Hypergraph Embedding Based on Random Walk with Adjusted Transition Probabilities

Kazuya Nagasato[1], Satoshi Takabe[1], and Kazuyuki Shudo[2]

[1] Tokyo Institute of Technology, Japan
[2] Kyoto University, Japan

Abstract. In this paper, we consider embedding hypergraphs using random walks. By executing a random walk on a hypergraph and inputting the resulting node sequence into a skip-gram used in natural language processing, a vector representation that captures the graph structure can be obtained. We propose a random walk method with adjustable transition probabilities for hypergraphs. As a result, we argue that it is possible to embed graph features more appropriately. Experimental results show that by tuning the parameters of the proposed method appropriately, highly accurate results can be obtained even for large hypergraphs for machine learning tasks such as node label classification.

Keywords: Hypergraph · Random walk · Embedding.

1 Introduction

Graphs are powerful mathematical models for quantitatively dealing with the relationships between people and things. For example, hyperlinks on the World Wide Web, friend networks on social networking services, and the structure of chemical compounds are often represented mathematically by graphs, which have been studied actively in recent years. Among these, recent developments in machine learning methods have attracted attention to embedding methods that represent entire graphs or graph nodes as low-order vectors for graphs, which are unstructured data, and have been applied to graph classification, node classification, and prediction of edges acquired by new nodes.

On the other hand, there are relationships in real networks that interact as a group, and these relationships cannot be represented by ordinary graphs in which an edge connects two nodes. Therefore, a mathematical model called a hypergraph, which is a generalization of graphs, is the target of research. Edges in a hypergraph can contain an arbitrary number of nodes, and higher-order relationships between nodes can be captured. The problem of node embedding for hypergraphs is important issue, and it is applied to recommendation systems, etc.

And random walks are useful as a means of capturing node characteristics of graphs and hypergraphs. A random walk on a graph transitions through the nodes that are connected to each other, and thus provides information on the subgraphs of a node's neighbors as a sequence of nodes.

Proc. 25th Int'l Conf. on Big Data Analytics and Knowledge Discovery (DaWaK 2023), August 2023

In this paper, we propose a random walk method with adjustable transition probabilities for hypergraphs and apply it to node embedding. Experimental results on node label estimation show that the proposed method can achieve similar estimation results to existing methods and reduce spatial computational complexity.

The remainder of this paper is organized as follows. Section 2 describes related work on node embedding in graphs using random walks and related work on random walks on hypergraphs. Section 3 describes the prior knowledge in this paper, and Section 4 describes the proposed method. In Section 5, we conduct experiments on a real data set and conclude in Section 6.

2 Related work

In recent years, there has been much research on graph embedding. Spectral embedding [5] is a method based on graph signal processing that produces a vector representation so that the distance of vectors between neighboring nodes is close. There have been many studies on graph neural networks [6], which are derived from this method. However, these methods use $n \times n$ adjacency matrices as input, which are computationally expensive and not suitable for large data sets.

DeepWalk [1] and node2vec [2] are methods for embedding graphs using random walks. These methods obtain a vector representation of each node by inputting a sequence of nodes obtained from a random walk into a skip-gram [4]. These methods are scalable and can perform embedding on large graphs, especially when executed in parallel.

Carletti et al. [3] show that selecting adjacent nodes with equal probability is not a wise decision for random walk transitions in hypergraphs because the connections between nodes belonging to a hyperedge are often stronger than the connections between consecutive individuals. Carletti et al. Therefore, Carletti et al. defined transition probabilities for neighboring nodes based on their weights, which are the number of nodes in the hyper-edge to which they belong minus themselves. Random walks based on this definition can visit many high-order hyperedges [3].

Chitra et al. defined transition probabilities and formulated random walks for hypergraphs with edge-dependent node weights [7]. Chitra et al. show that when node weights are independent of edges in a hypergraph, a random walk on a hypergraph is equivalent to a random walk on a general weighted graph projected onto a clique representation.

Gatta et al. proposed a hypergraph embedding method specifically for music recommendation [19]. They defined a hypergraph data model specific to music recommendation, performs a random walk on the hypergraph, and then inputs the resulting node sequence into a skip-gram to obtain a vector representation of the nodes. The results show that their technique significantly outperforms other state-of-the-art techniques, especially in scenarios where the cold-start problem arise.

Proc. 25th Int'l Conf. on Big Data Analytics and Knowledge Discovery (DaWaK 2023), August 2023

3 Preliminaries

3.1 Notation

In this paper, for a undirected hypergraph $\mathcal{H}(V,E)$, let $V=\{1,...,n\}$ be the set of nodes and $E=\{E_1,...,E_m\}$ be the set of hyperedges, where n is the number of nodes and m is the number of hyperedges. Each E_α is a subset of V and is equivalent to a general graph, especially when $|E_\alpha|=2$ holds for any α.

let

$$e_{i\alpha}=\begin{cases}1 & \text{for } i\in E_\alpha\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

be a hypergraph incidence matrix, the adjacency matrix A of a hypergraph is denoted by $A=ee^T$ and the hyperedge matrix C by $C=e^Te$ where e is a matrix representing the inclusion relationship between hyperedges and nodes, A_{ij} is the number of hyperedges shared by node i and node j, and $C_{\alpha\beta}$ is the number of nodes included in $E_\alpha\cap E_\beta$.

3.2 Hypergraph projection

In a hypergraph, a hyperedge can contain any number of nodes, and thus has a high degree of freedom, making it difficult to treat the hypergraph as it is. Therefore, there has been a lot of research using a method called Clique expansion, which treats hypergraphs as general unweighted undirected graphs by representing hyperedges as complete graphs (cliques) in which all nodes have edges with each other. However, a clique expansion is an irreversible transformation, which results in a considerable loss of information as a hypergraph. Therefore, in this paper, a random walk on an unweighted undirected graph, which is a clique expansion of a hypergraph, is used as a comparison target.

3.3 Random walk and stationary distribution

In general, if we define a node in a graph as a state and define the transition probability T_{ij} from node i to node j, the transition of a node in the graph can be regarded as a random walk. A stochastic process in which the next state is determined only by the current state is called a Markov chain. An ergodic Markov chain has a stationary distribution $\boldsymbol{p}$ and is obtained by solving

$$\boldsymbol{p}=\boldsymbol{p}T. \tag{2}$$

Equation (2) shows that the stationary distribution of the random walk is a right eigenvector corresponding to eigenvalue 1 of the transition probability matrix T.

3.4 Skip-gram

We employ skip-gram as a node embedding method [4]. Skip-gram is a neural network that learns a function $f : V \to \mathbb{R}^d$ to obtain a d-dimensional vector representation of nodes, and was designed to maximize word co-occurrence probability in natural language processing. In this study, we perform a random walk on a hypergraph and use the resulting node sequence as input to obtain a vector representation of the nodes that captures the relationships among the nodes.

4 Proposed method

In this chapter, we propose a random walk whose behavior can be controlled by parameters. By setting parameters that match the dataset and executing a random walk, a set of nodes that capture the graph structure can be obtained. Then, by inputting the obtained nodes into a skip-gram, a vector representation of each node can be obtained.

4.1 Random walk

If the current node is i, then the adjacent node j is assigned the weight $k^H_{ij}(\beta)$ expressed in Equation(3).

$$k^H_{ij}(\beta) = \sum_{\alpha} (C_{\alpha\alpha} - 1)^{\beta} e_{i\alpha} e_{j\alpha} \tag{3}$$

This definition is an extension of the previous study [3] by adding the parameter β as a power.

By normalizing so as to impose a uniform choice among the connected hyperedges, we get the expression for the transition probabilities

$$T_{ij} = \frac{k^H_{ij}(\beta)}{\sum_{l \neq i} k^H_{il}(\beta)}. \tag{4}$$

Equation(4) is a function of β. Depending on the value of β, a random walk with various behaviors is realized. $\beta = 1$ is equivalent to the previous study [3], and a behavior of visiting more high-order hyperedges is expected when β is larger than 1. When β is smaller, it is expected to visit more low-order hyperedges. When $\beta = 0$, adjacent nodes are visited with equal probability, which is equivalent to a random walk in a graph with a hyper-edge clique expansion.

From the above, our hypothesis in the proposed random walk can be summarized as follows.

- $\beta > 1$ Extracts many high-order hyperedge structures
- $\beta = 1$ Equivalent to previous studies [3]
- $\beta < 1$ Extracts many low-order hyperedge structures
- $\beta = 0$ Equivalent to a hypergraph with clique expansion

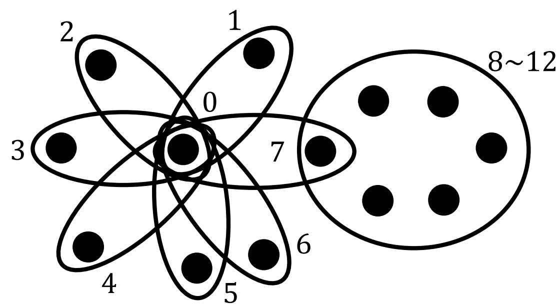

Fig. 1. Artificial hypergraphs. Each number represents a node number.

Also, when the hypergraph is ergodic for a given β, the random walk has a stationary distribution

$$p_j^\infty = \frac{d_j^H(\beta)}{\sum_j d_j^H(\beta)} \tag{5}$$

for all $j = 1, ..., n$, where $d_j^H(\beta)$ is defined as $d_j^H(\beta) = \sum_{l \neq j} k_{jl}^H(\beta)$.

5 Experiment

First, a simple artificial hypergraph is used to verify the behavior of the proposed random walk. Then, we check the accuracy in the label estimation task when the parameters are changed for the three datasets.

5.1 Transition probabilities in steady state

For the artificial hypergraph (Fig.1), the stationary distribution given by equation (5) was computed to verify the random walk behavior.

As a result, when $\beta = -1$, the probability of staying at high-order hyperedges was lower and the probability of staying at hubs included in low-order hyperedges was higher than in the random walk in the clique-expanded hypergraph. On the other hand, when $\beta = 2$, the probability of staying at a node included in a high-order hyperedge was higher than $\beta = 1$ in the previous study [3], and the probability of staying at a hub included in a low-order hyperedge was lower (Fig.2).

Also, examining the probability of staying at each node, higher β induced higher latent probabilities to higher order hyperedges, and vice versa (Fig.2).

These results indicate that our hypothesis is correct and that we can control the random walk behavior by changing the parameter β.

Proc. 25th Int'l Conf. on Big Data Analytics and Knowledge Discovery (DaWaK 2023), August 2023

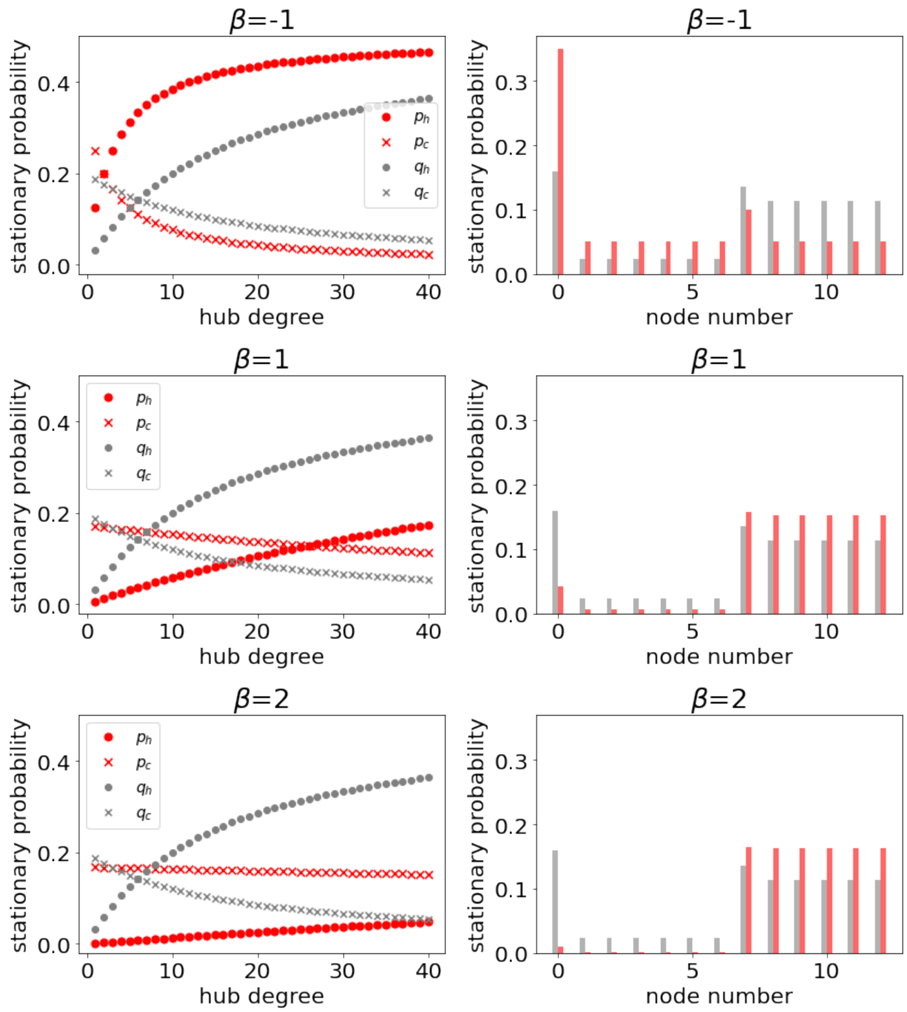

Fig. 2. The left column shows the stationary transition probabilities for node 0 and node 7 when the order of the hub is varied. p_h and q_h represent the stationary transition probabilities for each β at node 0 and node 7, respectively. p_c and q_c represent the stationary transition probabilities for $\beta = 0$ (clique expansion) at node 0 and node 7, respectively. The right column represents the stationary distribution when the degree of node 0 is fixed to 7. The red histograms are for each β, and the gray histograms correspond to $\beta = 0$.

Proc. 25th Int'l Conf. on Big Data Analytics and Knowledge Discovery
(DaWaK 2023), August 2023

Table 1. Hypergraphs used in our experiments.

dataset	n	m	avg$\|E_i\|$	max$\|E_i\|$	labels
senate-bills	294	29,157	8.0	99	2
contact-primary-school	242	12,704	2.4	5	11
mathoverflow-answers	73,851	5,446	24.2	1,784	1,456

5.2 Node label estimation

In this section, we perform the label estimation task on real data. First, we input the sequence of nodes obtained from the proposed random walk into a skip-gram to obtain a vector representation of each node. Then, we perform the task of estimating labels from the vector representation of the nodes using logistic regression. When executing the random walk, the transition probabilities are calculated for each state sequentially, and the walk is performed without explicitly describing the transition probability matrix. This significantly reduces the amount of spatial computation and allows experiments to be performed on large data sets.

The experimental procedure is as follows.

1. A sequence of nodes is obtained from each node in the hypergraph γ times, each with a walk length t, by the proposed random walk.
2. The obtained node sequence is input into a skip-gram to obtain a d-dimensional vector representation for each node.
3. The percentage of supervised data is set to 80% of the number of nodes, and logistic regression is used to estimate the labels of the remaining nodes.
4. (1)∼(3) is repeated s times for each β to obtain the mean and standard error of the F1 score.

We use the datasets [8]-[12] shown in Table 2. The senate-bills dataset is a hypergraph, where nodes are US Congresspersons and hyperedges are comprised of the sponsor and co-sponsors of bills put forth in the Senate. The contact-priary-school dataset represents students in proximity. Each hyperedge corresponds to a group of people that were all in proximity of one another at a given time, based on data from sensors worn by students and teachers. Each node is labeled as a teacher or the classroom to which the student belongs. The mathoverflow-answers dataset is a hypergraph where hyperedges are sets of questions answered by users on Math Overflow. Nodes are labeled by the tags used in the questions, and nodes often have multiple labels.

We set $\gamma \in \{8, 16, 32, 64\}$, $t = 20$, $d = 64$, $\beta \in \{-1, 0, 1\}$, $s = 20$ for senate-bills and contact-private-school datasets. For mathoverflow-answers we set $\gamma \in \{8, 16\}$, $t = 20$, $d = 128$, $\beta \in \{-1, 0, 1\}$, $s = 1$. When $\beta = 0$, the embedding technique is equal to the deepwalk in the clique expansion graph, and when $\beta = 1$, the random walk is equal to the [3] in the previous study.

Fig.3 shows the experimental results for the three data sets. Fig.3 shows that the F1 score varies depending on the value of β. In particular, for the senate-bills dataset, setting $\beta = -1$ yields a high F1 score even with a small

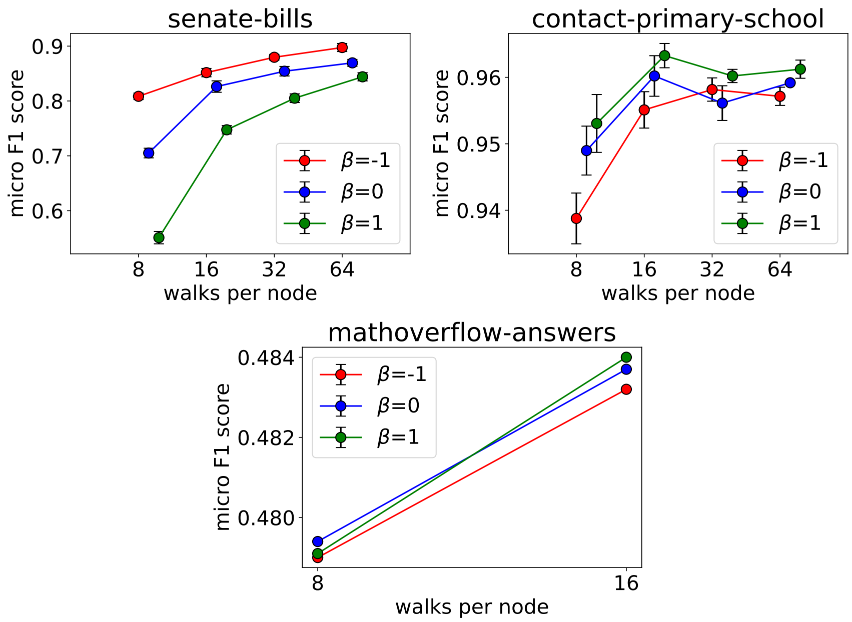

Fig. 3. Experimental results with label estimation for each dataset. Error bars in the figure represent standard errors.

number of samples, which is significantly higher than that of the existing method $\beta = 0$. The proposed method is also able to obtain results on the mathoverflow-answers dataset, which is a large dataset, due to its ability to reduce the spatial computational complexity, and it correctly answers nearly half of the labels on the $1,456$ label count.

5.3 Parameter dependence of F1 score

In this subsection, we show how the F1 score varies with the parameter β. For two datasets, senate-bills and contact-primary-school, we conducted experiments by moving the value of β from -3 to 3 and from -3 to 4 in 0.5 increments, respectively (Fig.4). The range of values of β was kept around 0 based on the hypothesis that random walks would not be able to extract relationships between nodes in these cases because random walks would not be able to escape from high-order hyperedges if β is too large or from low-order hyperedges if β is too small. Fig.4 shows that senate-bills and contact-primary-school are convex functions with maximum values around $\beta = -0.5$ and $\beta = 1.5$, respectively. Therefore, it may be possible to optimize the results using the proposed method by appropriately searching for the parameter β according to the data set.

Proc. 25th Int'l Conf. on Big Data Analytics and Knowledge Discovery (DaWaK 2023), August 2023

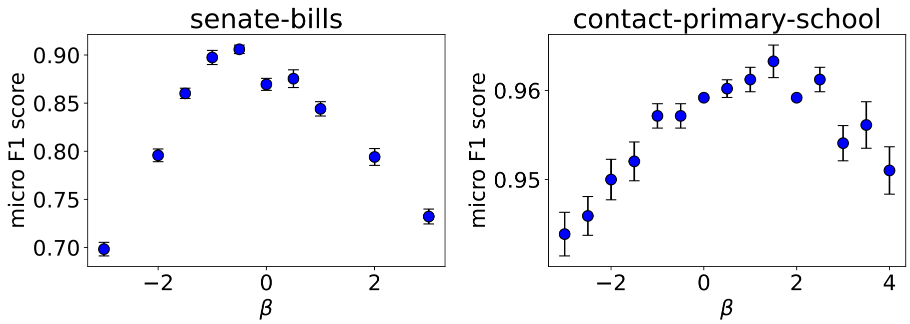

Fig. 4. Dependence of β on F1 score. Error bars in the figure represent standard error.

6 Conclusion

In this paper, we propose a random walk with tunable transition probabilities for hypergraphs and apply it to embedding.

By using an artificial hypergraph, we have confirmed that the proposed random walk can freely exhibit behavior by changing the parameters according to our hypothesis. The proposed random walk can embed even large datasets by executing the random walk on the hypergraph, and it shows higher accuracy than existing methods in the label estimation task using real data. The parameters of the proposed method may be optimized for each dataset, and more appropriate embedding can be achieved by optimizing the parameters.

References

1. B. Perozzi, R. AI-Rfou, and S. Skiena "DeepWalk: Online learning of social representations", In KDD, 2014, p. 701-710.
2. A. Grover and J. Leskovec "node2vec: Scalable feature learning for networks", In KDD. New York: ACM; 2016. p. 855–64
3. T. Carletti, F. Battiston, G. Cencetti and D. Fanelli "Random walks on hypergraphs", Physical Review E, 101 ,2020 , p. 022308.
4. T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space", In ICLR, 2013.
5. L. Tang and H. Liu. "Leveraging social media networks for classification", Data Mining and Knowledge Discovery, 23(3):447–478, 2011.
6. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu. "A Comprehensive Survey on Graph Neural Networks" , IEEE Trans. Neural Netw. Learn. Syst. 32 (1), 4–24.
7. U. Chitra and B. J Raphael. "Random walks on hypergraphs with edge-dependent vertex weights", In Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.
8. P S. Chodrow, N. Veldt and A R. Benson. "Generative hypergraph clustering: from blockmodels to modularity" ,Science Advances, 2021.
9. J H. Fowler. "Connecting the Congress: A Study of Cosponsorship Networks" , James H. Fowler.

10. J H. Fowler. “Legislative Cosponsorship Networks in the U.S. House and Senate” , Social Networks, 2006.
11. J. Stehle, N. Voirin, A. Barrat, V. Vattuto, L. Isella, J. Pinton, M. Quaggiotto, W V. Broeck, C. Regis, B. Lina and P. Vanhems. “High-Resolution Measurements of Face-to-Face Contact Patterns in a Primary School” , PLOS ONE, 2011.
12. N. Veldt, A R. Benson and J. Kleinberg. “Minimizing Localized Ratio Cut Objectives in Hypergraphs” , Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2020.
13. M. Gjoka, M. Kurant, C. T. Butts, A. Markopoulou. “Walking in Facebook: A Case Study of Unbiased Sampling of OSNs” , InProc. IEEE INFOCOM, 1–9.
14. K. Hayashi, S. G. Aksoy, C. H. Park and H. Park. “Hypergraph Random Walks, Laplacians, and Clustering” , CIKM ’20, October 19–23, 2020, Virtual Event, Ireland.
15. D. Zhou, J. Huang, and B.Scholkopf. “Learning with Hypergraphs: Clustering, Classification, and Embedding” , In NIPS, 2007, 1601–1608.
16. Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. “Hypergraph neural networks” , In AAAI, 2019, 3558–3565.
17. M. Belkin and P. Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation” , Neural computation, 15(6):1373–1396, 2003.
18. P. Goyal and E. Ferrara. “Graph embedding techniques, applications, and performance: A survey” , Knowledge-Based Systems, vol. 151, pp.78–94, 2018.
19. V. L. Gatta, V.Moscato, M.Pennone, M.Postiglione, and G.Sperli. “Music recommendation via hypergraph embedding” , IEEE Transactions on Neural Networks and Learning Systems, 2022.

Proc. 25th Int'l Conf. on Big Data Analytics and Knowledge Discovery (DaWaK 2023), August 2023