Information Fusion 50 (2019) 221–230

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

AHNG: Representation learning on attributed heterogeneous network

Mengyue Liu[a,*], Jun Liu[b,c], Yihe Chen[d], Meng Wang[e], Hao Chen[a], Qinghua Zheng[a,f]

[a] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
[b] Ministry of Education Key Lab For Intelligent Networks and Network Security, Xi'an, China
[c] Guang Dong Xi'an Jiaotong University Academy, Shunde, China
[d] University of Toronto 27 King's College Circle Toronto, Ontario M5S 1A1 Canada
[e] School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 211111, China
[f] National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Keywords:
Attributed heterogeneous network
Network embedding
Gaussian distribution

ABSTRACT

Network embedding aims to encode nodes into a low-dimensional space with the structure and inherent properties of the networks preserved. It is an upstream technique for network analyses such as link prediction and node clustering. Most existing efforts are devoted to homogeneous or heterogeneous plain networks. However, networks in real-world scenarios are usually heterogeneous and not plain, *i.e.*, they contain multi-type nodes/links and diverse node attributes. We refer such kind of networks with both heterogeneities and attributes as attributed heterogeneous networks (AHNs). Embedding AHNs faces two challenges: (1) how to fuse heterogeneous information sources including network structures, semantic information and node attributes; (2) how to capture uncertainty of node embeddings caused by diverse attributes. To tackle these challenges, we propose a unified embedding model which represents each node in an AHN with a Gaussian distribution (AHNG). AHNG fuses multi-type nodes/links and diverse attributes through a two-layer neural network and captures the uncertainty by embedding nodes as Gaussian distributions. Furthermore, the incorporation of node attributes makes AHNG inductive, embedding previously unseen nodes or isolated nodes without additional training. Extensive experiments on a large real-world dataset validate the effectiveness and efficiency of the proposed model.

1. Introduction

Network embedding is mainly designed to encode the graph data into a low-dimensional latent space [1]. As an upstream technique, network embedding benefits a lot of network analysis tasks, such as link prediction [2,3] and node clustering [4], as well as attracts considerable attention in various fields, ranging from linguistics [5,6], social sciences [7,8] to biological networks [9].

Most of the network embedding methods [10–13] rely on a basic assumption that the networks are homogeneous [14], that is, the networks contain the same type of nodes and links. However, this assumption is untenable as many real-world networks contain multi-type nodes and links, holding rich structure and manifold semantics information, showing heterogeneity. For example, a network in terms of film may contain several types of nodes like *director, movie, actor*, and *producer*, as well as different relationships among nodes, such as *direct, play*, and *produce*. In addition, nodes in heterogeneous networks are usually affiliated with diverse attributes (*e.g.*, director's profile information and movie's abstract) which play an important role in the formation of network structure [15]. In this paper, we refer such kind of heterogeneous networks affiliated with diverse attributes as *attributed heterogeneous networks* (AHN). Our goal is to design an effective model to represent AHNs in a low-dimensional space with the structure and inherent properties preserved.

1.1. Motivating example and challenges

Fig. 1 is a toy attributed heterogeneous bibliographic network which contains multi-typed nodes including *author* (a), *paper* (p) and *venue* (v), and diverse links including *write* (r_1), *accept* (r_2) and *cite* (r_3). Different types of nodes have different attributes. As shown in Fig. 1, f_1, f_2 indicate attributes of authors which are depicted in blue, attributes of papers are indicated by f_3, f_4 which are depicted in green, attributes of venues are indicated by f_5, f_6 which are depicted in orange. Accompanying with such abundant heterogeneous information, two challenges in AHN embedding have to be addressed:

How to fuse heterogeneous information. Three elements are needed to consider in AHN embedding: (1) multi-type nodes, (2) multi-type links, and (3) diverse attributes of nodes, involving semantic types, topology

[*] Corresponding author.
E-mail address: liumengyue@stu.xjtu.edu.cn (M. Liu).

https://doi.org/10.1016/j.inffus.2019.01.005
Received 15 September 2018; Received in revised form 13 January 2019; Accepted 26 January 2019
Available online 28 January 2019

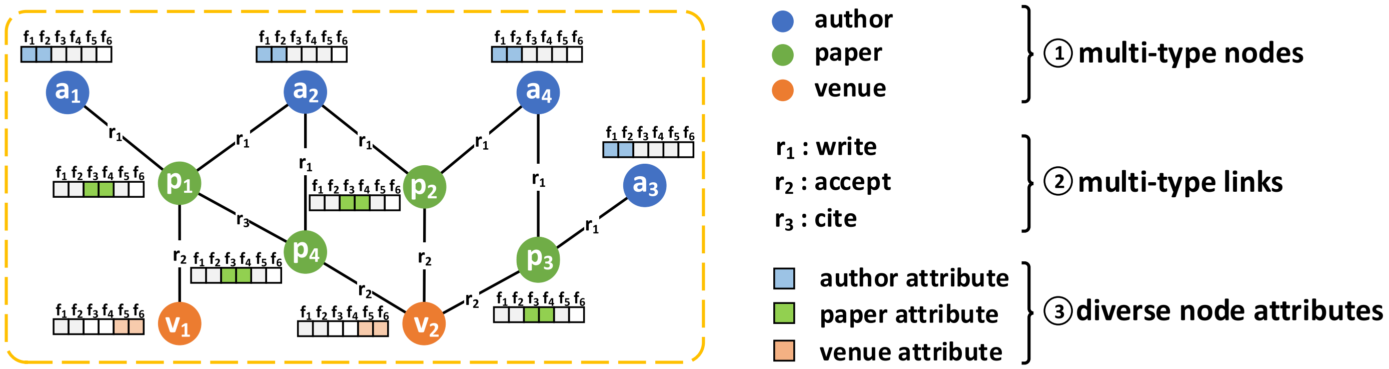


Fig. 1. An illustrative example of attributed heterogeneous networks. As shown in the figure, the toy attributed bibliographic network contains (1) multi-type nodes, (2) multi-type links, and each node is affiliated with (3) diverse attributes depicted in different colors. The attribute vector for each node is a concatenation of different types of attributes.

information and unstructured text. A vast majority of homogeneous network embedding methods [10,11,16,17] ignore all of the above three elements. Some methods, such as G2G [18] and LANE [19], take attributes into account but convert a heterogeneous network to a homogeneous one by a projection on a single node type. Such projection will lose valuable information. Few heterogeneous network embedding methods came out until the concept of meta-path [20] is presented. Meta-paths are able to effectively capture semantic information of heterogeneous networks. For example, in Fig. 1, there are two kinds of meta-path between a_2 and a_4, indicating different semantic information: one meta-path “$a_2 \xrightarrow{r_1} p_2 \xrightarrow{r_1^{-1}} a_4$” indicates a collaborative relationship between a_2 and a_4; another meta-path “$a_2 \xrightarrow{r_1} p_2 \xrightarrow{r_2} v_2 \xrightarrow{r_2^{-1}} p_3 \xrightarrow{r_1^{-1}} a_4$” indicates a_2 and a_4 may have common research interests, and the superscript -1 denotes the reverse of relation direction. Most meta-path based methods [21,22] simultaneously fuse multi-type nodes and links but ignore node attributes. Recent models regard node attributes as new kinds of nodes but greatly increase the complexity of heterogeneous network topology such as BL-MNE [23]. Therefore, it is arduous to seamlessly fuse multi-type nodes/links and diverse attributes in a unified AHN embedding model.

How to capture the uncertainty of node embeddings. Most existing network embedding methods represent each node as a vector, *i.e.*, depict a node as a single point in the latent space, leading to a certain representation of a node. However, the complex and diverse attributes of nodes actually make node representations uncertain. For example, in Fig. 1, a_2 publishes p_1 and p_2 in v_1 and v_2 respectively. When the attributes of these nodes are different or even opposite, *e.g.*, v_1 and v_2 belong to two different research areas respectively or p_1 and p_2 have diverse attributes, they will cause variant effects while introducing the representation of a_2. Recent years, some methods [18,24] embed nodes in homogeneous networks with distributions, but none of them is applicable for AHNs. Intuitively, an AHN embedding model should learn the representation of a node based on its neighbors, related links, and node attributes. Encoding these various information magnify the uncertainty of nodes.

1.2. Solutions and contributions

To effectively cope with the aforementioned challenges, we propose a novel model to represent each node in an attributed heterogeneous network with a Gaussian distribution in a latent space, called AHNG. AHNG encodes diverse attributes with a two-layer neural network and preserves the structure and semantic information of AHN based on meta-path based random walks [21], which can seamlessly fuse multi-nodes/links and node attributes. AHNG represents each node as a Gaussian distribution, which is capable of capturing the uncertainty of a node representation. The main contributions of our work are summarized as follows:

- We propose a novel model AHNG, which is able to fuse network structure, semantic information and node attributes of AHNs in a unified embedding model. AHNG shows inductiveness, *i.e.*, it can be generalized to previously unseen nodes or isolated nodes without additional training.
- We represent each node with a multi-dimensional Gaussian distribution to capture the uncertainty of node representations and experimentally prove that AHNG achieves a better performance by utilizing KL-divergence, an asymmetric measurement, to measure the dissimilarity among nodes.
- We conduct extensive experiments on a large real-world attributed heterogeneous networks to investigate the effectiveness and efficiency of the proposed AHNG.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 formulates the problem, and proposes a framework of AHNG in details. Experimental results are discussed in Section 4. Conclusions and future work are presented in Section 5.

2. Related work

In this work, we review the embedding methods in (1) homogeneous networks and (2) heterogeneous networks.

2.1. Homogeneous network embedding

A vast majority of existing methods focus on embedding nodes in homogeneous networks in previous years. The pioneer studies in graph embedding always preserve the graph property with a matrix and factorize the matrix to reduce the dimension of network data. For example, LLE [25] represents the connections between nodes with an adjacency matrix and computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs. Laplacian Eigenmaps [26] is a geometrically motivated algorithm for the construction of the representation for nodes. Recently, some matrix factorization based methods incorporate text features to learn embeddings such as TADW [27]. However, matrix factorization based methods suffer from both computational and statistical performance drawbacks.

With the advent of deep learning techniques and the invention of word2vec [28], researchers extend embedding methods in natural language processing from sequences of words to graphs. DeepWalk [10] and Node2vec [11] preserve higher-order proximity between nodes based on skip-gram model. LINE [29] preserves both first order and second order proximities between a node and its contexts. SDNE [30] employs deep auto-encoders to capture the highly non-linear network structure, which preserve the global and local structure. VAGE [31] applies a graph convolutional network encoder and an inner product decoder to the task of graph embedding. G2G [18] considers node attributes and utilizes encoders to learn embeddings. Deep learning based methods learn more and deeper semantic information of the network compared to shallow representation methods.

Previous network embedding methods in homogeneous networks cannot be directly applied to heterogeneous networks which are linked with real-world scenarios more naturally.

2.2. Heterogeneous networks embedding

Network embedding in heterogeneous networks emerged a few years ago and quickly became a flourishing research field. Existing network embedding techniques in heterogeneous networks can be broadly divided into three types: (1) random walk based (2) network factorization based (3) deep neural network based.

Due to the multi-type of nodes and links in heterogeneous networks, it is difficult for random walkers to find a walk mode to traverse the whole networks. Meta-paths limit the liberty of random walkers, reducing the traversal complexity in AHNs. For example, inspired by skip-gram model on homogeneous networks, metapath2vec [21] proposes a heterogeneous skip-gram model and utilizes meta-path based random walks to traverses the whole network. HIN2Vec [22] also utilizes specific meta-paths and proposes a neural network model to capture the rich semantics embedded in heterogeneous networks. However, meta-path based random walk models can not completely preserve the whole structure of a network when the number of walks *wk* for each node and walk lengths *len* are too small. In other words, unreachable nodes and isolated nodes can not be represented at all.

Network factorization based techniques divide a big heterogeneous network into several small bipartite graphs to ease the complexity of heterogeneous network. For example, PTE [32] divides a heterogeneous text network into word-word network, word-document network and word-label network, and then utilizes both labeled and unlabeled data to learn the embedding of text. However, network factorization based techniques have no unified standard when dividing different networks into several small ones.

Deep neural network based models provide a robust and effective embedding way based on deep learning techniques, which reflect more and deeper semantic information of the network. Recent years, some deep neural networks based methods embed heterogeneous networks with auxiliary informations. For example, HNE [1] learns representation for texts and images through a deep neural network and a convolutional neural network respectively, and then unifies them into a common space using linear transformation but neglects the semantic information among nodes. BL-MNE [23] regards attributes as new kinds of nodes and utilizes auto-encoders to encode each node, but increases the complexity of network topology. Meanwhile, few deep neural network based models in heterogeneous networks take uncertainty of node representations into account.

3. The proposed AHNG

3.1. Problem formulation and framework

Following the standard notation, we use the normal lowercase characters (*e.g.*, u) to denote scalars and bold lowercase characters (*e.g.*, **f**) to indicate vectors. Matrices are written as bold uppercase characters (*e.g.*, **F**) and uppercase italic characters (*e.g.*, *V*) for sets.

Definition 1. An **attributed heterogeneous network** is a directed graph $G = (V, E, \mathbf{F})$ with a node mapping function $\phi : V \to \mathcal{A}$, and a link mapping function $\varphi : E \to \mathcal{R}$, where each node $u \in V$ belongs to one type $A_a \in \mathcal{A}$ and each link $e \in E \subseteq V \times V$ belongs to one type $r \in \mathcal{R}$, and $|\mathcal{A}| + |\mathcal{R}| > 2$. Moreover, $\mathbf{F}_i$, a row vector of attribute matrix $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_i, \ldots \mathbf{F}_{|V|}]^\top$, denotes the attribute information of the *i*-th node $u_i \in V$, concatenating all types of node attributes in $\mathcal{A}$.

For example, in Fig. 2, $G = (V, E, \mathbf{F})$, where $V = \{a_1, a_2, a_3, a_4, p_1, p_2, p_3, p_4, v_1, v_2\}$, $E = \{a_1 p_1, p_1 v_1, \ldots\}$, ϕ: $V \to \{A, P, V\}$, φ: $E \to \{write, accept\}$, and $\mathbf{F} = \{\mathbf{F}_1, \ldots, \mathbf{F}_{10}\}$. As for node attributes, $\{f_1, f_2\}$ denote *author* attributes: *affiliations* and *research interest*; $\{f_3, f_4\}$ denote *paper* attributes: *paper title* and *abstract*; $\{f_5, f_6\}$ denote *venue* attributes: *venue name* and *venue introduction.* More specifically, every node is represented by an attribute vector $\mathbf{F}_i$ which contain six kinds of attributes $(f_1, \ldots, f_6)$. Suppose that we use an attribute vector $\mathbf{F}_1$ to represent an author u_1, the values of attributes f_1, f_2 are given by the author u_1 while the attributes f_3, f_4, f_5, f_6 are set as zeros because they are used to describe *paper* and *venue*. besides, the dimension of each attribute f_i is defined by users.

Definition 2. A **meta-path** [33] defines a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$ between type A_1 and A_{l+1}, which is denoted in the form of $\mathcal{P} : A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, and $l\,(l \geq 1)$ is the length of $\mathcal{P}$. Each subscript (e.g. $1, 2, \ldots, l$) represents the position of a type or a link in $\mathcal{P}$.

Meta-paths extract semantic information effectively. For example, as shown in Fig. 2, meta-path $A \xrightarrow{write} P \xrightarrow{accept^{-1}} V \xrightarrow{accept} P \xrightarrow{write^{-1}} A$ denotes two authors are interested in the same research, $A \xrightarrow{write} P \xrightarrow{write^{-1}} A$ denotes collaboration relationship, and $P \xrightarrow{write^{-1}} A \xrightarrow{write} P$ denotes two papers are written by a same author.

The problem of network embedding on attributed heterogeneous networks can be formally defined as follows:

Problem formulation. Given an attributed heterogeneous network $G = \{V, E, \mathbf{F}\}$, where $|V| = n$ and $\mathbf{F} \in \mathbb{R}^{n \times d}$, the embedding of G aims to encode each node $u_i \in V$ with a lower-dimensional Gaussian distribution $z_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}^l$, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{l \times l}$ with $l \ll n, d$.

Framework. AHNG aims to encode nodes into a latent low-dimensional space by leveraging three types of information: multi-type nodes, multi-type links, and node attributes. The framework of the proposed AHNG is shown in Fig. 2, which consists of three essential components: (I) *attribute encoder*: an attribute encoder is designed to encode diverse node attributes and output mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ for each node u_i in *V*; (II) *meta-path based random walker*: a random walker traverses the heterogeneous network based on preassigned meta-paths M_p to preserve the semantic and structure information of the AHN; (III) *learner*: to update the parameters of the attribute encoder and learn the node embedding.

3.2. Attribute encoder

A two-layer neural network $\hbar$, called *attribute encoder*, processes the node's attributes, outputs mean and covariance for each node. More specifically, as shown in Fig. 2, $\mathbf{f}_i$ represents the attribute vector for each node u_i, and the first layer outputs an intermediate representation y_i for u_i. The layer μ and layer Σ output mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$ respectively for node u_i. Formally, the relationship between these variables can be represented by the following equations:

$$\begin{cases} y_i = \sigma_1(\mathbf{W}_1\, \mathbf{f}_i^\top + \mathbf{b}_1) \\ \boldsymbol{\mu}_i = \mathbf{W}_\mu\, y_i + \mathbf{b}_2 \\ \boldsymbol{\Sigma}_i = \sigma_2(\mathbf{W}_\Sigma\, y_i + \mathbf{b}_3) \end{cases}, \qquad (1)$$

where σ_1 and σ_2 represent *relu* and *elu* active functions, and $\mathbf{W}_1, \mathbf{W}_\mu, \mathbf{W}_\Sigma$ represent the weight matrix of layer 1, μ and Σ respectively. The well-tuned parameters of the attribute encoder enable AHNG to be an inductive learning model.

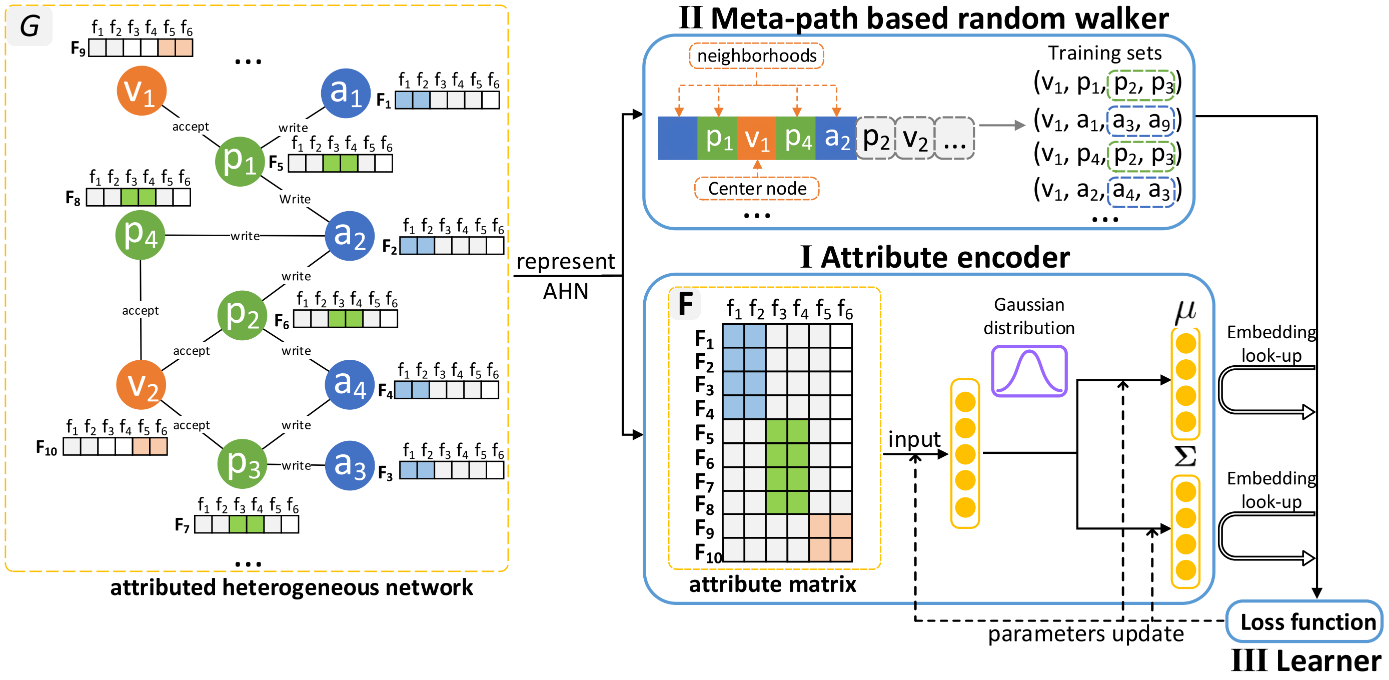


Fig. 2. The framework of the proposed AHNG.

3.3. Meta-path based random walker

We preserve structure and semantic information of AHN by demonstrating it as a set of random walk paths. Different from random walk-based models in homogeneous network, in which a random walker traverses the whole network without constraint, a meta-path based random walker [21] traverse an AHN based on preassigned meta-paths. For instance, given a meta-path $\mathcal{P}: A_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{t-1}} A_t \xrightarrow{R_t} A_{t+1} \cdots \xrightarrow{R_l} A_{l+1}$, and u_i^t denotes node u_i belonging to type t. The random walker goes to node u_i^t at step j, and at step $j+1$ the node u_{i+1} should satisfied with $(u_i^t, u_{i+1}) \in E$ and $\phi(u_{i+1}) = t+1$. As for those nodes who conform to conditions above, one of them is picked out randomly.

In order to cover every node in the network, meta-path based random walk models should assure both of the number of walks and walk lengths to be sufficient. For instance, metapath2vec achieves better performance when the number of walks (wk) and walk lengths (len) are larger than 800 and 100 respectively. However, due to the incorporation of attributes, we experimentally show that even when the two parameters are very slim such as $wk = 5$ and $len = 2$, AHNG still performs well.

3.4. Learner

Skip-gram [28] models maximize the probability of observing a neighborhood node (context) based on a center node embedding. These models based on the conditional independence assumption: given a learned embedding, observing neighboring nodes become independent [10,11,29]. We follow the clue and formulate AHNG with considering the heterogeneity of networks and attributes of nodes:

$$\arg\max_{\hbar} \sum_{u_i \in V} \sum_{t \in \mathcal{A}} \log Pr\Big(N_t(u_i) | \hbar(u_i)\Big), \tag{2}$$

$N_t(u_i) \subseteq V$ collects the t-type ($t \in \mathcal{A}$) neighborhoods which appears within the window-size w of the center node u_i, *i.e.*, $N_t(u_i) = \{u_j | d(u_i, u_j) \le w\}$, u_j^t denotes node u_j belonging to type t, and $d(u_i, u_j)$ denotes the shortest distance between u_i and u_j.

Based on the conditional independence assumption, we approximate the conditional probability in Eq. (2) as follows:

$$Pr\Big(N_t(u_i) | \hbar(u_i)\Big) = \prod_{u_j \in N_t(u_i)} Pr(u_j | \hbar(u_i)). \tag{3}$$

One approach for parameterizing the skip-gram model follows the literature in neural network language models, and the basic skip-gram formulation is defined using a softmax function: $Pr(u_j | \hbar(u_i)) = \frac{exp(z_j \cdot z_i)}{\sum_{j' \in V} exp(z_{j'} \cdot z_i)}$, where z_i and z_j denote the embeddings of a center node and its neighbor respectively, *i.e.*, $z_i = \hbar(u_i)$ and $z_j = \hbar(u_j)$. The production $z_j \cdot z_i$ can be regarded as a similarity measure between z_j and z_i. However, dot product only considers means without the incorporation of covariances. We assume that it is ubiquitous that the similarity between different types of nodes has directionality, *i.e.*, the "distance" between different types of nodes is asymmetric. For instance, when we mention a beginner of his/her research field, his/her research field will be brought up. However, when we mention this research field, the prestigious experts in this field will come in our mind rather than this beginner. It means that the distance from the scholar to the research field is unequal to the distance from this research field to this scholar.

The key notion of skip-gram is to learn center node embeddings that specializes in predicting the nearby nodes, thus the similarity between a center node and a context node is assumed to be asymmetric. Hence, we use Kullback–Leibler(KL) divergence, a naturally asymmetric measure, to incorporate covariance (denotes uncertainties of node representations) into the model:

$$\begin{aligned} \mathcal{H}(z_i, z_j) = D_{KL}(\mathcal{N}_j || \mathcal{N}_i) &= \int_{u \in \mathbb{R}} \mathcal{N}(u; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \log \frac{\mathcal{N}(u; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\mathcal{N}(u; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \mathrm{d}x \\ &= \frac{1}{2}\Big(\log \frac{|\boldsymbol{\Sigma}_j|}{|\boldsymbol{\Sigma}_i|} - l + \mathrm{tr}(\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_j) + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \Big), \end{aligned} \tag{4}$$

where $z_i, z_j \in \mathbb{R}^l$, $\boldsymbol{\Sigma}^{-1}$ and $\mathrm{tr}(\boldsymbol{\Sigma})$ indicate the inverse and the trace of a covariance matrix, respectively. We assume that two attributes are uncorrelated and each covariance matrix $\boldsymbol{\Sigma}_i$ is diagonal. $D_{KL}(\mathcal{N}_j || \mathcal{N}_i)$ denotes that an approximating distribution ($\mathcal{N}_i$) is used to model an unknown distribution ($\mathcal{N}_j$). The smaller $D_{KL}(\mathcal{N}_j || \mathcal{N}_i)$ is, the more similar the two distributions are.

In this way, $Pr(u_j|\hbar(u_i))$ could be rewrote as:

$$Pr(u_j|\hbar(u_i)) = \frac{exp[\mathcal{H}(z_i, z_j)]}{\sum_{j'\in V} exp[\mathcal{H}(z_i, z_{j'})]}. \tag{5}$$

Eq. (5) is computationally expensive due to the summation term. Hence, we utilize negative sampling [34] to redefine the loss function:

$$\mathcal{L} = \log \sigma[\mathcal{H}(z_i, z_j^t)] + \sum_{k=1}^{K} \mathbb{E}_{u_k^t \sim P_{neg}(u^t)}\Big(\log \sigma[-\mathcal{H}(z_i, z_k^t)]\Big), \tag{6}$$

where $\sigma(x) = \frac{1}{1+exp(-x)}$ and $z_j^t \in N_t(u_i)$. Negative samples denote nodes not in $N_t(u_i)$, and the type of a negative sample z_k^t should be as the same as z_j^t. For instance, in Fig. 2, when v_1 is the center node, P-type neighbors of v_1 is $N_A(v_1) = \{a_1, a_2\}$ and A-type neighbors of v_1 is $N_A(v_1) = \{a_1, a_2\}$, and the corresponding negative samplings are $\{p_2, p_3\}$ and $\{a_3, a_9\}$. K negative samples are collected from a noise distribution $P_{neg}(u^t)$ for each center node z_i, and K usually equals to the window size w. Inspired by [28], AHNG set the selecting probability of u_k^t as: $p(u_k^t) = \frac{counter(u_k^t)^{\frac{3}{4}}}{\sum_{u\in V} counter(u)^{\frac{3}{4}}}$, where the numerator indicates the occurrence frequency of node u_k^t in all meta-path based random walks, and denominator indicates the summation occurrence frequency of all nodes. We utilize gradient descent on the *attribute encoder* to optimize Eq. (6) until either convergence or the training epochs have been finished.

4. Experiments

In this section, we empirically evaluate the effectiveness of AHNG on three classic benchmark tasks: (I) link prediction, ((II)) node clustering, and (III) multi-class classification. The inductiveness of AHNG is verified at last.

4.1. Datasets

AMiner [1] [35] is a collection of bibliographic entities such as papers, authors and venues. To construct our own attributed heterogeneous network, we extract 18,243 papers (P) which are written by 41,274 authors (A), and accepted by 12 venues (V). These 12 venues are uniformly selected from four areas including *Database, Data Mining, Computer Vision* and *Machine Learning*. The raw data contains lots of authors with the same names, which makes us unable to accurately match authors with their corresponding papers. In order to eliminate the issue of name repetition, we compute the similarity of affiliations of authors and papers via calling *Levenshtein* package. The attribute information of 12 venues are represented by one hot encoding. The attribute information of a paper is represented by combining abstract of the paper and the encoding of its belonging venue. The attributes of a author is represented by combining research interests of the author and venues which he/she has ever published.

4.2. Baseline methods

We compare AHNG to several state-of-art methods from different aspects: (1) embedding methods in homogeneous networks; (2) embedding methods in heterogeneous networks; and (3) the variants of AHNG. Parameters of these baseline methods follow the settings in their respective experiments.

- **node2vec** [11] is a representative of skip-gram based embedding model in homogeneous networks. Node2vec defines a flexible notion of a nodes network neighborhood and efficiently embeds nodes via a biased random walk procedure.
- **LINE** [29] preserves both the local and global network structures in large-scale homogeneous networks. We utilize both first-order and second-order proximity and concatenate the vector representations learned by LINE(1st) and LINE(2nd) into a longer vector as LINE's authors did.
- **metapath2vec++** [21] is one of the state-of-the-art embedding algorithms for large-scale heterogeneous networks. It exploits metapaths to guide random walkers and proposes a heterogeneous skip-gram model to learn node embeddings.
- **AHNG_sy** is one variant model of the proposed AHNG, which utilizes a symmetric measurement, inner product, to compute the similarity between nodes. We compare AHNG with AHNG_sy to inspect the performance of asymmetric and symmetric measures.
- **AHNvec** is another variant model of AHNG which represents each node as a vector rather than a Gaussian distribution in a latent space. We compare AHNG with AHNvec to inspect the necessity of considering the uncertainty of nodes during the training process.

In the experiments, homogeneous network embedding methods are implemented by considering different types of nodes and links as the same type. For the meta-path based models, we specify the meta-path $A \xrightarrow{write} P \xrightarrow{accept^{-1}} V \xrightarrow{accept} P \xrightarrow{write} A$ to guide random walkers.

4.3. Effectiveness evaluation

For all embedding methods, the dimension of node embeddings l is set to 128. We follow the default parameter settings in each baseline methods: for node2vec, the number of walks per node *wk*, the walk length *len, p* and *q* for parameterized random walks are set to 10, 80, 1 and 1, respectively; for metapath2vec++, $wk = 1000$, $len = 100$ and negative samples $neg\_sample = 5$; for LINE, $neg\_sample = 5$; for the proposed AHNG and its variants, AHNG_sy and AHNvec, $wk = 5$, $len = 10$ and $neg\_sample = 2$, the context window $w = 2$. For each of the three classic empirical tasks, we examine the parameter sensitivity of the proposed AHNG by varying each of parameters with others fixed.

4.3.1. Link prediction

Setup. Link prediction is a mainstream task to evaluate the effectiveness of the embeddings. Our extracted AMiner network contains two types of links that need to be predicted: $A-P$, $P-V$. We confirm the connectivity of the entire graph, and then hide a set of existing/non-existing links from the original attributed heterogeneous network as the test set according to a hidden ratio $\epsilon \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$ to report the performance. A validation set is created for hyper-parameter tuning and early stopping, including equal number of 10% randomly selected existing and non existing links.

Two frequent metrics are used to compare the link prediction performance of different methods: *area under the ROC curve* (**AUC**), *average precision* (**AP**). The higher the values of AUC and AP are, the better the prediction performance is.

Results and discussion. The average link prediction results are presented in Table 1, and the best result in each column is highlighted in bold. As we can see, the values of AUC and AP of the proposed AHNG and its variants outperform all the baselines, explicitly proving that the learned embeddings are useful. When the hidden ratio reduces from 10% to 50%, the AUC values of node2vec, LINE and metapth2vec++ are reduced by 15.69%, 22.85% and 31.26% respectively. However, AHNG and its variants can still obtain very good and stable performance, and AUC values of AHNG declines by 2% approximately. The reason tends to be that these baseline models suffer from the information sparsity a lot as the hidden ratio ϵ increases, but AHNG and its variants conquer it by fusing attribute information. It is surprising that as a homogeneous network embedding method, node2vec performs so well in link prediction, and segmental results of node2vec are on a par with AHNG. Compared to

[1] https://www.aminer.cn/citation.

Table 1
AUC and AP values for link prediction on AMiner data.

Metric	Method	Hidden ratio ϵ				
		10%	20%	30%	40%	50%
AUC	node2vec	0.9701	0.9641	0.9567	0.9409	0.8132
	LINE	0.8260	0.8188	0.8053	0.7781	0.5975
	metapath2vec++	0.6029	0.5592	0.5504	0.4545	0.2903
	AHN2vec	0.9617	0.9553	0.9500	0.9461	0.9404
	AHNG_sy	**0.9703**	0.9687	0.9675	0.9651	0.9625
	AHNG	0.9698	**0.9688**	**0.9676**	**0.9653**	**0.9627**
AP	node2vec	**0.9669**	0.9572	0.9526	0.9474	0.8646
	LINE	0.8752	0.8666	0.8534	0.8083	0.5499
	metapath2vec++	0.5937	0.5416	0.5364	0.4896	0.3998
	AHN2vec	0.9597	0.9591	0.9457	0.9407	0.9385
	AHNG_sy	0.9659	0.9572	0.9549	**0.9536**	0.9512
	AHNG	**0.9669**	**0.9601**	**0.9553**	**0.9536**	**0.9520**

Table 2
Multi-class classification via a KNN classifier.

Metric	Method	Training ratio λ								
		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	node2vec	0.9005	0.9219	0.9518	0.9429	0.9468	0.9515	0.9558	0.9577	0.9618
	line	0.9323	0.9475	0.9556	0.9604	0.9640	0.9665	0.9693	0.9703	0.9704
	metapath2vec++	0.9701	**0.9721**	0.9731	0.9752	0.9756	0.9758	0.9760	**0.9777**	**0.9778**
	AHNvec	0.9681	0.9694	0.9710	0.9722	0.9724	0.9731	0.9741	0.9743	0.9746
	AHNG_sy	0.9660	0.9688	0.9691	0.9704	0.9729	0.9748	0.9753	0.9761	0.9769
	AHNG	**0.9719**	**0.9721**	**0.9733**	**0.9756**	**0.9758**	**0.9762**	**0.9770**	**0.9777**	**0.9778**
Macro-F1	node2vec	0.9002	0.9217	0.9514	0.9424	0.9463	0.9511	0.9556	0.9574	0.9610
	line	0.9321	0.9473	0.9552	0.9599	0.9636	0.9662	0.9691	0.9700	0.9699
	metapath2vec++	0.9700	0.9718	0.9727	0.9735	0.9737	0.9745	0.9752	0.9754	0.9777
	AHNvec	0.9682	0.9696	0.9711	0.9724	0.9723	0.9730	0.9741	0.9745	0.9750
	AHNG_sy	0.9662	0.9690	0.9694	0.9705	0.9725	0.9737	0.9745	0.9750	0.9772
	AHNG	**0.9702**	**0.9719**	**0.9731**	**0.9736**	**0.9740**	**0.9747**	**0.9753**	**0.9761**	**0.9778**

Table 3
Multi-class classification via a SVM classifier.

Metric	Method	Training ratio λ								
		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	node2vec	0.9483	0.9491	0.9517	0.9573	0.9584	0.9592	0.9595	0.9602	0.9617
	line	0.9344	0.9452	0.9507	0.9546	0.9569	0.9590	0.9612	0.9614	0.9620
	metapath2vec++	0.9596	**0.9615**	0.9635	0.9641	0.9662	**0.9679**	0.9695	0.9727	0.9730
	AHN2vec	0.9413	0.9487	0.9569	0.9573	0.9585	0.9601	0.9667	0.9696	0.9701
	AHNG_sy	0.9432	0.9496	0.9554	0.9588	0.9632	0.9643	0.9666	0.9689	0.9724
	AHNG	**0.9598**	0.9612	**0.9636**	**0.9642**	**0.9665**	**0.9679**	**0.9699**	**0.9728**	**0.9733**
Macro-F1	node2vec	0.9484	0.9491	0.9514	0.9572	0.9582	0.9588	0.9601	0.9610	0.9612
	line	0.9346	0.9447	0.9501	0.9541	0.9563	0.9583	0.9593	0.9610	0.9614
	metapath2vec++	0.9589	0.9607	0.9621	0.9635	0.9661	0.9699	0.9713	**0.9725**	0.9734
	AHN2vec	0.9427	0.9522	0.9563	0.9574	0.9581	0.9595	0.9664	0.9694	0.9702
	AHNG_sy	0.9508	0.9569	0.9588	0.9607	0.9643	0.9684	0.9696	0.9703	0.9730
	AHNG	**0.9591**	**0.9614**	**0.9625**	**0.9637**	**0.9666**	**0.9708**	**0.9719**	**0.9725**	**0.9735**

node2vec and metapath2vec, AHNG and its variants require the smallest *wk* and *len* to achieve the highest AUC and AP values, from their high efficiency. We assume that the efficiency also benefits from the fusion of attributes. Metapath2vec++ gets poor performance in this task but requires $wk = 1000$ and $len = 100$ which will cost much space and time. The AUC and AP values of AHNG_sy are on a par with AHNG, which indicates that the measurements may not be important for link prediction. The AUC and AP values of AHNvec are inferior to AHNG, indicating the necessity of capturing nodes uncertainty by embedding nodes with Gaussian embedding.

4.3.2. Multi-class classification

***Setup*.** We evaluate the node classification performance on AMiner dataset with ground-truth classes. In our AMiner attributed heterogeneous network, venues are divided into 4 categories[2] according to Google Scholar[3]. The label of each paper is as the same as its venue's label. The label of each author is assigned to the category with the majority of his/her publications. We assure each node in the dataset with a label indicating its research area.

The embeddings of nodes, generated from the above-mentioned network embedding methods, are used as features to classify each node into one of four labels. We randomly sample 10–90% with the interval 10 of nodes along with labels as the training data and use the remaining nodes to test the performance. We employ two classic classifier including SVM classifier and KNN classifier, and repeat each process ten times to obtain the average performance.

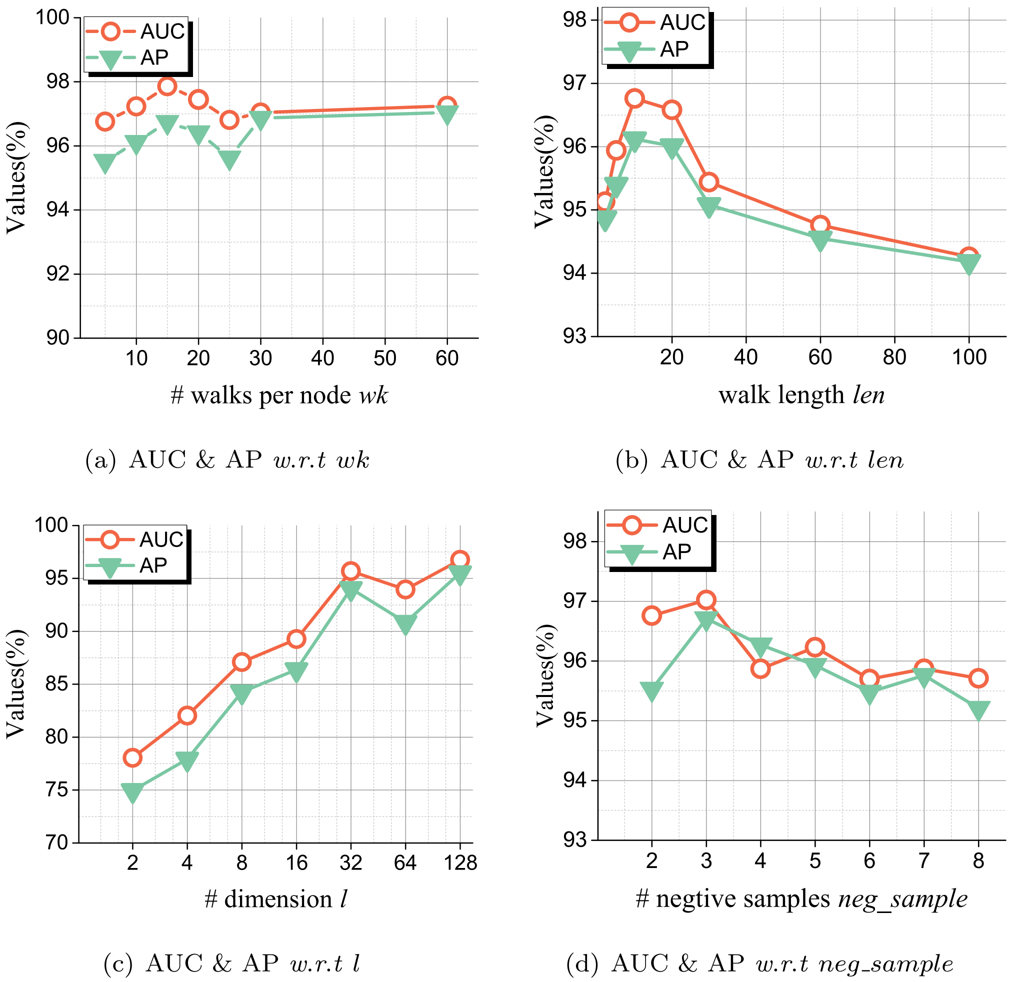
[2] 1.Database, 2.Data Mining, 3.Computer Vision, 4.Machine Learning.

[3] https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng.

Table 4
The variance interval of metapath2vec++ and AHNG across different training ratio in Table 2.

Metric	Method	Training ratio λ				
		10%	20%	30%	40%	50%
Micro-F1 (%)	metapath2vec++	97.0092 ± 0.0105	**97.2054 ± 0.0103**	97.3063 ± 0.0102	97.5211 ± 0.0101	97.5623 ± 0.0108
	AHNG	**97.1892 ± 0.0105**	**97.2102 ± 0.0104**	**97.3315 ± 0.0101**	**97.5641 ± 0.0101**	**97.5843 ± 0.0104**
Macro-F1 (%)	metapath2vec++	97.0002 ± 0.0101	97.1834 ± 0.0102	97.2665 ± 0.0103	97.3487 ± 0.0105	97.3725 ± 0.0110
	AHNG	**97.0223 ± 0.0104**	**97.1942 ± 0.0102**	**97.3138 ± 0.0102**	**97.3611 ± 0.0102**	**97.4045 ± 0.0103**

Metric	Method	Training ratio λ			
		60%	70%	80%	90%
Micro-F1 (%)	metapath2vec++	97.5778 ± 0.0102	97.6040 ± 0.0101	**97.7667 ± 0.0102**	**97.7815 ± 0.0104**
	AHNG	**97.6237 ± 0.0103**	**97.7022 ± 0.0103**	**97.7704 ± 0.0104**	**97.7849 ± 0.0102**
Macro-F1 (%)	metapath2vec++	97.446 ± 0.0102	97.5228 ± 0.0102	97.5442 ± 0.0101	97.7671 ± 0.0104
	AHNG	**97.4729 ± 0.0102**	**97.5344 ± 0.0103**	**97.6083 ± 0.0102**	**97.7836 ± 0.0102**



(a) AUC & AP *w.r.t wk*

(b) AUC & AP *w.r.t len*

(c) AUC & AP *w.r.t l*

(d) AUC & AP *w.r.t neg_sample*

Fig. 3. Parameter sensitivity in link prediction (hiding ratio ϵ = 30%).

The classification performance is measured by **Micro-F1** and **Macro-F1** metrics. The higher the values of these three metrics are, the better the multi-class classification performance is.

Results and discussion. The classification results are shown in Table 2 (KNN classifier) and 3 (SVM classifier). The best results in each column are also highlighted in bold. We observe that all methods perform well in this task but AHNG still outperforms all baselines. Overall, the heterogeneous network embedding methods perform better than homogeneous network embedding methods. Compared to link prediction experiments, metapath2vec++ has outstanding performance in multi-class classification and sometimes it reached the highest value. Although the F1 scores of metapath2vec++ are close to AHNG, AHNG requires smaller *wk* and *len* than metapath2vec++, which proves the efficiency of AHNG. The stable F1 scores imply that AHNG is sufficient with only a small percentage of labeled nodes available. The F1 scores of AHNG_sy and AHNvec have little difference from AHNG, indicating that adopting a symmetric measure or representing nodes with Gaussian distributions have little influence on nodes classification.

We notice that the results in Table 2 (especially bold ones) have minor varieties when we keep a few decimal places. Therefore, we especially calculate the variance/confidence interval of the 10 runs for metapath2vec++ and AHNG and keep more decimal places across different training ratio to revel their differences. The results (in %) are shown in Table 4 (mean preference 90% confidence intervals), proving that AHNG outperforms metapath2vec++.

Table 5
Node clustering results in AMiner data.

	node2vec	LINE	metapath2vec	AHNvec	AHNG_sy	AHNG
NMI	0.6672	0.2522	0.6832	0.7498	0.6221	**0.7669**
ARI	0.6693	0.1503	0.6971	0.7634	0.6253	**0.7846**

4.3.3. Node clustering

Setup. Different from link prediction, node clustering is an unsupervised learning task. We utilize the node embeddings learned by different methods as features, and apply K-Means as the base clustering model to partition the nodes into four different clusters ($K = 4$). The ground-truth clustering labels are as the same as the four categories used in the classification task above. All clustering experiments are conducted 10 times to report an average performance.

Both *adjusted rand index* (**ARI**) and *normalized mutual information* (**NMI**) are used as evaluation metrics. The value of ARI belongs to $[-1.0, 1.0]$. Random labeling has an ARI close to 0.0, and 1.0 stands for perfect match. NMI is a normalization of the Mutual Information score varying from 0 (no mutual information) to 1 (perfect correlation). The higher the ARI and NMI are, the better the clustering performance is.

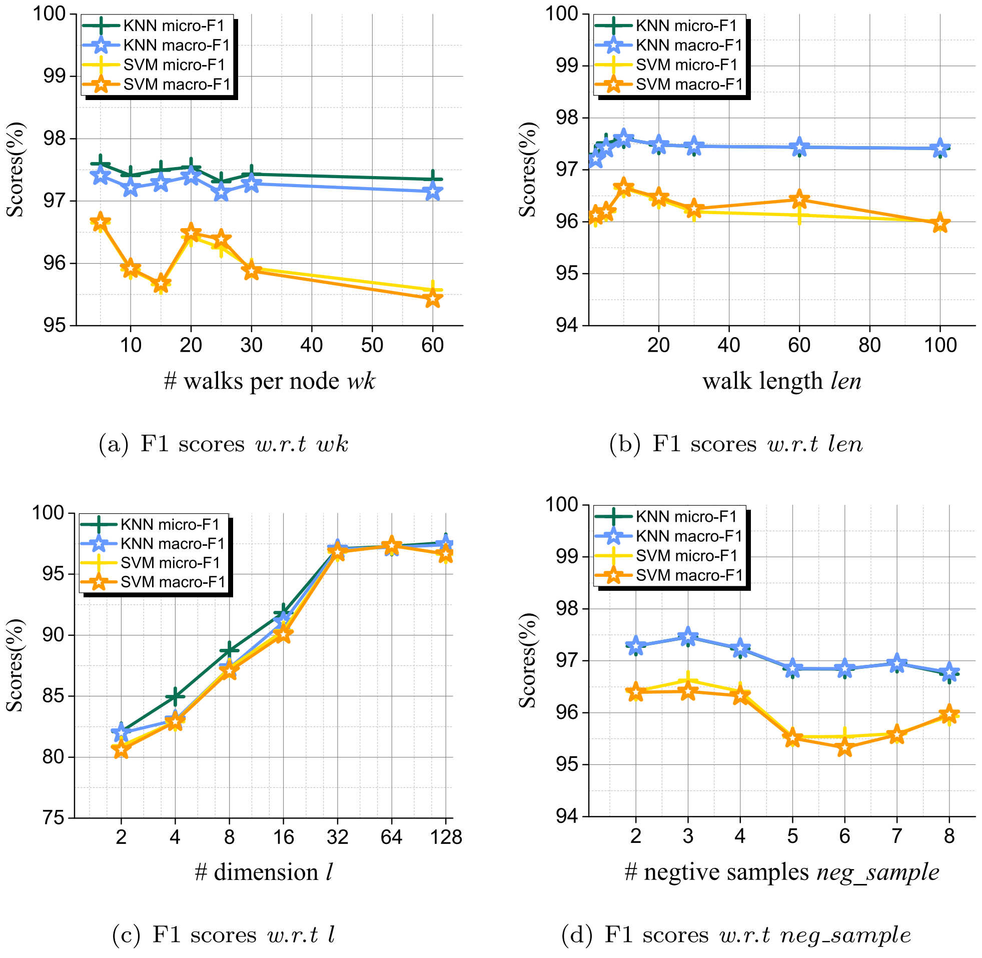
Results and discussion. Table 5 presents the average clustering results. From the results, we can see that AHNG significantly outperforms all other competitors, and obtains 9.97% and 8.37% improvements over node2vec and metapth2vec of NMI values respectively. Similarly, heterogeneous network embedding methods outperform the homogeneous network embedding methods. The insufficient NMI and ARI values of LINE prove the viewpoint presented by its authors: it is more difficult to set the weights in an unsupervised task. Therefore LINE (1st+2nd) is only applied to the scenario of supervised tasks. AHNG performs better than AHNG_sy and AHNvec, indicating the necessity of adopting an asymmetric measure and representing nodes with Gaussian distributions in clustering.

To sum up, the above three tasks imply that the proposed AHNG is an effective and efficient model. Since AHNG_sy performs well in nodes classification and link prediction, we think there could be different metrics and they may perform reasonably well. However, AHNG performs well on all three tasks especially on node clustering, proving the advantage of asymmetric measurements when clustering.

4.4. Parameter sensitivity analysis

We choose four major parameters in AHNG, *i.e.*, *wk, len, l* and neg_sample to conduct the parameter sensitivity experiments. The completeness of preserved network structure is relevant to *wk* and *len*, and the validity of the learned embeddings is relevant to *l*. *wk* is set to 5, 10, 15, 20, 25, 30, 60 in turn; *len* is set to 2, 5, 10, 20, 30, 60, 100 in turn; *l* is set to 2, 4, 8, 16, 32, 64, 128 in turn. Figs. 3–5 show the performance of parameter sensitivity in link prediction, multi-class classification and node clustering, respectively. We set $wk = 10$ and $len = 20$ to obtain a complete network structure when exploring the variations of *l* *w.r.t* the evaluation metrics. We have the following observations from these figures:

- **Link prediction**: Fig. 3(a) and (b) show that AHNG are able to achieve high AUC and AP values when *wk* and *len* are small. Furthermore, we find that when $wk = 5$, the structure of our AHN are rather incomplete. That is to say, the proposed AHNG can achieve good



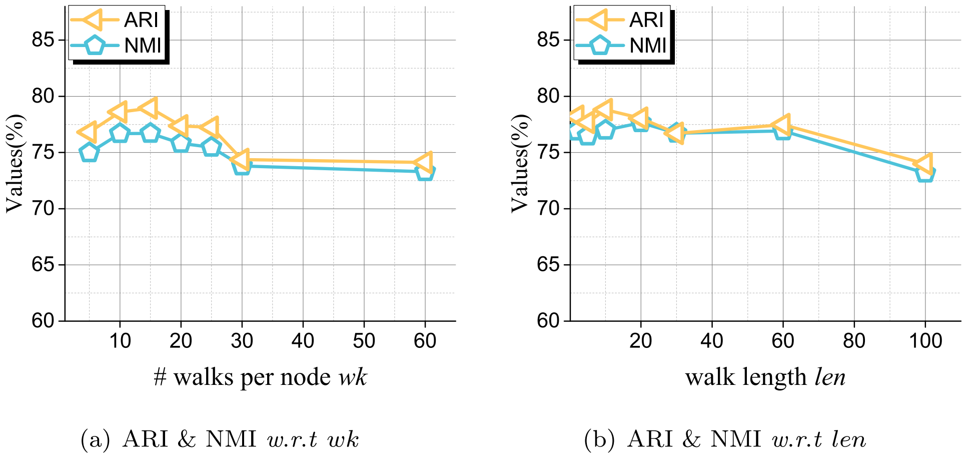
(a) F1 scores *w.r.t wk*

(b) F1 scores *w.r.t len*

(c) F1 scores *w.r.t l*

(d) F1 scores *w.r.t neg_sample*

Fig. 4. Parameter sensitivity in multi-class classification (training ratio $\epsilon = 50\%$).



(a) ARI & NMI *w.r.t wk*

(b) ARI & NMI *w.r.t len*

Fig. 5. Parameter sensitivity in node clustering (clusters $K = 4$).

performance even when the network structure is incomplete. Therefore, fusing the attributes can reduce the requirement for complete network structures, then overcome the defect of traditional random walk based models. In Fig. 3(b), the AUC and AP values descend a little as *len* increases which implies that the redundancy of network structural information may disturb the network representation learning. Fig. 3(c) also achieves stability and robustness in *l*, and is able to learn useful embeddings even with small embedding sizes. Fig. 3(d) indicates that neg_sample contributes to the AUC and AP values. Since *wk* and *len* are small, some positive samples are taken as negative, leading to the decreasing of AUC and AP values.

- **Multi-class classification**: A significant observation from Fig. 4(a) and (b) is that the F1 scores of AHNG are insensitive to the changes of *wk* and *len*. In Fig. 4(c), F1 scores of AHNG shows an increasing trend as *l* increases being consistent with the efficiency of AHNG. In Fig. 4(d), the F1 scores increases firstly and then decreases with the rise of neg_sample. We assume that neg_sample benefits the node representations indeed but the scale of networks limits its contribution.
- **Node clustering**: Fig. 5(a) and (b) show the ARI and NMI values regarding different *wk* and *len*, respectively. It can be seen from Fig. 5(a) that the clustering performance can reach the optimum value over the variation of *wk* and it remains nearly unchanged as *wk* increases. The reason is in accordance with link prediction: AHNG achieves good performance by utilizing node attributes to remedy the incompleteness of network structure. Fig. 5(b) indicates that a descent trend emerges with the extension of *len*. That is to say, the effectiveness of AHNG can be reduced by high redundancy of structural information.

4.5. *Inductiveness*

There are three elements needed to consider during the learning process of AHNG: network structure, semantic information and node attributes. The elegance of introducing node attributes is that for a node which was absent the learning process, its embedding can be obtained individually based on its attribute from the well-trained encoder network. That is, AHNG is an inductive model. We follow the experimental settings in [18] and perform the *inductiveness* experiment on AMiner attributed heterogeneous network: (I) randomly hide 10%, 25%, 50% of nodes from the network respectively; (II) represent the rest of nodes and get the well-trained encoder network; (III) pass the hidden nodes through the well-trained encoder network to collect their embeddings; (IV) evaluate the effectiveness in the link prediction task.

Table 6 shows the results of inductive link prediction on *AMiner*. Even though 50% nodes are hidden in networks, AHNG accomplishes a good AUC and AP values. In other words, AHNG is compatible with those nodes that does not participate in the whole embedding process. In general, the involvement of a new node requires the retraining of neural network based models, leading to enormous space and time costs. Fortunately, by incorporating abundant node attributes, AHNG shows inductiveness. The inductiveness of AHNG facilitates us to embed new nodes, which results in the brilliance of AHNG comparing with existing heterogeneous network embedding methods.

Table 6
Inductive link prediction on *AMiner*.

Metric	Hiding ratio ϵ		
	10%	25%	50%
AUC	0.9395	0.9236	0.9096
AP	0.9188	0.9041	0.8983

5. Conclusion

Most existing network embedding methods are designed for homogeneous networks and ignore the diverse nodes attributes. However, many real-world networks are naturally heterogeneous and are affiliated with abundant attributes. The manifold heterogeneous information including structured and unstructured information throws out a challenge on seamlessly fusing these information and constrains the applicability of conventional network embedding methods in jeopardy. Furthermore, the rich variety of node attributes leads to the uncertain representations of nodes. Taking the above problems into account, we study the problem of node embedding on attributed heterogeneous networks and propose a novel method AHNG in this paper. AHNG encodes the heterogeneous information with a neural network. Different from most traditional methods which represent nodes as vectors, AHNG embeds nodes with Gaussian distributions to capture the uncertainty of node representations. The collaboration of node attributes advances the skip-gram based methods and enables AHNG inductive. We conduct several classic experimental evaluations on the real-world datasets, and the results demonstrate that AHNG not only can embed nodes more accurately but also is much more computationally efficient than its competitors.

Acknowledgments

This work is supported by National Key Research and Development Program of China (2016YFB1000903), National Natural Science Foundation of China (61872287, 61532015, and 61672418), Innovative Research Group of the National Natural Science Foundation of China (61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Project of China Knowledge Centre for Engineering Science and Technology, Science and Technology Planning Project of

Guangdong Province (No. 2017A010101029), the program of China Scholarship Council (201706280198).

References

[1] S. Chang, W. Han, J. Tang, G.-J. Qi, C.C. Aggarwal, T.S. Huang, Heterogeneous network embedding via deep architectures, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 119–128.

[2] H.H. Song, T.W. Cho, V. Dave, Y. Zhang, L. Qiu, Scalable proximity estimation and link prediction in online social networks, in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, ACM, 2009, pp. 322–335.

[3] Y. Jia, Y. Wang, X. Jin, Z. Zhao, X. Cheng, Link inference in dynamic heterogeneous information network: a knapsack-based approach, IEEE Trans. Comput. Soc. Syst. 4 (3) (2017) 80–92.

[4] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community preserving network embedding., in: AAAI, 2017, pp. 203–209.

[5] A. Celikyilmaz, D. Hakkani-Tur, P. Pasupat, R. Sarikaya, Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems, Genre (2010).

[6] S. Guan, X. Jin, Y. Jia, Y. Wang, H. Shen, X. Cheng, Self-learning and embedding based entity alignment, in: Big Knowledge (ICBK), 2017 IEEE International Conference on, IEEE, 2017, pp. 33–40.

[7] T. Man, H. Shen, S. Liu, X. Jin, X. Cheng, Predict anchor links across social networks via an embedding approach., in: IJCAI, 16, 2016, pp. 1823–1829.

[8] L.C. Freeman, Visualizing social networks, J. Soc. Struct. 1 (1) (2000) 4.

[9] A. Theocharidis, S. Van Dongen, A.J. Enright, T.C. Freeman, Network visualization and analysis of gene expression data using biolayout express 3d, Nat. Protoc. 4 (10) (2009) 1535.

[10] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 701–710.

[11] A. Grover, J. Leskovec, node2vec: scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.

[12] S. Cao, W. Lu, Q. Xu, Grarep: learning graph representations with global structural information, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 891–900.

[13] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: a survey, Knowl Based Syst 151 (2018) 78–94.

[14] Y. Sun, J. Han, Mining heterogeneous information networks: a structural analysis approach, ACM SIGKDD Explor. Newsl. 14 (2) (2013) 20–28.

[15] T. La Fond, J. Neville, Randomization tests for distinguishing social influence and homophily effects, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 601–610.

[16] W.L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: methods and applications, arXiv:1709.05584 (2017).

[17] D. Zhang, J. Yin, X. Zhu, C. Zhang, Network representation learning: a survey, IEEE Trans. Big Data (2018).

[18] A. Bojchevski, S. Günnemann, Deep Gaussian embedding of attributed graphs: unsupervised inductive learning via ranking, arXiv:1707.03815 (2017).

[19] X. Huang, J. Li, X. Hu, Label informed attributed network embedding, in: ACM International Conference on Web Search and Data Mining, 2017, pp. 731–739.

[20] C. Shi, Y. Li, J. Zhang, Y. Sun, S.Y. Philip, A survey of heterogeneous information network analysis, IEEE Trans. Knowl. Data Eng. 29 (1) (2017) 17–37.

[21] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 135–144.

[22] T.-y. Fu, W.-C. Lee, Z. Lei, Hin2vec: explore meta-paths in heterogeneous information networks for representation learning, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 1797–1806.

[23] J. Zhang, C. Xia, C. Zhang, L. Cui, Y. Fu, S.Y. Philip, Bl-mne: emerging heterogeneous social network embedding through broad learning with aligned autoencoder, in: Data Mining (ICDM), 2017 IEEE International Conference on, IEEE, 2017, pp. 605–614.

[24] L. Vilnis, A. McCallum, Word representations via gaussian embedding, arXiv:1412.6623 (2014).

[25] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[26] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Advances in Neural Information Processing Systems, 2002, pp. 585–591.

[27] C. Yang, Z. Liu, D. Zhao, M. Sun, E.Y. Chang, Network representation learning with rich text information., in: IJCAI, 2015, pp. 2111–2117.

[28] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[29] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[30] D. Wang, P. Cui, W. Zhu, Structural network embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1225–1234.

[31] T.N. Kipf, M. Welling, Variational graph auto-encoders, arXiv:1611.07308 (2016).

[32] J. Tang, M. Qu, Q. Mei, Pte: predictive text embedding through large-scale heterogeneous text networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1165–1174.

[33] Y. Sun, B. Norick, J. Han, X. Yan, P.S. Yu, X. Yu, Pathselclus: integrating meta-path selection with user-guided object clustering in heterogeneous information networks, ACM Trans. Knowl. Discov. Data 7 (3) (2013) 11.

[34] Y. Goldberg, O. Levy, Word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method, arXiv:1402.3722 (2014).

[35] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 990–998.