Network Schema Preserving Heterogeneous Information Network Embedding

Jianan Zhao[1*], **Xiao Wang**[1*], **Chuan Shi**[1†], **Zekuan Liu**[1] and **Yanfang Ye**[2]

[1]School of CS, Beijing University of Posts and Telecommunications, Beijing, China

[2]Department of CDS, Case Western Reserve University, OH, USA

{zhaojianan, xiaowang, shichuan, liuzekuan}@bupt.edu.cn, yanfang.ye@case.edu

Abstract

As heterogeneous networks have become increasingly ubiquitous, Heterogeneous Information Network (HIN) embedding, aiming to project nodes into a low-dimensional space while preserving the heterogeneous structure, has drawn increasing attention in recent years. Many of the existing HIN embedding methods adopt meta-path guided random walk to retain both the semantics and structural correlations between different types of nodes. However, the selection of meta-paths is still an open problem, which either depends on domain knowledge or is learned from label information. As a uniform blueprint of HINs, the network schema comprehensively embraces the high-order structure and contains rich semantics. In this paper, we make the first attempt to study network schema preserving HIN embedding, and propose a novel model named NSHE. In NSHE, a network schema sampling method is first proposed to generate subgraphs (i.e., schema instances), and then multi-task learning task is built to preserve the heterogeneous structure of each schema instance. Besides preserving pairwise structure information, NSHE is able to retain high-order structure (i.e., network schema). Extensive experiments on three real-world datasets demonstrate that our proposed model NSHE significantly outperforms the state-of-the-art methods.

1 Introduction

Network embedding, which aims to project the nodes of a network into a low-dimensional space while preserving the structural properties of the network, has been a promising research field [Cui *et al.*, 2019]. Most of the existing network embedding methods focus on homogeneous network. However, with the proliferation of interaction systems, Heterogeneous Information Networks (HINs) [Sun *et al.*, 2011], which consist of multiple types of entities and links, have emerged as a powerful tool for modeling complex interaction behaviors. Recently, to handle the ubiquitous heterogeneous network data, representation learning on HIN has drawn increasing attention [Dong *et al.*, 2017; Fu *et al.*, 2017] and is widely used in various real-world applications including node classification, clustering, and recommendation.

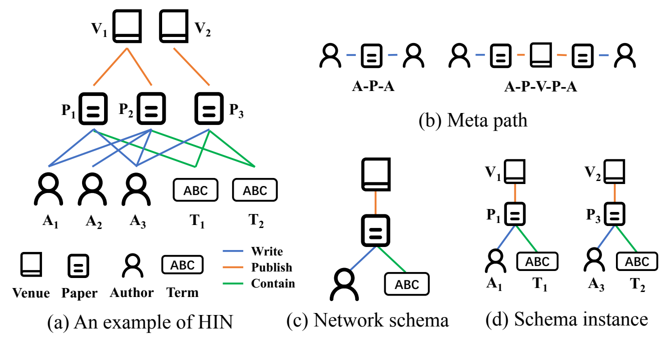


Figure 1: A toy example of an HIN on bibliographic data.

To learn the node representation on HINs, HIN embedding methods have been proposed [Dong *et al.*, 2017; Fu *et al.*, 2017; Shi *et al.*, 2019; Zhang *et al.*, 2018; He *et al.*, 2019], many of which exploit the meta-path(s) guided random walk to retain both the semantics and structural correlations between different types of nodes. Typically, a meta-path is a sequence of relations between two nodes in an HIN. For example, given a bibliography HIN (four types of nodes: author (A), paper (P), Venue (V) and term (T); three types of relations: "write", "publish", and "contain") shown in Figure 1(a), Figure 1(b) displays two meta-paths APA and APVPA which describe the co-author or co-venue structure between two authors, respectively. The meta-path based random walk will confine the node sequence along the predefined meta-path, and further capture the high-order semantic structure.

Despite the success of meta-path guided HIN embedding methods, the selection of meta-paths still remains an open yet challenging problem [Sun *et al.*, 2011]. The design of meta-path schemes significantly relies on domain knowledge. Manually selecting meta-paths based on prior knowledge may work for a simple HIN, while it is difficult to determine meta-paths for a complex HIN. Furthermore, different meta-paths will result in different embeddings from different points of view, which leads to another challenging problem,

*Both authors contributed equally to this research.

†Contact Author

i.e., how to effectively fuse different embeddings to generate uniform embeddings. Some existing works [Shi *et al.*, 2019; Wang *et al.*, 2019; Hu *et al.*, 2019b] use label information to guide the embedding fusion; unfortunately, this is not applicable in unsupervised scenarios.

To tackle the above challenges, we observe that the network schema [Sun *et al.*, 2011], as a uniform blueprint of HIN, comprehensively retains node types and their relations in an HIN. Since network schema is a meta template for HIN, guided by it, we can extract subgraphs (i.e., schema instances) from the HIN. An example is shown in Figure 1(c) and (d), from which we can see that the schema instance depicts the high-order structure information of these four nodes, besides the first-order structure information of two nodes (i.e., pairwise structure). Moreover, the schema instance also contains rich semantics, i.e., a schema instance (shown in Figure 1(d)) naturally describes the overall information, such as the author, the term, and the venue of a paper, as well as their relations. More importantly, different from meta-paths, network schema is a unique structure for an HIN, and thus we do not need domain knowledge to make a choice. These benefits of network schema motivate us to study network schema preserving HIN embedding.

However, it is a non-trivial task. First, *how to effectively preserve the network schema structure?* Network schema structure usually contains all types of nodes. The widely used random walk (with/without meta-path) strategy cannot guarantee to visit all types of nodes and links, therefore it is not applicable for preserving network schema. Moreover, the numbers of various types of nodes in a network schema structure are usually very different, leading to the bias problem. For example, a paper is associated with one venue, but with many terms. Moreover, *how to capture the heterogeneity of nodes and links inside network schema?* We need to delicately design a method not only preserving network schema structure but also considering node and link heterogeneity.

In this paper, we make the first attempt to investigate **N**etwork **S**chema preserving **H**eterogeneous information network **E**mbedding and propose a novel model named NSHE. Based on node embedding generated by heterogeneous graph convolutional network, NSHE optimizes the embedding via node pairs and schema instances sampled from the HIN. Particularly, in the network schema preserving component, we propose a network schema sampling method, which generates sub-graphs (i.e., schema instances) naturally preserving schema structure. Furthermore, for each schema instance, a multi-task learning model is built to predict each node in the instance with other nodes, which tackles the challenge of heterogeneity. Our major contributions are highlighted as follows:

- To the best of our knowledge, we make the first attempt to preserve the network schema structure for HIN embedding, which not only preserves high-order structure in HIN but also alleviates the meta-path selection dilemma in meta-path guided HIN embeddings.
- We propose a novel model NSHE, in which some delicate designs, e.g., network schema sampling and multi-task learning, are proposed to solve the schema structure preserving and the heterogeneity challenges.
- We conduct extensive experiments on three real-world datasets to validate the effectiveness of NSHE compared with the state-of-the-art methods.

2 Related Work

Our work is related to network embedding, which assigns nodes in a network to low-dimensional representations and effectively preserves the network structure. For example, the neighbor structure preserving network embedding [Perozzi *et al.*, 2014], the second-order structure preserving network embedding [Tang *et al.*, 2015; Wang *et al.*, 2016], and the community structure preserving network embedding [Wang *et al.*, 2017]. Most of these network embedding methods focus on homogeneous networks, and an elaborate review can be found in [Cui *et al.*, 2019].

With the thriving of heterogeneous network data in real-world applications, HIN Embedding methods have drawn increasing research attention recently. Most existing methods utilize the meta-paths to capture HIN structure. For instance, ESim [Shang *et al.*, 2016] accepts meta-paths as guidance to learn node embedding for similarity search. Metapath2Vec [Dong *et al.*, 2017] proposes meta-path guided random walk and heterogeneous Skip-Gram to handle the heterogeneity in HINs. HIN2Vec [Fu *et al.*, 2017] learns HIN embeddings via predicting different relations in HINs. RHINE [Lu *et al.*, 2019] distinguishes the meta-path based relations and deals with them using different models. HeteSpaceyWalk [He *et al.*, 2019] proposes a spacey random walk to preserve the Markov chain nature of meta-paths based random walks. However, these methods suffer from the meta-path selection and fusion conundrums. In addition, several methods perform HIN embedding without using meta-paths. JUST [Hussein *et al.*, 2018] develops a jump and stay strategy on random walks. HetGNN [Zhang *et al.*, 2019] adopts graph neural networks and preserves the first-order and second-order proximity. HeGAN [Hu *et al.*, 2019a] introduces adversarial learning in HIN embeddings. Though these methods perform HIN embedding without using meta-paths, all of them do not explicitly preserves the network schema structure.

3 Proposed Method

Consider an HIN $G = (V, E)$ composed of a node set V and an edge set E, along with the node type mapping function $\phi : V \to \mathcal{A}$, and the edge type mapping function $\psi : E \to \mathcal{R}$, where $\mathcal{A}$ and $\mathcal{R}$ denotes the node and edge types, $|\mathcal{A}| + |\mathcal{R}| > 2$. The task is to learn the representation of nodes $\mathbf{Z} \in \mathbb{R}^{|V| \times d}$, where d is the dimension of representation.

Figure 2 illustrates the framework of the proposed NSHE. NSHE preserves the pairwise and schema proximity concurrently. First, to fully exploit complex network structure and heterogeneous node feature together, we propose to learn node embedding via heterogeneous node aggregation. Second, we preserve the pairwise structure and the schema structure simultaneously. While directly performing random walk cannot generate the desired schema structure, we propose to sample schema instances and preserve the proximity inside

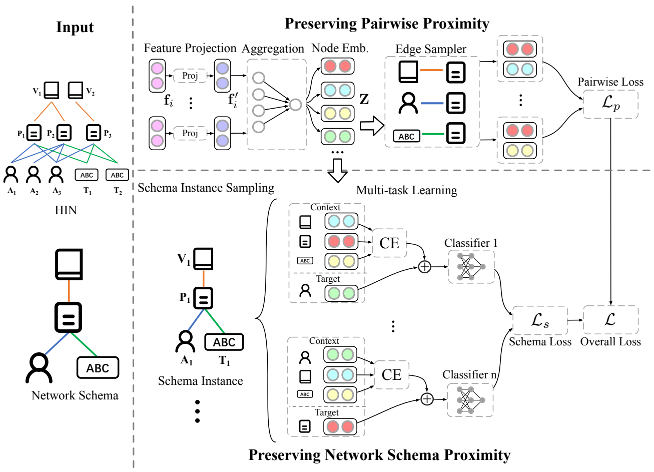


Figure 2: Overview of the NSHE model.

instances. Moreover, as different types of nodes in the instances carry different context, a multi-task learning model is designed to in turn predict a target node with other context nodes to handle heterogeneity inside schema instances. Finally, NSHE iteratively updates node embeddings via optimizing the aggregation of the pairwise and schema preserving loss.

3.1 Preserving Pairwise Proximity

Despite that we need to capture the network schema structure in HIN embedding, the pairwise proximity between nodes [Tang *et al.*, 2015], as one of the most direct expressions of an HIN, still needs to be preserved. It demonstrates that two nodes with a link, regardless of their types, should be similar. Specifically, considering the heterogeneity of different node feature, for each node v_i with feature $\mathbf{f}_i$ and type $\phi(v_i)$, we use a type-specific mapping matrix $\mathbf{W}_{\phi(v_i)}$ to map the heterogeneous feature to a common space:

$$\mathbf{f}_i' = \sigma(\mathbf{f}_i * \mathbf{W}_{\phi(v_i)} + \mathbf{b}_{\phi(v_i)}), \tag{1}$$

where $\sigma(\cdot)$ denotes an activation function, and $\mathbf{b}_{\phi(v_i)}$ stands for the bias vector of type $\phi(v_i)$. Based on Equation (1), all the nodes with different types are mapped to the common space, and we denote their mapped features as $\mathbf{H} = [\mathbf{f}_i']$. Then, we use a L-layer graph convolutional network to generate the node embeddings [Kipf and Welling, 2017] as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I}_{|V|})\mathbf{D}^{-\frac{1}{2}} \cdot \mathbf{H}^{(l)} \cdot \mathbf{W}^{(l)}\right), \tag{2}$$

where $\mathbf{A}$ is the adjacency matrix, and $\mathbf{A}_{i,j} = 1$ if $(v_i, v_j) \in E$, otherwise $\mathbf{A}_{i,j} = 0$. $\mathbf{D}$ is a diagonal matrix, where $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. $\mathbf{I}_{|V|}$ is the identity matrix of $\mathbb{R}^{|V|\times|V|}$. For the first layer, we denote $\mathbf{H}^{(0)} = \mathbf{H}$ and use the output of the L-layer graph convolutional networks as the node embedding, i.e., $\mathbf{Z} = \mathbf{H}^{(L)}$, where the i-th row of $\mathbf{Z}$ is the embedding $\mathbf{z}_{v_i}$ of node v_i.

The objective of preserving the pairwise proximity with parameters Θ can be described as:

$$\mathcal{O}_p = \arg\max_{\Theta} \prod_{v_i \in V} \prod_{v_j \in N_{v_i}} p\left(v_j | v_i; \Theta\right), \tag{3}$$

where $N_{v_i} = \{v_j | (v_i, v_j) \in E\}$. The conditional probability $p\left(v_j | v_i; \Theta\right)$ is defined as a softmax function:

$$p\left(v_j | v_i; \Theta\right) = \frac{\exp(\mathbf{z}_{v_j} \cdot \mathbf{z}_{v_i})}{\sum_{v_k \in V} \exp(\mathbf{z}_{v_k} \cdot \mathbf{z}_{v_i})}. \tag{4}$$

To calculate $p\left(v_j | v_i; \Theta\right)$ efficiently, we leverage the negative sampling method [Mikolov *et al.*, 2013] and optimize Θ with the logarithm of Equation (3), therefore the pairwise loss $\mathcal{L}_p$ can be calculated by:

$$\begin{aligned}\mathcal{L}_p = \frac{1}{|E|} \sum_{(v_i, v_j) \in E} \Big[&- \log \delta(\mathbf{z}_{v_j} \cdot \mathbf{z}_{v_i}) \\ &- \sum_{m=1}^{M_e} \mathbb{E}_{v_{j'} \sim P_n(v)} \log \delta(-\mathbf{z}_{v_{j'}} \cdot \mathbf{z}_{v_i}) \Big]\end{aligned} \tag{5}$$

where $\delta(x) = 1/(1 + \exp(-x))$, $P_n\left(v\right)$ is the noisy distribution, and M_e is the negative edge sampling rate. Through minimizing $\mathcal{L}_p$, NSHE preserves the pairwise proximity.

3.2 Preserving Network Schema Proximity

Network Schema Instance Sampling

Network schema is the blueprint of an HIN [Sun *et al.*, 2011]. Given an HIN $G = (V, E)$, a network schema $T_G = (\mathcal{A}, \mathcal{R})$ preserves all the node types $\mathcal{A}$ and relation types $\mathcal{R}$ inside G. Network schema proximity implies that all the nodes with different types in a network schema structure should be similar. However, as we mentioned before, the nodes in a network schema structure are usually biased, i.e., the number of nodes of a certain type is larger than those of other types. For example, in Figure 1(a), a paper has multiple authors, but only one venue. To alleviate such bias, we propose to sample a network schema instance defined as follows: A *network schema instance* S is the smallest sub-graph of an HIN, which contains all the node types and edge types defined by the network schema T_G, if existing. By this definition, each network schema instance is composed of all the node types $\mathcal{A}$ and relation types $\mathcal{R}$ defined by the schema, i.e., one node for each type. To illustrate, Figure 1(d) shows two instances sampled from the given HIN. The sampling process is as follows: Starting from a set S with one node, we keep adding a new node to S until $|S| = |\mathcal{A}|$, where the new node satisfies: (1) its type is different from the node types in S; (2) it connects with the node(s) in S.

Schema Preserving with Multi-task Learning

Now, we aim to preserve the network schema proximity by predicting whether a network schema instance exists in an HIN. To this end, assume we have a network schema instance $S = \{A_1, P_1, V_1, T_1\}$ as shown in Figure 2, we can predict whether A_1 exists given the set $\{P_1, V_1, T_1\}$, or whether P_1 exists given the set $\{A_1, V_1, T_1\}$, and so on. These two predictions are different, because of the node heterogeneity. Considering this, we are motivated to design a multi-task learning model to handle the heterogeneity within schema.

Without loss of generality, assume we have the schema instance $S = \{v_i, v_j, v_k\}$, if we aim to predict whether v_i exists given $\{v_j, v_k\}$, we call v_i the **target node** and $\{v_j, v_k\}$ the **context nodes**. Therefore, each node will have two roles:

one is as the target node and the other is as the context node, as well as two embeddings: target embedding and context embedding. To fully consider the heterogeneity, each node type $\phi(v_i)$ is associated with an encoder $\mathrm{CE}^{\phi(v_i)}$ to learn the context embeddings for the context nodes:

$$\mathbf{c}_{v_j} = \mathrm{CE}^{\phi(v_j)}(\mathbf{z}_{v_j}), \mathbf{c}_{v_k} = \mathrm{CE}^{\phi(v_k)}(\mathbf{z}_{v_k}), \tag{6}$$

where each CE stands for a fully connected layer of neural network. Then for the target node v_i, we concatenate its target embedding $\mathbf{z}_{v_i}$ with the context embeddings to obtain the schema instance embedding with target node v_i denoted as $\mathbf{z}_S^{v_i}$ as follows:

$$\mathbf{z}_S^{v_i} = \mathbf{z}_{v_i} \| \mathbf{c}_{v_j} \| \mathbf{c}_{v_k}. \tag{7}$$

After obtaining the embedding $\mathbf{z}_S^{v_i}$, we predict the probability of S with target node v_i, denoted as $y_S^{v_i}$, whether exists in the network:

$$y_S^{v_i} = \mathrm{MLP}^{\phi(v_i)}\left(\mathbf{z}_S^{v_i}\right), \tag{8}$$

where $\mathrm{MLP}^{\phi(v_i)}$ is the classifier for schema instances with target node type as $\phi(v_i)$. Similarly, when we treat v_j and v_k as the target nodes, respectively, $y_S^{v_j}$ and $y_S^{v_k}$ can also be obtained following the steps introduced above. Note that, here we take the schema instance with three nodes as an example to explain our method. However, it is easy to extend the model to schema instance with more nodes, since the process is the same.

The schema proximity loss $\mathcal{L}_s$ can be obtained by predicting the multi-tasks of the schema instances $\mathcal{S}$ sampled from HIN. Additionally, to avoid trivial solutions, we also draw M_s negative examples of target type for each schema instance via replacing the target node with another node in the same type. The loss of preserving network schema can be described as:

$$\mathcal{L}_s = -\frac{1}{|\mathcal{A}||\mathcal{S}|} \sum_{S \in \mathcal{S}} \sum_{v_i \in S} \left(R_S^{v_i} \log y_S^{v_i} + (1 - R_S^{v_i}) \log\left(1 - y_S^{v_i}\right)\right), \tag{9}$$

where $R_S^{v_i} = 1$ if S^{v_i} is a positive network schema instance, otherwise $R_S^{v_i} = 0$. By minimizing $\mathcal{L}_s$, the schema structure is preserved.

3.3 Optimization Objective

To preserve both the pairwise proximity and the network schema proximity of HINs, NSHE optimizes the overall loss $\mathcal{L}$ by aggregating the loss of preserving pairwise proximity $\mathcal{L}_p$ and preserving schema proximity $\mathcal{L}_s$:

$$\mathcal{L} = \mathcal{L}_p + \beta \mathcal{L}_s, \tag{10}$$

where β is a balancing coefficient. At last, we adopt the Adam [Kingma and Ba, 2015] algorithm to minimize the objective in Equation (10).

4 Experimental Results and Analysis

4.1 Experimental Setup

In order to demonstrate the effectiveness of the proposed model, we conduct extensive experiments including clustering and classification on three HINs shown below:

- **DBLP** [Lu *et al.*, 2019]: We extract a subset of DBLP which contains 9556 papers (P), 2000 authors (A), and 20 conferences (C). The authors and papers are divided into four areas: database, data mining, machine learning, and information retrieval. We do the two tasks for papers and authors and call them DBLP-P and DBLP-A, respectively. We use 4 meta-paths, namely APCPA, APA, PAP, and PCP, for meta-path related baselines.
- **IMDB** [Wang *et al.*, 2019]: We extract a subset of IMDB which contains 3676 movies (M), 4353 actors (A), and 1678 directors (D). Movies are divided into three classes, namely action, comedy, and drama according to their genre. We use two meta-paths, namely MAM and MDM, for meta-path related baselines.
- **ACM** [Wang *et al.*, 2019]: We extract papers published in KDD, SIGMOD, SIGCOMM, MobiCOMM, and VLDB and divide them into three classes: database, wireless communication, and data mining. Then we construct an HIN that contains 4019 papers (P), 7167 authors (A), and 60 conference subjects (S). Papers are labeled according to their conferences. Paper features are the bag-of-words representation of keywords. We use PAP and PSP for meta-path related baselines.

We compare NSHE with seven state-of-the-art embedding methods including two homogeneous network embedding methods, i.e., DeepWalk and LINE and five heterogeneous networks embedding methods, i.e., the last five algorithms:

- **DeepWalk** [Perozzi *et al.*, 2014]: It performs a random walk on networks and then learns the representation of nodes via the Skip-Gram model.
- **LINE** [Tang *et al.*, 2015]: It considers first-order or second-order proximity in networks, denoted as LINE-1st and LINE-2nd, respectively.
- **Metapath2Vec** [Dong *et al.*, 2017]: It adopts meta-paths based random walk and heterogeneous Skip-Gram model to perform node embedding.
- **HIN2Vec** [Fu *et al.*, 2017]: It learns the latent representation of nodes and meta-paths in an HIN by conducting multiple prediction training tasks jointly.
- **HERec** [Shi *et al.*, 2019]: It adopts meta-paths to filter node sequences of same type with different semantics and applies DeepWalk to perform network embedding.
- **DHNE** [Tu *et al.*, 2018]: It adopts deep auto-encoders and classifiers to preserve the first and second order proximity of hyper networks. Here we treat the network schema instances as hyper-edges.
- **HeGAN** [Hu *et al.*, 2019a]: It learns the representation of nodes in HIN via preserving the heterogeneous relations with adversarial learning.

Here, we briefly introduce the experimental settings. For our proposed model, the feature dimension in common space and the embedding dimension d is set as 128. The negative schema instance sample rate M_s in Section 3.2 is set as 4. We perform neighborhood aggregation via an one-layer-GCN, i.e., $L = 1$, and use two-layer-MLPs for schema in-

	DBLP-P		DBLP-A		IMDB		ACM	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DeepWalk	90.12	89.45	89.44	88.48	56.52	55.24	82.17	81.82
LINE-1st	81.43	80.74	82.32	80.20	43.75	39.87	82.46	82.35
LINE-2nd	84.76	83.45	88.76	87.35	40.54	33.06	82.21	81.32
DHNE	85.71	84.67	73.30	67.61	38.99	30.53	65.27	62.31
Metapath2Vec	92.86	92.44	89.36	87.95	51.90	50.21	83.61	82.77
HIN2Vec	83.81	83.85	90.30	89.46	48.02	46.24	54.30	48.59
HERec	90.47	87.50	86.21	84.55	54.48	53.46	81.89	81.74
HeGAN	88.79	83.81	90.48	89.27	58.56	57.12	83.09	82.94
NSHE	**95.24**	**94.76**	**93.10**	**92.37**	**59.21**	**58.35**	**84.12**	**83.27**

Table 1: Performance evaluation of multi-class classification.

	DBLP-P	DBLP-A	IMDB	ACM
DeepWalk	46.75	66.25	0.41	**48.81**
LINE-1st	42.18	29.98	0.03	37.75
LINE-2nd	46.83	61.11	0.03	41.80
DHNE	35.33	21.00	0.05	20.25
Metapath2Vec	56.89	68.74	0.09	26.71
HIN2Vec	30.47	65.79	0.04	2.28
HERec	39.46	24.09	0.51	40.70
HeGAN	60.78	68.95	6.56	43.35
NSHE	**65.54**	**69.52**	**7.58**	44.32

Table 2: Performance evaluation of node clustering.

stance classification. For models that use meta-paths in modeling, we choose the popular meta-paths adopted in previous methods and report the best result. For models that require node feature, we apply DeepWalk [Perozzi *et al.*, 2014] to generate node feature. The code and dataset is publicly available on Github[1].

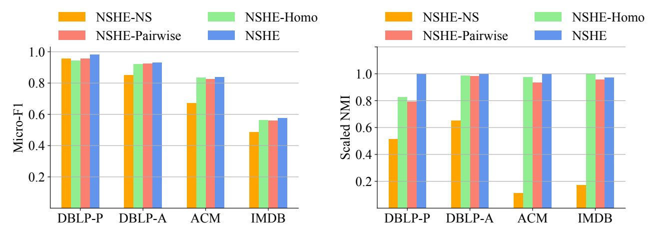
4.2 Node Classification

In this section, we evaluate the performance of node embedding with node classification tasks. After learning the node embeddings, we train a logistic classifier with 80% of the labeled nodes and use the remaining data for testing. We use Micro-F1 and Macro-F1 score as the metrics for evaluation. The results are shown in Table 1, from which we have the following observations: (1) Generally speaking, HIN embedding methods perform better than homogeneous network embedding methods, which proves the benefits of considering heterogeneity. (2) Though NSHE does not utilize any prior knowledge, it consistently outperforms the baselines. It demonstrates the effectiveness of our proposed method in classification tasks.

4.3 Node Clustering

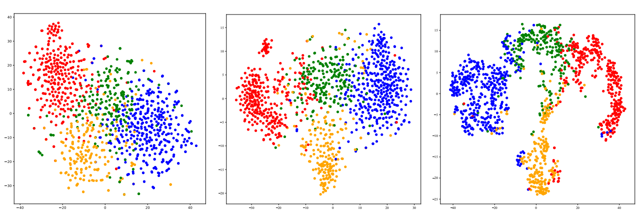
We further conduct clustering tasks to evaluate the embeddings learned by NSHE. Here we utilize the K-Means model to perform node clustering and set the number of clusters for K-Means as the number of classes. The performance in terms of NMI is shown in Table 2. Similarly, the proposed method

[1]https://github.com/Andy-Border/NSHE



(a) Node classification (b) Node clustering

Figure 3: Performance evaluation of variants of NSHE.



(a) DeepWalk (b) Metapath2Vec (c) NSHE

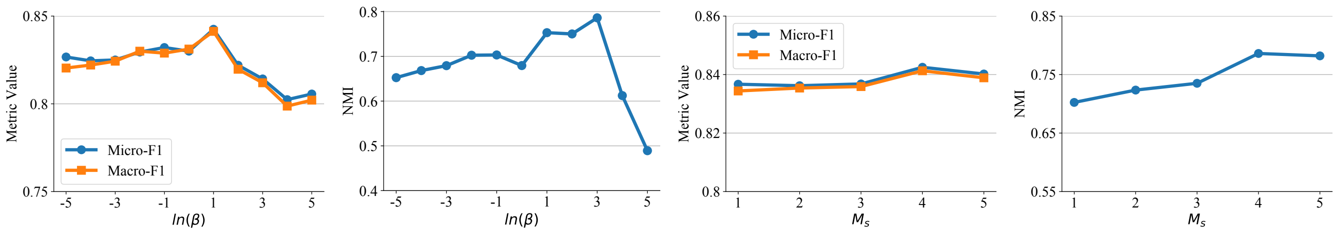
Figure 4: Embedding visualization of different methods on DBLP-A. Each point indicates one author and its color indicates the research area.

NSHE significantly outperforms others in most cases, which further demonstrates the effectiveness of NSHE.

4.4 Comparison of Variants of NSHE

In order to verify the effectiveness of the delicate designs in NSHE, we propose three variants of NSHE as follows:

- **NSHE-Pairwise** considers the pairwise loss only, i.e., $\mathcal{L} = \mathcal{L}_p$. Therefore, the NSHE-Pairwise model does not explicitly preserve the high-order structure of network schema.
- **NSHE-NS** leverages the structure of network schema only, i.e., $\mathcal{L} = \mathcal{L}_s$. Therefore, the NSHE-NS model does not explicitly preserve the pairwise structure.
- **NSHE-Homo** treats the heterogeneous network schema instances as homogeneous. That is, NSHE-Homo uses one MLP classifier for all of the network schema instances classification tasks.



(a) Impact of β on classification (b) Impact of β on clustering (c) Impact of M_s on classification (d) Impact of M_s on clustering

Figure 5: Parameter sensitivity of NSHE w.r.t. the balancing coefficient β and the number of negative schema instances M_s.

We use the same parameter for variants of NSHE and show the classification and clustering performances of each model in Figure 3. For better visualization, we scale the NMI value to [0,1] according to the best performance of the variants. It is obvious that NSHE outperforms the variants in most cases. NSHE performs better than NSHE-NS and NSHE-pairwise, indicating that it is necessary to preserve both first-order (i.e., pairwise) and high-order (i.e., network schema) structure information. Moreover, the first-order structure, as the most basic expression of HIN, is more essential, since NSHE-Pairwise generally performs better than NSHE-NS. The better performance of NSHE against NSHE-Homo confirms the benefit of multi-task learning for handling heterogeneity in network schema instances.

4.5 Visualization

To examine the network representations intuitively, we visualize the embeddings of author nodes in DBLP using the t-SNE [Maaten and Hinton, 2008] algorithm. We select DeepWalk and Metapath2Vec as the representative baselines for homogeneous and heterogeneous embedding methods. For Metapath2Vec, the APCPA meta-path is adopted, since it achieves the best results on DBLP. The visualization of embeddings generated by different methods is shown in Figure 4, from which we can observe the following phenomenons: The heterogeneous methods (i.e. Metapath2Vec and NSHE) outperform the homogeneous method DeepWalk, whose embedding has blurry boundaries and may lead to inaccuracy in classification tasks. Moreover, without given any prior knowledge or supervision, NSHE still separates the authors in different research areas with distinct boundaries, which further demonstrates that the preservation of both pairwise and the schema structure in HIN is effective.

4.6 Parameter Analysis

In this section, we investigate the sensitivity of parameters and report the results of NSHE in terms of classification on ACM and clustering on DBLP-P with different parameters.

Balance coefficient β. The balance coefficient β, described in Section 3.3, balances the importance between the pairwise similarity and the high-order proximity of network schema. Larger β stands for greater high-order influence on the model. Figure 5(a) and Figure 5(b) show the impact of different β on classification and clustering tasks, respectively. As we can see, for both tasks, it is better to balance the importance of different terms. Specifically, while increasing β, the performance of both tasks firstly increases and reaches a peak with $\beta = e$ and $\beta = e^3$ for classification and clustering respectively. Notably, the optimal β differs in different datasets and tasks, which indicates that the importance of the pairwise and the high-order proximity varies in different tasks.

Negative schema instance rate M_s. The negative instance rate, described in Section 3.2, is the ratio between negative and positive network schema instances. Figure 5(c) and Figure 5(d) show the impact of different M_s on classification and clustering tasks, respectively. The results indicate that NSHE needs a suitable negative sample rate to preserve the high-order structure. As we can see, with the growth of the negative sample rate, the performance raises first and then starts to drop slowly after the negative sample rate equals to 4. The reason is that the large negative schema sampling rate will cause imbalance of data which causes trivial solutions (all negative prediction) of classifiers. In this case, the schema proximity cannot be well preserved.

5 Conclusion

In this paper, we make the first attempt to study the network schema preserving embedding in HINs. Network embedding via network schema preserves the semantics of network schema and does not suffer from limited domain knowledge. We propose NSHE, which learns embeddings that preserve pairwise structure and network schema structure concurrently. Particularly, NSHE adopts a network schema instance sampling method to deal with the bias of different types of nodes and uses multi-tasks classifiers to preserve the heterogeneity within HINs. Experimental results including classification and clustering demonstrate the effectiveness of NSHE.

6 Acknowledgement

J. Zhao, C. Shi, X. Wang, and Z. Liu's work is supported by the National Natural Science Foundation of China (No. U1936220, 61772082, 61702296, 61806020, 61972442), the National Key Research and Development Program of China (2018YFB1402600), the CCF-Tencent Open Research Fund, and the Fundamental Research Funds for the Central Universities. Y. Ye's work is partially supported by the NSF IIS-1951504, CNS-1940859, CNS-1946327, CNS-1814825 and OAC-1940855, and the NIJ 2018-75-CX-0032.

References

[Cui *et al.*, 2019] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Trans. Knowl. Data Eng.*, 31(5):833–852, 2019.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, pages 135–144, 2017.

[Fu *et al.*, 2017] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*, pages 1797–1806, 2017.

[He *et al.*, 2019] Yu He, Yangqiu Song, Jianxin Li, Cheng Ji, Jian Peng, and Hao Peng. Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding. In *CIKM*, pages 639–648, 2019.

[Hu *et al.*, 2019a] Binbin Hu, Yuan Fang, and Chuan Shi. Adversarial learning on heterogeneous information networks. In *KDD*, pages 120–129, 2019.

[Hu *et al.*, 2019b] Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *EMNLP-IJCNLP*, pages 4820–4829, 2019.

[Hussein *et al.*, 2018] Rana Hussein, Dingqi Yang, and Philippe Cudré-Mauroux. Are meta-paths necessary?: Revisiting heterogeneous graph embeddings. In *CIKM*, pages 437–446, 2018.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Lu *et al.*, 2019] Yuanfu Lu, Chuan Shi, Linmei Hu, and Zhiyuan Liu. Relation structure-aware heterogeneous information network embedding. In *AAAI*, pages 4456–4463, 2019.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.

[Shang *et al.*, 2016] Jingbo Shang, Meng Qu, Jialu Liu, Lance M. Kaplan, Jiawei Han, and Jian Peng. Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. *CoRR*, abs/1610.09769, 2016.

[Shi *et al.*, 2019] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.*, 31(2):357–370, 2019.

[Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[Tu *et al.*, 2018] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. Structural deep embedding for hyper-networks. In *AAAI*, pages 426–433, 2018.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *KDD*, pages 1225–1234, 2016.

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pages 203–209, 2017.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019.

[Zhang *et al.*, 2018] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Metagraph2vec: Complex semantic path augmented heterogeneous network embedding. In *PAKDD*, pages 196–208, 2018.

[Zhang *et al.*, 2019] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *KDD*, pages 793–803, 2019.