RHINE: Relation Structure-Aware Heterogeneous Information Network Embedding

Chuan Shi, *Member, IEEE,* Yuanfu Lu, Linmei Hu*, Zhiyuan Liu, *Member, IEEE*
Huadong Ma, *Senior Member, IEEE*

Abstract—Heterogeneous information network (HIN) embedding aims to learn the low-dimensional representations of nodes while preserving structures and semantics in HINs. Although most existing methods consider heterogeneous relations and achieve promising performance, they usually employ one single model for all relations without distinction, which inevitably restricts the capability of HIN embedding. In this paper, we argue that heterogeneous relations have different structural characteristics, and propose a novel Relation structure-aware HIN Embedding model, called RHINE. By exploring four real-world networks with thorough analysis, we present two structure-related measures which consistently distinguish heterogeneous relations into two categories: Affiliation Relations (ARs) and Interaction Relations (IRs). To respect the distinctive structural characteristics of relations, in RHINE, we propose different models specifically tailored to handle ARs and IRs, which can better capture the structures in HINs. Finally, we combine and optimize these models in a unified manner. Furthermore, considering that nodes connected via heterogeneous relations may have multi-aspect semantics and each relation focuses on one aspect, we introduce relation-specific projection matrices to learn node and relation embeddings in separate spaces rather than a common space, which can better preserve the semantics in HINs, referring to a new model RHINE-M. Experiments on four real-world datasets demonstrate that our models significantly outperform the state-of-the-art methods in four tasks.

Index Terms—Heterogeneous information network, Network embedding, Social network analysis.

◆

1 INTRODUCTION

NETWORK embedding aims to embed nodes into a low-dimensional space, while preserving the network structures and properties, which has shed a light on the analysis of networks and become a focal point of study interests in both academic and industrial domains [1], [2], [3]. Recent works on network embedding have achieved promising performance in many data mining tasks, such as classification [4], [5], [6], [7], recommendation [8], [9] and link prediction [10], [11], [12], [13], etc. However, most of these methods generally focus on homogeneous networks which only contain one single type of nodes and edges.

In reality, many real-life information networks consist of multiple types of nodes and edges, widely known as heterogeneous information networks (HINs) [14], [15]. Due to its flexibility in modeling heterogeneous data, HIN has been proposed as a powerful information modeling method. As shown in Figure 1(a), we present an example of HIN, i.e., the DBLP network. We can see that the DBLP network contains four types of nodes: Author (A), Paper (P), Conference (C) and Term (T), and multiple types of relations: writing/written relations, and publish/published relations, etc. In addition, there are composite relations represented by meta-paths [14], [16] such as *APA* (co-author relation) and *APC* (authors write papers published in conferences),
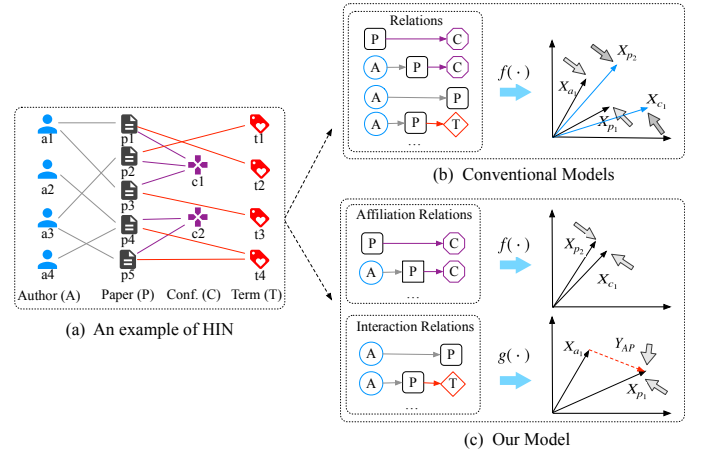

Fig. 1. The illustration of an HIN and the comparison between conventional methods and our method (non-differentiated relations v.s. differentiated relations).

which are widely used to exploit rich semantics in HINs. Thus, compared to homogeneous networks, HINs fuse more information and contain richer semantics. HIN embedding, which provides a new perspective for heterogeneous data analysis, has thus attracted considerable research attention. A natural and straightforward idea to embed an HIN is to directly apply homogeneous network embedding models for heterogeneous data, but this will inevitably lead to reduced performance in downstream tasks.

Some researchers have been working on exploring methods for HIN embedding [17], [18], [19], [20], [21], [22], and have demonstrated the effectiveness of HIN embedding in

- *Chuan Shi, Yuanfu Lu, Linmei Hu (corresponding author) and Huadong Ma are with the Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, China.*
  E-mail: {shichuan, luyuanfu, hulinmei, mhd}@bupt.edu.cn
- *Zhiyuan Liu is with the Department of Computer Science and Technology, Tsinghua University, China.*
  E-mail: liuzy@tsinghua.edu.cn

HIN analysis applications. Roughly speaking, some methods employ meta-path based random walk to generate node sequences for optimizing the similarity between nodes [21], [23], [24]. Some methods decompose the HIN into simple networks and then optimize the proximity between nodes in each sub-network [17], [22], [25]. There are also some neural network based methods that learn non-linear mapping functions for HIN embedding [19], [20], [26], [27]. Also, some methods are designed for specific tasks, such as link prediction [28] and recommendation [22]. Although these methods are designed for heterogeneous networks and have achieved performance improvement to some extent, they usually have an assumption that one single model can handle all relations and nodes, through keeping the representations of two nodes close to each other, as illustrated in Figure 1(b).

However, various heterogeneous relations in an HIN have significantly different structural characteristics, which should be carefully handled with different models. Let's see a toy example in Figure 1(a). The relations in the network include atomic relations (e.g., *AP* and *PC*) and composite relations (e.g., *APA* and *APC*). Intuitively, *AP* relation and *PC* relation reveal rather different characteristics in structure. That is, some authors write some papers in the *AP* relation, which shows a peer-to-peer structure. While that many papers are published in one conference in the *PC* relation reveals the structure of one-centered-by-another. Similarly, *APA* and *APC* indicate peer-to-peer and one-centered-by-another structures respectively. The intuitive examples clearly illustrate that relations in an HIN indeed have different structural characteristics.

It is non-trivial to consider different structural characteristics of relations for HIN embedding, due to the following challenges: (1) How to distinguish the structural characteristics of relations in an HIN? Various relations (atomic relations or meta-paths) with different structures are involved in an HIN. Quantitative and explainable criteria are desired to explore the structural characteristics of relations and distinguish them. (2) How to capture the distinctive structural characteristics of different categories of relations? Since the various relations have different structures, modeling them with one single model may lead to some loss of information. We need to specifically design appropriate models which are able to capture their distinctive characteristics. (3) The different models for the differentiated relations should be easily and smoothly combined to ensure simple optimization in a unified manner.

In this paper, we first present a **R**elation structure-aware **HIN E**mbedding method, called **RHINE**, which is capable of capturing the distinctive structural characteristics of relations in HINs. In specific, we first explore the structural characteristics of heterogeneous relations in HINs with intuitive observations and thorough quantitative analysis. Intuitively, we find a huge difference between the degree distributions of nodes w.r.t. various relations, which indicates the distinctive structural characteristics of relations in HINs. From four real datasets, we systematically analyze the structural characteristics of relations in HINs, and present two structure-related measures which can consistently distinguish the various relations into two categories: Affiliation Relations (ARs) with one-centered-

by-another structures and Interaction Relations (IRs) with peer-to-peer structures. In order to capture the distinctive structural characteristics of the relations, we then propose two specifically designed models. For ARs where the nodes share similar properties [29], [30], we calculate Euclidean distance as the proximity between nodes, so as to make the nodes directly close in the low-dimensional space. On the other hand, for IRs which bridge two compatible nodes, we model them as translations between the nodes and calculate translation-based distance as the proximity between nodes. Since the Euclidean distance and translation-based distance are consistent in terms of mathematical form, they can be optimized in a unified and elegant way. Furthermore, considering that nodes connected via heterogeneous relations may have multi-aspect semantics and each relation focuses on one aspect, we introduce relation-specific projection matrices into the above two models to learn node and relation embeddings in separate spaces rather than a common space. We denote the extended model as **RHINE-M**. With the simultaneous projection and united training, RHINE-M ensures the unified measure of two proximities between nodes (i.e., the Euclidean distance and translation-based distance) in the specific space.

It is worthwhile to highlight our contributions as follows:

- To the best of our knowledge, we make the first attempt to explore the different structural characteristics of relations in HINs and present two structure-related criteria which can consistently distinguish heterogeneous relations into ARs and IRs.
- We propose a novel relation structure-aware HIN embedding model (RHINE) and its extended version RHINE-M, which fully respects the distinctive structural characteristics of ARs and IRs by exploiting appropriate models and combining them in a unified and elegant manner.
- We conduct comprehensive experiments to evaluate the performance of our models. Experimental results demonstrate that our models significantly outperform state-of-the-art network embedding models in various tasks.

Please notice that the preliminary work has been accepted for oral presentation at the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19) [31]. Based on the conference paper, we substantially extend the original work from the following four aspects: (1) Considering that nodes connected via heterogeneous relations may have multifaceted semantics and various relations focus on different aspects, we extend the original RHINE into a novel RHINE-M model with relation-specific projection matrices. RHINE-M learns node embeddings with respect to different relations in separate latent spaces, rather than a common space. Besides, in five data mining tasks, we demonstrate the superior performance of RHINE-M on four datasets. (2) In order to more intuitively illustrate the difference in various relations, we further analyze the distinctive structural characteristics of heterogeneous relations in HINs and conduct qualitative analysis from the perspective of the node degree distribution on four real-world HINs. (3) To further verify the effectiveness of the proposed models, we significantly enrich experiments with two new baselines (i.e., HERec and JUST), a new dataset (i.e., Amazon) and a new tasks

(i.e., node recommendation). Concretely, we evaluate our proposed RHINE and RHINE-M on the new dataset, and our methods consistently achieve outstanding performances in five tasks. Compared with two new baselines (i.e., HERec and JUST), our methods also perform better than them. (4) A comprehensive survey of related work is provided in our paper, ranging from general network embedding to HIN embedding. Besides, we carefully polish our paper and improve the language quality.

The rest of the paper is organized as follows. In Section 2, we summarize and compare the related works. Section 3 describes notations used in the paper and presents some preliminary knowledge. Then, we describe four real-world HINs and analyze the structural characteristics of relations in Section 4. Section 5 presents our proposed models, i.e., RHINE and RHINE-M in detail. Extensive experiments are done to validate the proposed models in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

In this section, we first introduce the related methods of general network embedding, and then discuss the recent works on HIN embedding.

2.1 Network Embedding

The goal of network embedding is to project a network into a low-dimensional vector space under the principle of preserving the original structural information and properties in networks [1], [2], [3]. Hence, network embedding can be traced back to the dimensionality reduction techniques. Traditional dimensionality reduction techniques typically learn the latent low-dimensional vectors for nodes or edges by decomposing a network [32], [33]. Graph factorization [34] represents a graph as a matrix where matrix elements correspond to edges between nodes, and then learns a low-dimensional representation of a graph through matrix factorization. However, decomposition-based models are inflexible and not scalable. They suffer from the computational cost of decomposing a large-scale matrix, making them neither practical nor effective for addressing data mining tasks in large networks.

Recently, rapid development on deep learning has shed a light on network representation learning and significant effort has been devoted to designing neural network-based representation learning models [4], [5], [6], [7], [10], [11], [13], [35], [36]. Inspired by word2vec [37], [38], Perozzi et al. propose DeepWalk [4] based on truncated random walks and Skip-gram model. DeepWalk views node sequences generated from random walks as "sentences", and nodes as "words", and then maximizes the co-occurrence probability among nodes. Based on DeepWalk, Grover et al. propose node2vec [10] which obtains nodes' neighbors by employing breadth-first and depth-first sampling on homogeneous networks. In addition, to learn representation of large-scale network, Tang et al. present LINE [5] which captures both the first-order and second-order proximities in networks. However, DeepWalk, node2vec and LINE only focus on learning the representation of nodes in homogeneous information networks, leading to the fact that they could not be directly applied for HIN embedding.

2.2 HIN Embedding

As a newly emerging network model, HINs can naturally model complex objects and their rich relations. HIN embedding, which aims to embed multiple types of nodes into a low-dimensional space, has received growing attention. Considerable researches have been done on representation learning for HINs [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Broadly speaking, the existing works on HIN embedding can be categorized into four types: random walk based methods [21], [23], [24], decomposition based methods [17], [22], [25], deep neural network based methods [19], [20], [26], [27] and task-specific methods [18], [20], [22], [27], [28].

2.2.1 Random Walk based Methods

Random walk based methods are inspired by word2vec [37], [38], where a node vector should be able to reconstruct the vectors of its neighborhood nodes which are defined by co-occurrence rate. Typically, metapath2vec [24] formalizes meta-path based random walks to construct heterogeneous neighborhoods of a node and leverages Skip-gram of word2vec [38] to learn the representation of networks. HIN2Vec [21] conducts random walk and learns latent vectors of nodes and meta-paths by conducting multiple prediction training tasks jointly. Shang et al. propose ESim [23] to perform random walks based on user-defined meta-paths on HINs, and learn the vector representation of nodes appearing in the instance by maximizing the probability of meta-path instances.

2.2.2 Decomposition based Methods

Decomposition based methods separate an HIN into multiple simple homogeneous networks, and respectively embed these networks into low-dimensional spaces. As an extension of LINE, PTE [17] is proposed to suit HIN embedding. It decomposes a HIN to a set of edgewise bipartite networks and then performs network embedding individually by using LINE. EOE [25] decomposes the complex academic heterogeneous network into a word co-occurrence network and an author cooperative network, and simultaneously performs representation learning on node pairs in subnetworks.

2.2.3 Deep Neural Network based Methods

Deep neural network based methods benefit from the powerful modeling capabilities of deep models, which employ different deep neural models, such as MLP, CNN and Autoencoder, etc. to model heterogeneous data. For instance, HNE [19] utilizes CNN and MLP to extract the features of text and image data respectively, and then projects different types of data into the same space through the transfer matrix to overcome the challenges of modeling heterogeneous data. SHINE [20] uses the autoencoder to encode and decode the heterogeneous information in the social network, the emotional network and the portrait network respectively to obtain the feature representation, and then fuses these representations through an aggregate function to obtain the final node embeddings.

2.2.4 Task-specific Methods

Task-specific methods mainly focus on solving a specific task (e.g., link prediction or recommendation) with representation learning on HINs. In order to predict links between nodes with different types in HINs, PME [28] projects different types of nodes into the same relation space and conducts heterogeneous link prediction. For recommendation in e-commerce, HERec [22] integrates matrix factorization with HIN embedding and predicts ratings for items. Fan et al. [39] proposes a embedding model metagraph2vec, where both the structures and semantics are maximally preserved for malware detection.

To sum up, all the above mentioned models deal with various relations without distinguishing their different properties and handle them with one single model. In this paper, we explore and distinguish the structural characteristics of relations with thorough qualitative and quantitative analysis. For relations with distinct structural and semantic characteristics, we propose to handle them with specifically designed models.

3 PRELIMINARIES

In this section, we introduce some basic concepts and formalize the problem of HIN embedding.

Definition 1. **Heterogeneous Information Network (HIN)** [15]. An HIN is defined as a graph $G = (V, E, T, \phi, \varphi)$, in which V and E are the sets of nodes and edges, respectively. Each node v and edge e are associated with their type mapping functions $\phi : V \rightarrow T_V$ and $\varphi : E \rightarrow T_E$, respectively. T_V and T_E denote the sets of node and edge types, where $|T_V| + |T_E| > 2$, and $T = T_V \cup T_E$.

In HINs, two objects can be connected via different semantic paths, called meta-paths, which describe composite relations between objects. We define the concept of meta-paths and node-relation triples as follows.

Definition 2. **Meta-path** [14]. Given an HIN $G = (V, E, T, \phi, \varphi)$, a meta-path $m \in M$ is defined as a sequence of node types t_{v_i} or edge types t_{e_j} in the form of $t_{v_1} \xrightarrow{t_{e_1}} t_{v_2} ... \xrightarrow{t_{e_l}} t_{v_{l+1}}$ (abbreviated as $t_{v_1} t_{v_2} ... t_{v_{l+1}}$), which describes a composite relation between v_1 and v_{l+1}.

Definition 3. **Node-Relation Triple**. In an HIN G, relations R include atomic relations (e.g., links) and composite relations (e.g., meta-paths). A node-relation triple $\langle u, r, v \rangle \in P$, describes that two nodes u and v are connected by a relation $r \in R$. Here P represents the set of all node-relation triples.

Example 1. As shown in Figure 1(a), a meta-path $A \xrightarrow{write} P \xrightarrow{Published} C$ (abbreviated as APC) describes a composite relation between authors and conferences, which indicates that *'authors write papers published in conferences'*. In Figure 1(a), both $\langle a_1, AP, p_1 \rangle$ and $\langle a_2, APC, c_2 \rangle$ are node-relation triples of DBLP.

Definition 4. **Heterogeneous Information Network Embedding**. Given an HIN $G = (V, E, T, \phi, \varphi)$, the goal of HIN embedding is to project nodes into a latent low-dimensional representation space while preserving the network structure and properties. Formally, we aim to develop a mapping function $f : V \rightarrow \mathbb{R}^d$ that projects each node $v \in V$ to a low-dimensional vector in $\mathbb{R}^d$, where $d \ll |V|$.

4 STRUCTURAL CHARACTERISTICS ANALYSIS OF RELATIONS

In this section, we first describe four real-world HINs and analyze the structural characteristics of relations in HINs. Then we present two structure-related measures which can consistently distinguish various relations quantitatively.

4.1 Dataset Description

Before analyzing the structural characteristics of relations, we first briefly introduce four datasets used in this paper, including DBLP[1], Yelp[2], AMiner[3] [40] and Amazon [4]. The detailed statistics of these datasets are illustrated in Table 1.

DBLP is an academic network, which contains four types of nodes: author (A), paper (P), conference (C) and term (T). We extract node-relation triples based on the set of relations {*AP, PC, PT, APC, APT*}. Yelp is a social network, which contains five types of nodes: user (U), business (B), reservation (R), service (S) and star level (L). We consider the relations {*BR, BS, BL, UB, BUB*}. AMiner is also an academic network, which contains four types of nodes, including author (A), paper (P), conference (C) and reference (R). We consider the relations {*AP, PC, PR, APC, APR*}. Amazon is a product purchasing dataset, from which we extract a subset that contains four types of nodes, including user (U), item (I), brand (B) and tag (T). We consider the relations {*UI, IB, IT, UIB, UIT*}. Notice that we can actually analyze all the relations based on meta-paths. However, not all meta-paths have a positive effect on embeddings [41], [42] and the selection of meta-path is still an open question [43]. In addition, our work focuses on the exploration of structural characteristics of various relations in HINs, rather than the selection of meta-paths. Hence, following previous works [23], [24], we choose the popular and meaningful meta-paths in this work.

4.2 Affiliation Relations and Interaction Relations

In order to explore the structural characteristics of relations, we present mathematical analysis on the above datasets.

4.2.1 Data Observation

Since the degree of nodes can well reflect the structures of networks [44], we first conduct some data observation with respect to node degree distributions on four datasets. To be specific, given a relation r, we have the node types at both ends of the relation r, denoted as t_u and t_v. Then, we calculate two distributions of node degree under the relation r, denoted as $Dis(t_u|r)$ and $Dis(t_v|r)$. Here, we plot the degree distribution of some typical relations on four datasets due to similar trends of other relations. For DBLP dataset, we show the degree distributions of nodes with type A and type C with respect to *APC* in Figure 2(a), and the degree distributions of nodes with type A and type T with respect to *APT* in Figure 2(d). For Yelp and AMiner, we show the degree distributions of nodes in Figure 2(b)(e) and Figure 2(c)(f), respectively.

---

1. https://dblp.uni-trier.de
2. https://www.yelp.com/dataset/
3. https://www.aminer.cn/citation
4. http://snap.stanford.edu/data/index.html

TABLE 1
Statistics of the four datasets. t_u denotes the type of node u, $\langle u, r, v \rangle$ is a node-relation triple.

Datasets	Nodes	Number of Nodes	Relations ($t_u \sim t_v$)	Number of Relations	Avg. Degree of t_u	Avg. Degree of t_v	Measures		Relation Category
							$D(r)$	$S(r)$	
DBLP	Term (T) Paper (P) Author (A) Conference (C)	8,811 14,376 14,475 20	PC APC AP PT APT	14,376 24,495 41,794 88,683 260,605	1.0 2.9 2.9 6.2 18.0	718.8 2089.7 2.9 10.7 29.6	718.8 720.6 1.0 1.7 1.6	0.05 0.085 0.0002 0.0007 0.002	AR AR IR IR IR
Yelp	User (U) Service (S) Business (B) Star Level (L) Reservation (R)	1,286 2 2,614 9 2	BR BS BL UB BUB	2,614 2,614 2,614 30,838 528,332	1.0 1.0 1.0 23.9 405.3	1307.0 1307.0 290.4 11.8 405.3	1307.0 1307.0 290.4 2.0 1.0	0.5 0.5 0.1 0.009 0.07	AR AR AR IR IR
AMiner	Paper (P) Author (A) Reference (R) Conference (C)	127,623 164,472 147,251 101	PC APC AP PR APR	127,623 232,659 355,072 392,519 1,084,287	1.0 2.2 2.2 3.1 7.1	1263.6 3515.6 2.8 2.7 7.9	1263.6 1598.0 1.3 1.1 1.1	0.01 0.01 0.00002 0.00002 0.00004	AR AR IR IR IR
Amazon	Tag (T) Item (I) User (U) Brand (B)	22,140 8,493 15,619 22	IB UIB UI IT UIT	8,493 16,789 23,493 39,528 117,618	1.0 1.5 1.5 4.6 7.8	386.1 1067.9 2.7 1.8 5.5	386.1 711.9 1.8 2.6 1.4	0.05 0.05 0.0002 0.0002 0.0003	AR AR IR IR IR

(a) Degree distribution w.r.t. APC on DBLP (b) Degree distribution w.r.t. BL on Yelp (c) Degree distribution w.r.t. APC on AMiner (d) Degree distribution w.r.t. UIB on Amazon

(e) Degree distribution w.r.t. APT on DBLP (f) Degree distribution w.r.t. UB on Yelp (g) Degree distribution w.r.t. APR on AMiner (h) Degree distribution w.r.t. UIT on Amazon
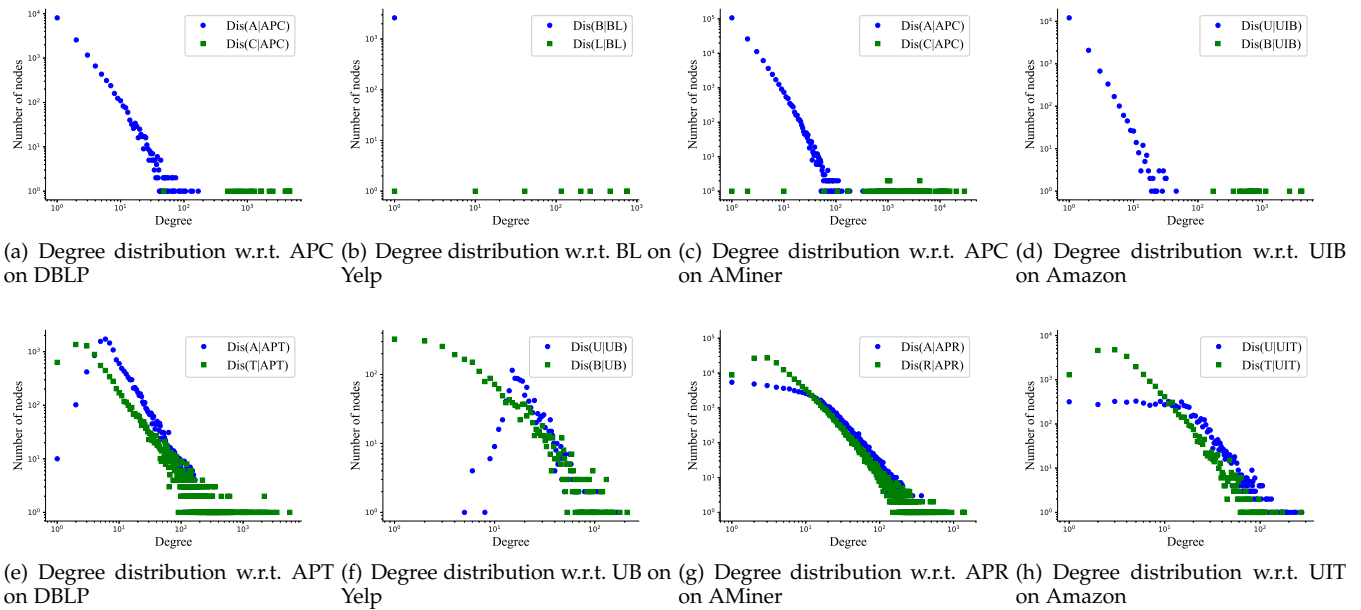
Fig. 2. Degree distribution of nodes connected via different relations on four datasets.

Observing the degree distribution of nodes connected via different relations on the same dataset (i.e., compare the up and down figures in Figure 2), we can find that different relations have obviously distinct structural characteristics. Taking the DBLP dataset as an example, we can observe from Figure 2(a)(d) that the degree distributions of nodes with respect to relation *APC* and *APT* are quite different. In terms of the relation *APC* (i.e. Figure 2(a)), we find that the degree distribution of nodes with type A is significantly different from that of nodes with type C. This fact implies that two types of nodes connected via relation *APC* are extremely unbalanced, and such relation means an inequivalent structure. On the other hand, the

two degree distributions of nodes with respect to relation *APT* are almost identical, which shows that the similar and compatible structural roles of two types of nodes connected via relation *APT*. With the above analysis in terms of the degree distribution of nodes, we find that different relations in an HIN have quite distinctive structural characteristics, which should be carefully analyzed and considered for HIN embedding.

4.2.2 Quantitative Analysis

Data observation only gives us an intuitive understanding, and more accurate and reasonable quantitative analysis is a must. Hence, we define a degree-based measure $D(r)$

(a) An example of HIN   (b) Structural characteristics analysis of relations in an HIN   (d) RHINE-M   (e) Node embeddings
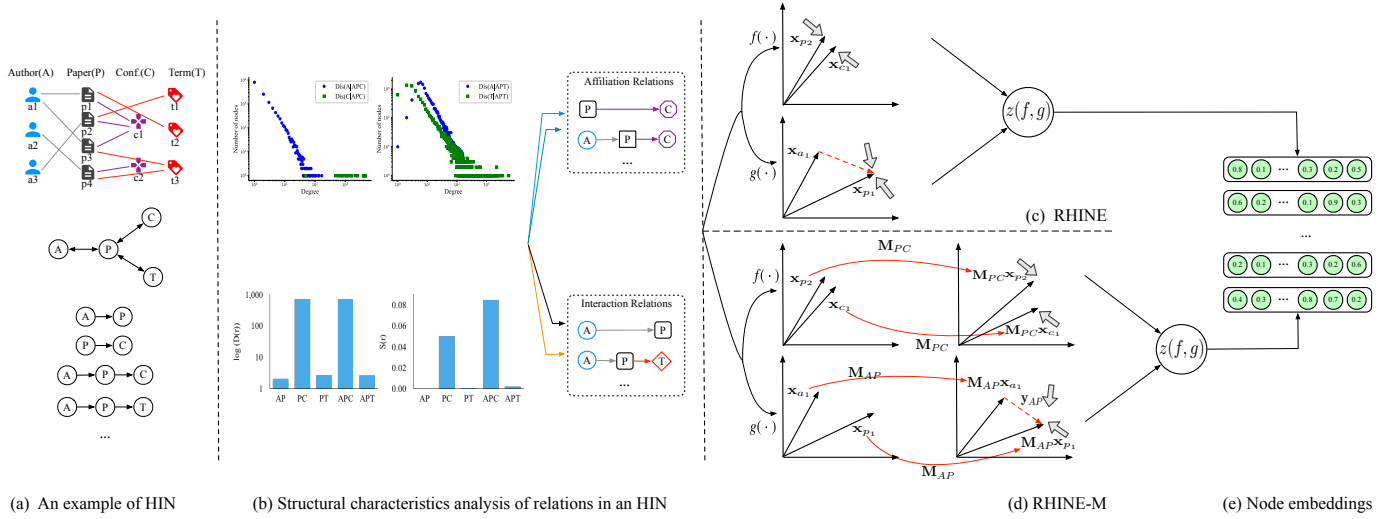
Fig. 3. The overall architecture of the proposed models. (a) An HIN can be abstracted as a network schema describing various relations in the network. (b) Analysis of structural characteristics clearly divides heterogeneous relations into ARs and IRs in an HIN. (c) The proposed RHINE individually handles ARs and IRs with two carefully designed models. (d) The proposed RHINE-M extends RHINE with relation-specific projection matrices. (e) RHINE and RHINE-M derive node embeddings, respectively.

to explore the distinction of various relations in an HIN. Specifically, we compare the average degrees of two types of nodes connected with the relation r, via dividing the larger one by the smaller one ($D(r) \geq 1$). Formally, given a relation r with nodes u and v (i.e., node relation triple $\langle u, r, v \rangle$), t_u and t_v are the node types of u and v, we define $D(r)$ as follows:

$$D(r) = \frac{\max[\bar{d}_{t_u}, \bar{d}_{t_v}]}{\min[\bar{d}_{t_u}, \bar{d}_{t_v}]}, \quad (1)$$

where $\bar{d}_{t_u}$ and $\bar{d}_{t_v}$ are the average degrees of nodes of the types t_u and t_v respectively.

A large value of $D(r)$ indicates quite inequivalent structural roles of two types of nodes connected via the relation r (one-centered-by-another), while a small value of $D(r)$ means compatible structural roles (peer-to-peer). In other words, relations with a large value of $D(r)$ show much stronger affiliation relationships. Nodes connected via such relations share much more similar properties [29]. While relations with a small value of $D(r)$ implicate much stronger interaction relationships. Therefore, we call the two categories of relations as *Affiliation Relations* (ARs) and *Interaction Relations* (IRs), respectively.

In order to better understand the structural difference between various relations, we take the DBLP network as an example. As shown in Table 1, for the relation PC with $D(PC) = 718.8$, the average degree of nodes with type P is 1.0 while that of nodes with type C is 718.8. It shows that papers and conferences are structurally inequivalent. Papers are centered by conferences. While $D(AP) = 1.1$ indicates that authors and papers are compatible and peer-to-peer in structure. This is consistent with our common sense. Semantically, the relation PC means that *'papers are published in conferences'*, indicating an affiliation relationship. Differently, AP means that *'authors write papers'*, which explicitly describes an interaction relationship.

In fact, we can also define some other measures to capture the structural difference. For example, we compare

the relations in terms of sparsity, which can be defined as:

$$S(r) = \frac{N_r}{N_{t_u} \times N_{t_v}}, \quad (2)$$

where N_r represents the number of relation instances following r. N_{t_u} and N_{t_v} mean the number of nodes with type t_u and t_v, respectively. The measure can also consistently distinguish the relations into two categories: ARs and IRs. The detailed statistics of all the relations in the four HINs are shown in Table 1.

Evidently, Affiliation Relations and Interaction Relations exhibit rather distinct characteristics: (1) ARs indicate one-centered-by-another structures, where the average degrees of the types of end nodes are extremely different. They imply an affiliation relationship between nodes. (2) IRs describe peer-to-peer structures, where the average degrees of the types of end nodes are compatible. They suggest an interaction relationship between nodes.

5 RELATION STRUCTURE-AWARE HIN EMBEDDING

In this section, we present a novel Relation structure-aware HIN Embedding model (RHINE), which individually handles two categories of relations (ARs and IRs) with different models in order to preserve their distinct structural characteristics, as illustrated in Figure 3.

5.1 Basic Idea

Through our exploration with thorough mathematical analysis, we find that the heterogeneous relations can be typically divided into ARs and IRs with different structural characteristics. In order to respect their distinct characteristics, we need to specifically design different while appropriate models for the different categories of relations.

For ARs, we propose to take Euclidean distance as a metric to measure the proximity of the connected nodes in the low-dimensional space. There are two motivations

behind this: (1) First of all, ARs show affiliation structures between nodes, which indicate that nodes connected via such relations share similar properties [29], [30]. Hence, nodes connected via ARs could be directly close to each other in the vector space, which is also consistent with the optimization of Euclidean distance [45]. (2) Additionally, one goal of HIN embedding is to preserve the high-order proximity. Euclidean distance can ensure that both first-order and second-order proximities are preserved as it meets the condition of the triangle inequality [46].

Different from ARs, IRs indicate strong interaction relationships between compatible nodes, which themselves contain important structural information of two nodes. Thus, we propose to explicitly model an IR as a translation between nodes in the low-dimensional vector space. Additionally, the translation based distance is consistent with the Euclidean distance in the mathematical form [47]. Therefore, they can be smoothly combined in a unified and elegant manner.

5.2 Different Models for ARs and IRs

In this subsection, we introduce two different models exploited in RHINE for ARs and IRs, as shown in Figure 3(c).

5.2.1 Euclidean Distance for Affiliation Relations

Nodes connected via ARs share similar properties [29], therefore nodes could be directly close to each other in the vector space. We take the Euclidean distance as the proximity measure of two nodes connected by an AR.

Formally, given an affiliation node-relation triple $\langle p, s, q \rangle \in P_{AR}$ where $s \in R_{AR}$ is the relation between p and q with weight w_{pq}, the distance between p and q in the latent vector space is calculated as follows:

$$f(p, q) = w_{pq}||\mathbf{x}_p - \mathbf{x}_q||_2^2, \qquad (3)$$

in which $\mathbf{x}_p \in \mathbb{R}^d$ and $\mathbf{x}_q \in \mathbb{R}^d$ are the embedding vectors of p and q, respectively. As $f(p, q)$ quantifies the distance between p and q in the low-dimensional vector space, we aim to minimize $f(p, q)$ to ensure that nodes connected by an AR should be close to each other. Hence, we define the margin-based loss [47] function as follows:

$$L_{EuAR} = \sum_{s \in R_{AR}} \sum_{\langle p, s, q \rangle \in P_{AR}} \sum_{\langle p', s, q' \rangle \in P'_{AR}} \max[0, \gamma + f(p, q) - f(p', q')], \qquad (4)$$

where $\gamma > 0$ is a margin hyperparameter. P_{AR} is the set of positive affiliation node-relation triples, while P'_{AR} is the set of negative affiliation node-relation triples.

5.2.2 Translation-based Distance for Interaction Relations

Interaction Relations demonstrate strong interactions between nodes with compatible structural roles. Thus, different from ARs, we explicitly model IRs as translations between nodes.

Formally, given an interaction node-relation triple $\langle u, r, v \rangle$ where $r \in R_{IR}$ with weight w_{uv}, we define the score function as:

$$g(u, v) = w_{uv}||\mathbf{x}_u + \mathbf{y}_r - \mathbf{x}_v||, \qquad (5)$$

Algorithm 1: The Optimization Model Algorithm

Input: An HIN $G = (V, E, T, \phi, \varphi)$, AR set R_{AR}, IR set R_{IR}, interaction triple batch size $batch\_size_{IR}$, affiliation triple batch size $batch\_size_{AR}$, margin γ, embedding dimension d, negative samples k, epochs I

Output: Node embedding matrix $\mathbf{X} \in \mathbb{R}^{|V| \times d}$ and relation embedding matrix $\mathbf{Y}^{|R| \times d}$.

Initialize

$\mathbf{X} \in \mathbb{R}^{|V| \times d}$ and $\mathbf{Y}^{|R| \times d}$ with uniform distribution;

Interaction node-relation triples $P_{IR} \leftarrow \emptyset$;

Affiliation node-relation triples $P_{AR} \leftarrow \emptyset$.

for *each interaction relation r in R_{IR}* **do**
  | P_{IR}.add(GenerateTriples(r))
end

for *each affiliation relation s in R_{AR}* **do**
  | P_{AR}.add(GenerateTriples(s))
end

$\mathbf{X}, \mathbf{Y}$ = RHINE(P_{IR}, P_{AR})

return $\mathbf{X}, \mathbf{Y}$;

RHINE(P_{IR}, P_{AR})

$iter \leftarrow 0$

repeat
  | Sample($P_{IR}, batch\_size_{IR}$);
  | Compute gradients; Update node embeddings $\mathbf{X}$ and relation embeddings $\mathbf{Y}$;
  | Sample($P_{AR}, batch\_size_{IR}$);
  | Compute gradients; Update node embeddings $\mathbf{X}$;
  | $iter \leftarrow iter + 1$;
until *iteration* $>= I$;

return $\mathbf{X}, \mathbf{Y}$;

---

where $\mathbf{x}_u$ and $\mathbf{x}_v$ are the node embeddings of u and v respectively, and $\mathbf{y}_r$ is the embedding of the relation r. Intuitively, this score function penalizes deviation of $(\mathbf{x}_u + \mathbf{y}_r)$ from the vector $\mathbf{x}_v$.

For each interaction node-relation triple $\langle u, r, v \rangle \in P_{IR}$, we define the margin-based loss function as follows:

$$L_{TrIR} = \sum_{r \in R_{IR}} \sum_{\langle u, r, v \rangle \in P_{IR}} \sum_{\langle u', r, v' \rangle \in P'_{IR}} \max[0, \gamma + g(u, v) - g(u', v')] \qquad (6)$$

where P_{IR} is the set of positive interaction node-relation triples, while P'_{IR} is the set of negative interaction node-relation triples.

5.3 A Unified Model for HIN Embedding

Finally, we smoothly combine the two models for different categories of relations by minimizing the following loss function:

$$L = L_{EuAR} + L_{TrIR} \qquad (7)$$

$$= \sum_{s \in R_{AR}} \sum_{\langle p, s, q \rangle \in P_{AR}} \sum_{\langle p', s, q' \rangle \in P'_{AR}} \max[0, \gamma + f(p, q) - f(p', q')]$$
$$+ \sum_{r \in R_{IR}} \sum_{\langle u, r, v \rangle \in P_{IR}} \sum_{\langle u', r, v' \rangle \in P'_{IR}} \max[0, \gamma + g(u, v) - g(u', v')].$$

5.3.1 Sampling Strategy

As shown in Table 1, the distributions of ARs and IRs are quite unbalanced. Additionally, the proportion of relations are unbalanced within ARs and IRs. Traditional edge sampling may suffer from under-sampling for relations with a small amount or over-sampling for relations with a large amount. To address the problems, we draw positive samples according to their probability distributions. As for negative samples, we follow previous work [47], [48] to construct a set of negative node-relation triples $P'_{(u,r,v)} = \{(u', r, v)|u' \in V\} \cup \{(u, r, v')|v' \in V\}$ for the positive node-relation triple (u, r, v), where either the head or tail is replaced by a random node, but not both at the same time.

5.3.2 Optimization Algorithm

With the objective function (i.e., Eq. 7), we adopt stochastic gradient descent algorithm for optimization. We optimize Euclidean Distance based model and Translation-based distance model, step by step. To be specific, we first initialize node embedding matrix $\mathbf{X}$ and relation embedding matrix $\mathbf{Y}$ with uniform distribution. Then we sample positive and negative interaction node-relation triples, and update $\mathbf{X}$ and $\mathbf{Y}$ correspondingly. In the same way, we sample positive and negative affiliation node-relation triples, and only update $\mathbf{X}$. Finally, the node embedding matrix $\mathbf{X}$ and relation embedding matrix $\mathbf{Y}$ are returned. The detail of the optimization algorithm is illustrated in Algorithm 1. The code of our model is publicly available at website [1].

5.4 The Extended Model RHINE-M

The proposed RHINE takes into account the structural characteristics of heterogeneous relations for HIN embedding, which embeds various nodes and relations in a common low-dimension latent space. However, nodes of the same type connected via heterogeneous relations may have multi-aspect semantics and each relation focus on one aspect. For instance, an author (i.e., a node of type A) may indicate a writer or participant, depending on whether he/she is in the *AP* relation or *APC* relation. Hence, we need to project node embeddings into a relation-specific representation space. Following this idea, we extend the original RHINE to RHINE-M with relation-specific projection matrices, as shown in Figure 3(d).

In specific, we introduce a relation-specific projection matrix $\mathbf{M}_r \in \mathbb{R}^{d \times d}$ to project nodes from the node representation space to corresponding relation-specific spaces. The projection matrix makes it much more effective to measure proximities between nodes in the relation-specific space, thus RHINE-M is capable of capturing structures and semantics in an HIN at a finer-grained manner. Hence, we redefine Eq. (3) as follows:

$$f(p, q) = w_{pq}||\mathbf{M}_s\mathbf{x}_p - \mathbf{M}_s\mathbf{x}_q||_2^2, \qquad (8)$$

where $\mathbf{M}_s$ is the relation s-specific projection matrix. With the above definition, we can directly enable nodes connected via ARs closer and push nodes without ARs farther in the certain relation space.

1. https://github.com/rootlu/RHINE

Similarly, we leverage projection matrices to enhance the embeddings learned with translation-based distance. Thus, we reformulate Eq. (5) as follows:

$$g(u, v) = w_{uv}||\mathbf{M}_r\mathbf{x}_u + \mathbf{y}_r - \mathbf{M}_r\mathbf{x}_v||, \qquad (9)$$

where $\mathbf{M}_r$ is also a projection matrix corresponding to the relation r.

Notice that here we apply the relation-specific projection matrix for both the Euclidean distance and the translation-based distance, which not only enables node proximities to be measured in the unified space, but also ensures that our RHINE-M can be jointly trained. Assuming that we only apply the transformation matrix to the Euclidean distance, the node proximity measured in our model will not be in the same space. That is, the node proximity based on the Euclidean distance is measured in the node space, while the node proximity of the translation-based distances is measured in the relation-specific space, which is inconsistent with our motivation.

With the above definition, we can project nodes from node space to relation-specific space, which makes nodes that actually hold the relation close with each other, otherwise far away. By replacing Eqs. (3) (5) with Eqs. (8) (9) in Eq. (7), we initialize RHINE-M with node embeddings learned using RHINE and optimize RHINE-M in the same way as RHINE. At last, we can learn the node embeddings for downstream tasks.

5.5 Discussion of RHINE and RHINE-M

By taking the structural characteristics of heterogeneous relations into consideration, the proposed RHINE specifically tailors the Euclidean distance and translation-based distance to respect the distinctive structural characteristics of relations in HINs. With the Euclidean distance for ARs, nodes connected via affiliation relations, i.e., one centered by another, are directly pushed close in the latent low-dimensional space. On the other hand, translation-based distance bridges two compatible nodes with relation representations. Jointly training the Euclidean distance and translation-based distance ensures that our RHINE can capture the structural characteristics of heterogeneous relations.

RHINE assumes embeddings of nodes and relations being in the same latent low-dimensional space. However, nodes connected via heterogeneous relations may have multifaceted semantics and various relations focus on different aspects. Hence, it is intuitive that heterogeneous nodes and relations in HINs should be embedded in distinct spaces. Inspired by TransH [49] and TransR [48], we introduce a relation-specific projection matrix to map nodes from the node space to the corresponding relation-specific space. Different from TransR modeling knowledge graph with translation mechanism, we not only apply relation-specific matrix for translation-based distance but also the Euclidean distance. As mentioned before, with the simultaneous projection and united training, RHINE-M ensures the unified measure of two proximities between nodes (i.e., the Euclidean distance and translation-based distance) in the specific space.

6 EXPERIMENTS

In this section, we conduct extensive experiments to demonstrate the effectiveness of our model RHINE.

6.1 Datasets

As described in Subsection 4.1, we conduct experiments on four datasets, including DBLP, Yelp, AMiner and Amazon. The statistics of them are summarized in Table 1.

- **DBLP** is an academic dataset in computer science, consisting of 14,456 authors, 14,375 papers, 8,811 terms and 20 conferences. In DBLP, 200 papers are labeled with their research areas such as data mining. We conduct clustering and classification on papers.
- **Yelp** is a social media dataset provided by Yelp Challenge. We extract information related to restaurants of three sub-categories: "American (New) Food", "Fast Food" and "Sushi Bars" [42], and construct a HIN, which includes 1,286 users, 2,614 businesses, 9 star levels, 2 service types and 2 reservation types. We conduct clustering and classification on businesses.
- **AMiner** is also an academic network in computer science, which is much larger than DBLP, including 164,472 authors, 127,623 papers, 147,251 references and 101 conferences. We take 10 domains related to computer science as labels of papers, and perform clustering and classification on papers.
- **Amazon** is a product purchasing dataset. We extract information and construct a HIN, which contains 15,619, users, 8,493 items, 22,140 tags and 22 brands. We take 9 categories as labels of items (e.g., sports, electronics, etc), and perform clustering and classification on items.

6.2 Baseline Methods

We compare our proposed model with seven state-of-the-art network embedding methods, the first two of which are designed for homogeneous networks and the rest are capable of modeling heterogeneous information networks. We use codes of the baseline methods provided by their authors.

- **DeepWalk** [4] [2] performs a random walk on networks and then learns low-dimensional node vectors via the skip-gram model.
- **LINE** [5] [3] considers first-order and second-order proximities in networks. We denote the model that only uses first-order or second-order proximity as LINE-1st or LINE-2nd, respectively.
- **PTE** [17] [4] decomposes an HIN to a set of bipartite networks and then learns the low-dimensional representation of the network.
- **ESim** [23] [5] takes a given set of meta-paths as input to learn a low-dimensional vector space. For a fair comparison, we use the same meta-paths with equal weights in Esim and our model RHINE.

- **HIN2Vec** [21] [6] learns the latent vectors of nodes and meta-paths in an HIN by conducting multiple prediction training tasks jointly.
- **Metapath2vec** [24] [7] leverages meta-path based random walks and skip-gram model to perform node embedding. We leverage the meta-paths *APCPA*, *UBSBU* and *APCPA* in DBLP, Yelp and AMiner respectively, which perform best in the evaluations.
- **HERec** [22] [8] designs a type constraint strategy to filter the node sequence and utilizes Skip-gram to embed the heterogeneous information network.
- **JUST** [50] [9] is a heterogeneous graph embedding technique using random walks with jump and stay strategies to learn node embeddings in an more efficient manner.

6.2.1 Parameter Settings

For a fair comparison, we set the embedding dimension $d = 100$ and the size of negative samples $k = 3$ for all models. For DeepWalk, HIN2Vec, metapath2vec and JUST based on random walk, we set the number of walks per node $w = 10$, the walk length $l = 100$ and the window size $\tau = 5$. For our models RHINE and RHINE-M, margin γ is set to 1 and learning rate $\alpha = 0.005$. We set the batch size and the number of epochs to 128 and 400, respectively.

6.3 Node Clustering

We conduct node clustering experiments to illustrate how the latent representations learned by embedding methods can benefit the node clustering task in HINs [51], [52].

6.3.1 Experimental Setting

In this task, we learn the representations of nodes using network embedding methods mentioned above. Based on the learned node embeddings, we leverage the K-means algorithm to cluster the nodes and evaluate the clustering results in terms of normalized mutual information (NMI) [53]. All clustering experiments are conducted 10 times. We report the average performance in Table 2.

6.3.2 Results

As shown in Table 2, our model RHINE significantly outperforms all the compared methods. (1) Compared with the best competitors, the clustering performance of our model RHINE improves by 18.79%, 6.15%, 7.84%, 6.12% on DBLP, Yelp, AMiner and Amazon, respectively. It demonstrates the effectiveness of our model RHINE by distinguishing the various relations with different structural characteristics in an HIN. In addition, it also validates that we utilize appropriate models for different categories of relations. (2) In all baseline methods, homogeneous network embedding models achieve the lowest performance, because they ignore the heterogeneity of relations and nodes. (3) RHINE and RHINE-M significantly outperform existing HIN embedding models (i.e., ESim, HIN2Vec, metapath2vec, HERec and JUST) on all datasets. We believe the reason is that

---

2. https://github.com/phanein/deepwalk
3. https://github.com/tangjianpku/LINE
4. https://github.com/mnqu/PTE
5. https://github.com/shangjingbo1226/ESim

6. https://github.com/csiesheep/hin2vec
7. https://ericdongyx.github.io/metapath2vec/m2v.html
8. https://github.com/librahu/HERec
9. https://github.com/eXascaleInfolab/JUST

TABLE 2
Performance Evaluation of Node Clustering.

Methods	DBLP	Yelp	AMiner	Amazon
DeepWalk	0.3884	0.3043	0.5427	0.7062
LINE-1st	0.2775	0.3103	0.3736	0.5997
LINE-2nd	0.4675	0.3593	0.3862	0.6709
PTE	0.3101	0.3527	0.4089	0.6692
ESim	0.3449	0.2214	0.3409	0.6403
HIN2Vec	0.4256	0.3657	0.3948	0.6794
metapath2vec	0.6065	0.3507	0.5586	0.7145
HERec	0.5893	0.3313	0.5123	0.6878
JUST	0.5634	0.3516	0.5021	0.6894
RHINE	0.7204	0.3882	0.6024	0.7435
RHINE-M	**0.7323**	**0.3934**	**0.6152**	**0.7582**

our proposed methods with appropriate models for different categories of relations can better capture the structural and semantic information of HINs. Furthermore, RHINE-M performs slightly better than RHINE due to the separate relation-specific representation spaces.

6.4 Link Prediction

Link prediction aims to estimate the likelihood of the existence of a link between two nodes in a network. Formally, given a node pair $\langle u, v \rangle$, we aim to predict whether there exists a relation r between them in the network [54].

6.4.1 Experimental Setting

We model the link prediction problem as a binary classification problem that aims to predict whether a link exists. In this task, we conduct co-author (*A-A*) and author-conference (*A-C*) link prediction for DBLP and AMiner. For Yelp and Amazon, we predict user-business (*U-B*) and user-item (U-I) links which indicate whether a user reviews a business/item. We first randomly separate the original network into training network and testing network, where the training network contains 80% relations to be predicted (i.e., *A-A*, *A-C*, *U-B* and *U-I*) and the testing network contains the rest. Then, we train the embedding vectors on the training network and evaluate the prediction performance on the testing network.

6.4.2 Results

The results of link prediction task are reported in Table 3 with respect to AUC and F1 score. It is clear that both RHINE and RHINE-M perform better than all baseline methods on four datasets. The reason behind the improvement is that our models based on Euclidean distance modeling relations can capture both the first-order and second-order proximities. In addition, our models distinguishes multiple types of relations into two categories in terms of their structural characteristics, and thus can learn better embeddings of nodes, which are beneficial for predicting complex relationships between two nodes. Since RHINE-M learn node embeddings in relation-specific spaces, it can captures much more semantics and structural information.

6.5 Multi-Class Classification

Multi-class classification is a common task to evaluate the performance of representation learning on networks. In this

task, we use the labeled data to train a classifier and evaluate the performance on test set [55], [56].

6.5.1 Experimental Setting

In this task, we employ the same labeled data used in the clustering task. After learning the node vectors, we train a logistic classifier with 40%, 60% and 80% of the labeled nodes and test with the remaining data. We use Micro-F1 and Macro-F1 score as the metrics for evaluation [24].

6.5.2 Results

We summarize the results of classification in Table 4. As we can observe, (1) RHINE and RHINE-M achieve better performance than all baseline methods on all datasets except Aminer. It improves the performance of node classification by about 4% on both DBLP and Yelp averagely. On Amazon dataset, our models RHINE and RHINE-M continue to perform well in most cases. In terms of AMiner, our models perform slightly worse than ESim, HIN2vec and metapath2vec. This may be caused by over-capturing the information of relations *PR* and *APR* (*R* represents references). Since an author may write a paper referring to various fields, these relations may introduce some noise. (2) Although ESim, HIN2Vec and JUST can model multiple types of relations in HINs, they fail to perform well in most cases. Our model RHINE and RHINE-M achieve good performance due to the respect of distinct characteristics of various relations. (3) The stable performances of our methods against different training ratio indicates the robustness of our learned node embeddings when served as features for node classification.

6.6 Node Recommendation

Node recommendation is a wide application of network embedding, such as business or item recommendation in social network (e.g., Yelp and Amazon). Note that the node recommendation is different from link prediction in several ways. Specifically, node recommendation can be regarded as a ranking task while link prediction can be taken as a binary classification. Additionally, node recommendation evaluates the performance of network embedding methods from the node view, which is totally different from link prediction in terms of edges [8], [22].

6.6.1 Experimental Setting

In this task, we aim to recommend conferences to an author in DBLP and AMiner networks (i.e., *A-C*), and we also perform business and item recommendation for a user in Yelp network and Amazon network (i.e., *U-B* and *U-I*). In specific, we first divide the original data set into a training set and test set, as in the link prediction task, and learn node embeddings on training set. Then, for each node v_i in test set, we calculate the similarity between the embedding of node v_i and those of other nodes in the network, referring as the ranking score between two nodes. After that, we sort the ranking scores of node v_i against other nodes and derive the top-k nodes with the highest scores as candidates. At last, we compare the ground truth with the candidates and use Hit@k to evaluate the quality of recommendation.

TABLE 3
Performance Evaluation of Link Prediction.

Methods	DBLP (A-A)		DBLP (A-C)		Yelp (U-B)		AMiner (A-A)		AMiner (A-C)		Amazon (U-I)	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
DeepWalk	0.9131	0.8246	0.7634	0.7047	0.8476	0.6397	0.9122	0.8471	0.7701	0.7112	0.9045	0.8978
LINE-1st	0.8264	0.7233	0.5335	0.6436	0.5084	0.4379	0.6665	0.6274	0.7574	0.6983	0.8810	0.8619
LINE-2nd	0.7448	0.6741	0.8340	0.7396	0.7509	0.6809	0.5808	0.4682	0.7899	0.7177	0.8879	0.8793
PTE	0.8853	0.8331	0.8843	0.7720	0.8061	0.7043	0.8119	0.7319	0.8442	0.7587	0.9014	0.8991
ESim	0.9077	0.8129	0.7736	0.6795	0.6160	0.4051	0.8970	0.8245	0.8089	0.7392	0.8992	0.8875
HIN2Vec	0.9160	0.8475	0.8966	0.7892	0.8653	0.7709	0.9141	0.8566	0.8099	0.7282	0.9012	0.8981
metapath2vec	0.9153	0.8431	0.8987	0.8012	0.7818	0.5391	0.9111	0.8530	0.8902	0.8125	0.9388	0.9145
HERec	0.9178	0.8523	0.8874	0.8132	0.7923	0.5742	0.9058	0.8514	0.8834	0.8067	0.9295	0.9017
JUST	0.9018	0.8103	0.8174	0.7532	0.7711	0.5932	0.8823	0.8141	0.8045	0.7603	0.8955	0.8912
RHINE	0.9315	0.8664	0.9148	0.8478	0.8762	0.7912	0.9316	0.8664	0.9173	0.8262	0.9561	0.9207
RHINE-M	**0.9421**	**0.8726**	**0.9218**	**0.8508**	**0.8796**	**0.7994**	**0.9460**	**0.8714**	**0.9207**	**0.8322**	**0.9634**	**0.9371**

TABLE 4
Performance Evaluation of Multi-class Classification. Tr.Ra is the training ratio. Ma.-F1 and Mi.-F1 mean Macro-F1 and Micro-F1. D.W. L-1st, L-2nd, H2Vec and mp2vec represent DeepWalk, LINE-1st, LINE-2nd, HIN2Vec and metapath2vec, respectively.

Datasets	Metrics	Tr.Ra	D.W.	L-1st	L-2nd	PTE	ESim	H2Vec	mp2vec	HERec	JUST	RHINE	RHINE-M
DBLP	Ma.-F1	40%	0.6754	0.7237	0.7488	0.7723	0.8069	0.9073	0.9084	0.9005	0.8612	0.9261	**0.9301**
		60%	0.7283	0.7311	0.7338	0.7768	0.8161	0.8600	0.8941	0.8963	0.8733	0.9257	**0.9387**
		80%	0.7475	0.8091	0.7559	0.8852	0.8867	0.8631	0.8976	0.9101	0.8745	0.9344	**0.9445**
	Mi.-F1	40%	0.6937	0.7250	0.7583	0.7629	0.8000	0.9083	0.9098	0.9000	0.8567	0.9125	**0.9237**
		60%	0.7125	0.7500	0.7375	0.7875	0.8125	0.8625	0.8950	0.8915	0.8654	0.9051	**0.9254**
		80%	0.7500	0.8250	0.7500	0.8750	0.8750	0.8500	0.9000	0.9021	0.8705	0.9250	**0.9304**
Yelp	Ma.-F1	40%	0.6708	0.4607	0.5393	0.5397	0.6799	0.6104	0.5613	0.5509	0.5583	0.6909	**0.6998**
		60%	0.6717	0.4681	0.5299	0.5407	0.6830	0.6032	0.5478	0.5557	0.5632	0.7021	**0.7110**
		80%	0.6723	0.4872	0.5304	0.5389	0.6836	0.6075	0.5337	0.5517	0.5701	0.7132	**0.7205**
	Mi.-F1	40%	0.6732	0.6680	0.6637	0.7297	0.7247	0.7342	0.7074	0.7265	0.6002	0.7476	**0.7523**
		60%	0.6893	0.6537	0.6857	0.7323	0.7358	0.7189	0.7171	0.7304	0.6201	0.7562	**0.7598**
		80%	0.7012	0.6639	0.7377	0.7342	0.7399	0.7361	0.7208	0.7323	0.6295	0.7572	**0.7634**
AMiner	Ma.-F1	40%	0.9421	0.9473	0.9392	0.9649	0.9898	**0.9955**	0.9895	0.9901	0.9535	0.9798	0.9832
		60%	0.9423	0.9456	0.9467	0.9736	0.9904	**0.9961**	0.9912	0.9942	0.9684	09845	0.9893
		80%	0.9386	0.9494	0.9468	0.9791	0.9910	**0.9962**	0.9934	0.9956	0.9745	0.9884	0.9923
	Mi.-F1	40%	0.9525	0.9471	0.9397	0.9754	0.9936	**0.9958**	0.9901	0.9874	0.9684	0.9804	0.9848
		60%	0.9533	0.9527	0.9484	0.9813	0.9943	**0.9962**	0.9916	0.9953	0.9734	0.9845	0.9897
		80%	0.9512	0.9569	0.9491	0.9874	0.9948	**0.9965**	0.9936	0.9948	0.9844	0.9807	0.9912
Amazon	Ma.-F1	40%	0.9618	0.9596	0.9636	0.9657	0.9713	0.9875	0.9814	0.9822	0.9732	0.9958	**0.9969**
		60%	0.9645	0.9601	0.9623	0.9703	0.9759	0.9896	0.9871	0.9814	0.9766	0.9963	**0.9972**
		80%	0.9703	0.9652	0.9679	0.9721	0.9832	**0.9982**	0.9898	0.9853	0.9801	0.9967	0.9978
	Mi.-F1	40%	0.9634	0.9651	0.9658	0.9634	0.9689	0.9759	0.9866	0.9845	0.9699	0.9864	**0.9870**
		60%	0.9691	0.9672	0.9649	0.9678	0.9702	0.9819	0.9856	0.9878	0.9732	0.9961	**0.9892**
		80%	0.9734	0.9721	0.9714	0.9701	0.9734	**0.9974**	0.9872	0.9882	0.9764	0.9931	0.9939

6.6.2 Results

Figure 4 reports the results of node recommendation on four datasets. We can observe that our proposed RHINE and RHINE-M continuously perform better than the compared methods in terms of Hit@k. We believe that the significant improvement is due to that our models preserve the higher order proximities in an HIN. What's more, the proposed RHINE and RHINE-M take the distinctive structural characteristics and semantic information into consideration, which effectively guarantees the embedding accuracy of various types of nodes connected via heterogeneous relations.

6.7 Comparison of Variant Models

In order to verify the effectiveness of distinguishing the structural characteristics of relations, we design three variant models based on RHINE as follows:

- **RHINE**$_{Eu}$ which does not distinguish multiple relations in HINs, and only leverages Euclidean distance to embed heterogeneous information networks.
- **RHINE**$_{Tr}$ models all nodes and relations in HINs with translation-based distance, regardless of the different categories of relations, which is just like TransE [47].
- **RHINE**$_{Re}$ leverages Euclidean distance to model Interaction Relations while translation-based distance for Affiliation Relations.

We set the parameters of variant models as the same as those of our proposed model RHINE. The results of the three tasks are shown in Figure 5. It is evident that our model outperforms RHINE$_{Eu}$ and RHINE$_{Tr}$, indicating that it is beneficial for learning the representations of nodes by distinguishing the heterogeneous relations. Besides, we find that RHINE$_{Tr}$ achieves better performance than RHINE$_{Eu}$.
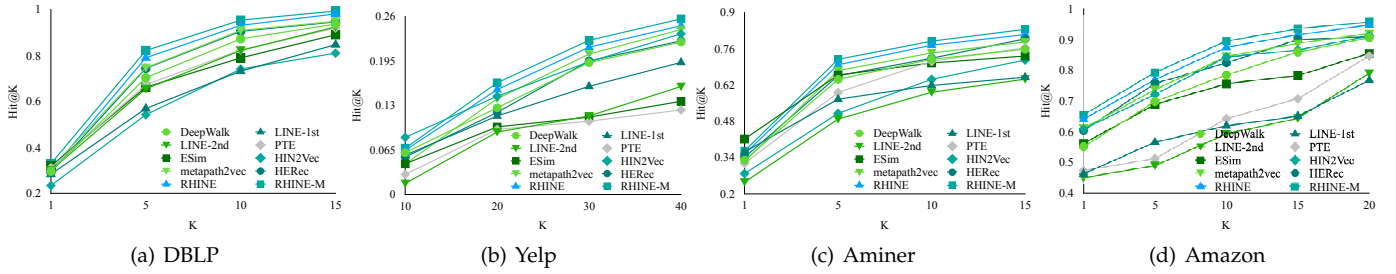
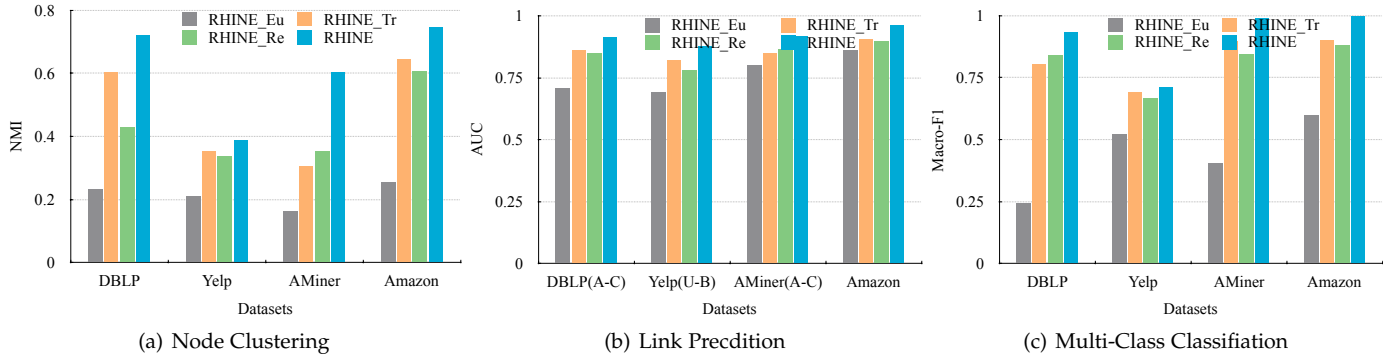Fig. 4. Performance Evaluation of Node Recommendation.

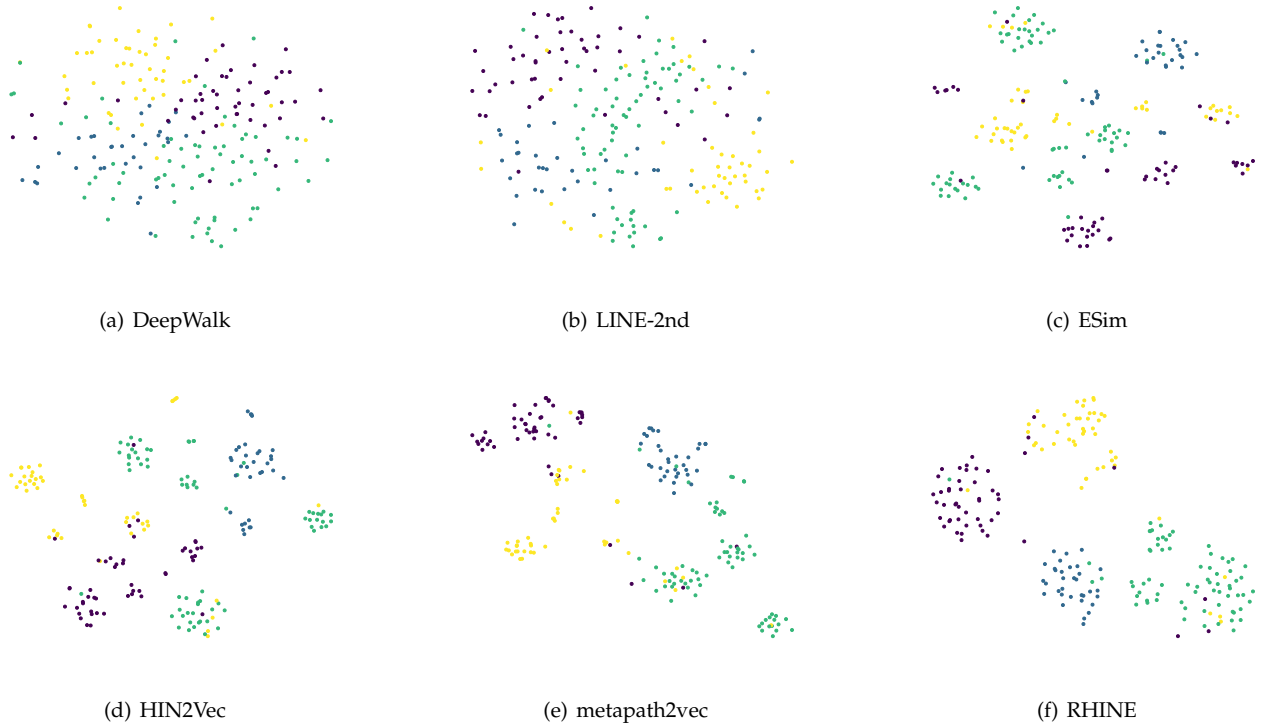
Fig. 5. Performance Evaluation of Variant Models.

Fig. 6. Visualization of Node Embeddings.

This is due to the fact that there are generally more peer-to-peer relationships (i.e., IRs) in the networks. Directly making all nodes close to each other leads to much loss of information. Compared with the reverse model RHINE$_{Re}$, RHINE also achieves better performance on all tasks, which

implies that two models for ARs and IRs are well designed to capture their distinctive characteristics.

(a) NMI w.r.t. #Dimensions   (b) NMI w.r.t. #Negative samples
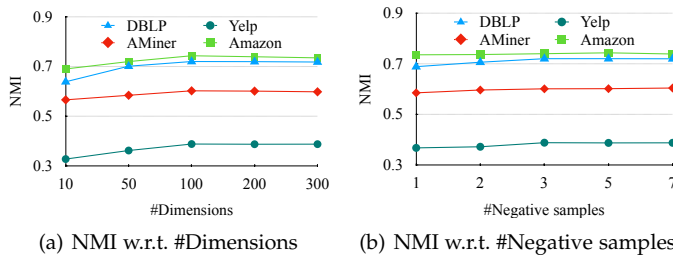
Fig. 7. Parameter Analysis.

6.8 Visualization

In order to understand the representation of the networks intuitively, we visualize the vectors of nodes (.e., papers) in DBLP. To be specific, we utilize t-SNE [57] to layout a vector in a 2-dimensional space, and report the results of DeepWalk, metapath2vec and our RHINE in Figure 6.

It is evident that our model RHINE can separate nodes in different datasets clearly with obvious borders. As we can see, papers in the latent space are geographically clustered into four groups, which is consistent with four domains of papers. Moreover, each group is well separated from others, which demonstrates that our model learns superior node embeddings by distinguishing the heterogeneous relations in HINs. In contrast, DeepWalk and LINE barely split papers into different groups. Since ESim, HIN2Vec and Metapath2vec are designed for heterogeneous information network, they perform better than DeepWalk and LINE, but the boundary is blurry.

6.9 Parameter Analysis

In order to evaluate the influence of different parameters in our model, we investigate the sensitivity of them on node clustering task. Specifically, we investigate the sensitivity of two parameters, including the number of negative samples and the number of embedding dimension. We vary the number of embedding dimensions as 10, 50, 100, 200 and 300, and the number of negative samples as 1, 2, 3, 5 and 7. The results of node clustering are reported in Figure 7.

As shown in Figure 7(a), the performance of our model improves with the increase of the number of embedding dimensions, and then tends to be stable once the dimension of the embeddings reaches around 100. It is evident that our model are capable to capture rich information of various relations in HINs using a low-dimensional representation. Similarly, Figure 7(b) shows that as the number of negative examples increases, the performance of our model first grows and then becomes stable when the number reaches 3. Overall, the change trend is smooth, indicating that the proposed model is not very sensitive to the two parameters.

7 CONCLUSION

In this paper, we make the first attempt to explore and distinguish the structural characteristics of relations for HIN embedding. We present two structure-related measures which can consistently distinguish heterogeneous relations into two categories: Affiliation Relations and Interaction Relations. To respect the distinctive structures of relations,

we propose a novel relation structure-aware HIN embedding model (RHINE), which individually handles these two categories of relations. Considering the heterogeneity of nodes and relations, we further introduce relation-specific projection matrices to learn node and relation embeddings in separate spaces (i.e., RHINE-M), rather than a common space. Experimental results demonstrate that RHINE and RHINE-M outperform state-of-the-art baselines in various tasks. In the future, we will explore other possible measures to differentiate relations so that we can better capture the structural information of HINs. In addition, we will exploit deep neural network based models for different relations.

REFERENCES

[1] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," *arXiv preprint arXiv:1801.05852*, 2017.

[2] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[3] H. Cai, V. W. Zheng, and K. Chang, "A comprehensive survey of graph embedding: problems, techniques and applications," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of SIGKDD*. ACM, 2014, pp. 701–710.

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of WWW*, 2015, pp. 1067–1077.

[6] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of CIKM*. ACM, 2015, pp. 891–900.

[7] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proceedings of SIGKDD*. ACM, 2017, pp. 385–394.

[8] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of SIGKDD*. ACM, 2016, pp. 1105–1114.

[9] D. Zhu, P. Cui, Z. Zhang, J. Pei, and W. Zhu, "High-order proximity preserved embedding for dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 11, pp. 2134–2144, 2018.

[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of SIGKDD*. ACM, 2016, pp. 855–864.

[11] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of SIGKDD*. ACM, 2016, pp. 1225–1234.

[12] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of AAAI*, 2016, pp. 1145–1152.

[13] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *Proceedings of WSDM*, 2018, pp. 459–467.

[14] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of VLDB*, vol. 4, no. 11, pp. 992–1003, 2011.

[15] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[16] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *Acm Sigkdd Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2982898, IEEE Transactions on Knowledge and Data Engineering

14

[17] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *Proceedings of SIGKDD*. ACM, 2015, pp. 1165–1174.

[18] Y. Jacob, L. Denoyer, and P. Gallinari, "Learning latent representations of nodes for classifying in heterogeneous social networks," in *Proceedings of WSDM*. ACM, 2014, pp. 373–382.

[19] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proceedings of SIGKDD*. ACM, 2015, pp. 119–128.

[20] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "Shine: signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of WSDM*. ACM, 2018, pp. 592–600.

[21] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proceedings of CIKM*. ACM, 2017, pp. 1797–1806.

[22] C. Shi, B. Hu, X. Zhao, and P. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[23] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, "Meta-path guided embedding for similarity search in large-scale heterogeneous information networks," *arXiv preprint arXiv:1610.09769*, 2016.

[24] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of SIGKDD*. ACM, 2017, pp. 135–144.

[25] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks," in *Proceedings of WSDM*. ACM, 2017, pp. 741–749.

[26] J. Zhang, C. Xia, C. Zhang, L. Cui, Y. Fu, and S. Y. Philip, "Bl-mne: Emerging heterogeneous social network embedding through broad learning with aligned autoencoder," in *Proceedings of ICDM*. IEEE, 2017, pp. 605–614.

[27] X. Han, C. Shi, S. Wang, S. Y. Philip, and L. Song, "Aspect-level deep collaborative filtering via heterogeneous information networks." in *Proceedings of IJCAI*, 2018, pp. 3393–3399.

[28] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: projected metric embedding on heterogeneous networks for link prediction," in *Proceeding of SIGKDD*, 2018, pp. 1177–1186.

[29] K. Faust, "Centrality in affiliation networks," *Social networks*, vol. 19, no. 2, pp. 157–191, 1997.

[30] J. Yang and J. Leskovec, "Community-affiliation graph model for overlapping network community detection," in *Proceedings of ICDM*. IEEE, 2012, pp. 1170–1175.

[31] Y. Lu, C. Shi, L. Hu, and Z. Liu, "Relation structure-aware heterogeneous information network embedding," in *Proceedings of AAAI*, 2019, pp. 4456–4463.

[32] M. Balasubramanian and E. L. Schwartz, "The isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, pp. 7–7, 2002.

[33] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proceedings of NIPS*, 2002, pp. 585–591.

[34] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *Proceedings of WWW*, 2013, pp. 37–48.

[35] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information." in *Proceedings of IJCAI*, 2015, pp. 2111–2117.

[36] C. Tu, W. Zhang, Z. Liu, and M. Sun, "Max-margin deepwalk: Discriminative learning of network representation." in *Proceedings of IJCAI*, 2016, pp. 3889–3895.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of NIPS*, 2013, pp. 3111–3119.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[39] Y. Fan, S. Hou, Y. Zhang, Y. Ye, and M. Abdulhayoglu, "Gotcha - sly malware!: Scorpion A metagraph2vec based malware detection system," in *Proceedings of SIGKDD*, 2018, pp. 253–262.

[40] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of SIGKDD*. ACM, 2008, pp. 990–998.

[41] Y. Sun and J. Han, "Mining heterogeneous information networks: principles and methodologies," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 1–159, 2012.

[42] X. Li, Y. Wu, M. Ester, B. Kao, X. Wang, and Y. Zheng, "Semi-supervised clustering in attributed heterogeneous information networks," in *Proceedings of WWW*, 2017, pp. 1621–1629.

[43] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 7, no. 3, p. 11, 2013.

[44] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[45] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227–248, 1980.

[46] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proceedings of WWW*, 2017, pp. 193–201.

[47] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of NIPS*, 2013, pp. 2787–2795.

[48] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion." in *Proceedings of AAAI*, vol. 15, 2015, pp. 2181–2187.

[49] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of AAAI*, 2014, pp. 1112–1119.

[50] R. Hussein, D. Yang, and P. Cudré-Mauroux, "Are meta-paths necessary?: Revisiting heterogeneous graph embeddings," in *Proceedings of CIKM*. ACM, 2018, pp. 437–446.

[51] C. Wang, Y. Song, A. El-Kishky, D. Roth, M. Zhang, and J. Han, "Incorporating world knowledge to document clustering via heterogeneous information networks," in *Proceedings of SIGKDD*, 2015, pp. 1215–1224.

[52] Y. Shi, X. He, N. Zhang, C. Yang, and J. Han, "Higher-order clustering in heterogeneous information networks," *CoRR*, vol. abs/1811.11320, 2018.

[53] C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, "Hetesim: A general framework for relevance measure in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2479–2492, 2014.

[54] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *Proceedings of ASONAM*. IEEE, 2011, pp. 121–128.

[55] X. Kong, B. Cao, and P. S. Yu, "Multi-label classification by mining label and instance correlations from heterogeneous information networks," in *Proceeding of SIGKDD*, 2013, pp. 614–622.

[56] Y. Zhang, Y. Xiong, X. Kong, S. Li, J. Mi, and Y. Zhu, "Deep collective classification in heterogeneous information networks," in *Proceedings of WWW*, 2018, pp. 399–408.

[57] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

Chuan Shi received the B.S. degree from the Jilin University in 2001, the M.S. degree from the Wuhan University in 2004, and Ph.D. degree from the ICT of Chinese Academic of Sciences in 2007. He joined the Beijing University of Posts and Telecommunications as a lecturer in 2007, and is a professor and deputy director of Beijing Key Lab of Intelligent Telecommunications Software and Multimedia at present. His research interests are in data mining, machine learning, and evolutionary computing. He has published more than 40 papers in refereed journals and conferences.

Yuanfu Lu received the B.S. degree from Nanjing University of Posts and Telecommunications in 2017. He is currently a master student in Beijing University of Posts and Communications. His research interests are in data mining and machine learning.

Zhiyuan Liu is an assistant researcher of the Department of Computer Science and Technology, Tsinghua University, Beijing. He got his B.E. degree and Ph.D. degree in computer science from Tsinghua University, Beijing, in 2006 and 2011 respectively. His research interests are natural language processing and social computation. He has published over 40 papers in international journals and conferences including ACM Transactions, IJCAI, AAAI, ACL and EMNLP. He obtained Excellent Doctoral Dissertation awarded by Chinese Association for Artificial Intelligence in 2012, and Excellent Post-doctoral Fellow Award at Tsinghua University in 2013.

Linmei Hu is an assistant professor in School of Computer Sciences, Beijing University of Posts and Communications. She received her Ph.D. degree from Tsinghua University in 2018. Her research interests focus on natural language processing and data mining. She was awarded Beijing Excellent Ph.D. Student in 2018.

Huadong Ma (SM'16) received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Science, in 1995. He is currently a Professor and the Director of the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, and an Executive Dean of the Institute of Network Technology, Beijing University of Posts and Telecommunications, China. His current research focuses on multimedia system and networking, Internet of Things, and sensor networks. He has authored or co-authored over 100 papers and four books on his fields. He is a member of the ACM.