Ruan CY, Wang Y, Ma J *et al.* Adversarial heterogeneous network embedding with metapath attention mechanism. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(6): 1217–1229 Nov. 2019. DOI 10.1007/s11390-019-1971-3

Adversarial Heterogeneous Network Embedding with Metapath Attention Mechanism

Chun-Yang Ruan[1,2], Ye Wang[3], Jiangang Ma[4], Yanchun Zhang[1,2,5], and Xin-Tian Chen[1,2]

[1]*Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai 200433, China*
[2]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[3]*College of Computer Science, National University of Defense Technology, Changsha 410073, China*
[4]*School of Science, Engineering and Information Technology, Federation University Australia, Melbourne 3000, Australia*
[5]*College of Engineering and Science, Victoria University, Melbourne 3000, Australia*

E-mail: cyruan16@fudan.edu.cn; ye.wang10@live.vu.edu.au; j.ma@federation.edu.au
{yanchunzhang, 16212010003}@fudan.edu.cn

Received January 23, 2019; revised September 24, 2019.

Abstract Heterogeneous information network (HIN)-structured data provide an effective model for practical purposes in real world. Network embedding is fundamental for supporting the network-based analysis and prediction tasks. Methods of network embedding that are currently popular normally fail to effectively preserve the semantics of HIN. In this study, we propose AGA2Vec, a generative adversarial model for HIN embedding that uses attention mechanisms and meta-paths. To capture the semantic information from multi-typed entities and relations in HIN, we develop a weighted meta-path strategy to preserve the proximity of HIN. We then use an autoencoder and a generative adversarial model to obtain robust representations of HIN. The results of experiments on several real-world datasets show that the proposed approach outperforms state-of-the-art approaches for HIN embedding.

Keywords heterogeneous information network, network embedding, attention mechanism, generative adversarial network

1 Introduction

Network embedding focuses on the learning of a low-dimensional representation that reflects the core information of a network. It is used in several network-based applications, such as visualization, node classification, and link prediction and recommendation[1]. Current methods of network embedding often incorporate structural information to embed homogeneous networks. However, large-scale heterogeneous information network (HIN), including social network, academic network and biomedical network, is becoming ubiquitous in the real world. The various types of nodes and edges carry rich semantics other than basic structural information. Prevalent methods of network embedding are limited to learning the representation of HIN because they cannot preserve network semantics. HIN-based applications often use meta-paths to overcome the lack of semantics[2]. Meta-path[3] is a recently proposed proximity model in HIN that consists of different paths between nodes. Inspired by this, several methods of HIN embedding have been proposed. They use meta-paths to preserve semantic and structural information in the process of HIN embedding[3−6].

Although the above approaches to HIN embedding perform well on a number of tasks, they suffer from the following drawbacks.

1) *Insensitivity to Semantics of Meta-Paths.* To obtain stable and robust HIN embedding, the semantics of multiple meta-paths must be integrated[1,5,7]. During this integration, as the significance of various meta-paths can be relatively different, their weights need to be carefully decided. For example, if we consider the collaboration between two authors of an article, they

---

Regular Paper
This work was supported by the National Natural Science Foundation of China under Grant No. 61672161, and the Youth Research Fund of Shanghai Municipal Health and Family Planning Commission of China under Grant No. 2015Y0195.

may be very tightly tied to their paper and loosely tied to the conferences at which they present. However, considering the general relationship between the authors, they may have relatively stronger ties to the conferences in which both have presented. A question then arises: which types of relations should we use to capture the proximity between two authors? Different meta-paths lead to different author-connection networks, which may lead to different proximity for embedding results. Prevalent methods usually treat different meta-paths equally. This is not reasonable for most large-scale heterogeneous information networks.

2) *Lack of Additional Constraints for HIN Embedding.* Recent work on HIN embedding is effective in semantic and structural preservation by using different well-designed objectives. However, it suffers from inadequate constraints on the distribution of embedding[8]. Considering the method in [9] as an example, it calculates the truncated proximity of HIN by using meta-paths, and obtains the prior joint probability $\widehat{p}(v_i, v_i)$ between nodes v_i and v_j. To learn the low-dimensional representation u_i (or u_j) of node v_i (or v_j), the method simply minimizes the distance between these probability distributions $\widehat{p}(v_i, v_i)$ and $p(u_i, u_j)$. This implies that $p(u_i, u_j)$ can be very irregular, and can lead to difficulty in generating new samples, or can even render this unfeasible.

To overcome these challenges, we propose a novel model based on attention mechanisms[10] and generative adversarial networks[11] to embed HIN into a low-dimensional vector space, called AGA2Vec. Specifically, AGA2Vec contains the following two main embedding mechanisms.

1) *Integrated Semantics for Embedding.* It is the first mechanism that aims to dynamically capture the weights of different meta-paths and to preserve the proximity of HIN via meta-path propagation. HIN-based methods usually achieve user-guided semantics. With user-guidance, a model will be able to learn the most appropriate meta-paths or their weighted combinations[12]. However, current methods of network embedding often use the unsupervised or semi-supervised models, and some studies that add the user-guided semantics into the training are challenging. To solve this problem, we use attention mechanisms to learn the weights of different meta-paths in the training process, which is inspired by the recent progress of the attention mechanism for machine translation[10].

2) *Enhanced Regularization for Embedding.* That is another way to focus on learning the robust representations of nodes via generative adversarial networks (GANs)[11]. Specifically, we design a generator via the semantic-based autoencoder that tries to simulate the potential interrelations of nodes and constructs the representations for these nodes. We use a discriminator to determine whether a sample originates from the real distribution or the low-dimensional representations of the network. Enhanced regularization effectively reduces the amount of information that may be lost in the decoding, enabling the model to learn a stable and robust representation of the data. HIN embedding is capable of capturing both local and global semantic information in the embedding vectors. Note that each dimension of the embedding vectors is a distribution over entities, and is able to preserve the user-guided semantics. Overall, to incorporate the advantage of attention-based weight learning and adversarial regularization, we propose AGA2Vec that makes good use of the virtues of multiple meta-paths that can describe the semantic relationship and extract feature information between entities to capture semantic and structural information in vector spaces. We compare AGA2Vec with several state-of-the-art approaches on both empirical and synthetic datasets, and obtain improvements that demonstrate the usefulness of our approach for HIN embedding. AGA2Vec incorporates the attention mechanisms and generative adversarial learning simultaneously in the studies of HIN embedding. The main contributions of our paper can be summarized as follows.

- We show that the weights of meta-paths in an HIN can influence the performance of HIN embedding, and propose an attention-based approach to learn the weights in the embedding process. It is flexible for weighted meta-path proximity.
- We leverage adversarial mechanisms to impose a prior distribution on the embedding space, and to learn stable and robust HIN embedding.
- We extensively prove our model through various heterogeneous network mining tasks on three datasets and a case study. The results show the effectiveness and robust improvements in comparison with other state-of-the-art methods.

2 Related Work

2.1 HIN Embedding

Several methods of network embedding have been recently developed from different perspectives[13]. However, many such studies have focused only on learning node vectors in homogeneous information networks[14−17]. Moreover, while all these researchers claimed that their approaches can capture the embedded structures of information networks, those models tend to consider only aggregated information among nodes or limited types of relations[5]. For example, GraphGAN[14] captures the nearby neighborhood of each node by breadth-first search (BFS). BHONEM[18] sets a binary higher-order method based on domain knowledge and designs an objective function by neural networks.

Although general network embedding methods can be applied by treating every node in the networks as the same type, developing more dedicated methods for modelling the heterogeneous types of nodes and relations in a unified manner is still an interesting and challenging problem[4]. Through several studies[1,4−6,19,20] on HIN embedding have been proposed, some only focus on the limited-types of meta-paths of nodes. For example, SHINE[6] and HNE[19,21] learn the representations of nodes by capturing one-hop neighborhood relations between nodes. Some models[4,5,7,22] tend to preserve the different meta-paths between nodes, but ignore the weights of different meta-paths. Only HINE[1] tries to normalize weights of multi-typed meta-paths and captures different semantics between nodes. However, it relies heavily on user-guidance to determine a user-given meta-path set and the frequency of each meta-path for embedding. Moreover, some parts of its objective function include calculating the frequency normalized weight, and this negatively affects the time complexity. The attention mechanism provides a new approach to solving the weight problem, which focuses on the most pertinent information for global information. Recent work[23,24] uses node- or semantic-level attention to embed the HIN. However, [23, 24] need to embed each meta-path before attention and the specific task. Meta-graph[25] is currently the most powerful approach to measuring the proximity in HIN. [26] advocates a metagraph concept to capture richer structural contexts and semantics between distant nodes, and uses metagraph to guide the generation of random walks and to learn latent embeddings of multi-typed HIN nodes. [27] combines the meta-graph and the meta-path to obtain node embedding, but they ignore the complexity of matrix factorization in HIN[1]. However, current relevance work based on meta-graph only considers the complex structural information and takes the problem of computation complexity. In the HIN relevance tasks, we select the meta-path or meta-graph as a tool according to our demand analysis.

2.2 Generative Adversarial Networks

Recent advances in generative adversarial networks (GANs)[11] have shown that they are a powerful framework to learn complex data distributions. The core idea can be formulated as a minimax-game, in which the generator aims to match data samples from some prior distributions to the data space, while the discriminator is dedicated to distinguishing fake samples from the real data[8]. GANs have been successfully applied to computer vision, such as in image classification[28] and image generation[29,30]. However, few attempts have been made to apply GANs to representation learning owing to a lack of explicit structure for inference. By projecting samples in the original data space back into the space of latent features, EBGAN, BiGAN and ALI can learn robust representations in many applications, such as image classification and document retrieval[8]. DCGANs can learn expressive image representations from both the generator and the discriminator networks based on convolutional layers[31]. Some models use the adversarial learning process to regularize the representations. One successful practice is the use of the GraphGAN[14] and ANE[8], both of which can learn powerful representations from network data without any supervision. However, these methods are not directly suitable for learning HIN representations because of a lack of semantic preservation.

3 Problem Definition and Notations

In this section, we formally introduce the preliminary concepts, and define the problem of HIN embedding.

Definition 1. *An HIN is defined as a typed graph $G = (V, E, \phi, \psi, \omega)$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges in V, $\phi : V \rightarrow O$ and $\psi : E \rightarrow R$ are the type-mapping functions for nodes and edges, respectively, and ω is the set of weights in E. Each object $v \in V$ belongs to a node type $\phi(v) \in O$, and each edge $e \in E$ belongs to an edge type $\psi(e) \in R$. When $|O| + |R| > 1$, the network is called an HIN; otherwise, it is a homogeneous information network. Fig.*1 *illustrates a toy HIN.*

Definition 2. *In an HIN, two nodes can be interrelated via different semantic paths called meta-paths*[12]*, which are defined as follows. Meta-path ρ is a sequence of node types $a_1, a_2, \cdots, a_n$ and/or edge types $r_1, r_2, \cdots, r_{n-1}$, $\rho = a_1 \xrightarrow{r_1} a_2 \xrightarrow{r_2} \cdots a_i \xrightarrow{r_i} \cdots \xrightarrow{r_{n-1}}$*

a_n. *When* $\forall i = 1, \cdots i, \cdots, n, a_i = \phi(v_i)$ *and* $r_i = \psi(v_i, v_{i+1})$, *an instance of the meta-path is a path* l *that passes through nodes* $v_1, v_2, \cdots, v_n$.

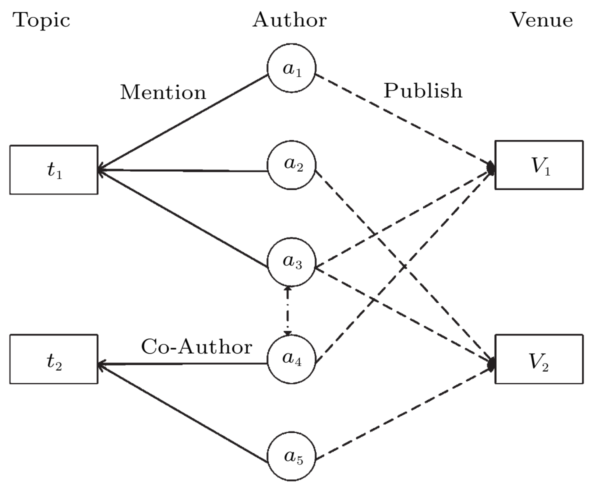

Fig.1. Toy heterogeneous cite network schema with three types of objects: topics, authors, and venues.

Example 1. Owing to the heterogeneity in an HIN, the meta-path consists of various edge types that may not fully align with one another. Different meta-paths lead to different author relation graphs, which in turn may lead to different underlying semantics. In Fig.2(a), authors are connected through topics and form two facets: a_1, a_2, a_3 and a_4, a_5; in Fig.2(b), authors are connected through venues and form two different facets: a_1, a_3, a_4 and a_2, a_3, a_5. By contrast, in Fig.2(c), a connection graph combining both meta-paths generates one semantic: a_2, a_3.

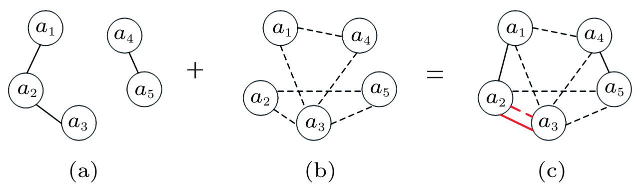

Fig.2. Example author-relations networks under different meta-paths. (a) ATA. (b) AVA. (c) ATA + AVA.

Definition 3. *A meta-path based proximity matrix captures various transition probabilities in an HIN. Given an example of meta-path* ρ, $\widehat{\boldsymbol{A}} \in \mathbb{R}_{\rho}^{|V|\times|V|}$ *is an adjacency matrix of HIN used to preserve the first-order proximity*[16] *of the network. We can define the transition matrix* $\bar{\boldsymbol{A}} \in \mathbb{R}_{\rho}^{|V|\times|V|}$, *where each element* $\bar{A}_{i,j}$ *is the probability of a transition from node* v_i *to node* v_j *within one or* t *steps in the meta-path* ρ. $\bar{\boldsymbol{A}}$ *based on different meta-paths can preserve* k*-order proximity between nodes* v_i *and* v_j.

Example 2. For the toy HIN in Fig.1, we choose a meta-path "ATA" and nodes "a_1", "a_2", "a_3" and "t_1" as an example. The "ATA"-based matrices $\widehat{\boldsymbol{A}}$ and $\bar{\boldsymbol{A}}$ are formulated as follows.

$$\widehat{\boldsymbol{A}}_{\mathrm{ATA}} = \begin{array}{c c} & \begin{array}{cccc} a_1 & a_2 & a_3 & t_1 \end{array} \\ \begin{array}{c} a_1 \\ a_2 \\ a_3 \\ t_1 \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array},$$

$$\bar{\boldsymbol{A}}_{\mathrm{ATA}} = \begin{array}{c c} & \begin{array}{cccc} a_1 & a_2 & a_3 & t_1 \end{array} \\ \begin{array}{c} a_1 \\ a_2 \\ a_3 \\ t_1 \end{array} & \begin{pmatrix} 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 0 \\ 1/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix} \end{array}.$$

Definition 4. *Given HIN* G *and a meta-path* ρ, *HIN embeddding aims to learn a function* $f_\rho : V \rightarrow \mathbb{R}^d$ *that projects each node* $v \in V$ *to a low-dimensional space, where the dimension* $d \ll |V|$. *The problem of embedding learning from all meta-paths in HIN involves learning of the corresponding function* $f = \oplus_{|\rho|}\pi_\rho f_\rho$. *Note that for different node types, the corresponding* f_ρ *might have different dimensions by definition. The determination of quality weights* π *for meta-paths is crucial for a particular HIN embedding.*

4 Proposed Model

To address the problem of embedding learning from meta-paths in HIN, we propose a flexible framework to distinguish the semantic information regarding each meta-path. In this section, we first provide a brief overview of the proposed AGA2Vec, and then formulate our method of HIN embedding using an attention-based generative adversarial model.

4.1 Overview of Framework

As shown in Fig.3, AGA2Vec consists of three components: an attention-based meta-path decider, an autoencoder-based generator, and an adversarial learning component. Let vector $\boldsymbol{X}_{ik}$ be a column in each $\bar{\boldsymbol{A}}_k$ for $k = 1, 2, \cdots, K$, which denotes the k-th meta-path based semantic and the structural information for node v_i. The attention-based meta-path decider attempts to preserve the full information. The vector $\boldsymbol{X}_i$ is the original representation of the node v_i, which consists of all $\boldsymbol{X}_{ik}$. We place $\boldsymbol{X}_i$ into an autoencoder, a standard autoencoder consisting of encoder and decoder layers. The encoder projects node $\boldsymbol{X}_i$ into a latent vector $\boldsymbol{Y}_i$, and the decoder reconstructs the data points of each

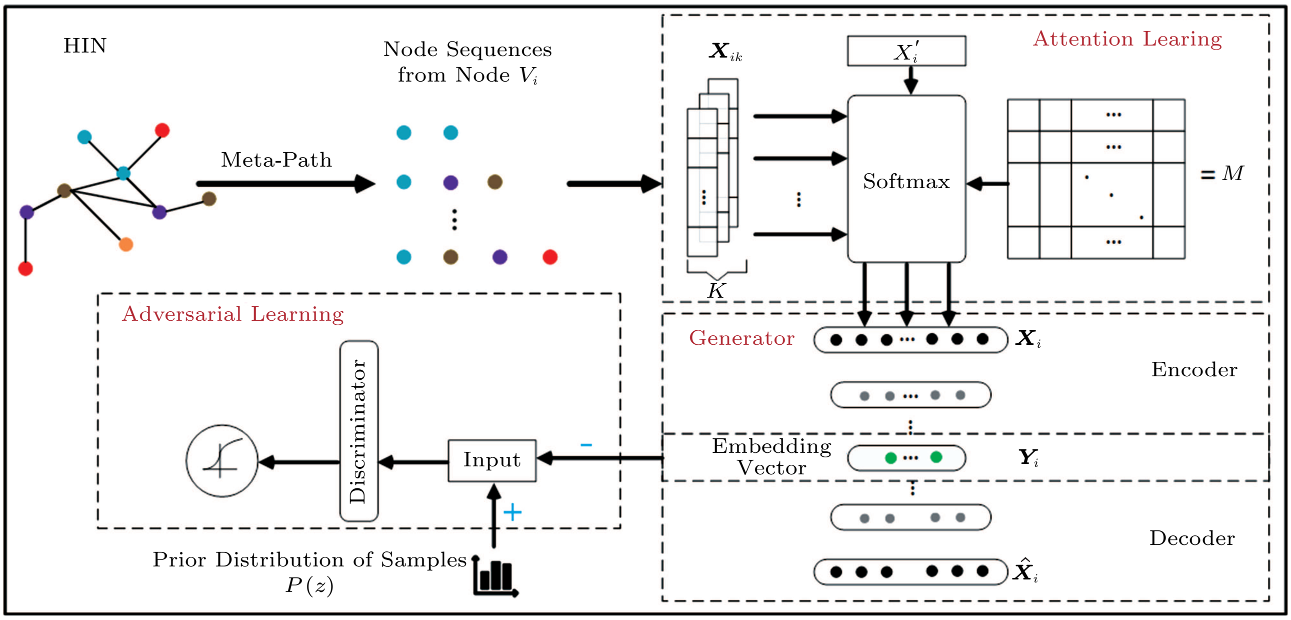

Fig.3. Architecture and dataflow of AGA2Vec. AGA2Vec includes an attention-based decider, an autoencoder-based generator and a discriminator. Nodes marked in red, orange, purple, blue and brown represent the different types of nodes. Green dots represent the embedding vector.

input as a vector $\widehat{\boldsymbol{X}}_i$. Then, the adversarial learning component regularizes the generated $\boldsymbol{Y}_i$ by matching the aggregated posterior, which helps enhance the robustness of representation $\boldsymbol{Y}_i$.

4.2 Attention-Based Meta-Path Decider

To integrate meta-path information objectively and to preserve the semantics between nodes, we learn the quality weight for each meta-path by using the attention mechanism and the matrix multiplication. For node v_i, $\boldsymbol{X}_i$ is defined as the weighted summation of every vector $\boldsymbol{X}_{ik}$, $k = 1, 2, \cdots, K$, corresponding to the meta-path index for each node. $\boldsymbol{X}_i$ is formulated as follows:

$$\boldsymbol{X}_i = \sum_{k=1}^{K} \lambda_{ik} \cdot \boldsymbol{X}_{ik}.$$

For each original vector $\boldsymbol{X}_{ik}$ of node v_i, we compute a quality weight λ_{ik} that can be interpreted as the probability that $\boldsymbol{X}_{ik}$ is assigned by a node. Intuitively, with the learned proper weights λ_{ik} for each node, AGA2Vec can obtain the most informative meta-paths. Following recently proposed attention-based models for neural machine translation[10], we define the weight of the k-th meta-path for a node using the softmax function as follows:

$$\lambda_{ik} = \frac{e^{\theta_{ik}}}{\sum_{j=1}^{|V|} e^{\theta_{jk}}}, \quad \theta_{ik} = \boldsymbol{X}_{ik}^{\mathrm{T}} \odot \boldsymbol{M} \odot X_i',$$

$$X_i' = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{X}_{ik},$$

where X_i' is the average value of different meta-path based transition vectors for node v_i that can preserve the global semantic and the structural information. $\boldsymbol{M} \in \mathbb{R}^{|V|\times|V|}$ is the matrix used to match the relevance between global information X_i' and each meta-path-based transition vector $\boldsymbol{X}_{ik}$ of node v_i, which is learned as part of the training process. If $\boldsymbol{X}_{ik}$ and X_i' have a large dot product, node v_i believes that meta-path k is an informative link, and the weight of meta-path k for node v_i is large.

4.3 Autoencoder-Based Generator

Having obtained the robust node original representation $\boldsymbol{X}_i \in \mathbb{R}^{|V|}$, we adopt a deep autoencoder to learn a low-dimensional node embedding. An autoencoder as a generator performs an encoding step, followed by a decoding step. In the former, a function $f(\cdot)$ is applied to the original representation $\boldsymbol{X}_i$ in the input space and $f(\cdot)$ sents $\boldsymbol{X}_i$ to a new feature space. An activation function is typically used in this process to model non-linearities between the vector spaces. In the decoding step, a reconstruction function $g(\cdot)$ is used to reconstruct the original input vectors back from the $\mathbb{R}^d$ space. $\widehat{\boldsymbol{X}}_i$ is the reconstructed vector representation. Following training, representations of the bottleneck layer $\boldsymbol{Y}_i$ can be considered as a low-dimensional embedding of the input node v_i.

This autoencoder is trained to minimize the reconstruction error. Therefore, we reduce the distance between vectors $\boldsymbol{X}_i$ and $\widehat{\boldsymbol{X}_i}$. However, there are a limited number of links in the real-world networks, which can lead to network sparsity[32]. As a result, the number of zero elements in the adjacency matrix $\widehat{\boldsymbol{A}}$ is much greater than that of the non-zero elements for an input node v_i, which may degrade the reconstruction performance. Following the work in [32], we choose Binary Cross Entropy with a value parameter as the objective function to render the reconstructed embedding $\widehat{\boldsymbol{X}_i}$ similar to the target node embedding $\boldsymbol{X}_i$. Then, the unregularized objective function of each instance v_i is formulated as follows:

$$\begin{aligned} &J_i \\ &= -1/|V| \sum_{j}^{|V|} \zeta_{ij}(x_{ij} \log(\widehat{x}_{ij}) + (1 - x_{ij}) \log(1 - \widehat{x}_{ij})), \end{aligned}$$

where the value parameter ζ_{ij} is used to calculate the weight coefficient of the penalty imposed on each element in $\boldsymbol{A}$. If $A_{ij} \neq 0$, $\zeta_{ij} = \omega_{ij} \geqslant 1$; otherwise, $\zeta_{ij} = 0$. To accumulate all nodes, the objective function can be defined as follows:

$$J = \sum_{i}^{|V|} J_i. \tag{1}$$

With this objective function (1), global HIN structures are preserved by latent representation. However, this process is insufficient to embed the network by reconstructing global structures only, as it ignores useful information of the network local structures[33]. Given a pair of nodes v_i, v_j connected by a specific edge, these nodes have a higher similarity. Intuitively, the similarity between v_i and v_j in $\mathbb{R}^d$, which denotes the local structures, is strongly constrained by the information of the edge. We borrow the idea of Laplacian Eigenmaps to capture the local HIN structures. Then, (1) can be rephrased as follows:

$$J(\theta) = \sum_{i}^{|V|} J(\theta)_i + \sum_{i,j}^{|V|} \zeta_{ij} \parallel \widehat{\boldsymbol{X}_i} - \boldsymbol{X}_i \parallel_2^2 . \tag{2}$$

The standard autoencoder is a powerful tool with multiple non-linear functions to learn data embedding. The unsupervised autoencoder, however, is sensitive to noisy network data that result in very different codes for similar inputs because of the lack of constraint for embedding distribution.

4.4 Adversarial Learning

The adversarial learning component is employed to regularize the naive representations from the generator. An autoencoder comprises an encoder and a decoder, both of which have their own set of learnable parameters. The encoder is used to obtain a latent code $\boldsymbol{Y}_i$ from the input $\boldsymbol{X}_i$, where the number of dimensions of the latent code should be smaller than that of the input. The decoder takes in this latent code and reconstructs the original input. Adversarial learning is employed to solve the problem of instability of the autoencoder by imposing an adversarial regularization on representation of the bottleneck layer of it. And the distribution of the latent code may then be shaped to match a desired prior distribution. Adversarial learning can reduce the amount of information that may be held in the encoding process, forcing the model to learn a robust representation for HIN. In general, the generative adversarial model consists of a generator $G(\cdot)$ and a discriminator $D(\cdot)$. Our main goal is to force $\boldsymbol{Y}_i$ as the output of the encoder to follow a given prior distribution $p(z)$. We use the encoder as our generator, and the discriminator to ascertain if the samples are from a prior distribution or from the output of the encoder. $G(\cdot)$ and $D(\cdot)$ play the following two-player minimax-game with a score function $V(G, D)$:

$$\begin{aligned} &\min_G \max_D V(G, D) \\ &= \mathbb{E}_p(z)(\log D(\cdot)) + \mathbb{E}_q(z)(\log(1 - D(\cdot))), \end{aligned} \tag{3}$$

where $p(x)$ is the distribution of real data samples, and $q(x)$ is the distributions of the encoded data samples. In our work, the training of the general conventional adversarial method in (3) is unstable. According to the suggestion of improved WGAN[34], we use the Wasserstein distance to define the overall objective function of adversarial learning. The cost can be defined as follows:

$$\begin{aligned} &\min_G \max_D V(G, D) \\ &= \mathbb{E}_{p(y)}(D(Y_i)) - \mathbb{E}_{q(y)}(1 - D(Y_i)), \end{aligned} \tag{4}$$

where $p(y)$ is the distribution of the edges, and $q(y)$ is the distributions of unobserved edges generated by $G(\cdot)$. In detail, $D(Y_i)$ is the evaluation function that calculates the trust score for a given edge $e = v_i, v_j$, and $D(Y_i)$ can yield $e' = Y_i$, $Y_j \in G(Y_i) \sim p_\theta(v_j|v_i)$. A naive discriminator $D(x, y)$ is a function (e.g, cosine similarity) used to minimize the distance between the real probability distribution and that in the embedded space. It can be formulated as follows:

$$D(x, y) = \cos(x, y) = \frac{x \odot y}{|x| \odot |y|},$$

where x represents the original data and $y \in \mathbb{R}^d$ is their latent representation of data. In an HIN, considering the semantics and structures, we define the distance between the prior probability distribution $p(x)$ and the resulting probability distribution $q(x)$ as follows:

$$\begin{aligned} D(\cdot) = &-\sum_{(v_i,v_j)\in E} p_1(v_i, v_j) \log q_1(v_i, v_j) - \\ &\sum_{(v_i,v_j)\in E} p_2(v_i, v_j) \log q_2(v_i, v_j), \end{aligned} \quad (5)$$

where $q_1(v_i, v_j)$ is the transition probability between the nodes v_i and v_j in the embedding space. The prior probability is defined as $p_1(v_i, v_j) = \omega_{ij}$. $q_1(v_i, v_j)$ is defined as $q_1(v_i, v_j) = 1/e^{-\boldsymbol{Y}_i^{\mathrm{T}} Y_j}$. $p_2(v_i, v_j) = \omega_{ij} / \sum_{j=1}^{|V|} \omega_{ij}$, and $q_2(v_i, v_j) = e^{\boldsymbol{Y}_i^{\mathrm{T}} Y_j} / \sum_{s=1}^{|V|} e^{\boldsymbol{Y}_i^{\mathrm{T}} Y_s} \boldsymbol{Y}_i$. Directly optimizing the objective function in (5) is problematic because $q_2(v_i, v_j)$ needs to sum up all nodes of the HIN. To address these problems, we adopt negative sampling to optimize $q_2(v_i, v_j)$. Formally, $\log q_2(v_i, v_j)$ is rewritten as follows:

$$\begin{aligned} &\log q_2(v_i, v_j) \\ &= \log \sigma(\boldsymbol{Y}_i^{\mathrm{T}} Y_j) + \sum_{m=1}^{\boldsymbol{M}} \mathbb{E}_{v_m \sim p_m(v)[\log(-\boldsymbol{Y}_i^{\mathrm{T}} Y_m)]}. \end{aligned}$$

4.5 Model Training

We train our model AGA2Vec in a step-by-step manner. 1) To train the generator, the encoder and the decoder are trained simultaneously to minimize the reconstruction loss as shown in (2). 2) The discriminator $\boldsymbol{D}$ is trained to effectively distinguish the true input data z from the false space $\boldsymbol{Y}_i$, where the data are generated from the sample distribution and by minimizing the loss function in (2). 3) The final step is to force the encoder to trick the discriminator by minimizing another loss function represented by (4). Note that we connect the output of the encoder as the input to the discriminator. To avoid over-fitting in the training procedure, we add the dropout layer to our model.

5 Experiments

In this section, we first introduce the datasets and methods of comparison used in our experiments. We then evaluate the proposed embedding methods on three empirical datasets. We also discuss the experimental results and investigate sensitivity across the hyper-parameters.

5.1 Data and Evaluation Measures

We use the following three publicly available empirical heterogeneous information network datasets.

- *DBLP*①. It is the network used most frequently in the study of HINs. It has four node types: Paper, Author, Conference, Term, and four edge types: authorOf, publishedIn, containsTerm, and cites.
- *BlogCatalog*②. It is the network of blogs, including blogger, blogs, and categories of blogs. There are two relations: blogger-follow-blogger and the blogger interest categories.
- *TRE*③. It is a medicine dataset integrating herbs, symptoms, diseases, and their relations from traditional Chinese medicine texts.

Of these methods, HIN-based methods need to specify the meta-paths to be used. We present these meta-paths in Table 1. Following [35], we only select short meta-paths for the four steps because long meta-paths are likely to introduce noisy semantics. We use different metrics to evaluate the different tasks. For clustering, we use normalized mutual information (NMI)[12] to evaluate the performance. For the link prediction, we use mean average precision (MAP)[32] to evaluate the performance. For the case study and the parameter sensitivity, we use *precision*@k[33] to evaluate performance.

Table 1. Selected Meta-Paths for Three Datasets in Trainning

Dataset	Meta-Paths
DBLP	AP, APA, AVA, APAPA, APVPA, APTPA, VPAPV, VPTPV
BlogCatalog	UU, UG, UGU, GUG
TRE	HS, HD, DS, HSH, HDH, DSD, HSDSH

5.2 Baselines

We consider the following baselines to verify the effectiveness and the robustness of the proposed AGA2Vec:

①https://dblp.uni-trier.de/xml/, Oct. 2019.
②http://socialcomputing.asu.edu/datasets/BlogCatalog3, Oct. 2019.
③https://www.aminer.cn/tcmrelextr, Oct. 2019.

• *Deepwalk*[36]: an approach to network embedding that converts the graph structure to linear sequences through fixed length random walks and learns sequences with skip-gram;

• *LINE*[16]: an approach to network embedding that maintains the first and the second order proximities between the nodes;

• *SemNE*[15]: an order sensitive network embedding method that integrates node order information and annotation data;

• *Adversarial NE*[8]: aiming to capture the structural properties of the network and contributing to learning robust representations by matching the posterior distribution of the latent representations with given priors;

• *HINE*[1]: an HIN embedding approach to learning the semantic representations of nodes via the Markov random field and LDA;

• *HIN2Vec*[5]: a typical meta-path based HIN embedding model that carries out multiple prediction training tasks jointly based on a target set of relations to learn the latent vectors of nodes and meta-paths.

Note that we set the parameters for the baselines to get the best results in experiments.

5.3 Node Clustering

For the node clustering task, we first learn node embedding, and then execute the K-means clustering algorithm based on the embedding. The results of clustering on the DBLP, BlogCatalog and TRE datasets are presented in Table 2. The results show that all the embedding methods performed well on BlogCatalog but comparatively worse on the other two datasets. On all datasets, DeepWalk, LINE and SemNE achieved similar results. Of all methods of homogeneous network embedding (Deepwalk, LINE, SemNE, and Adversarial NE), Adversarial NE significantly improved the performance because of its adversarial learning. On DBLP and TRE, HINE and HIN2Vec yielded similarly better performances. Overall, we can see that AGA2Vec outperformed all the other methods in the task of clustering. AGA2Vec achieved an approximately 35.8% improvement. This indicates that the weight of the meta-path used in embedding makes models easy to overfit on HIN. We also conduct specific comparisons for clustering. Given the clusters[37], "data mining", "machine learning", and "computer vision", we quantitatively calculate the NMI of nodes being clustered for each cluster. As shown in Fig.4(a), our model gene-

Table 2. Performance Evaluation (NMI) of Node Clustering of Different Methods

Dataset	Deepwalk	LINE	SemNE	Adversarial NE	HINE	HIN2Vec	AGA2Vec
DBLP	0.468	0.384	0.475	0.512	0.547	0.563	0.642
BlogCatalog	0.475	0.481	0.482	0.546	0.562	0.585	0.573
TRE	0.482	0.406	0.432	0.532	0.542	0.577	0.649

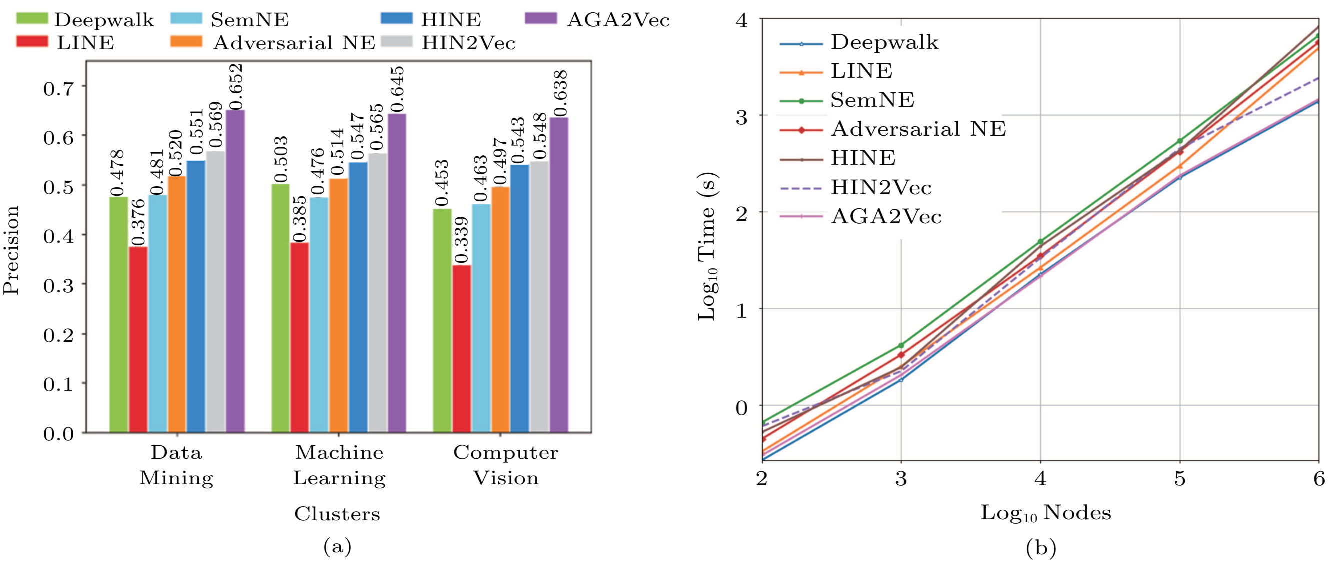

Fig.4. Comparison with baselines for specific clustering on DBLP and scalability testing on the TRE dataset. (a) Specific clustering performance of AGA2Vec. (b) Scalability comparison with average degree of 10.

rated the best performance. This indicates that it can accurately capture the semantics and the structure of nodes in low-dimensional vector spaces.

5.4 Link Prediction

We compare our methods with the baselines for the link prediction task, and report the results in terms of MAP scores. In this task, we want to predict whether a pair of nodes should have had an edge connecting them. Link prediction is useful in many domains, such as in medicine networks where it helps predict drug interaction at the molecular level. We first randomly hid a portion of links from the input network. We used different embedding methods to obtain the low dimensional vector space of the remaining sub-HIN. We chose the hidden edges between node pairs as positive samples. We also randomly sampled an equal number of node pairs not directly connected as negative samples. The similarity score between nodes in a sampled node pair was calculated according to their representation vectors. We used MAP to measure the consistency between the labels and the similarity scores. Table 3 shows the experimental results of the link prediction task. After hiding 10%, 20%, 40%, 50%, and 60% of edges in an orderly manner from the original HIN, our proposed embedding method AGA2Vec outperformed other state-of-the-art baselines by approximately 6.5% in DBLP, 9.9% in BlogCatalog and 5.0% in the TRE dataset, which clearly shows that AGA2Vec is more robust against parse HIN.

5.5 Case Study

The case study involved herbs recommended for given symptoms in TRE. The performance of the case shows that node recommendation can adequately reveal the quality of the learned representations. We provide the qualitative results whereby the top-k herbs are corresponding to the symptom "anemopyretic cold" from over thousands of herbs. We also compare the performance of AGA2Vec and HINE for this recommendation. Table 4 shows the top-12 herbs to the symptom "anemopyretic cold", where the label assigned by the

Table 3. Performance Evaluation (MAP) on Link Prediction of Different Methods

Dataset	Hiding Ratio (%)	Deepwalk	LINE	SemNE	Adversarial NE	HINE	HIN2Vec	AGA2Vec
DBLP	10	0.812	0.801	0.803	0.815	0.832	0.838	0.845
	20	0.807	0.794	0.795	0.804	0.830	0.831	0.841
	40	0.771	0.783	0.790	0.797	0.824	0.829	0.835
	50	0.701	0.712	0.714	0.751	0.769	0.772	0.810
	60	0.612	0.623	0.626	0.701	0.712	0.726	0.783
BlogCatalog	10	0.779	0.785	0.792	0.806	0.823	0.831	0.832
	20	0.772	0.781	0.787	0.799	0.817	0.828	0.829
	40	0.732	0.737	0.729	0.763	0.801	0.804	0.818
	50	0.603	0.597	0.612	0.656	0.722	0.726	0.771
	60	0.573	0.553	0.581	0.620	0.692	0.702	0.753
TRE	10	0.763	0.761	0.765	0.815	0.846	0.850	0.857
	20	0.761	0.757	0.762	0.812	0.843	0.847	0.851
	40	0.741	0.732	0.745	0.748	0.757	0.765	0.779
	50	0.591	0.597	0.603	0.607	0.621	0.626	0.631
	60	0.498	0.501	0.532	0.573	0.594	0.613	0.612

Table 4. Top-k Similar Lists for Symptom "Anemopyretic Cold"

Rank	Herb	Score	Label	Herb	Score	Label
1	Goldenrod	0.10130	1	Mentha haplocalyx	0.3460	1
2	Honeysuckle	0.01640	1	Fructus arctii	0.0670	1
3	Chrysanthemum	0.01320	1	Picria felterrae	0.0662	1
4	Picria felterrae	0.01120	1	Quassia	0.0643	1
5	Quassia	0.00110	1	Herba laggerae	0.0641	1
6	Mentha haplocalyx	0.00640	1	Goldenrod	0.0601	1
7	Fennel	0.00430	**0**	Forsythia suspensa Vahl	0.0566	1
8	Dried tangerine peel	0.00210	**0**	Lonicera confusa DC.	0.0563	1
9	Clove	0.00130	**0**	Honeysuckle	0.0562	1
10	Forsythia suspensa Vah	0.00120	1	Chrysanthemum	0.0550	1
11	Lonicera confusa DC.	0.00104	1	Cicada slough	0.0540	1
12	Fructus arctii	0.00101	1	Folium mori	0.0529	1

physician was used to determine whether the herb was suited to this symptom. If correct, the value of the label was 1. HINE cannot generate a satisfactory top-12 list because it cannot adequately rank the top herbs for the symptom, and cannot generate three incorrect herbs. AGA2Vec provides a better list and ranks the herbs more adequately according to the suitability. In addition, we report the results of recommendation in terms of the *precision*@k of our model and other methods on TRE. We present the performance comparison of different methods in Fig.5. The results indicate that the weights of the meta-path are effective in improving recommendation performance, and the proposed AGA2Vec can effectively capture HIN semantics in a more plausible way.

5.6 Parameter Sensitivity

We explore the sensitivity of the performance with respect to the dimensions of latent representations, suitable meta-path length l, and the trade-off parameter α that balances the weight of the first-order and the second-order proximities in HIN. We report *precision*@k on DBLP.

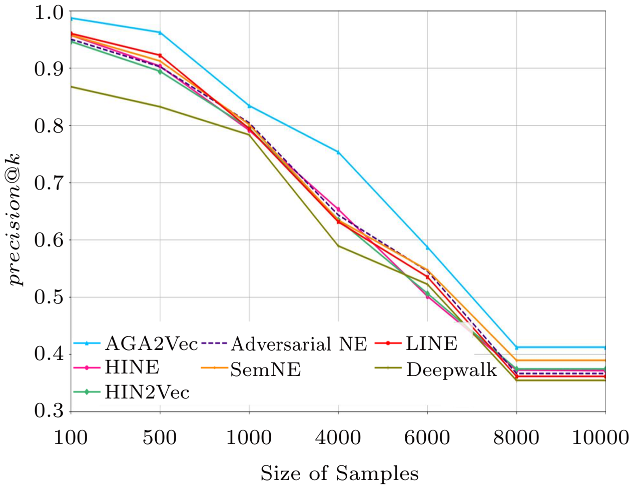

Fig.5. Herbs recommendation based on symptoms.

Fig.6(a) shows the performance of our proposed model with different dimensions. The results show that increasing the number of dimensions improved the

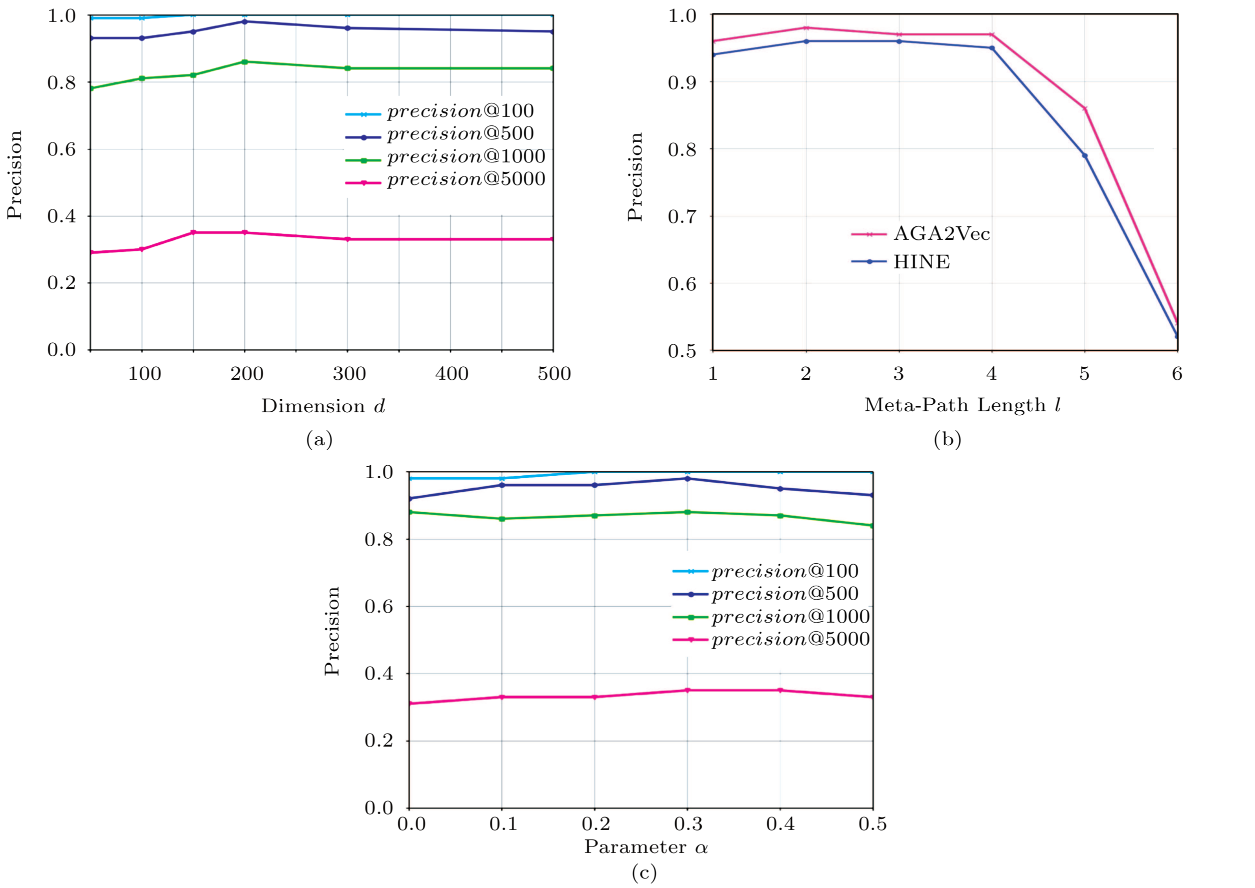

Fig.6. Effect of parameters: (a) embedding dimension d, (b) meta-path length l, and (c) trade-off parameter α.

performance. However, when the number of dimensions continuously increased, the performance started to drop. A large number of dimensions might have caused noises to degrade the performance. The performance of our model, however, dropped slowly because it took advantage of adversarial learning. To determine whether the number of dimensions for the latent representations needs a large quantity of testing tasks, our method is not very sensitive to this parameter, which is good for model application. From Fig.6(b), we observe that slightly increasing the length can improve performance. However, performance was degraded when the length became too long. This result validates the short-path theory[35], and this indicates that our model can generally capture more information than HINE. We then show how the value of α affects the performance in Fig.6(c). When $\alpha = 0$, our model only captured the global information of HIN. And, the larger $\alpha \neq 0$, the more the local information captured by our model. Fig.6(c) shows that the performance of $\alpha = 0.3$ was superior to that of $\alpha = 0$. It shows that both global and local information is essential for HIN embedding methods to preserve the structure of HIN.

5.7 Scalability

In order to illustrate its scalability, we applied AGA2Vec to learn node representation on all datasets. We computed the average runtime with increasing sizes from 100 to 1 000 000 nodes and the average degree of 10. For each model, the average runtime comprises of preprocessing for computing transition probabilities for walk paths and the runtime of node embedding. In Fig.4(b) we empirically observe that AGA2Vec scales linearly with the increase in the number of nodes generating representations for one million nodes in less than three hours. In order to speed up training the deep model, we used GAN with negative sampling.

6 Conclusions

This paper proposed a method for HIN embedding AGA2Vec. We first proposed a novel attention-based decider to dynamically determine the weights of different meta-paths. We then proposed a naive node embedding model based on the autoencoder, which is a generator to integrate the meta-path information into network embedding processes. Finally, we employed adversarial learning to significantly and robustly improve the results of naive embedding. The results of experiments on clustering, link prediction, and a case study on empirical datasets demonstrated the effectiveness of the proposed AGA2Vec.

Acknowledgements We thank the reviewers for their careful consideration and helpful comments.

References

[1] Chen Y X, Wang C G. HINE: Heterogeneous information network embedding. In *Proc. the 22nd International Conference on Database Systems for Advanced Applications*, March 2017, pp.180-195.

[2] Meng C P, Cheng R, Maniu S, Senellart P, Zhang W D. Discovering meta-paths in large heterogeneous information networks. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.754-764.

[3] Shi C, Hu B B, Zhao X, Yu P. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2019, 31(2): 357-370.

[4] Chen H X, Yin H Z, Wang W Q, Wang H, Nguyen Q V H, Li X. PME: Projected metric embedding on heterogeneous networks for link prediction. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, August 2018, pp.1177-1186.

[5] Fu T Y, Lee W C, Lei Z. HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proc. the 2017 ACM on Conference on Information and Knowledge Management*, November 2017, pp.1797-1806.

[6] Wang H W, Zhang F Z, Hou M, Xie X, Guo M Y, Liu Q. SHINE: Signed heterogeneous information network embedding for sentiment link prediction. In *Proc. the 11th ACM International Conference on Web Search and Data Mining*, February 2018, pp.592-600.

[7] Shi Y, Gui H, Zhu Q, Kaplan L M, Han J W. AspEm: Embedding learning by aspects in heterogeneous information networks. In *Proc. the 2018 SIAM International Conference on Data Mining*, May 2018, pp.144-152.

[8] Dai Q Y, Li Q, Tang J, Wang D. Adversarial network embedding. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, February 2018, pp.2167-2174.

[9] Huang Z P, Mamoulis N. Heterogeneous information network embedding for meta path based proximity. arXiv:1701.05291, 2017. https://arxiv.org/pdf/1701.05291.pdf, August 2019.

[10] Domhan T. How much attention do you need? A granular analysis of neural machine translation architectures. In *Proc. the 56th Annual Meeting of the Association for Computational Linguistics,* July 2018, pp.1799-1808.

[11] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A C, Bengio Y. Generative adversarial nets. In *Proc. the 2014 Annual Conference on Neural Information Processing Systems*, December 2014, pp.2672-2680.

[12] Sun Y Z, Norick B, Han J W, Yan X F, Yu P S, Yu X. PathSelClus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data*, 2013, 7(3): Article No. 11.

[13] Li J H, Wang C D, Huang L, Huang D, Lai J H, Chen P. Attributed network embedding with micro-meso structure. In *Proc. the 23rd International Conference on Database Systems for Advanced Applications*, May 2018, pp.20-36.

[14] Wang H W, Wang J, Wang J L, Zhao M, Zhang W N, Zhang F Z, Xie X, Guo M Y. GraphGAN: Graph representation learning with generative adversarial nets. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, February 2018, pp.2508-2515.

[15] Li C Z, Li Z J, Wang S Z, Yang Y, Zhang X M, Zhou J S. Semi-supervised network embedding. In *Proc. the 22nd International Conference on Database Systems for Advanced Application*, March 2017, pp.131-147.

[16] Tang J, Qu M, Wang M Z, Zhang M, Yan J, Mei Q Z. LINE: Large-scale information network embedding. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.1067-1077.

[17] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp.855-864.

[18] Cheng D W, Tu Y, Ma Z W, Niu Z B, Zhang L Q. BHONEM: Binary high-order network embedding methods for networked-guarantee loans. *J. Comput. Sci. Technol.*, 2019, 34(3): 657-669.

[19] Chang S Y, Han W, Tang J L, Qi G J, Aggarwal C C, Huang T S. Heterogeneous network embedding via deep architectures. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2015, pp.119-128.

[20] Guo L, Wen Y F, Wang X H. Exploiting pre-trained network embeddings for recommendations in social networks. *J. Comput. Sci. Technol.*, 2018, 33(4): 682-696.

[21] Zhou D F, Fan J X, Lin C K, Cheng B L, Zhou J Y, Liu Z. Optimal path embedding in the exchanged crossed cube. *J. Comput. Sci. Technol.*, 2017, 32(3): 618-629.

[22] Dong Y X, Chawla N V, Swami A. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proc. the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2017, pp.135-144.

[23] Ji H Y, Shi C, Wang B. Attention based meta path fusion for heterogeneous information network embedding. In *Proc. the 15th Pacific Rim International Conference on Artificial Intelligence*, August 2018, pp.348-360.

[24] Wang X, Ji H Y, Shi C, Wang B, Ye Y F, Cui P, Yu P S. Heterogeneous graph attention network. In *Proc. the 2019 World Wide Web Conference*, May 2019, pp.2022-2032.

[25] Fang Y, Lin W Q, Zheng V W C, Wu M, Chang K C C, Li X L. Semantic proximity search on graphs with metagraph-based learning. In *Proc. the 32nd IEEE International Conference on Data Engineering*, May 2016, pp.277-288.

[26] Zhang D K, Yin J, Zhu X G, Zhang C Q. Metagraph2vec: Complex semantic path augmented heterogeneous network embedding. In *Proc. the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, June 2018, pp.196-208.

[27] Sun L C, He L F, Huang Z P, Cao B K, Xia C Y, Wei X K, Yu P S. Joint embedding of meta-path and meta-graph for heterogeneous information networks. In *Proc. the 2018 IEEE International Conference on Big Knowledge*, November 2018, pp.131-138.

[28] Donahue J, Krähenbühl P, Darrell T. Adversarial feature learning. In *Proc. the 5th International Conference on Learning Representations*, April 2017.

[29] Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.95-104.

[30] Li H Y, Dong W M, Hu B G. Facial image attributes transformation via conditional recycle generative adversarial networks. *J. Comput. Sci. Technol.*, 2018, 33(3): 511-521.

[31] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. the 4th International Conference on Learning Representations*, May 2016.

[32] Xiao Y, Xiao D, Hu B B, Shi C. ANE: Network embedding via adversarial autoencoders. In *Proc. the 2018 IEEE International Conference on Big Data and Smart Computing*, January 2018, pp.66-73.

[33] Wang D X, Cui P, Zhu W W. Structural deep network embedding. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp.1225-1234.

[34] Gulrajani I, Ahmed Fa, Arjovsky M, Dumoulin V, Courville A C. Improved training of Wasserstein GANs. In *Proc. the 2017 Annual Conference on Neural Information Processing Systems*, December 2017, pp.5769-5779.

[35] Sun Y Z, Han J W, Yan X F, Yu P S, Wu T Y. PathSim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 2011, 4(11): 992-1003.

[36] Perozzi B, Al-Rfou' R, Skiena S. DeepWalk: Online learning of social representations. In *Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2014, pp.701-710.

[37] Tang J, Zhang J, Yao L M, Li J Z, Zhang L, Su Z. ArnetMiner: Extraction and mining of academic social networks. In *Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2008, pp.990-998.

Chun-Yang Ruan is a Ph.D. student at the School of Computer Science, Fudan University, Shanghai. He received his Master's degree in computer application from Zhengzhou University, Zhengzhou, in 2016. His main research interests include graph embedding and medical data mining.

Ye Wang received her Ph.D. degree in computer science from Victoria University, Melbourne, in 2019. Currently, she works at National University of Defense Technology, Changsha, as a junior lecture. Her research interests mainly include data mining and knowledge discovery, social network analysis and medical informatics.

Jiangang Ma is a lecturer with School of Science, Engineering and Information Technology in Federation University Australia, Melbourne. He received his Ph.D. degree in computer science from Victoria University, Melbourne, in 2010. His main research interests include data science, algorithms design and health informatics.

Yanchun Zhang received his Ph.D. degree in computer science from The University of Queensland, Brisbane, in 1991. He is a full professor of Victoria University, Melbourne. His research areas cover e-research, Internet and Web services, especially cooperative transactions management, distributed databases, data mining, web information systems, web services, and e-health.

Xin-Tian Chen is a Master student at the School of Computer Science, Fudan University, Shanghai. His main research interests include network representation learning, and traditional Chinese medicine.