Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 June 2022 doi:10.20944/preprints202206.0225.v1

Article

Heterogeneous Network Embedding Based on Random Walks of Type & Inner Constraint

Xiao Chen [1,*], Tong Hao [2, 3,*], Li Han [1], Meng Leng [1], Jing Chen [3], and Jingfeng Guo [3]

[1] Research Center of Marine Sciences, Hebei Normal University of Science and Technology, Qinhuangdao 066004, China; chenxiao0604@hevttc.edu.cn (X.C.); hanli@hevttc.edu.cn (L.H.); leng3831@hevttc.edu.cn (M.L.)
[2] School of Public Health and Health Sciences, Tianjin University of Traditional Chinese Medicine, Tianjin 301600, China; haotong@stumail.ysu.edu.cn (T.H.)
[3] College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China; xychenjing@ysu.edu.cn (J.C.); jfguo@ysu.edu.cn (J.G.)
* Correspondence: chenxiao0604@hevttc.edu.cn (X.C.); haotong@stumail.ysu.edu.cn (T.H.)

Abstract: In heterogeneous networks, the random walks based on meta-path requires prior knowledge and lacks flexibility. And the random walks based on non-meta-path only considers the number of node types, but does not consider the influence of schema and topology between node types in real networks. To solve the above problems, this paper proposes a novel model HNE-RWTIC (Heterogeneous Network Embedding Based on Random Walks of Type & Inner Constraint). Firstly, to realize the flexible walks, we design a Type strategy, which is the node type selection strategy based on the co-occurrence probability of node types. Secondly, to achieve the uniformity of node sampling, we design an Inner strategy, which is the node selection strategy based on the adjacency relationship between nodes. The Type & Inner strategy can realize the random walks based on meta-path, the flexibility of the walks, and can sample the node types and nodes uniformly in proportion. Thirdly, based on the above strategy, a transition probability model is constructed; then, we obtain the nodes embedding based on the random walks and Skip-Gram. Finally, in classification and clustering tasks, we conducted a thorough empirical evaluation of our method on three real heterogeneous networks. Experimental results shown that F1-Score and NMI of HNE-RWTIC outperform state-of-the-art approaches.

Keywords: heterogeneous network embedding; random walks; non-meta-path; type and node constraints

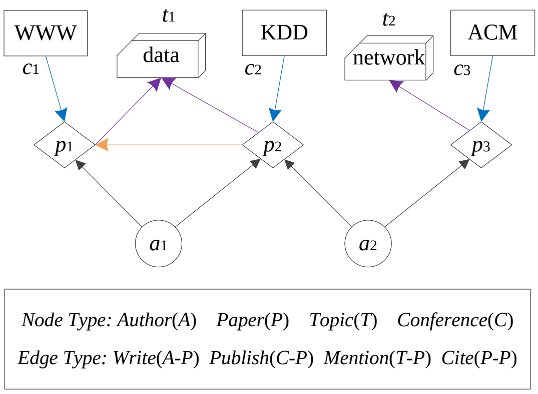
MSC: 68T30

1. Introduction

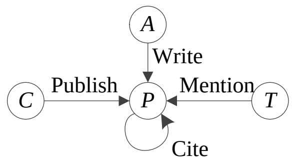
Many systems in the real world can be modeled as Heterogeneous Information Networks (HINs) [1], such as literature and technology networks, social media networks and medical information networks, etc. Among them, the classic DBLP literature and technology network (short for DBLP), in Figure 1(a), can be abstractly described as a heterogeneous network. Compare with homogeneous networks [2], HINs contain multiple types of entities and relationships, and richer semantic information. Therefore, HINs have been widely used in various fields.

With the increase of network scale, traditional methods (such as adjacency matrix) become the bottleneck of network analysis and mining tasks [3]. Due to the increase of semantic and structural information in HINs, the methods of homogeneous networks embedding either cannot be used directly or their complexity increases greatly. In contrast to this, the heterogeneous network embedding [4] can preserve the key structural attributes and the semantic attributes, and mine the potential semantic information. This is also of great significance for completing various network application tasks, such as classification [5, 6], clustering [7, 8] link prediction [9, 10], and so on. Therefore, heterogeneous network embedding has become a current research hotspot.

At present, some methods for heterogeneous network embedding have been proposed, which are mainly divided into three categories, methods based on decomposition [11-14], deep learning [15-20] and random walks [21-26]. Among them, heterogeneous network embedding based on random walk is classical and widely used. It mostly relies on meta-paths guided random walks [22-25]. Meta-path is the embodiment of semantics in HINs. For example, the meta-path information of DBLP is shown in Figure 1(b). There are many meta-paths in HINs. Semantics of "A-P-A" is co-authorship, and "A-P-C-P-A" is two papers published by two authors in the same conference, and so on. Different meta-paths can capture different semantic information, but the number of them increases exponentially as the length increases. The selection of meta-paths either requires domain experts, or the optimization of a set of predefined meta-paths. So that it is necessary in order to try a lot of situations. Determined meta-path limits the flexibility of the random walks. All of these bring significant challenges for the random walks based on meta-path in practical application.



Length	Meta-path	Type
1	*P-P*	Symmetric
	P-A/C/T --- ---	Asymmetric
2	*P-P-P* *P-A/C/T-P* *A/C/T-P-A/C/T*	Symmetric
	A-P-P/C/T *C-P-P/A/T* *T-P-P/A/C* … …	Asymmetric
3	*A/C/T/P-P-P-A/C/T/* ……	Symmetric
	A-P-P-P/C/T *P-P-P-A/C/T* *A/C/T-P-A/C/T-P* … …	Asymmetric
… …	… …	… …



(a) (b) (c)

Figure 1. Instance of DBLP heterogeneous network. (**a**) DBLP network; (**b**) Meta-path; (**c**) Network schema.

Dingqi Yang et al. [26] proposed JUST (**JU**mp & **ST**ay) to solve some problems existing in the meta-path. This is the first method of random walks based on non-meta-path for HINs. It applies a **Jump/Stay** (**Jump** to other type /**Stay** on current node type) strategy when selecting the next node. In Figure 2, the current node type is P, and L=3 means it has stayed in P for 3 times. In this case, the stay probability of the next node in P is α^3. If α^3 is greater than the given threshold, it will remain in P. Otherwise, the Jump strategy is performed. Q_{hist} of size m to memorize up-to-m previously visited types. Q_{hist}={P, A} indicates that the recently visited types are P and A with m=2. Then we randomly sample one type from {T, V} as the target type, where next node is sampled.

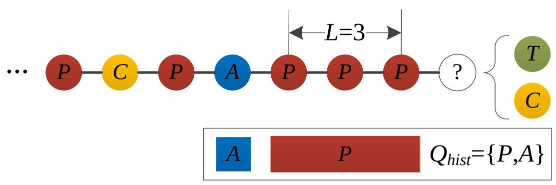


Figure 2. Schematic diagram of random walks strategy of JUST.

In Figure 2, we found some problems in JUST. (1) For the Stay strategy, 0/1 represents the cases that cannot/must Stay in the current node type. In other cases, α^l limits the probability of staying in the current node type, but without considering the schema of networks. For example, in DBLP, it can be seen from Figure 1(c) that only *P* can stay in its own type, while *A*, *T* and *C* cannot stay. (2) For the Jump strategy, the node types that in {Q-Q_{hist}} are given priority selection. But, the types are not considered to meet the Jump requirements. If the current random walk sequence is "···-*P*-*P*-*P*-*A*-?···", and Q_{hist}={*P*, *A*}, then the next node type will be *T* or *C*. But actually, *A* only has edge with *P*. So, *A* can only jump to *P*, where it makes no sense to consider type first. (3) JUST only provides the selection of node type, but not consider the node selection. When the above problems arise, JUST shows great limitations.

In view of the above problems, we design a novel random walks strategy based on Type & Inner constraint, which considers the node type and the adjacency relationship between nodes.

The main contributions of this paper can be summarized as following.

- We propose a novel model HNE-RWTIC. It performs the random walks based on Type & Inner constraint, and adopts Skip-Gram to learn dense and low-dimensional embedding in HINs.
- We propose a novel random walks strategy based on Type & Inner constraint. In Type strategy, the node types selection considers the co-occurrence probability of nodes. In Inner strategy, the nodes selection considers the adjacency relationship between nodes. The strategy realizes the flexibility of node types selection in HINs (see 5), and the uniformity of proportional sampling between types and nodes (see 6.2.4).
- We build a transition probability model based on the Type & Inner strategy. In the model, parameters control the selection of node type and node. Then, some properties are obtained. They indicate the relationship between the parameter value and the node type or node selection.
- Using DBLP, AMiner-Top and Yelp, we conduct the experiments in classification and clustering tasks. By comparing with five classic networks embedding algorithms, the correctness and effectiveness of HNE-RWTIC are verified.

2. Related Work

In 2014, based on the natural language processing (NLP) model Word2Vec [27], Bryan Perozzi et al. first proposed the network embedding model Deepwalk [3]. Up to now, many methods of network embedding have been proposed. Among them, the methods are mainly divided into three categories: methods based on decomposition, deep learning and random walks. The first methods usually decompose HINs into simple networks, learns the embedding of these networks respectively, and integrates them. The second methods use the deep neural network model to obtain the embedding. The third methods usually combine the random walk and Skip-Gram for embedding. Due to the limited space, we mainly introduce the method based on random walk which is closely related to this paper.

According to the different strategies, the heterogeneous network embedding methods based on random walks can be divided into two categories.

- Random walks based on meta-path. For example, Metapath2Vec [1] manually select "*A*-*P*-*A*" or "*A*-*P*-*C*-*P*-*A*" as meta-path to guide the random walks in literature and technology networks. HERec (Heterogeneous network Embedding based approach for Recommendation) [14] uses a set of predefined meta-paths to guide the random walks. ESim (Embedding for Similarity) [22] considers the information of multiple meta-paths and attempts to learn the optimal weight combine to guide the random walks. HIN2Vec (Heterogeneous Information Networks to Vectors) [25] uses the different type relations between nodes to combine meta-paths shorter than a certain length to guide the random walks. HeteSpaceyWalk [28] systematically formalizes the random walks based on meta-path into a high-order Markov chain process, and proposes a heterogeneous personalized space random walk. These techniques are designed to optimize meta-paths. But they still require specific meta-path. The experimental results show that the quality of node embedding is sensitive to the meta-path.

- Random walks based on non-meta-path. For example, JUST can balance the sampling distribution of different node types in random walks. However, JUST only considers how to select the node type, and does not consider how to select the nodes within type. The advantage is that the generated walk sequences are regarded as articles in NLP, and the existing NLP model can be utilized to learning. The disadvantage is the case that the quality of the random walks affects the performance of embedding.

There are few methods for heterogeneous network embedding. Most of them are based on random walks. However, there are also a lot of problems to be improved in this kind of method. Inspired by the idea of selecting node type in JUST and the priority search strategy in Node2Vec [2], we propose a novel random walks strategy that can balance the selection of node type and node.

3. Preliminary Knowledge and Problem Definition

Definition 1. *Heterogeneous network is defined as* $G = (V, E, \mathbf{A}, \mathbf{R})$*. Where,* V *is the node set,* E *is the edge set,* $\mathbf{A}=\{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n, \ldots, \mathbf{A}_N\}$ $(N\leq|V|)$ *is the node type set, and* $\mathbf{R}$ *is the edge type set. For each node* $v_i \in V$*, it belongs to a specific node type, denoted by* $\varphi(v_i)=\mathbf{A}_n \in \mathbf{A}$ $(1\leq n\leq N)$*. Where* $N=|\mathbf{A}|$ *is the number of node types. For each edge* $e_j=(v_i, v_j) \in E$*, it belongs to a specific relation type, denoted by* $\psi(e_j) \in \mathbf{R}$*. Where,* $M=|\mathbf{R}|$ *is the number of edge types. It is generally believed that heterogeneous information networks satisfy* $M>1$ *or* $N>1$*.*

Definition 2. *The network schema* [29] *is denoted as* $T_G=(\mathbf{A}, \mathbf{R})$*. Which is a meta template for a heterogeneous network* $G=(V, E, \mathbf{A}, \mathbf{R})$ *with the object type mapping* $\varphi: V \rightarrow \mathbf{A}$ *and the link type mapping* $\psi: E \rightarrow \mathbf{R}$*.*

Definition 3. *Given a heterogeneous network* $G=(V, E, \mathbf{A}, \mathbf{R})$*, the heterogeneous network embedding is to learn a mapping function* $f: V \rightarrow X \in R^{|V|\times d}$*,* $d<<|V|$*, so as to obtain the vector representation of nodes in the network. The vector representation can capture the structural and semantic relationships between nodes in the network.*

The purpose of this paper is to study the method of heterogeneous network embedding based on random walks. Firstly, the random walks strategy is determined and described as a transition probability model. Secondly, the sequence W is obtained through the random walks. Then, the obtained W is combined with the Skip-Gram model to learn the embedding of nodes in HINs.

4. HNE-RWTIC Model

In this section, the random walks strategy and transition probability model are introduced in detail, and the algorithm description of HNE-RWTIC is given.

4.1. Random Walks Strategy

Due to the characteristics of HINs and the problems existing in meta-path and JUST, we design a random walks strategy based on Type & Inner. This strategy is divided into three steps:

- **Node type partitioning strategy**. According to the network schema and the research purpose, the node types are divided into objective class and non-objective class.
- **Type strategy.** In the strategy of node type selection, the co-occurrence probability of three consecutive node types in walking sequence is considered. Three consecutive node types are the previous node type, the current node type and the next node type. And the next node type is selected with the largest probability value.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 June 2022 doi:10.20944/preprints202206.0225.v1

- **Inner strategy.** In the strategy of nodes selection, based on the adjacency relationship of three consecutive nodes, the probability value of backtracking, breadth or depth is calculated. Then, the next node is selected by the largest value.

Thus, the node type partitioning strategy can solve the JUST problem (2), which the preferred type not meeting the jump requirement. Type strategy can solve the JUST problem (1), which the problem of being confined in the current node type. Inner strategy can solve the JUST problem (3), which is not considering how to select the next node. The next work is how to build the transition probability model based on the strategy.

4.2. Transition Probability Model

4.2.1. Node Type Partitioning

In this paper, the HINs are unsigned. Therefore, the node type needs to satisfy $N \geq 2$. In order to better select the node type and the next node, node type partitioning is necessary in HINs.

Based on network schema and application, node types are divided into objective and non-objective class. Where, the objective class is the type of the entity being studied or the type connected to most classes in the network, denoted as O. The rests are non-objective class, denoted as $\bar{O}$. Then, in Definition 1, the set of node types is also denoted as $\mathbf{A}=O \cup \bar{O}$ $(O \cap \bar{O} = \emptyset)$. Where, $O = \{O_1, O_2, \dots, O_{n_1}\}$, $\bar{O} = \{\bar{O}_1, \bar{O}_2, \dots, \bar{O}_{n_2}\}$ which are equivalent to $O=\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n\}$, $\bar{O}=\{\mathbf{A}_{n+1}, \mathbf{A}_{n+2}, \dots, \mathbf{A}_N\}$. So, $n_1+n_2=N$, $n_1=n$, $n_2=N-n$. In this study, $n_1 \geq 1$ and $n_2 \geq 1$ are required.

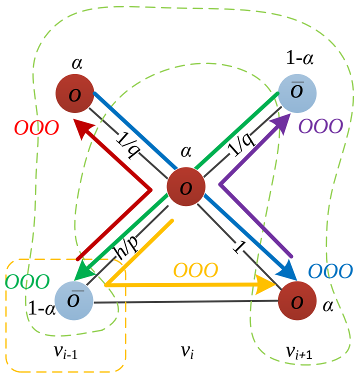


Figure 3. Illustration of the random walks model.

In HINs, $\forall v_i \in V$, if $\varphi(v_i) \in O$, the type of node v_i is the objective class. Otherwise, the type of node v_i is the non-objective class, denoted by $\varphi(v_i) \in \bar{O}$. During the random walks, the stay probability of the node type is in Equation (1).

$$P_{Stay} = \begin{cases} \alpha, & O \\ 1-\alpha, & \bar{O} \end{cases} \tag{1}$$

In Equation (1), $\alpha \in [0, 1]$ is the probability that the node stays at O, and 1-α is the probability that the node stays at $\bar{O}$, as shown in Figure 3.

4.2.2. Transition Probability Model

Given a G, start node v_0 and walks length L, we carry out the random walks. v_{i-1} and v_i are the i-1 and i node in the path. The transition probability of v_{i+1} is shown in Equation (2).

$$P(v_{i+1}|v_i, v_{i-1}) = P_{Type}(\varphi(v_{i+1})|\varphi(v_i), \varphi(v_{i-1}))P_{Inner}(v_{i+1}|v_i, v_{i-1}), \tag{2}$$

Where, P_{Type} is the selection probability of the $\varphi(v_{i+1})$, P_{Inner} is the selection probability of node v_{i+1}. v_{i-1}, v_i and v_{i+1} represent the previous node, the current node and the next node. $\varphi(v_{i-1})$, $\varphi(v_i)$ and $\varphi(v_{i+1})$ represent the previous node type, the current node type and the next node type.

Note: P_{Type}, P_{Inner}, v_{i-1}, v_i, v_{i+1}, $\varphi(v_{i-1})$, $\varphi(v_i)$ and $\varphi(v_{i+1})$, are used as simple descriptions below.

1. The probability of selecting node type

In the random walks, we use the parameters α and k to control the transition probability between node types. Given a G, when $\varphi(v_i)$ and $\varphi(v_{i-1})$ are known, the probability of $\varphi(v_{i+1})$ is shown in Equation (3).

$$P_{Type}(\varphi(v_{i+1})|\varphi(v_i),\varphi(v_{i-1})) = \begin{cases} \alpha^3, & OOO \\ (1-\alpha)\alpha^2, & OO\bar{O} \text{ or } \bar{O}OO \text{ or } O\bar{O}O \\ k(1-\alpha)^2\alpha, & \bar{O}O\bar{O} \end{cases} \quad (3)$$

In Equation (3) and Figure 3, when OOO, the probability of $\varphi(v_{i+1})$ is α^3. OOO is short for $\varphi(v_{i-1})\in O$, $\varphi(v_i)\in O$ and $\varphi(v_{i+1})\in O$, and other cases are the same. Then, the probability of $\varphi(v_{i+1})\in\bar{O}$ is $\alpha^2(1-\alpha)$ when $OO\bar{O}$. The probability of $\varphi(v_{i+1})\in O$ is $(1-\alpha)\alpha^2$ when $\bar{O}OO$. The probability of $\varphi(v_{i+1})\in O$ is $\alpha(1-\alpha)\alpha$ when $O\bar{O}O$. The probability of $\varphi(v_{i+1})\in\bar{O}$ is $(1-\alpha)\alpha(1-\alpha)$ when $\bar{O}O\bar{O}$.

In Equation (3) and Figure 3, the probability of $\varphi(v_{i+1})$ is divided into five cases. We can see when $\varphi(v_i)\in O$, there are four cases. Otherwise, there is only one case. Since in the current HINs study, there are no edges between $\bar{O}$. When $\varphi(v_i)\in\bar{O}$, $\varphi(v_{i+1})$ can only be O.

In Equation(3), when $\varphi(v_{i-1})\in\bar{O}$ and $\varphi(v_{i+1})\in\bar{O}$, parameter k is used to adjust the influence of $\bar{O}$ on the selection of $\varphi(v_{i+1})$ as shown in Equation (4). When $N=|\mathbf{A}|=2$, there are only two types in HINs. In this case, $k=1$. When $N>2$, the $\bar{O}$ contains multiple seed types, $\varphi(v_{i-1})=\varphi(v_{i+1})$ or $\varphi(v_{i-1})\neq\varphi(v_{i+1})$ exists. When $\varphi(v_{i-1})=\varphi(v_{i+1})$, let $k=k_1$, $k_1\in(0,+\infty)$. Otherwise, $k=1/k_1$.

$$k = \begin{cases} 1, & N=2 \\ k_1, & N>2 \text{ and } \varphi(v_{i-1})=\varphi(v_{i+1}) \\ \dfrac{1}{k_1}, & N>2 \text{ and } \varphi(v_{i-1})\neq\varphi(v_{i+1}) \end{cases} \quad (4)$$

2. The probability of selecting nodes

After determining the node type, we consider the adjacency relationship between v_{i+1}, v_i and v_{i-1}, and adopt parameters h, p and q to control the backtracking, breadth or depth of the node. Therefore, the transition probability of v_{i+1} is shown in Equation (5).

$$P_{Inner}(v_{i+1}|v_i,v_{i-1}) = \begin{cases} \dfrac{h}{p}, & d(v_{i-1},v_{i+1})=0 \\ 1, & d(v_{i-1},v_{i+1})=1 \\ \dfrac{1}{q}, & d(v_{i-1},v_{i+1})=2 \end{cases} \quad (5)$$

In Equation (5) and Figure 3, $d(v_{i-1}, v_{i+1})$ is the shortest distance from v_{i-1} to v_{i+1}. $d(v_{i-1}, v_{i+1})=2$ represents v_{i+1} is a neighbor of v_i but not a neighbor of v_{i-1}, then the probability of v_{i+1} is $1/q$. Where, $q\in(0,+\infty)$ controls the breadth or depth. When $q>1$, the node adopts breadth-first. Otherwise, the node adopts depth-first. $d(v_{i-1}, v_{i+1})=1$ represents v_{i+1} is a common neighbor of v_i and v_{i-1}, then the probability of v_{i+1} is 1. $d(v_{i-1}, v_{i+1})=0$ represents v_{i+1} is v_{i-1}, then the probability of v_{i+1} is h/p. Where, $p\in(0,+\infty)$ and $h\in\{0, 1\}$ are return parameters that control the probability of returning v_{i-1}. When $p>max(q, 1)$, it tends not to return v_{i-1}. When $p<min(q, 1)$, it tends to return v_{i-1}. h is set as shown in Equation (6). When OOO ($\varphi(v_{i-1})=\varphi(v_{i+1})$) or $\bar{O}O\bar{O}$ ($\varphi(v_{i-1})=\varphi(v_{i+1})$), $h=1$. In this case, v_{i+1} may be v_{i-1}. In the rest of the cases, $h=0$, v_{i+1} cannot be v_{i-1}.

$$h = \begin{cases} 1, & (OOO \text{ or } \bar{O}O\bar{O}) \text{ and } (\varphi(v_{i-1})=\varphi(v_{i+1})) \\ 0, & other \end{cases} \quad (6)$$

4.3. Algorithm Description of HNE-RWTIC

The description of algorithm HNE-RWTIC is shown in Algorithm 1. In Algorithm 1, two key works are included. (1) According to $\varphi(v_i)$, $\varphi(v_{i-1})$, parameters α and k, the next node type $\varphi(v_{i+1})$ is selected, at line 12. (2) According to parameters q, p and h, the next node v_{i+1} is selected, at line 13.

Algorithm 1 HNE-RWTIC

Input: G, probability parameter α, control parameter k_1, return parameter p, controlling search mode parameter q, walks length L, each node as the start node times r, node vector dimension *dim*, window size *window*.

Output: The embedding of nodes $\Phi = R^{|V| \times d}$.

```
1.   W = Ø
2.   V = shuffle(G.nodes())
3.   for i = 1 to r do
4.       for v in V do
5.           path = Ø
6.           for walk_iter = 1 to L do
7.               v_i = path[-1]
8.               if walk_iter == 1 then
9.                   path = path∪v
10.              else if walk_iter < L then
11.                  v_{i-1} = path[-2]
12.                  next_node_type(G, L, v_i, v_{i-1}, α, k_1)
13.                  next_node(G, q, p, v_{i-1}, next_node_type)
14.                  path = path∪next_node
15.              W = W∪path
16.  end for
17.  Φ = SkipGram(W, dim, window)
18.  return Φ
```

5. Properties and Analysis of Transition Probability Model

5.1. Properties

After constructing the transition probability model based on Type & Inner strategy, we get the following properties of parameters α, k, p, q and h selected with $\varphi(v_{i+1})$ and v_{i+1}.

Property 1. *When $\varphi(v_{i-1}) \in \{O \cup \bar{O}\}$ and $\varphi(v_i) \in O$, with the increase of $\alpha \in [0, 1]$, the selection of $\varphi(v_{i+1})$ tends to change from $\bar{O}$ to O.*

We can see from Property 1 and Equation (3), without considering the influence of parameter k, $\varphi(v_{i+1})$ is affected by α, as shown in Figure 4(a).

1. For $\varphi(v_{i-1}) \in O$, when α<0.5, $\varphi(v_{i+1})$ tends to jump $\bar{O}$. When α>0.5, $\varphi(v_{i+1})$ tends to stay at O.
2. For $\varphi(v_{i-1}) \in \bar{O}$, when α<0.5, $\varphi(v_{i+1})$ tends to jump $\bar{O}$. When α>0.5, $\varphi(v_{i+1})$ tends to stay at O.
3. When α=0.5, $\varphi(v_{i+1})$ is selected randomly.

Property 2. *When $\varphi(v_{i-1})\varphi(v_i)\varphi(v_{i+1})$ is $\bar{O}O\bar{O}$ and N>2, with the increase of $k \in (0, +\infty)$, $\varphi(v_{i+1})$ and $\varphi(v_{i-1})$ tend to change from different to same.*

 doi:10.20944/preprints202206.0225.v1

We can see from Property 2 and Equation (4), the relationship between $\varphi(v_{i+1})$ and $\varphi(v_{i-1})$ is affected by k, as shown in Figure 4(b).

1. When $0<k_1<1$, $\varphi(v_{i+1})$ does not tend to equal $\varphi(v_{i-1})$, that is $\varphi(v_{i+1})\neq\varphi(v_{i-1})$.
2. When $k_1>1$, $\varphi(v_{i+1})$ tends to equal $\varphi(v_{i-1})$, that is $\varphi(v_{i+1})=\varphi(v_{i-1})$.
3. When $k_1=1$, $\varphi(v_{i+1})$ is randomly selected.

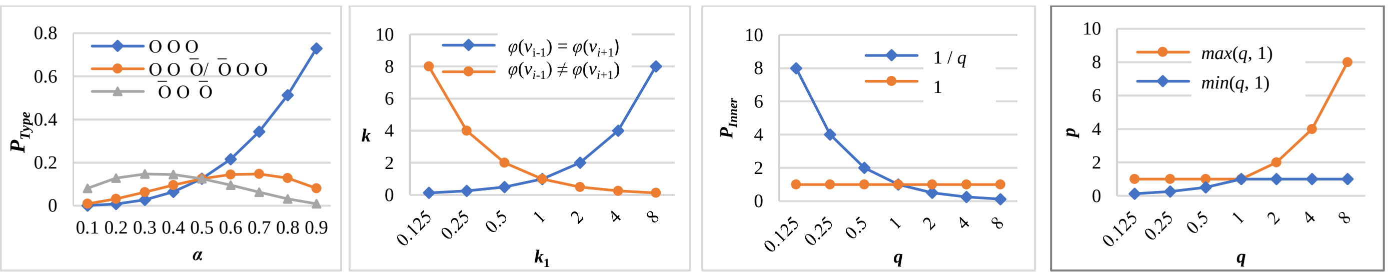
Property 3. *With the increase of $q \in (0, +\infty)$, the selection of v_{i+1} tends to change from depth to breadth.*

Property 4. *With the increase of $p \in (0, +\infty)$, the selection of v_{i+1} tends to change from backtracking to non-backtracking.*

According to Property 3 and 4, and Equation (5), ignoring the influence of h, we consider the adjacency relationship of v_{i-1}, v_i and v_{i+1}. Parameter q controls the preference of breadth-first or depth-first. p controls are backtracking. The influence of p and q on v_{i+1} are shown in Figure 4(c) and Figure 4(d).

1. When $0<q<1$, v_{i+1} tends to depth-first. When $q>1$, v_{i+1} tends to breadth-first.
2. When $p>max(q, 1)$, v_{i+1} does not tend to return v_{i-1}. When $p<min(q, 1)$, v_{i+1} tends to return v_{i-1}.
3. When $p=q=1$, v_{i+1} is randomly selected.

Property 5. *When $d(v_{i-1}, v_{i+1}) = 0$, parameter h controls whether v_{i-1} can be returned. In this case, when h=0, v_{i+1} cannot return v_{i-1}. When h=1, v_{i+1} might return v_{i-1}.*



(a) (b) (c) (d)

Figure 4. Properties analysis. (**a**) α; (**b**) k; (**c**) q; (**d**) p.

5.2. Analysis

Meta-path is usually used to guide the random walks in HINs. For example, in DBLP, we generally choose "*APA*" or "*APCPA*" as the meta-path in experiments. It can be seen that the meta-path essentially only considers node types. We also describe the selection strategy of node type in Equation (3) and (4). According to Property 1 and 2, the strategy can realize random walks with specified meta-path and other paths. In this section, DBLP is taken as an example to make a comparative analysis with meta-path "*APCPA*".

In DBLP, the types of "*APCPA*" are A, P and C. According to the network schema, O={P}, $\bar{O}$={A, C}. For Equation (4), in this case, N=3. so, $k=k_1$ or $k=1/k_1$.

According to "*APCPA*", the starting node type of the random walks can only be A. When $k_1<1$, according to Property 2 and Equation (4), it can be known that $k_1<1/k_1$. At this time, the path type is $\bar{O}O\bar{O}$. Since $\bar{O}$={A, C} and $\varphi(v_{i-1})\neq\varphi(v_{i+1})$, therefore, $\bar{O}O\bar{O}$=*APC*. In this case, $\varphi(v_{i-1})=P$ and $\varphi(v_i)=C$. According to the network schema (Figure 1(c)), $\varphi(v_{i+1})=P$, the sequence *APCP* can be obtained. Now, $\varphi(v_{i-1})=C$ and $\varphi(v_i)=P$. Finally, according to Property 2 and Equation (4), $\varphi(v_{i+1})=A$. To sum up, we get "*APCPA*".

Similarly, other meta-paths can be realized by this random walks strategy. Therefore, the Type & Inner strategy is more flexible than meta-path.

6. Experiments

6.1. Experimental Setup

In experiments, three real HINs datasets are used, which are DBLP [27], Aminer-Top [21] and Yelp [16]. Five classical network embedding algorithms are selected, which are Node2Vec [2], HIN2Vec [25], Metapath2Vec [1], JUST [26] and HeGAN [15].

In the above algorithms, the general parameters are consistent with those in the DeepWalk, which are each node as the start node times r=10, walks length L=100 and node vector dimension dim=128.

The unique parameters of the five algorithms are described as following. In Node2Vec, the bias parameters are p=1 and q=1. In HIN2Vec, the range of the meta-path length is 1-4. In Metapath2Vec, DBLP and Aminer-Top network use the meta-path "*A-P-C-P-A*" suggested by the author, and Yelp uses "*B-U-B*". In JUST, the stay probability is set to $\alpha \in [0.2, 0.5]$, and the number of node types recorded recently visited is set to m=1. And in HeGAN, iteration time is $epoch$=20; the training times of discriminator and generator are n_G=5 and n_D=15, Gaussian variance is σ^2=1.

6.2. Experimental Results and Analysis

In this section, we conducted experimental comparison with five classical algorithms in three HINs from classification and clustering. Then, HNE-RWTIC is analyzed by parameter sensitivity analysis. Finally, the uniformity experiment was carried out on DBLP.

6.2.1. Classification

The goal of classification is to predict the most likely tag of some nodes based on the nodes with labels. In experiments, we divided the datasets into training sets and test sets. The training set considers 100%, 80%, 60%, 40% and 20% of the dataset, and the rest for testing. And we train a one-vs-rest logistic regression classifier to predict the labels of the test nodes. And we compare the predicted results with their true labels. 4 categories of authors in DBLP and 3 categories of businesses in Yelp are classified. There are 8 categories of authors in Aminer-Top, but because the number of authors with tags in some categories is too low, we chose the 3 categories with the most tags to classify. We repeated the experiment for 10 times and took the mean value of Micro-F1 and Macro-F1 as the experimental results in Table 1. Bold indicates the maximum value and italics the next maximum value.

In Table 1, the results show that:

- Compared with the other five classical algorithms, HNE-RWTIC has better classification results in the three networks.
- With the increase of the proportion of the training set, the values of Micro-F1 and Macro-F1 of each algorithm increased significantly. When the training set is 100%, the result is the best.
- HNE-RWTIC performed better than the baseline methods in 80% of the training sets. In DBLP, Aminer-Top and Yelp, the values of Micro-F1 and Macro-F1 are the highest: 2.25% and 2.43%, 0.85% and 0.99%, 3.77% and 5.02% than those of other algorithms.
- Compared with DBLP and Aminer-Top, the Micro-F1 and Macro-F1 value of HNE-RWTIC in Yelp improved more.

Because JUST and random walks based on meta-path only consider the node type and ignore the node distribution balance. The problem of unbalanced node distribution is less affected when the node types and the number of edges in the network are large, because rich semantic relationships can make up for this shortcoming. However, Yelp only contains two types of nodes and one type of edges, and the quality of walking sequences will be seriously affected. Detailed analysis refers to the analysis of parameters p and q in 5.2.3.

Table 1. Classification compared with different algorithms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 June 2022 doi:10.20944/preprints202206.0225.v1

Data sets	Index	Train set(%)	Node2Vec	HIN2Vec	Metapath2Vec++	JUST	HeGAN	HNE-RWTIC
DBLP	Micro-F1	100	0.9099	0.9120	0.8916	*0.9125*	0.9026	**0.9157**
		80	*0.8555*	0.8508	0.8545	0.8461	0.8414	**0.8639**
		60	0.8340	0.8420	*0.8492*	0.8393	0.8378	**0.8589**
		40	0.8085	0.8267	**0.8455**	0.8006	0.8057	*0.8272*
		20	0.7908	0.7907	**0.8304**	0.7872	0.7793	*0.7978*
	Macro-F1	100	0.9097	0.9123	0.8919	*0.9126*	0.9028	**0.9159**
		80	0.8546	0.8510	*0.8552*	0.8453	0.8399	**0.8642**
		60	0.8336	0.8417	*0.8497*	0.8393	0.8371	**0.8585**
		40	0.8084	0.8270	**0.8460**	0.8004	0.8057	*0.8273*
		20	0.7912	0.7913	**0.8308**	0.7876	0.7794	*0.7988*
AMINER - TOp	Micro-F1	100	**0.9058**	0.9030	0.8993	0.8797	*0.9036*	0.9000
		80	0.8971	0.8965	0.8959	0.8894	*0.8977*	**0.8979**
		60	0.8957	0.8957	0.8954	0.8777	*0.8959*	**0.8961**
		40	**0.8949**	0.8936	0.8941	0.8789	0.8924	*0.8945*
		20	0.8899	0.8907	**0.8912**	0.8705	0.8807	*0.8910*
	Macro-F1	100	**0.8969**	*0.8940*	0.8896	0.8682	*0.8943*	0.8907
		80	0.8862	0.8876	0.8854	0.8789	*0.8879*	**0.8888**
		60	0.8849	0.8852	0.8851	0.8664	*0.8856*	**0.8865**
		40	**0.8857**	0.8845	0.8837	0.8681	0.8819	*0.8846*
		20	0.8795	*0.8809*	0.8806	0.8582	0.8696	**0.8810**
Yelp	Micro-F1	100	0.7410	0.7441	**0.7490**	0.7184	0.7330	*0.7460*
		80	0.6805	*0.6989*	0.6987	0.6743	0.6973	**0.7121**
		60	0.6715	0.6936	*0.6944*	0.6672	0.6890	**0.7107**
		40	0.6637	*0.6813*	0.6762	0.6368	0.6803	**0.6939**
		20	0.6257	0.6500	0.6363	0.5941	*0.6661*	**0.6800**
	Macro-F1	100	0.7099	*0.7180*	**0.7187**	0.6785	0.6950	0.7157
		80	0.6385	*0.6625*	0.6596	0.6197	0.6521	**0.6698**
		60	0.6319	*0.6560*	0.6546	0.6150	0.6475	**0.6677**
		40	0.6203	*0.6456*	0.6382	0.5845	0.6326	**0.6481**
		20	0.5834	0.6096	0.5959	0.5425	*0.6159*	**0.6356**

6.2.2. Clustering

The goal of clustering is to aggregate similar nodes into the same community. In experiments, K-means was used to cluster nodes, and NMI was used to evaluate the results. Due to the sparse network and large community size, the NMI value is very small. This makes it difficult to compare different approaches. Therefore, referring to Metapath2Vec, we only selected the nodes of the two largest communities to calculate NMI values for DBLP and Yelp, and three largest communities for AMiner-top. Their community proportions are 53.2%, 82.01% and 81.09%, respectively. We repeated the experiment for 10 times and took the average value of NMI in Table 2. The bold indicates the maximum value and italics the next maximum value.

Table 2. Clustering compared with different algorithms.

Dataset	Node2vec	HIN2Vec	Metapath 2Vec++	JUST	HeGAN	HNE-RWTIC
DBLP	***0.7974***	0.6179	0.7600	0.7784	0.7666	**0.8091**
AMiner-Top	***0.4586***	0.4206	0.4519	0.3999	0.4495	**0.4689**
Yelp	0.0014	***0.0016***	0.0015	0.0014	0.0015	**0.0018**

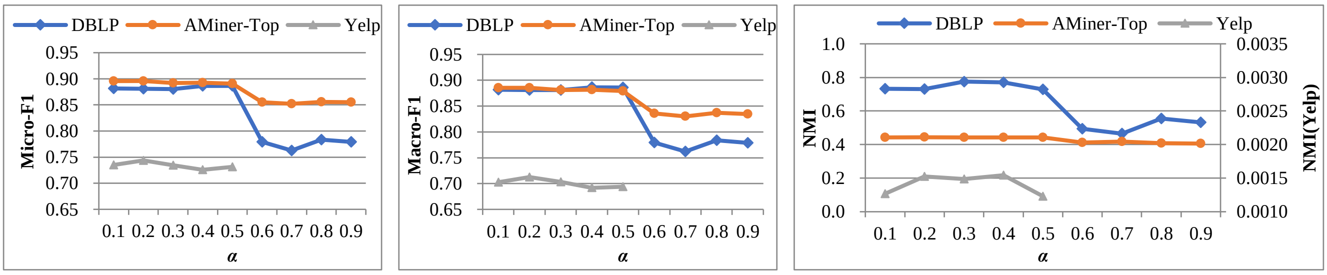
In Table 2, HNE-RWTIC has better clustering results in the three networks. It has the highest NMI values of 19.12%, 6.91% and 0.0439% higher than other algorithms in DBLP, Aminer-Top and Yelp respectively.

6.2.3. Parametric Sensitivity Analysis

In the classification and clustering of the three datasets, Micro-F1, Macro-F1 and NMI were used to analyze the sensitivity of the four main parameters of HNE-RWTIC, which are α, k_1, p, and q. The results are shown in Figure 8-11. The ordinate in the figure respectively represent the results of classification micro-F1 and macro-F1 values and clustering NMI values.

1. Probability parameter α

Experimental results of α are shown in Figure 8. The abscissa represents the range of parameter α, $\alpha \in [0.1, 0.9]$, and the step size is 0.1. In Yelp, when α>0.5, it tends to stay in the current node type, but there is no *B-B* or *U-U* edge in the real network, so $\alpha \in [0.1, 0.5]$.



(a) (b) (c)

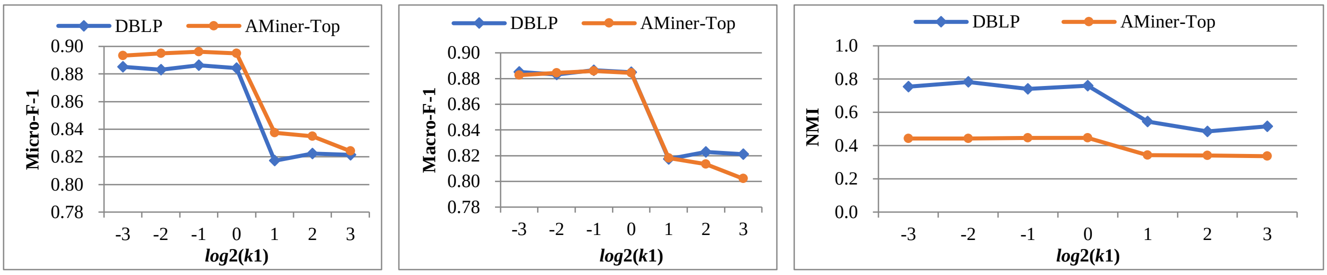
Figure 8. Sensitivity analysis of α. (**a**) Micro-F1 of classification; (**b**) Macro-F1 of classification; (**c**) NMI value of clustering.

Figure 8 shows when $\alpha \in [0.1, 0.5]$, both classification and clustering have good results. In DBLP, Aminer-Top and Yelp, when α is 0.4, 0.4 and 0.2, the classification results are the best. At this point, the values of Micro-F1 and Macro-F1 are 0.8863 and 0.8864, 0.8958 and 0.8855, 0.7437 and 0.7125, respectively. When α is 0.3, 0.2 and 0.4, the clustering results are the best, and NMI value is 0.7752, 0.4436 and 0.0015, respectively.

When the parameter α is smaller, the experimental results are better. The smaller α makes the nodes tend to different types in random walks, that is, to heterogeneous edges. On the contrary, when α is larger, the quality of node embedding will be affected because there are many homogeneous edges. The results show the balance of edge types can be achieved by adjusting α.

2. Control parameter k_1

Experimental results of k_1 in DBLP and Aminer-Top are shown in Figure 9. The abscissa represents the range of *log* values for parameter k_1, $log_2(k_1) \in [-3, 3]$, and the step size is 1.



(a) (b) (c)

Figure 9. Sensitivity analysis of k_1. (**a**) Micro-F1 of classification; (**b**) Macro-F1 of classification; (**c**) NMI value of clustering.

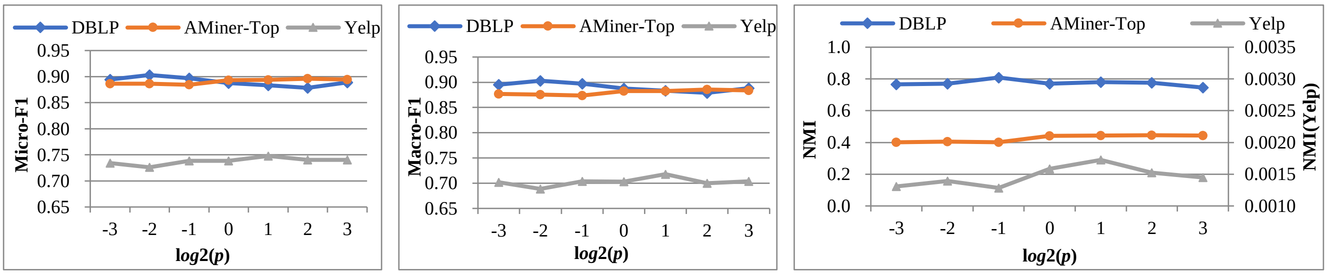
Figure 9 shows that when $log_2(k_1) \in [-3, 0]$, both classification and clustering have good results. In DBLP and Aminer-top, when k_1=0.5, the classification results are the best, and the Micro-F1 and Macro-F1 values are 0.8863 and 0.8865, 0.8962 and 0.8859, respectively. When k_1 is 0.25 and 0.5, the clustering results of nodes are the best, and the NMI values are 0.7826 and 0.4461.

Since Yelp has only two types of nodes, the value of k_1 is always 1. So parameter analysis of k_1 is not carried out.

When the parameter k_1 is smaller, the experimental results are better. The smaller value of k_1 indicates that the node types tend to different, that is, more types of heterogeneous edges. On the contrary, when k_1 is larger, the quality of node embedding will be affected when there are too few heterogeneous edge types. The results show the balance of the number of edge types can be achieved by adjusting k_1.

3. Return parameter p

Experimental results of p are shown in Figure 10. The abscissa represents the range of *log* values for p, $log_2(p) \in [-3, 3]$, and the step size is 1.



(a) (b) (c)

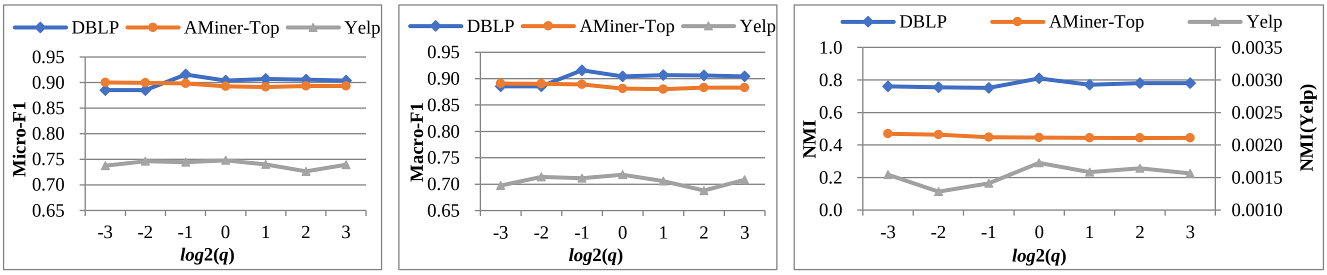
Figure 10. Sensitivity analysis of p. (**a**) Micro-F1 of classification; (**b**) Macro-F1 of classification; (**c**) NMI value of clustering.

Figure 10 shows that when $log_2(p) \in [0, 2]$, in DBLP and Yelp, both classification and clustering have good results. When $log_2(p) \in [-2, 0]$, in Aminer-top, both classification and clustering have good results. In DBLP, Aminer-top and Yelp, when p is 0.25, 4 and 2, the classification effect is the best, and the Micro-F1 and Macro-F1 values are 0.9031 and 0.9030, 0.8960 and 0.8856, 0.7479 and 0.7179, respectively. When p is 0.5, 4 and 2, clustering effect is the best, and NMI value is 0.8091, 0.4446 and 0.0017, respectively.

A larger p indicates that the next node in the random walks tend to migrate to other nodes, that is, to visit more nodes. On the contrary, when p is smaller, the next node tends to return v_{i-1}, which will affect the distribution of nodes. The results show the balance of node distribution can be achieved by adjusting p.

4. Controlling search mode parameter q.

Experimental results of q are shown in Figure 11. The abscissa represents the range of *log* values for q, $log_2(q) \in [-3, 3]$, and the step size is 1.



(a) (b) (c)

Figure 11. Sensitivity analysis of q. (**a**) Micro-F1 of classification; (**b**) Macro-F1 of classification; (**c**) NMI value of clustering.

Figure 11 shows that when $log_2(q) \in [-2, 0]$, classification has good results. When $log_2(q) \in [0, 2]$, in DBLP and Yelp, the clustering has good results, and the clustering of Aminin-top has good results when $log_2(q) \in [0, 2]$. In DBLP, Aminer-top and Yelp, when q was 0.5, 0.125 and 1, the classification effect was the best, and the Micro-F1 and Macro-F1 values were 0.9157 and 0.9159, 0.9002 and 0.8907, 0.7479 and 0.7179, respectively. When q is 1, 0.125, and 1, the clustering effect is the best, and the NMI value is 0.8091, 0.4689, and 0.0017, respectively.

When $q \in [0.5, 1]$, the results of classification are better. When $q \in [1, 4]$, the results of clustering are better. q>1 indicates that nodes in the random walk tend to breadth-first. q<1 indicates nodes tend to depth-first. The results show the distribution of nodes within type can be balanced by adjusting q.

Through the sensitivity analysis of the above parameters, the parameter values are set as follows in this paper: $\alpha \in [0.1, 0.5]$, $k_1 \in [0.125, 1]$, $p \in [1, 4]$ and $q \in [0.5, 1]$ or $p \in [0.5, 1]$ and $q \in [1, 4]$.

6.2.4. Uniformity of Sampling

This section analyzes the sampling situation from node type and node of the same type. Taking DBLP as an example, when α=0.4, k_1=0.5, p=0.25, q=0.5, L=100, r=10, 10 experiments were carried out. The mean values of each node type and each node were taken for analysis.

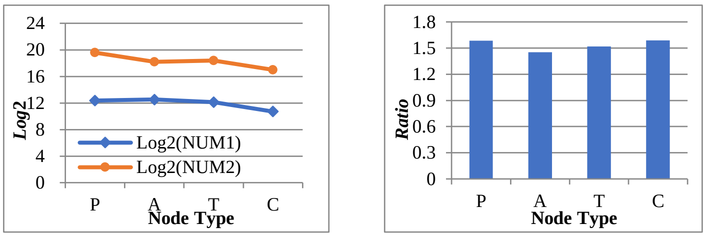
1. Distribution of node types

The distribution of four types of nodes in DBLP is shown in Table 3. NUM1 is the number of nodes P, T, C and A. The number is 5237, 5915, 4479 and 18, respectively. Since the node of C is much smaller than the others, non-uniform data processing is required. According to the ratio of the sampling number of P, A and T to the actual number, the filling value for C type is 1671. NUM2 is the average number of samples taken by P, T, C and A, which are 78400, 314161, 346665 and 13,073, respectively.

To make the results more intuitive, we take Log_2 for NUM1, NUM2, and NUM2/NUM1, and the comparison results are shown in Figure 12. The trend of Log_2(NUM1) and Log_2(NUM2) is basically the same from Figure 12(a). In Figure 12(b), the value of Log_2(NUM2)/Log_2(NUM1) is close to 1.5. In other words, the proportion of sampling quantity of four types in the network is basically the same as the actual quantity. Therefore, the strategy can achieve proportional and uniform sampling of node types.

Table 3. Distribution of node types.

Node Type	P	A	T	C	C(filling)
NUM1	5237	5915	4479	18	1671
NUM2	78400	314161	346665	130073	
NUM2/NUM1	149.70407	51.42199	77.39793	92.84133	

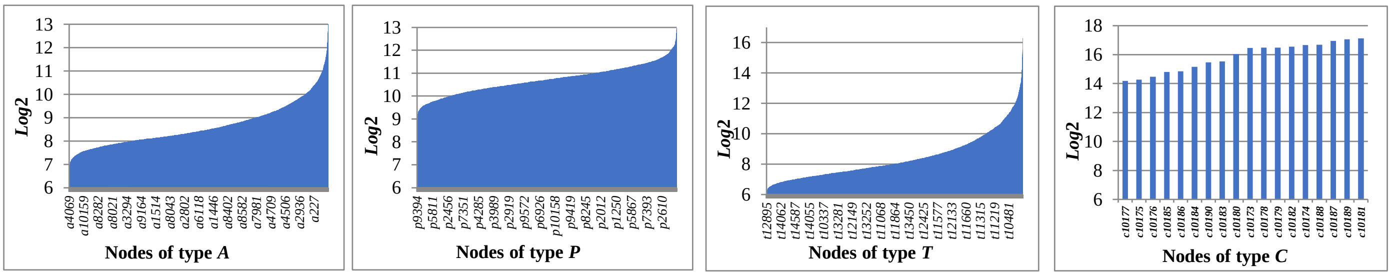


(a) (b)

Figure 12. Uniformity of node types. (**a**) Log_2 of NUM1 and NUM2; (**b**) Log_2 ratio of NUM2 to NUM1

2. Distribution of nodes of the same type

The sampling distribution of nodes of the same type in DBLP is shown in Figure 13. The abscissa in the figure is the node identification. The ordinate is the Log_2 of the average sampling value of the node, and the minimum value is set to 6 for comparison purpose. In Figure 13(a), 90.95% of the sampling values of A-type nodes are distributed between 7 and 10. In Figure 13(b), 84.79% of the sampling values of P-type nodes are in 10-12. In Figure 13(c), 85.74% of the sampling values of T-type nodes are in 6-10. In Figure 13(d), 100% of the sampling values of C-type nodes are in 14-17. In summary, for A, P, T and C, at least 85% or more of each type of nodes are uniformly distributed. Therefore, the strategy can achieve proportional and uniform sampling of nodes within type.



(**a**) (**b**) (**c**) (**d**)

Figure 13. Distribution uniformity of nodes of the same type. (**a**) Sampling times for A; (**b**) Sampling times for P; (**c**) Sampling times for T; (**d**) Sampling times for C.

7. Conclusions

In this paper, HNE-RWTIC is proposed based on the random walks' strategy with Type & Inner constraints. In three real networks, we conducted experimental verification from classification and clustering tasks respectively. The experimental results show that: (1) In classification task, compared with the five classic algorithms, HNE-RWTIC has the best results in the case of 80% training sets. In DBLP, Aminer-Top and Yelp, Macro-F1 value is increased by 2.43%, 0.99% and 5.02% at most, respectively. (2) In clustering task, in DBLP, Aminer-Top and Yelp networks, the NMI value is increased by 19.12%, 6.91% and 0.0421% at most, respectively. (3) In parameter sensitivity analysis, parameters α and k_1 affect the number and type of heterogeneous edges, and parameters p and q balance the node distribution within the type. We adjust the parameters to achieve the flexibility of the walks and improve the quality of the walks sequence. (4) In node sampling experiments, the sampling distribution of node types and nodes of the same type is consistent with the proportion of nodes in the real network. Thus, HNE-RWTIC can balance homogeneous edges and heterogeneous edges, and balance node distribution in different types. Next, how to further improve the strategy of heterogeneous types and node constraints in more complex HINs, and how to combine the dynamic network with the ideas of heterogeneous types and node constraints is the important contents of our next work.

 doi:10.20944/preprints202206.0225.v1

Author Contributions: Conceptualization, X.C. and T.H.; methodology, X.C., T.H and J.G.; validation, L.H. and M.L.; data curation, T.H.; writing—original draft preparation, X.C. and T.H.; writing—review and editing, J.G. and J.C.; visualization, L.H. and M.L.; funding acquisition, J.G. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: Please add: This work is supported by the National Natural Science Foundation of China, grant number 62172352 and 61871465; the Central Guidance on Local Science and Technology Development Fund of Hebei Province, grant number 226Z0102G; the Fundamental Research Funds for the Hebei Province Universities, grant number 2020JK005; the Doctoral Research Initiation Foundation of Hebei Normal University of Science and Technology of China, grant number 2019YB011; the Natural Science Foundation of Hebei Provincial, grant number F2017209070 and F2019203157, and the Key project of science and technology research in Hebei Province, grant number ZD2019004.

Data Availability Statement: DBLP at https://dblp.uni-trier.de/xml/, AMiner at https://www.aminer.cn/knowledge_graph_cn, Yelp at https://www.yelp.com/dataset.

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

References

1. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, Canada, 13-17 August, 2017.
2. 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, United States, 13-17 August 2016.
3. Perozzi, B.; AlRfou, 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, United States, 24-27 August 2014.
4. Yang, C.; Xiao, Y.; Zhang, Y., Sun, Y; Han, J. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, **2020**, 1−19.
5. Ji, M.; Han, J.; Danilevsky, M. Ranking-Based Classification of Heterogeneous Information Networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, 21-24 August 2011.
6. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep Learning Based Text Classification: A Comprehensive Review. arXiv:2004.03705, **2021**, 1-42.
7. Opsahl, T.; Panzarasa, P. Clustering in weighted networks. *Social networks*, **2009**, 31(2): 155-163.
8. Yang, L.; Zhan, X.; Chen, D.; Yan, J.; Loy, C.C.; Lin, D. Learning to Cluster Faces on an Affinity Graph. In Proceedings of IEEE/CVF Conference on Computer Vision & Pattern Recognition, Long Beach, CA, United States,15-20 June 2019.
9. Liben-Nowell, D.; Kleinberg, J. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, **2007**, 58(7): 1019-1031.
10. Rossi, A.; Firmani, D.; Matinata, A.; Matinata, A.; Merialdo, P. Knowledge graph embedding for link prediction: a comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, **2021**, 15(2): 1-49.
11. Xu, L.; Wei, X.; Cao, J.; Yu, P.S. Embedding of Embedding (EOE) Joint Embedding for Coupled Heterogeneous Networks. In Proceedings of the 10th ACM International Conference on Web Search and Data Mining, New York, United States, 6-10 February 2017.
12. Chen, H.; Yin, H.; Wang, W.; Wang, H.; Nguyen, Q.; Li, X. PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, United Kingdom, 19-23 August 2018.
13. Shi, C.; Lu, Y.; Hu, L.; Liu, Z.; Ma, H. RHINE: Relation structure-aware heterogeneous information network embedding. *IEEE Transactions on Knowledge and Data Engineering*, **2020**, 1-15.
14. Shi, C.; Hu, B.; Zhao, W.X.; Yu, P.S. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, **2019**, 31(2): 357-370.
15. Hu, B.; Fang, Y.; Shi, C. Adversarial Learning on Heterogeneous Information Networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, United States, 4-8 August 2019.
16. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, **2021**, 32(1): 4-24.
17. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous Graph Attention Network. In Proceedings of WWW' 19: The World Wide Web Conference, New York, United States, 13-17 May 2019.
18. Hu, L.; Li, C.; Shi, C.; Yang, C.; Shao, C. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, **2020**, 57(2): 102142.
19. Li, Q.; Shang, Y.; Qiao, X.; Dai, W. Heterogeneous Dynamic Graph Attention Network. In Proceedings of 11th IEEE International Conference on Knowledge Graph, Nanjing, China, 9-11 August 2020.
20. Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: a review of methods and applications. *AI Open*, **2020**, 1: 57-81.

 doi:10.20944/preprints202206.0225.v1

21. Lee, S.; Park, C.; Yu, H. BHIN2vec: Balancing the Type of Relation in Heterogeneous Information Network. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3-7 November 2019.
22. Shang, J.; Qu, M.; Liu, J.; Kaplan, L.M.; Han, J.; Peng, J. Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. arXiv:1610.09769, **2016**, 1-9.
23. Yin, Y.; Ji, L.; Huang, R.; Cheng, X. Heterogeneous Network Representation Learning Method Based on Meta-path. In Proceedings of the 4th International Conference on Cloud Computing and Big Data Analytics, Chengdu, China, 12–15 April 2019.
24. Fu, Y.; Xiong, Y.; Yu, P.S.; Tao, T.; Zhu, Y. Metapath Enhanced Graph Attention Encoder for HINs Representation Learning. In Proceedings of 6th IEEE International Conference on Big Data, Kyoto, Japan, 27 February - 2 March 2019.
25. Fu, T.; Lee, W.C.; Lei, Z. Hin2vec: Explore Meta-Paths in Heterogeneous Information Networks for Representation Learning. In Proceedings of the 26th ACM on Conference on Information and Knowledge Management, Singapore, 6-10 November 2017.
26. Hussein, R.; Yang, D.; Cudré-Mauroux, P. Are Meta-Paths Necessary? Revisiting Heterogeneous Graph Embeddings. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22-26 October 2018.
27. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *Computer Science*, **2013**, 1-12.
28. He, Y.; Song, Y.; Li, J.; Ji, C.; Peng, J.; Peng, H. Hetespaceywalk: A Heterogeneous Spacey Random Walk for Heterogeneous Information Network Embedding. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3-7 November 2019.
29. Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; Yu, P.S. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge & Data Engineering*, **2017**, 29(1): 17-37.