for scientific research collaboration from hypernetwork perspective. *Teh. Vjesn.* **2016**, *2*, 397–404.
18. Gallagher, S.R.; Goldberg, D.S. Clustering coefficients in protein interaction hypernetworks. In Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics, Washington, DC, USA, 22–25 September 2013; pp. 552–560.
19. Li, D.; Xu, Z.; Li, S.; Sun, X. Link prediction in social networks based on hypergraph. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 41–42.
20. Chen, C.; Liu, Y.Y. A survey on hyperlink prediction. *arXiv* **2022**, arXiv:2207.02911.
21. Sharma, A.; Srivastava, J.; Chandra, A. Predicting multi-actor collaborations using hypergraphs. *arXiv* **2014**, arXiv:1401.6404.
22. Vaida, M.; Purcell, K. Hypergraph link prediction: Learning drug interaction networks embeddings. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1860–1865.
23. Liu, J.; Song, L.; Wang, G.; Shang, X.Q. Meta-HGT: Metapath-aware HyperGraph Transformer for heterogeneous information network embedding. *Neural Netw.* **2023**, *157*, 65–76. [\[CrossRef\]](#)
24. Kang, X.; Li, X.; Yao, H.; Li, D.; Jiang, B.; Peng, X.; Wu, T.; Qi, S.H.; Dong, L.J. Dynamic hypergraph neural networks based on key hyperedges. *Inf. Sci.* **2022**, *616*, 37–51. [\[CrossRef\]](#)
25. Fan, H.; Zhang, F.; Wei, Y.; Li, Z.; Zou, C.; Gao, Y.; Dai, Q. Heterogeneous hypergraph variational autoencoder for link prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 4125–4138. [\[CrossRef\]](#)
26. Cui, P.; Wang, X.; Pei, J.; Zhu, W.W. A survey on network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *5*, 833–853. [\[CrossRef\]](#)
27. Chang, S.y.; Han, W.; Tang, J.L.; Qi, G.J.; Aggarwal, C.C.; Huang, T.S. Heterogeneous network embedding via deep architectures. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 119–128.
28. Arsov, N.; Mirceva, G. Network embedding: An overview. *arXiv* **2019**, arXiv:1911.11726.
29. Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.W.; Yang, S.Q. Community preserving network embedding. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
30. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2001; p. 14.

31. 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.
32. 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.
33. Donnat, C.; Zitnik, M.; Hallac, D.; Leskovec, J. Learning structural node embeddings via diffusion wavelets. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1320–1329.
34. Hu, B.; Fang, Y.; Shi, C. Adversarial learning on heterogeneous information networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage AK, USA, 4–8 August 2019; pp. 120–129.
35. Tsitsulin, A.; Mottin, D.; Karras, P.; Müller, E. Verse: Versatile graph embeddings from similarity measures. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 539–548.
36. Wang, S.H.; Tang, J.L.; Aggarwal, C.R.; Chang, Y.; Liu, H. Signed network embedding in social media. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 327–335.
37. Yadati, N.; Nitin, V.; Nimishakavi, M.; Yadav, P.; Louis, A.; Talukdar, P. NHP: Neural hypergraph link prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1705–1714.
38. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
39. Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; Talukdar, P.P. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In Proceedings of the Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 1509–1520.
40. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
41. Tu, K.; Cui, P.; Wang, X.; Wang, F.; Zhu, W.w. Structural deep embedding for hyper-networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; p. 1.
42. Ma, T.; Suo, Q. Review of Hypernetwork Based on Hypergraph. *Oper. Res. Manag. Sci.* **2021**, *2*, 232.
43. Service Centre of Huiyuan Sharing Academic Resources. University Library Dataset. Available online: <http://hdl.handle.net/20.500.12291/10022> (accessed on 22 July 2021).
44. Zhang, C.X.; Song, D.j.; Huang, C.; Swami, A.; Chawla, N.V. Heterogeneous graph neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage AK, USA, 4–8 August 2019; pp. 793–803.
45. Ding, Y.; Zhang, Z.L.; Zhao, X.F.; Cai, W.; He, F.; Cai, Y.M.; Cai, W.W. Deep hybrid: Multi-graph neural network collaboration for hyperspectral image classification. *Def. Technol.* **2022**, *in press*.
46. Wu, C.; Wu, F.; Cao, Y.; Huang, Y.; Xie, X. Fedgcn: Federated graph neural network for privacy-preserving recommendation. *arXiv* **2021**, arXiv:2102.04925.
47. Huang, C.; Xu, H.C.; Xu, Y.; Dai, P.; Xia, L.H.; Lu, M.Y.; Bo, L.F.; Xing, H.; Lai, X.P.; Ye, Y.F. Knowledge-aware coupled graph neural network for social recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 4115–4122.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.