



# Semantic-aware heterogeneous information network embedding with incompatible meta-paths

Susu Zheng<sup>1</sup> · Donghai Guan<sup>1</sup> · Weiwei Yuan<sup>1</sup>

Received: 10 September 2020 / Revised: 22 March 2021 / Accepted: 31 May 2021 /

Published online: 29 October 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Heterogeneous information network (HIN) embedding represents heterogeneous nodes as vectors in the low-dimensional space. Meta-path is used to measure the nodes similarity to guide HIN embedding. Existing works assume that different meta-paths share the same semantic space and directly fuse the different meta-paths for node similarity calculation. This ignores the incompatibility of different meta-paths, which cannot reflect the real relationship between nodes. To solve the problems of existing works, a novel **Semantic-Aware HIN Embedding (SAHE)** is proposed to fuse incompatible meta-paths for node similarity measurement. The key idea of the proposed method is to measure the relative similarity relationship on each meta-path in its own semantic space, and aggregate these similarity relationships to obtain the node similarity to calculate HIN embedding. The Kendall tau distance is used to aggregate the different similarity relationship in multiple semantic spaces. The semantic preference is extracted as a constraint to optimize the aggregated similarity matrix. The Kullback-Leibler Divergence (KL Divergence) is used to learn nodes embedding by measuring the node similarity distribution in the embedded space. Experiments on three real HIN datasets verify that the superiority of the proposed model is superior to other state-of-the-art methods on the node classification and the node clustering tasks.

**Keywords** Incompatibility · Meta-path · Network embedding · Heterogeneous information network

---

✉ Weiwei Yuan  
yuanweiwei@nuaa.edu.cn

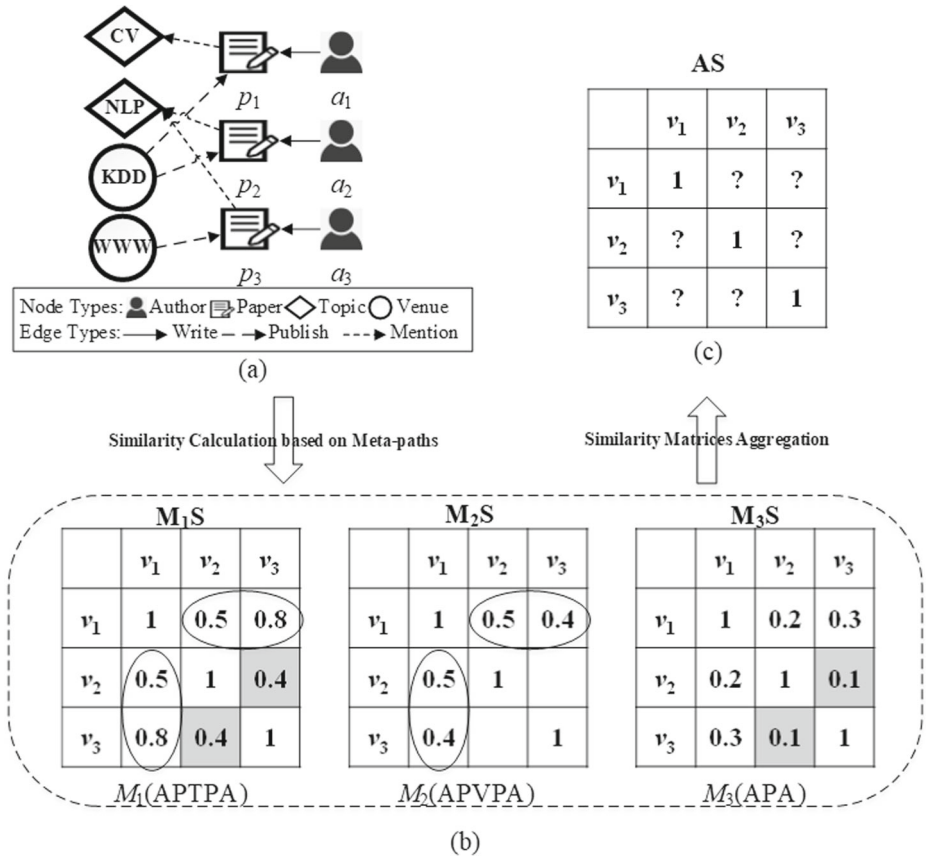
Susu Zheng  
zhengsusu@nuaa.edu.cn

Donghai Guan  
dhguan@nuaa.edu.cn

<sup>1</sup> Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

# 1 Introduction

Heterogeneous information network (HIN) embedding learns the representation of nodes in the low dimensional vector space by preserving its rich semantic information [38]. The key idea of HIN embedding is to measure the similarity between nodes in heterogeneous information network. The more similar the two nodes are, the closer the two nodes are in the mapping space[3, 29]. Meta-path based methods has been widely used in HIN embedding, since meta-paths can obtain rich semantics through multiple connections between nodes in HIN[7, 11, 18]. The basic idea of meta-path is to establish a path between nodes based on semantic relatedness over HINs. The meta-path is used to sample two nodes and compute the semantic similarity of the two nodes to learn node embedding. In HIN, each meta-path can capture its own semantic information [9, 15]. Take Figure 1 as an example, supposing there is a DBLP HIN, author  $a_2$  connected with  $a_3$  following the meta-path  $M_1$  (APTPA), and author  $a_1$  connected with  $a_2$  following the meta-path  $M_2$  (APVPA). Thus, the underlying semantics are, author  $a_2$  and  $a_3$  are related due to the same topic of their papers, while author  $a_1$  and  $a_2$  are related because their papers in the same venue.



**Figure 1** An example of incompatibility problem in HIN embedding, in which (a) gives a DBLP Network schema, (b) lists three similarity matrices based on three meta-paths. The key problem is how to aggregate the similarity matrices with incompatible meta-paths into the matrix of (c)

Existing methods [8, 15] usually assume that different meta-paths share the same semantic space, and ignore the incompatibility problem between different meta-paths. These methods either average the similarity calculated by multiple meta-paths [6], or directly concatenate the embedding of each meta-path to get an overall embedded representation [4]. The similarities between nodes calculated according to different meta-paths may be different — two nodes may be similar according to one meta-path, while not similar according to another meta-path. The distances of nodes in mapping spaces with different meta-paths are also different. This similarity difference results from inconsistent semantics of different meta-paths. The similarity relationships of nodes in different similarity matrices are inconsistent, which leads to the incompatibility problem of meta-paths. Methods ignoring this incompatibility can lead to unreliable results.

Figure 1 presents an example of node embedding in HIN to illustrate incompatibility problem of meta-path. In DBLP HIN, it is assumed that the node similarity matrices  $\mathbf{M}_1\mathbf{S}$ ,  $\mathbf{M}_2\mathbf{S}$  and  $\mathbf{M}_3\mathbf{S}$  are calculated based on three different meta-paths  $M_1$ ,  $M_2$  and  $M_3$ . There are contradictions in the similarity matrices calculated based on different meta-paths. For two pairs of nodes, the similarity relationships they calculated based on different meta-paths are opposite. The similarity between  $(v_1, v_2)$  in  $\mathbf{M}_1\mathbf{S}$  and  $\mathbf{M}_2\mathbf{S}$  are both 0.5, the similarity of  $(v_1, v_2)$  in  $\mathbf{M}_1\mathbf{S}$  is smaller than the similarity of  $(v_1, v_3)$ , but the similarity of  $(v_1, v_2)$  in  $\mathbf{M}_2\mathbf{S}$  is higher than the similarity of  $(v_1, v_3)$ . Because the semantics of different meta-paths are different, the strength of the same similarity value of in different similarity matrices may be different. On the other hand, different similarity values may represent the same similarity strength. The similarity of  $(v_2, v_3)$  in  $\mathbf{M}_1\mathbf{S}$  and  $\mathbf{M}_3\mathbf{S}$  is 0.4 and 0.1 respectively, but the similarity strength is the weakest in their respective similarity matrices. This shows that the similarity calculated according to different meta-paths is not compatible. Averaging the similarity calculated by multiple meta-paths[6] or directly concatenating the node embedding with each meta-path[4] cannot handle the incompatibility problem above, and may affect the node classification performances and node clustering performances. In addition, existing methods use the method of zero-filling on the similarity matrix completion problem, as in the case of  $\mathbf{M}_2\mathbf{S}$ , which does not truly reflect the similarity between nodes.

To solve the problems of the existing works, this paper proposes a novel Semantic-Aware HIN Embedding (SAHE) method, which aggregates the incompatible meta-paths in their own semantic spaces. The key idea of the proposed model is to convert the node similarity into the similarity relationship on each meta-path in its own semantic space, to aggregate multiple meta-path based similarity matrices. Instead of calculating the node embedding by using node similarity directly in the same semantic space, the proposed method can avoid the noise problem caused by the similarity difference, so that incompatibility of different meta-paths can be solved. The SAHE method first use Pathsim method to calculate the meta-path based similarity matrices according to different meta-paths, and then defines the Kendall tau distance by the similarity relationship, to measure the distance between the aggregated similarity matrix and the meta-path based similarity matrices. Next, the semantic preference is extracted as a constraint to optimize the aggregated similarity matrix. Finally, the KL divergence[21] is used to minimize the distribution difference between the aggregated similarity matrix and the node embedding to obtain the node representation.

The main contributions of this work are as follows:

- 1) This work studies the problem of incompatible meta-paths in HIN embedding. The semantic similarity between nodes can be preserved with the consideration of the incompatible problem to improve the embedding performance.

- 2) A novel Semantic-Aware HIN embedding (SAHE) is proposed to embed HIN. We measure the similarity relationships on each meta-path in its own semantic space to solve the incompatibility problem.
- 3) We extract the semantic preference ranking from meta-path based matrices and use it to optimize the aggregated similarity matrix.

The rest of the paper is organized as follows: we first review the related works in Section 2. Then Section 3 introduces the preliminaries. Section 4 presents the details of the SAHE model. Section 5 shows the experimental results and performance analysis. Finally, the conclusion and future works are shown in Section 6.

## 2 Related works

The research of heterogeneous information network (HIN) has recently received widespread attention because it can represent rich semantic information. HIN can describe the network with multiple types of nodes and edges, which is not possible in a homogeneous network [4, 32, 38]. HIN is becoming a new research direction in the field of data mining since HIN integrates more effective information and contains richer semantics in nodes and edges. By mining the information of the HIN, more data in the network can be fully analyzed. How to mine effective information in different types of nodes and edges has become a new challenge. The analysis of heterogeneous information networks is difficult due to its complexity.

HIN embedding aims at learning the representation of heterogeneous nodes in the mapped low-dimensional vector space, and it can improve the performance of data mining, such as classification, clustering and recommendation. However, previous works mainly focus on learning the vector representation of nodes in homogeneous information networks. Deepwalk [17] is inspired by the idea of natural language processing. It treats nodes as words and uses sequences generated by random walks as sentences, and then applies the Skip-gram model to these random walk sequences to learn node embedding. Node2vec [10] is an embedding method similar to Deepwalk, which also uses the Skip-gram model. Unlike Deepwalk, Node2vec has a unique random walk strategy. Node2vec combines depth-first search strategy and width-first search strategy to sample nodes and generate random walk sequences. LINE [28] considers not only the first-order neighbor relationship, but also the second-order neighbor relationship when learning node embedding. That is, even if two nodes are not directly connected, they can establish an indirect connection through their first-order public friends. In addition, it uses negative sampling techniques to optimize the model. GCN [13] learns node embedding through graph convolutional neural network, and GCN cannot learn the weight of each neighbor node during convolution. GAT [30] was proposed to solve the above-mentioned problems of GCN. GAT uses the attention mechanism to learn the weight of each neighbor node and aggregate the neighbor information of the node. The MVC-DNE [35] method based on deep learning performs network embedding by fusing data from two views of structure and attributes. Most of the above methods can learn node embedding of homogeneous information networks, without considering the types of nodes and edges. If these methods are applied directly to high-complexity HIN, the embedding effect may be reduced.

Some exiting methods have been proposed on HIN embedding [2, 6, 12, 24, 25, 36]. Metapath2vec [6] method is proposed as an extension of Deepwalk in HIN embedding. It sets different meta-paths for random walks according to the semantic information in HIN.

Similar to the Deepwalk model, it applies the Skip-gram to the random walk sequences of heterogeneous nodes to obtain the nodes embedding. Unlike most methods that use Skip-gram models to learn random walk sequences, Hin2vec [8] builds a neural network model. This model learns nodes embedding and meta-paths embedding by maximizing the possibility of node relations. Hin2vec model does not rely on artificially set meta-paths, but automatically learns meta-paths from network data. EOE [34] learns the potential node representations of two networks and uses a harmonious embedded matrix to transform the representations of different networks into the same space. These methods ignore the incompatibility of different meta-paths and miss the different semantic relationships of nodes.

Though several works notice the incompatible semantics of HIN, they ignore that different meta-paths have their own semantic spaces and cannot solve the problem of node similarity aggregation among different semantic spaces. A HIN embedding model based on deep learning method called Esim [22] is proposed, which transcribes semantics in HINs by meta-paths. It fuses multiple meta-paths by assigning different weights to different meta-paths. However, it relies heavily on the weight of the manually set meta-path to represent learning, which may not conform to the real network and cannot achieve a better embedding effect. HERec [23] uses a fusion function to fuse the node embeddings learned by different meta-paths into the final node embedding. DMGI [16] is a multi-relational network embedding method that uses an attention mechanism to learn node embedding. Multi-relational network is a special type of heterogeneous network, with a single type of nodes and multiple types of edges. DMGI cannot learn the node embeddings of general heterogeneous information networks (that is, there are multiple types of nodes and edges). DyHNE [33] gives different weights to various meta-paths, so as to evaluate the importance of different meta-path to node embedding. The HAN [32] method uses semantic-level attention and node-level attention to learn the importance of meta-paths and node neighbors at the same time, and aggregates the node representations of multiple meta-paths to obtain the final node representation. These methods of combining weight coefficient cannot solve the incompatibility problem of meta-paths. Since meta-path can learn the similarity between higher-order neighbors and better capture the semantic information of various aspects of HIN, it is necessary to study the incompatibility of meta-paths.

### 3 Preliminaries

**Definition 3.1** (Heterogeneous information network) Let a graph  $G = (V, E)$  represents a heterogeneous information network,  $V$  is the set of all nodes  $v_i$  and  $E$  is the set of all nodes  $e_i$  in the graph  $G$ . Each node  $v_i \in V$  has its own node type and each edge  $e_i \in E$  has its own edge type.  $T$  and  $R$  represent the set of node types and the set of node types respectively. The necessary and sufficient condition for a heterogeneous information network is  $|T| + |R| > 2$ .

Figur 1(a) illustrates a small bibliographic HIN with  $|T| = 4$  and  $|R| = 3$ . It contains four edge types that connect nodes, and the four types of nodes are: author (A), paper (P), venue (V), and topic (T). To simplify the model, it is assumed that each node belongs to a single type.

**Definition 3.2** (Network schema) The network schema is defined as  $T_G = (T, R)$ , which is the meta template for a heterogeneous information network  $G = (V, E)$ . There are node-

to-node type mapping and edge-to-edge type mapping in the network schema, which are  $\varphi: V \rightarrow T$  and  $\psi: E \rightarrow R$ .

Figure 2 shows the schema of the DBLP HIN in Figure 1.

**Definition 3.3** (Meta-path) A meta-path  $M$  is denoted as a path as a path pattern connecting different types of nodes  $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \dots \xrightarrow{R_{l-1}} T_l$ , where  $R_1, R_2, \dots, R_l$  represent edge type of node type  $T_1$  to  $T_l$ .

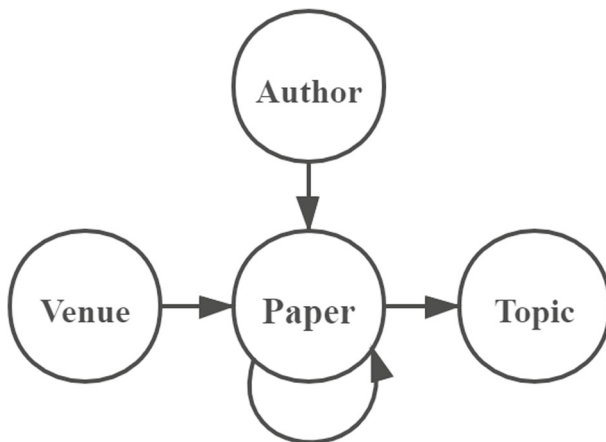
Figure 1(a) shows a meta-path  $A \xrightarrow{\text{write}} P \xrightarrow{\text{publish}} V \xrightarrow{\text{publish}^{-1}} P \xrightarrow{\text{write}^{-1}} A$  in DBLP, where the semantic information represented by this meta-path is authors(A) publish papers (P) in the same venue (V), short as “APVPA”.  $M$  is the set of all meta-path  $M_m$ . A path instance  $m = (v_1 v_2 \dots v_l)$  following the meta-path  $M_m \in M$  represents a path between  $v_1$  and  $v_l$  in network  $G$ , and short as  $m_{v_1 \sim v_l}$ .

**Definition 3.4** (Node Embedding of HIN) In a HIN  $G = (V, E)$ , each node is mapped to a  $d$ -dimensional space  $S^d$  and represented by a vector  $v^d$ , where  $d \ll |V|$ . The purpose of the proposed model is to learn node embedding  $v^d$  in heterogeneous information network.

## 4 The proposed model

To solve the incompatibility of different meta-paths in HIN embedding, this work proposes a SAHE model based on similarity relationship, which is used to aggregate multiple meta-paths to get the node representation. The key idea of SAHE method is convert the node similarity to the similarity relationship on each meta-path in its own semantic space. Figure 3 shows the framework of the SAHE model. The model consists of four main steps: node similarity calculation, node similarity aggregation, semantic preference extraction and embedding learning.

The following subsections detail the steps of this method.



**Figure 2** The HIN schema of DBLP with four node types and four directed edge types

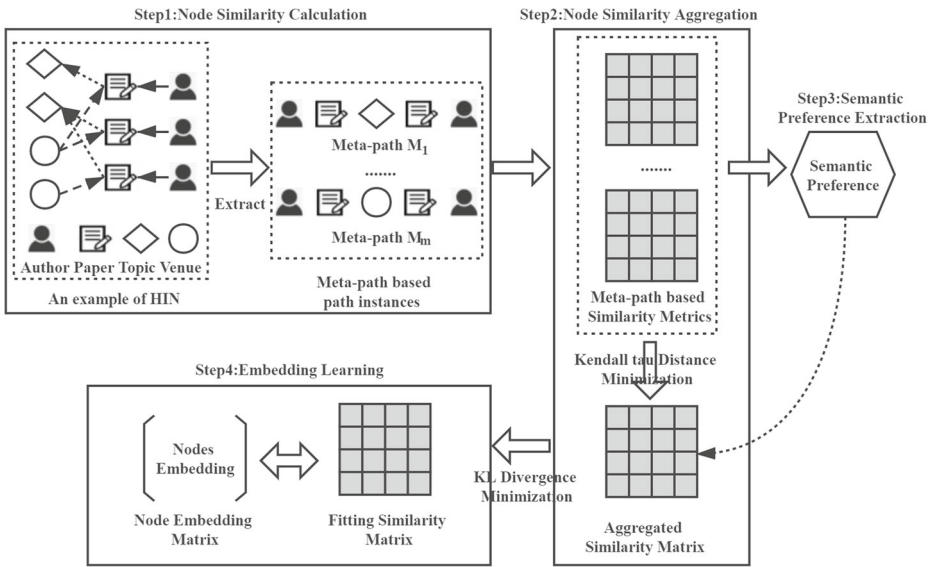


Figure 3 The framework of the proposed SAHE model

### 4.1 Node similarity calculation on single meta-path

The proposed model first calculates the node similarity based on each meta-path. Due to the skewed distribution of the HIN, the similarity method based on random walks is biased toward those nodes with large degrees. It cannot reflect the real network topology. Pathsim is used in our model because it considers all nodes connected through meta-path, thus avoiding the skew distribution problem. PathSim is a method commonly used to measure node similarity in HIN. It uses symmetric meta-paths to extract the connected path between two nodes to measure node similarity, which not only uses related heterogeneous information, but also extracts the rich semantics of nodes and edges [5, 27].

The basic idea of PathSim is that if two nodes in the network are connected by more meta-path instances, the similarity between them is higher. Given a meta-path  $M_m$  and two nodes of the same type  $v_i$  and  $v_j$ , PathSim is defined as follows:

$$ms(v_i, v_j) = \frac{2 \times |m_{v_i \sim v_j}|}{|m_{v_i \sim v_i}| + |m_{v_j \sim v_j}|}, \tag{1}$$

where  $ms(v_i, v_j)$  denotes the similarity between node  $v_i$  and  $v_j$ , simply expressed as  $ms_{ij}$ .  $m_{v_i \sim v_j}$ ,  $m_{v_i \sim v_i}$ ,  $m_{v_j \sim v_j} \in M_m$ ,  $m_{v_i \sim v_j}$ ,  $m_{v_i \sim v_i}$  and  $m_{v_j \sim v_j}$  are the path instances following meta-path  $M_m$  between  $(v_i, v_j)$ ,  $(v_i, v_i)$  and  $(v_j, v_j)$ , respectively.

As shown in (1), given a meta-path  $M_m$ ,  $ms_{ij}$  is defined into two parts: the similarity between nodes depends on the number of path instances between them following meta-path  $M_m$ ; number of path instances from node to itself. For each meta-path  $M_m$ , Pathsim is used to calculate the node similarity and obtain a symmetric matrix  $\mathbf{M}_m \mathbf{S}$  with a diagonal equal to 1:  $ms_{ij} = ms_{ji}$  and  $ms_{ii} = 1$ .

## 4.2 Node similarity aggregation on multiple meta-paths

After calculating the node similarity on single meta-path, our key step is to aggregate the node similarity on these incompatible meta-paths. According to the definition of node similarity in (1), the proposed method determines the meta-path based similarity matrices of  $|M|$  meta-paths. However, the nodes similarity with different meta-paths is not always related to each other: two nodes may be similar according to one meta-path, while not similar according to another meta-path. A Kendall tau distance is defined to aggregate different meta-paths by converting the node similarity to the similarity relationship.

Kendall tau distance is generally used to calculate the reverse ordinal numbers between two sequences [31]. It has many advantages in measuring sequence differences and attracts a lot of attention in many fields. We extend it to convert node similarity into similarity relationship and calculate the consistency of the two matrices. The smaller the Kendall tau distance, the two matrices more consistent. Our method obtains the aggregated similarity matrix by minimizing the Kendall tau distance, which has the smallest difference with all similarity matrices. Therefore, the problem of incompatible meta-paths is solved. Since each meta-path has its own semantics, the similarity relationship in the meta-path similarity matrix is compared by defining the Kendall tau distance. First, the Kendall tau distance between the two similarity matrices is defined as

$$K(\mathbf{M}_m\mathbf{S}, \mathbf{AS}) = \sum_{i=1}^{|V|} K(\mathbf{M}_m\mathbf{S}_i, \mathbf{AS}_i), \quad (2)$$

where  $\mathbf{AS}$  is the aggregated similarity matrix, and the Kendall tau distances of each row vector in the two matrices is defined as follows:

$$K(\mathbf{M}_m\mathbf{S}_i, \mathbf{AS}_i) = \begin{cases} 0, & \text{if } (ms_{ij} > ms_{ik} \wedge as_{ij} > as_{ik}) \\ & \vee (ms_{ij} < ms_{ik} \wedge as_{ij} < as_{ik}) \\ & \vee (ms_{ij} = ms_{ik} \wedge as_{ij} = as_{ik}) \\ 1, & \text{if } (ms_{ij} > ms_{ik} \wedge as_{ij} < as_{ik}) \\ & \vee (ms_{ij} < ms_{ik} \wedge as_{ij} > as_{ik}) \\ & \vee (ms_{ij} > ms_{ik} \wedge as_{ij} = as_{ik}) \\ & \vee (ms_{ij} < ms_{ik} \wedge as_{ij} = as_{ik}) \\ & \vee (ms_{ij} = ms_{ik} \wedge as_{ij} > as_{ik}) \\ & \vee (ms_{ij} = ms_{ik} \wedge as_{ij} < as_{ik}) \end{cases}. \quad (3)$$

If the relative similarity relationship in the two row vectors is the same, the distance remains unchanged. If the similarity relationship in the two row vectors is reverse, then the distance is accumulated. The sum of the distances between the meta-path based similarity matrices and the aggregated similarity matrix is:

$$K(\mathbf{M}_m\mathbf{S}, \mathbf{AS}) = \sum_{m=1}^{|M|} \sum_{i=1}^{|V|} K(\mathbf{M}_m\mathbf{S}_i, \mathbf{AS}_i). \quad (4)$$

Based on the above analysis, the similarity aggregation problem is defined as: given  $|M|$  similarity matrices  $\mathbf{M}_m\mathbf{S}$ , according to  $|M|$  meta-paths, find an aggregated similarity matrix



**AS.** Minimize the distance between  $M_mS$  and  $AS$ , and stochastic gradient descent [1] is used to solve this optimization problem:

$$AS^* = \arg \min K (M_mS, AS)$$

$$= \min \sum_{m=1}^{|M|} \sum_{v=1}^{|V|} K (M_mS_i, AS_i) . \tag{5}$$

In example of Figure 1, the aggregated similarity matrix calculated by SAHE is shown in Figure 4. We can calculate the Kendall tau distance between  $AS$  and  $(M_1S, M_2S, M_3S)$ . Since there is an reverse similarity relationship between row vector  $v_1$  in  $M_2S$  and  $AS$  :  $m_2s_{12} > m_2s_{13}$  while  $as_{12} < as_{13}$  ,  $K (M_2S_1, AS_1)=1$ . No reverse similarity relationship between other row vector in  $M_2S$  and  $AS$ ,  $K (M_2S, AS_1) =1$ . The similarity relationship in the other two similarity matrices  $M_1S$  and  $M_3S$  is the same as the aggregated similarity matrix  $AS$ :  $K (M_1S, AS_1)+ K (M_2S, AS_1)+ K (M_3S, AS_1)=1$ . Only one pair of nodes has the reverse similarity relationship between  $(M_1S, M_2S, M_3S)$ , which means the aggregated similarity matrix  $AS$  maintain high consistency with meta-path based similarity matrices.

### 4.3 Semantic preference extraction

Semantic preferences are extracted from the multiple meta-path based similarity matrices to optimize the aggregated similarity matrix. The majority principle [14], as the name suggests, is the obedience of minority to majority. It is widely used in economics and sociology due to its low complexity and easy to understand. The majority principle represents the majority opinion, and the semantic relationship is determined by obtaining more semantic information of meta-path. The majority principle is extended to the semantic preference ranking: if there are three nodes  $\{v_i, v_j, v_k\}$ , nodes  $(v_i, v_j)$  are more similar in most similarity matrices than node  $(v_i, v_k)$ , then according to the majority preference, the aggregated similarity matrix should satisfy the ranking: similarity of nodes  $(v_i, v_j)$  is higher than the nodes  $(v_i, v_k)$ 's. In other words,  $v_i$  prefers  $v_j$  to  $v_k$  in the aggregated similarity matrices. Formally, if  $|ms_{ij} > ms_{ik}| > |ms_{ij} < ms_{ik}|$ , then  $v_j > v_k$ , where ' $>$ ' denotes the prefer relationship. If nodes  $(v_i, v_j)$  are less similar in more similarity matrices than node  $(v_i, v_k)$ , then according to the majority preference, the aggregated similarity matrix should satisfy the ranking:

**Figure 4** The aggregated similarity matrix  $AS$  calculated by SAHE in the example of Figure 1

	$v_1$	$v_2$	$v_3$
$v_1$	1	0.48	0.67
$v_2$	0.48	1	0.31
$v_3$	0.67	0.31	1

similarity of nodes  $v_i, v_j$  is lower than the nodes  $(v_i, v_k)$ 's. Formally, if  $|ms_{ij} > ms_{ik}| < |ms_{ij} < ms_{ik}|$ , then  $v_k > v_j$ . Particularly,  $|ms_{ij} > ms_{ik}| = |ms_{ij} < ms_{ik}|$  appears in some cases.  $P_j$  and  $N_j$  are defined to solve this problem:

$$P_j = \sum_{k=1}^{|v|} |ms_{ij} > ms_{ik}|, j \neq k. \quad (6)$$

$$N_j = \sum_{k=1}^{|v|} |ms_{ij} < ms_{ik}|, j \neq k. \quad (7)$$

$P_j$  represents the number of nodes whose similarity lower than node  $v_j$  with node  $v_i$ . If  $P_j > P_k$ , then  $v_j > v_k$ , else  $v_k > v_j$ .  $N_j$  represents the number of nodes whose similarity higher than node  $v_j$  with node  $v_i$ . If  $N_j < N_k$ , then  $v_j > v_k$ , else  $v_k > v_j$ . Based on the above semantic preference ranking, for one node  $v_i$ , the similarity relationship ranking between any two other nodes  $(v_j, v_k)$  and  $v_i$  can be determined. This ranking can be used as a constraint to optimize the aggregated matrix. Algorithm 1 describes the process of semantic preference ranking.

In example of Figure 1, three similarity relationship rankings are calculated based on semantic preferences ranking.  $v_1 : v_1 > v_3 > v_2$ ,  $v_2 : v_2 > v_1 > v_3$ ,  $v_3 : v_3 > v_1 > v_2$ . It can be clearly seen that this ranking is consistent with the similarity ranking in Figure 4.

---

**Algorithm 1** Semantic preference ranking for aggregated similarity matrix optimization.

---

**Input:** Node Similarity Matrices  $\mathbf{M}_i\mathbf{S}$

**Output:** Semantic Preference Ranking  $PR_i$

```

for similarity  $(ms_{ij}, ms_{ik})$  in  $\mathbf{M}_i\mathbf{S}$  do
  if  $|ms_{ij} > ms_{ik}| > |ms_{ij} < ms_{ik}|$  then
     $PR_i = PR_i \cup \{v_j > v_k\}$ 
  else if  $|ms_{ij} > ms_{ik}| < |ms_{ij} < ms_{ik}|$  then
     $PR_i = PR_i \cup \{v_k > v_j\}$ 
  else if  $P_j > P_k$  then
     $PR_i = PR_i \cup \{v_j > v_k\}$ 
  else if  $P_j < P_k$  then
     $PR_i = PR_i \cup \{v_k > v_j\}$ 
  else if  $N_j < N_k$  then
     $PR_i = PR_i \cup \{v_j > v_k\}$ 
  else if  $N_j > N_k$  then
     $PR_i = PR_i \cup \{v_k > v_j\}$ 
  end if
end for
return  $PR_i$ 

```

---

#### 4.4 Embedding learning

Given a heterogeneous information network  $G = (V, E)$  and multiple meta-paths  $M^m \in M$ , an aggregated similarity matrix  $\mathbf{AS}$  preserves node similarities obtained by the algorithm 1. The purpose of the SAHE method is to learn the embedding of nodes in HIN, which can

be explained by minimizing the distribution difference between the aggregated similarity matrix  $\mathbf{AS}$  and the node embedding matrix  $\mathbf{H}$ . Since the elements stored in the two matrices are inconsistent, it is not easy to calculate the distribution difference. The activation function sigmoid is used to convert the embedding matrix  $\mathbf{H}$  into a similarity matrix  $\mathbf{HS}$ .

$$hs(v_i, v_j) = \frac{1}{1 + e^{-h_i^T h_j}}, \quad (8)$$

where  $h_i$  and  $h_j$  are the embedding vector of node  $v_i$  and  $v_j$ , respectively. The embedding matrix can be converted to a fitting similarity matrix, denoted as  $\mathbf{HS}$ , where  $hs_{ij} = hs(v_i, v_j)$ .

KL divergence [26] represents the loss of information when fitting a theoretical distribution to a real distribution. Aggregated similarity matrix  $\mathbf{AS}$  is the real distribution while fitting similarity matrix  $\mathbf{HS}$  is the theoretical distribution. Our goal is to minimize the KL divergence between  $\mathbf{AS}$  and  $\mathbf{HS}$ :

$$L = KL(\mathbf{AS} \parallel \mathbf{HS}). \quad (9)$$

Since similarities between any node  $v_i$  and any other node  $v_j$  in the similarity matrix can represent the probability distribution, the loss denotes as:

$$L = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} KL(as_{ij} \parallel hs_{ij}) = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} as_{ij} \log \frac{1}{hs_{ij}}. \quad (10)$$

The optimization objective is to find a node embedding matrix  $\mathbf{H}^*$  by minimizing the following loss function and stochastic gradient descent [1] is used to solve this optimization problem:

$$\mathbf{H}^* = \min \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} as_{ij} \log \frac{1}{hs_{ij}}. \quad (11)$$

The details of the optimization of embedding learning are shown in algorithm 2. A binary classifier is used to distinguish between node samples coming from the aggregated similarity distribution  $\mathbf{AS}$  and a noise distribution. And an auxiliary random variable  $D$  is used for this classification,  $D = 1$  for a node from the aggregated similarity distribution and  $D = 0$  for a sample extract from the noise distribution.  $\sigma$  is the sigmoid function and  $n$  is the number of nodes we extract from noise distribution. In our experiments, we use  $n = 3$ .

**Algorithm 2** Embedding learning.

**Input:** Heterogeneous information network  $G$ , Aggregated similarity matrix AS, embedding dimensionality

**Output:** Node embedding matrix H

```

1: Initialize  $H \in R^{|V| \times d}$ 
2: repeat
3:   Sample a node  $v_i \in G$ 
4:   Sample positive example  $v_j$ 
5:    $h_i, h_j \leftarrow \text{Update}(u, v, 1)$ 
6:   for  $n \leftarrow 1 \dots s$  do
7:     Sample negative example  $v_{j'}$ 
8:      $h_i, h_{j'} \leftarrow \text{Update}(u, v, 0)$ 
9:   end for
10: until converged
11: return H
12: function UPDATE( $u, v, D$ )
13:    $g \leftarrow (D - \sigma(h_u \cdot h_v))$ 
14:    $h_u \leftarrow g * h_u$ 
15:    $h_v \leftarrow g * h_v$ 
16: end function

```

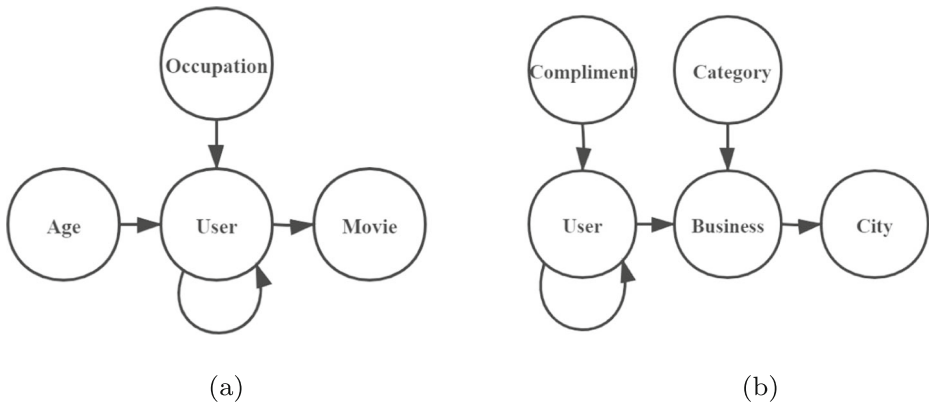
## 5 Experimental results

The embedding performances of the SAHE method are verified on three real-world HIN datasets. First, the experiments compare the node classification and clustering performances of the proposed model with the other latest existing works. Second, the influences of parameters are verified on the performances of the proposed method.

### 5.1 Datasets

Three datasets from different fields, including the DBLP network [19], the Movielens network [37], and the Yelp network [20] are used in the experiments to evaluate the performance of the proposed model.

- **DBLP** is an author-centric dataset in the field of computer science. The DBLP network consists of four types of nodes: author(A), paper(P), topic(T) and venue(V). The edge types include authorship(P-A), topic of paper (P-T), publishing venue (P-V), and the cite relation (P-P). The schema is shown in Figure 2.
- **Movielens** is a movie-centric dataset in movie field consisting of four types of nodes: movies(M), users(U), age(A), and occupation(O), three edge types: user's like (U-M), user's age (U-A) and user's occupation (U-O). Figure 5(a) shows the schema of the Movielens HIN.
- **Yelp** is a business-centric dataset containing user information, business information, and user reviews of businesses. We extract some data include business(B), user(U), city(Ci), compliment(Co) as nodes, and evaluation relation (U-B), user's like(U-Co) and location (B-Ci) as edges. Figure 5(b) shows the schema of the Yelp HIN.



**Figure 5** Schemas of two heterogeneous information networks

The network of the original dataset is sparse and has a skewed distribution. A large number of nodes are in the network and most nodes are with a small degree. It is difficult to analyze the entire network. We therefore randomly sample the original network and set a minimum degree to filter out nodes whose degree are less than the minimum degree. The minimum degree in DBLP, Movielens and Yelp is 3,4,4 respectively. The basic statistics and meta-paths of these three HIN networks are summarized in Table 1. The length of meta-path is the number of relations in the meta-path. For example, in the DBLP network, the meta-path APA can be described using the length-2 meta-path.

## 5.2 Baseline methods

We compare the proposed methods with two types of embedding methods: methods for homogeneous information networks and methods for heterogeneous information networks. These methods have been introduced in the related work in Section 2. Here we present the details of the method in the experiment. For the fairness of the experiment, the dimension  $d$  of the embedding vector space of all methods is set to be 64.

For homogeneous information networks, **Deepwalk** [17], **Node2vec** [10] and **Line** [28] method are used to compare with the proposed method. These three methods are network embedding method based on random walk. To apply these methods directly to HIN embedding, the heterogeneity of nodes will be ignored, and heterogeneous nodes will be regarded as homogeneous nodes. For these three methods, we used the default settings from the original article.

For heterogeneous information networks, meta-path based methods **Metapath2vec** [6], **Esim** [22], **HAN** [32] and non-meta-path based method **Hin2vec** [8] are used to compare

**Table 1** The basic statistics and meta-paths of three HIN datasets

Datasets	$ V $	$ E $	#labels	Meta-paths
DBLP	9150	45165	4	APA, APTPA, APVPA
Movielens	1682	28270	5	MUMUM, MUAUM, MUOUM
Yelp	7285	35191	10	BUB, BCiB, BUCoB

with the proposed method. In order to verify the ability of the proposed method to solve the problem of meta-path incompatibility, the Kendall tau distance is deleted and the average similarity calculated by different meta-paths is used as a variant of SAHE, called  $\text{SAHE}_{\text{avg}}$ , and the variant is compared with SAHE. For each dataset, the meta-paths used by Metapath2vec, Esim and HAN in the experiment are the same as SAHE. Since different datasets have different weight settings, various weights are assigned to Esim and the best performing one are chosen on different datasets. The parameter settings of the Hin2vec method are the same as in the original article.

### 5.3 Node classification

Node classification classifies the nodes into different categories, where the input of the node classifier are the learned embeddings. In the experiments, we mainly focus on the central node in the datasets for classification. Author, movie and business are used as the central node for classification on the DBLP, Movielens and Yelp datasets respectively. A group of nodes are randomly selected as the labeled training nodes, and the remaining nodes are used as test nodes. The embeddings of the training nodes are used as input and a logistic regression classifier is trained to predict the most likely labels for the test nodes. The network datasets used in the experiments are divided into training and test sets, and the ratio of the training set gradually increases from 5% to 50%. Each experiment is 10 repeated trials and the average Micro-F1 and Macro-F1 scores are recorded. The experiment results are shown in Tables 2 and 3.

As is shown in Tables 2 and 3, when the ratio of training data increases, the performance is better in general which explains that the accuracy is positive related to training ratio. (1) The results show that the node classification performance of the SAHE method is always better than the baselines. The performance improvement obtained by SAHE on the best benchmark (HAN) is about 2%-8%. When the training ratio is less than 20%, the performance of the SAHE method is much higher than the baseline methods. This means that the proposed method requires only a small amount of training data to obtain effective embedding results. (2) The performance of DeepWalk, Node2vec and Line is lower than Metapath2vec, Esim, Hin2vec, HAN and SAHE. Thus, the methods for homogeneous network embedding are significantly worse than heterogeneous network embedding methods in node classification task. Because the homogeneous network embedding methods ignore the heterogeneity of nodes. (3) By comparing the performance of heterogeneous information network baselines (Metapath2vec, Esim, HAN, Hin2vec), HAN is the best in baselines because the fusion embedding learned through the attention mechanism can improve the embedding ability. Metapath2vec is the least performance method among the heterogeneous network comparison methods, because this method only extracts the semantic information of a meta-path. (4) The node classification performances of SAHE method are superior than that of  $\text{SAHE}_{\text{avg}}$  method in all cases. This shows that the proposed method improves the performance of network embedding by solving the problem of meta-path incompatibility.

### 5.4 Link prediction

We conduct experiments on three datasets to compare the link prediction performance of node embeddings learned by SAHE and the baseline methods. In this task, we predict the citation relationship (P-P) between papers in DBLP, and the friend relationship (U-U) between users in Movielens and Yelp. We first randomly divide the edges of the dataset into a training set and a test set. Here, we set 75% of the edges as the training set and the

**Table 2** Micro-F1 scores of node classification task

Datasets	Methods	Training ratio						
		5%	10%	15%	20%	30%	40%	50%
DBLP	Node2vec	0.5945	0.6786	0.7147	0.7496	0.7648	0.7913	0.7952
	Line	0.5842	0.6789	0.6989	0.7189	0.7493	0.7543	0.7611
	Metapath2vec	0.6024	0.7052	0.7103	0.7457	0.7741	0.7916	0.8197
	Esim	0.6693	0.7247	0.7372	0.7539	0.7647	0.7857	0.8011
	Hin2vec	0.6541	0.7034	0.7343	0.7784	0.8091	0.8204	0.8219
	HAN	0.6752	0.7137	0.7531	0.7846	0.8105	0.8326	0.8565
	SAHE <sub>avg</sub>	0.6674	0.7316	0.7371	0.7564	0.7681	0.7817	0.8059
	SAHE	<b>0.7561</b>	<b>0.7995</b>	<b>0.8053</b>	<b>0.8265</b>	<b>0.8419</b>	<b>0.8671</b>	<b>0.8809</b>
Movielens	Node2vec	0.5585	0.5881	0.5978	0.6063	0.644	0.6565	0.6661
	Line	0.5321	0.5685	0.5968	0.6187	0.6302	0.6474	0.6628
	Metapath2vec	0.573	0.5887	0.5914	0.6079	0.6268	0.6594	0.6942
	Esim	0.5752	0.6235	0.6314	0.6308	0.6537	0.6604	0.6829
	Hin2vec	0.5627	0.6119	0.6145	0.6305	0.6637	0.6875	0.6962
	HAN	0.5816	0.6382	0.6428	0.6558	0.672	0.6903	0.7256
	SAHE <sub>avg</sub>	0.5715	0.6334	0.6341	0.6462	0.6651	0.6708	0.7011
	SAHE	<b>0.6117</b>	<b>0.6525</b>	<b>0.6576</b>	<b>0.6732</b>	<b>0.6943</b>	<b>0.7259</b>	<b>0.7481</b>
Yelp	Node2vec	0.2916	0.3426	0.3509	0.3554	0.3802	0.388	0.3923
	Line	0.2745	0.3307	0.3318	0.3422	0.3584	0.3699	0.3836
	Metapath2vec	0.3189	0.3548	0.3519	0.3527	0.3744	0.3809	0.3914
	Esim	0.3238	0.3604	0.3639	0.3664	0.3847	0.3816	0.3938
	Hin2vec	0.3096	0.3628	0.3661	0.3885	0.404	0.4121	0.4325
	HAN	0.3211	0.3562	0.3818	0.4323	0.4104	0.4214	0.4386
	SAHE <sub>avg</sub>	0.3265	0.3643	0.3773	0.3784	0.3891	0.3922	0.4105
	SAHE	<b>0.3528</b>	<b>0.3817</b>	<b>0.4143</b>	<b>0.4279</b>	<b>0.4307</b>	<b>0.4333</b>	<b>0.4521</b>

remaining edges as the test set. Then, the node embedding is used as a feature of the classifier to train the training set and evaluate the link prediction performance on the test set. The AUC and AP scores are used as indicators to reflect the quality of node embedding. The higher the values of AUC and AP are, the better the link prediction performance is. Table 4 shows the performances of link prediction of various methods on three datasets and the best performance is highlighted in bold.

Similar as the node classification task, the SAHE method significantly perform better than all baseline methods on three datasets. Compared with the baselines, the AUC and AP scores of link prediction of SAHE increased by about 2%- 22% in DBLP, 3%- 11% in Movielens, and 4%- 12% in Yelp. It shows that the proposed method can improve the performance of network embedding by capturing heterogeneous semantic information. And the link prediction performances of SAHE method are superior than that of SAHE<sub>avg</sub> method in all cases. This results verify the ability of the proposed method to solve the problem of meta-path incompatibility on link prediction task. Different from the node classification task, on the DBLP data set, node2vec performs better than the traditional heterogeneous method metapath2vec. This may be due to the fact that the node2vec method obtains more

**Table 3** Macro-F1 scores of node classification task

Datasets	Methods	Training ratio						
		5%	10%	15%	20%	30%	40%	50%
DBLP	Node2vec	0.5638	0.6463	0.6864	0.7162	0.7415	0.7686	0.7776
	Line	0.5584	0.6617	0.6774	0.6961	0.7302	0.73	0.7399
	Metapath2vec	0.5443	0.6861	0.6874	0.7315	0.7588	0.7868	0.8027
	Esim	0.5458	0.7239	0.7221	0.7436	0.7516	0.7775	0.7828
	Hin2vec	0.6097	0.6768	0.7326	0.7629	0.7894	0.8018	0.8143
	HAN	0.6815	0.6985	0.7348	0.7764	0.8069	0.8242	0.8317
	SAHE <sub>avg</sub>	0.6562	0.7204	0.7283	0.7345	0.7608	0.7854	0.8009
	SAHE	<b>0.7503</b>	<b>0.7638</b>	<b>0.7946</b>	<b>0.8112</b>	<b>0.8235</b>	<b>0.8348</b>	<b>0.8693</b>
Movielens	Node2vec	0.5627	0.6076	0.6383	0.6425	0.6779	0.6931	0.6796
	Line	0.5373	0.5532	0.6076	0.6362	0.6497	0.6502	0.6722
	Metapath2vec	0.5542	0.6129	0.6438	0.6746	0.6906	0.7281	0.7338
	Esim	0.5389	0.6217	0.6509	0.6629	0.683	0.7025	0.7271
	Hin2vec	0.5691	0.6257	0.6337	0.6656	0.6998	0.7137	0.7411
	HAN	0.5754	0.6375	0.6812	0.6856	0.7088	0.7423	0.7612
	SAHE <sub>avg</sub>	0.5485	0.6272	0.6514	0.6713	0.6777	0.7174	0.7275
	SAHE	<b>0.6946</b>	<b>0.6741</b>	<b>0.6954</b>	<b>0.7107</b>	<b>0.7345</b>	<b>0.7539</b>	<b>0.7737</b>
Yelp	Node2vec	0.2807	0.3248	0.3288	0.3365	0.3551	0.3632	0.3829
	Line	0.2512	0.3196	0.3089	0.3195	0.3313	0.3548	0.3603
	Metapath2vec	0.2904	0.321	0.3278	0.3368	0.3315	0.3523	0.3759
	Esim	0.3016	0.3348	0.3407	0.3422	0.3579	0.3603	0.3765
	Hin2vec	0.2868	0.3484	0.3538	0.3608	0.3813	0.4044	0.4189
	HAN	0.3124	0.3615	0.3603	0.3874	0.4012	0.4153	0.4228
	SAHE <sub>avg</sub>	0.3085	0.3385	0.3521	0.3579	0.3518	0.3604	0.3625
	SAHE	<b>0.3414</b>	<b>0.3743</b>	<b>0.3854</b>	<b>0.4019</b>	<b>0.4164</b>	<b>0.4207</b>	<b>0.4332</b>

structural and semantic information in the DBLP data set compared with the single metapath metapath2vec method.

## 5.5 Node clustering

We also conduct experiments on three datasets to verify the performances of the proposed SAHE method on node clustering task. Node clustering regroups nodes into clusters according to the similarity between nodes. The labeled node field of the clustering task is the same as the node classification task, such as the author type in DBLP, the movie type in Movielens and the business type in YELP. Specifically, the nodes embedding is treated as nodes feature while the k-means algorithm is used to cluster the nodes. The real group consists of nodes with the same label and is compared with the group obtained by the clustering task. The performance of the clustering task is evaluated with Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI). The average NMI and ARI scores for the 10 repeated trials are shown in Table 5.



**Table 4** AUC and AP scores of the link prediction task

Methods	DBLP		Movielens		Yelp	
	AUC	AP	AUC	AP	AUC	AP
Node2vec	0.5879	0.6213	0.6111	0.609	0.6048	0.6146
Line	0.5654	0.5541	0.5992	0.5865	0.5849	0.6295
Metapath2vec	0.5831	0.5807	0.6213	0.6294	0.6403	0.6396
Esim	0.6031	0.6119	0.6232	0.6206	0.6168	0.6152
Hin2vec	0.7514	0.7432	0.6507	0.6248	0.6264	0.6363
HAN	0.7892	0.7853	0.6682	0.6577	0.6667	0.6722
SAHE <sub>avg</sub>	0.6025	0.6601	0.6328	0.6585	0.6684	0.6517
SAHE	<b>0.8062</b>	<b>0.7963</b>	<b>0.6805</b>	<b>0.6928</b>	<b>0.7075</b>	<b>0.8194</b>

The experimental results show that the node clustering performances of the proposed SAHE method is consistently better than other baselines on different datasets. Compared with the baselines, the NMI and ARI scores of node clustering of SAHE increased by about 3%- 9% in DBLP, 1%- 9% in Movielens, and 1%- 7% in Yelp.

Similar to the node classification task, the clustering effect of HAN is still the second best. The methods for homogeneous network embedding are significantly worse than heterogeneous network embedding method in most cases. And the node clustering performances of SAHE method are superior than that of SAHE<sub>avg</sub> method in all cases. We observe that the performance of Hin2vec is not good in clustering experiments, sometimes even worse than the embedding method for homogeneous information network, this shows that the performance of HIN is unstable.

Comparing the performance of methods for heterogeneous information network, HAN, metapath2vec and Esim achieves higher NMI and ARI than Hin2vec on the DBLP, Movielens and Yelp datasets. This result shows that the method of manually setting the meta-path

**Table 5** NMI and ARI scores of the node clustering task

Methods	DBLP		Movielens		Yelp	
	NMI	ARI	NMI	ARI	NMI	ARI
Node2vec	0.5028	0.4944	0.0922	0.0675	0.158	0.1372
Line	0.4771	0.4634	0.1334	0.0913	0.1362	0.1268
Metapath2vec	0.5146	0.4979	0.0962	0.0689	0.1321	0.1153
Esim	0.5381	0.5268	0.0776	0.0607	0.1789	0.1554
Hin2vec	0.4912	0.4765	0.055	0.0253	0.1637	0.1501
HAN	0.5215	0.5307	0.115	0.1026	0.1789	0.1653
SAHE <sub>avg</sub>	0.5148	0.5011	0.1056	0.0862	0.1653	0.1670
SAHE	<b>0.5689</b>	<b>0.5571</b>	<b>0.1443</b>	<b>0.1189</b>	<b>0.2065</b>	<b>0.1784</b>

**Table 6** Influence of Node embedding dimension  $d$  on SAHE

Datasets	Metrics	$d=16$	$d=32$	$d=64$	$d=128$	$d=256$
DBLP	NMI	0.5517	0.5586	<b>0.5689</b>	0.5416	0.5403
	ARI	0.5383	0.5328	<b>0.5571</b>	0.5529	0.5217
Movielens	NMI	0.1279	0.1323	0.1443	<b>0.1537</b>	0.1205
	ARI	0.0969	0.1038	0.1189	<b>0.1266</b>	0.0884
Yelp	NMI	0.1889	0.1925	<b>0.2065</b>	0.1917	0.1908
	ARI	0.1661	0.1679	<b>0.1784</b>	0.1645	0.1633

(Metapath2vec, Esim) is better than the method of automatically selecting the meta-path (such as Hin2vec). It may be related to the fact that the manually set meta-path is more consistent with the semantic relationship in the real network. In other words, meta-path selection strategies require expert prior knowledge, as they are not only related to the direct connection of nodes in the network.

## 5.6 Parameter analysis

The experiments evaluate the node clustering task of the SAHE model with different values of node embedding dimension  $d$  and different numbers of meta-paths  $|M|$ . Other parameters, such as those in the optimization process, are not analyzed because they are not relevant to the contributions of this paper. In order to study the influence of  $d$ , we observe the clustering performance by increasing  $d$  from 16 to 256. The results of the node classification task are similar to the clustering results. Tables 6 and 7 only show the results of node clustering.

We observe that when  $d$  is less than 64, a smaller  $d$  leads to worse performance, which because the small size is not enough to capture more valid information for the node. The clustering performance is best at  $d = 64$  or  $d = 128$ , and then decreases with increases the of  $d$ , this may be due to over-fitting caused by over-dimension. The experiment at  $d = 64$  showed good performance on DBLP and Yelp, while  $d = 128$  performed well on Movielens. In different networks, different dimensions are required.

Table 7 shows that as the number of meta-paths increases, the clustering performance improves. This is reasonable since multiple meta-paths can obtain richer information from heterogeneous information networks. It further verifies that the proposed SAHE model can solve the incompatibility of different meta-paths.

**Table 7** Influence of meta-paths number  $|M|$  on SAHE

Datasets	Metrics	$ M =1$	$ M =2$	$ M =3$
DBLP	NMI	0.5348	0.5577	<b>0.5689</b>
	ARI	0.5178	0.5449	<b>0.5571</b>
Movielens	NMI	0.1193	0.1334	<b>0.1443</b>
	ARI	0.0972	0.1099	<b>0.1189</b>
Yelp	NMI	0.1726	0.1932	<b>0.2065</b>
	ARI	0.1412	0.1667	<b>0.1784</b>

## 6 Conclusions and future works

In this paper, a novel method named Semantic-Aware HIN Embedding (SAHE) is proposed to learn nodes embedding in heterogeneous information networks. Particularly, we solve the incompatibility problem of different meta-paths existing in real-world HINs. The core of the SAHE method is to convert the node similarity to the similarity relationships on each meta-path in its own semantic space. Since the similarity matrix calculated from the same meta-paths has no incompatibility problems, minimizing the distance between the aggregated similarity matrix and the meta-path based similarity matrices can solve the incompatibility problem. In addition, an innovative semantic preferences ranking is proposed as a constraint to optimize the aggregated similarity matrix. Extensive experimental evaluations confirm that the SAHE model can capture more semantic information of complex HINs, and overall exceeds the baseline.

To improve the proposed method, future works will focus on the following aspects. First, the proposed method selects meta-path manually, we plan to upgrade it to select the most influential meta-paths automatically and ensure that these meta-paths can preserve rich semantic information in HIN. Second, the proposed model will be extended to solve the problem of node representation on more complex networks, such as dynamic heterogeneous networks, attributed heterogeneous networks and large-scale heterogeneous networks.

**Acknowledgements** This research is supported by National Natural Science Foundation of China (No. 61672284), Natural Science Foundation of Jiangsu Province (No. BK20171418), Postdoctoral Science Foundation of China (No. 2016M591841) and Jiangsu Planned Projects for Postdoctoral Research Funds (No. 1601225C).

## References

1. Amiri, M.M., Gündüz, D.: Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air. *IEEE Trans. Signal Process.* **68**, 2155–2169 (2020). <https://doi.org/10.1109/TSP.2020.2981904>
2. Cen, Y., Zou, X., Zhang, J., Yang, H., Zhou, J., Tang, J.: Representation learning for attributed multiplex heterogeneous network. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1358–1368 (2019). <https://doi.org/10.1145/3292500.3330964>
3. Chen, H., Yin, H., Chen, T., Wang, W., Li, X., Hu, X.: Social boosted recommendation with folded bipartite network embedding. *IEEE Transactions on Knowledge and Data Engineering.* <https://doi.org/10.1109/TKDE.2020.2982878> (2020)
4. Chen, X., Yu, G., Wang, J., Domeniconi, C., Li, Z., Zhang, X.: Activehne: Active heterogeneous network embedding. *arXiv:1905.05659* (2019)
5. Do, P., Pham, P.: Dw-pathsim: a distributed computing model for topic-driven weighted meta-path-based similarity measure in a large-scale content-based heterogeneous information network. *Journal of Information and Telecommunication* **3**(1), 19–38 (2019). <https://doi.org/10.1080/24751839.2018.1516714>
6. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144 (2017). <https://doi.org/10.1145/3097983.3098036>
7. Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., Li, Y.: Metapath-guided heterogeneous graph neural network for intent recommendation. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2478–2486 (2019). <https://doi.org/10.1145/3292500.3330673>
8. Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1797–1806 (2017). <https://doi.org/10.1145/3132847.3132953>

9. Fu, Y., Xiong, Y., Philip, S.Y., Tao, T., Zhu, Y.: Metapath enhanced graph attention encoder for hins representation learning. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 1103–1110, IEEE (2019). <https://doi.org/10.1109/BigData47090.2019.9006097>
10. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855–864 (2016). <https://doi.org/10.1145/2939672.2939754>
11. Hussein, R., Yang, D., Cudré-Mauroux, P.: Are meta-paths necessary? revisiting heterogeneous graph embeddings. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 437–446 (2018). <https://doi.org/10.1145/3269206.3271777>
12. Jiang, Z., Gao, Z., Lan, J., Yang, H., Lu, Y., Liu, X.: Task-oriented genetic activation for large-scale complex heterogeneous graph embedding. In: Proceedings of The Web Conference 2020, pp. 1581–1591 (2020). <https://doi.org/10.1145/3366423.3380230>
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv:1609.02907 (2016)
14. Kondratev, A.Y., Nesterov, A.S.: Measuring majority power and veto power of voting rules. Public Choice **183**(1), 187–210 (2020). <https://doi.org/10.1007/s11127-019-00697-1>
15. Molaie, S., Zare, H., Veisi, H.: Deep learning approach on information diffusion in heterogeneous networks. Knowl.-Based Syst. **189**, 105153 (2020). <https://doi.org/10.1016/j.knsys.2019.105153>
16. Park, C., Kim, D., Han, J., Yu, H.: Unsupervised attributed multiplex network embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 5371–5378 (2020)
17. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710 (2014). <https://doi.org/10.1145/2623330.2623732>
18. Pham, P., Do, P.: W-metapath2vec: The topic-driven meta-path-based model for large-scaled content-based heterogeneous information network representation learning. Expert Syst. Appl. **123**, 328–344 (2019). <https://doi.org/10.1016/j.eswa.2019.01.015>
19. Pradhan, T., Pal, S.: Cnaver: A content and network-based academic venue recommender system. Knowl.-Based Syst. **189**, 105092 (2020). <https://doi.org/10.1016/j.knsys.2019.105092>
20. Rahmani, H.A., Aliannejadi, M., Mirzaei Zadeh, R., Baratchi, M., Afsharchi, M., Crestani, F.: Category-aware location embedding for point-of-interest recommendation. In: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 173–176 (2019). <https://doi.org/10.1145/3341981.3344240>
21. Rioux, G., Scarvelis, C., Choksi, R., Hoheisel, T., Marechal, P.: Blind deblurring of barcodes via kullback-leibler divergence. IEEE Trans. Pattern Anal. Mach. Intell. <https://doi.org/10.1109/TPAMI.2019.2927311> (2019)
22. Shang, J., Qu, M., Liu, J., Kaplan, L.M., Han, J., Peng, J.: Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. arXiv:1610.09769 (2016)
23. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. **31**(2), 357–370 (2018)
24. Shi, Y., Gui, H., Zhu, Q., Kaplan, L., Han, J.: Aspem: Embedding learning by aspects in heterogeneous information networks. In: Proceedings of the 2018 SIAM International Conference on Data Mining, pp. 144–152. SIAM (2018). <https://doi.org/10.1137/1.9781611975321.16>
25. Shi, Y., Zhu, Q., Guo, F., Zhang, C., Han, J.: Easing embedding learning by comprehensive transcription of heterogeneous information networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2190–2199 (2018). <https://doi.org/10.1145/3219819.3220006>
26. Song, Y., Deng, Y.: Divergence measure of belief function and its application in data fusion. IEEE Access **7**, 107465–107472 (2019)
27. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment **4**(11), 992–1003 (2011)
28. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077 (2015). <https://doi.org/10.1145/2736277.2741093>
29. Tu, K., Ma, J., Cui, P., Pei, J., Zhu, W.: Autone: Hyperparameter optimization for massive network embedding. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 216–225 (2019). <https://doi.org/10.1145/3292500.3330848>
30. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv:1710.10903 (2017)

31. Vieira, C., Ribeiro, F., Vaz de Melo, P.O., Benevenuto, F., Zagheni, E.: Using facebook data to measure cultural distance between countries: The case of brazilian cuisine. In: Proceedings of The Web Conference 2020, pp. 3091–3097 (2020). <https://doi.org/10.1145/3366423.3380082>
32. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The World Wide Web Conference, pp. 2022–2032 (2019). <https://doi.org/10.1145/3308558.3313562>
33. Wang, X., Lu, Y., Shi, C., Wang, R., Cui, P., Mou, S.: Dynamic heterogeneous information network embedding with meta-path based proximity. *IEEE Transactions on Knowledge and Data Engineering* (2020)
34. Xu, L., Wei, X., Cao, J., Yu, P.S.: Embedding of embedding (eoe) joint embedding for coupled heterogeneous networks. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 741–749 (2017). <https://doi.org/10.1145/3018661.3018723>
35. Yang, D., Wang, S., Li, C., Zhang, X., Li, Z.: From properties to links: Deep network embedding on incomplete graphs. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 367–376 (2017)
36. Yang, L., Xiao, Z., Jiang, W., Wei, Y., Hu, Y., Wang, H.: Dynamic heterogeneous graph embedding using hierarchical attentions. In: European Conference on Information Retrieval, pp. 425–432. Springer (2020)
37. Yu, Y., Wang, Z., Yuan, B.: An input-aware factorization machine for sparse prediction. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 1466–1472. AAAI Press (2019)
38. Zhang, C., Wang, G., Yu, B., Xie, Y., Pan, K.: Proximity-aware heterogeneous information network embedding. *Knowledge-Based Systems*, 105468. <https://doi.org/10.1016/j.knosys.2019.105468> (2020)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.