HINE: Heterogeneous Information Network Embedding

Yuxin Chen[1(✉)] and Chenguang Wang[2]

[1] Key Laboratory of High Confidence Software Technologies (Ministry of Education), EECS, Peking University, Beijing, China
chen.yuxin@pku.edu.cn

[2] IBM Research Almaden, San Jose, CA, USA
chenguang.wang@ibm.com

Abstract. Network embedding has shown its effectiveness in embedding homogeneous networks. Compared with homogeneous networks, heterogeneous information networks (HINs) contain semantic information from multi-typed entities and relations, and are shown to be a more effective model for real world data. The existing network embedding methods fail to explicitly capture the semantics in HINs. In this paper, we propose an HIN embedding model (HINE), which consists of local and global semantic embedding. Local semantic embedding aims to incorporate entity type information via embedding the local structures and types of the entities in a supervised way. Global semantic embedding leverages multi-hop relation types among entities to propagate the global semantics via a Markov Random Field (MRF) to impact the embedding vectors. By doing so, *HINE* is capable to capture both local and global semantic information in the embedding vectors. Experimental results show that HINE significantly outperforms state-of-the-art methods.

Keywords: Heterogeneous information network · Network embedding · Semantic embedding

1 Introduction

Network embedding has recently been proposed as a new representation of networks. The representation consists of low-dimensional vectors carrying the most important information about the network. It thus benefits lots of network-based applications, such as visualization [18], node classification [3], as well as link predication [15] and web search [32]. The common factor shared by various network embedding approaches (e.g., DeepWalk [23], LINE [27] and Node2vec [11]) is: *the network structure embedding*.

We are grateful to Tengjiao Wang for invaluable guidance, support and contribution in regard to this research and resulting paper. This research is supported by the Natural Science Foundation of China (Grant No. 61572043), and the National Key Research and Development Program (Grant No. 2016YFB1000704).

S. Candan et al. (Eds.): DASFAA 2017, Part I, LNCS 10177, pp. 180–195, 2017.
DOI: 10.1007/978-3-319-55753-3_12

The existing network embedding approaches are mainly focusing on leveraging structural information to embed homogeneous networks. Compared to homogeneous networks, heterogeneous information networks (HINs) have been demonstrated as a more efficient way to model real world data for many applications, such as similarity search [26,34,38], classification [35,43], clustering [33] et al. The reason is that HINs are graphs consisting of multi-typed entities and relations. The various type information carries rich semantics about networks other than the basic structural information. It is thus of great need to study HIN embedding.

It is non-trivial to apply the existing homogeneous network embedding methods to HINs, due to the following two reasons.

Incorrect embedding results. Only considering structural information in HIN embedding will not only lose the semantics provided by HINs, but also lead to incorrect embedding vectors. For example, two entities "New York City" and "The New York Times" will probably have dissimilar embedding vectors by only considering the structural information, since the near neighbors (i.e., local structure) of two entities are different. However HINs could provide relation type *publishedIn* (as global information) between the two entities, thus the embedding vectors of two entities should be similar.

Lack of user-guided semantics. HIN based approaches often require user-guided semantics [20]. For example, in similarity search [42], users are often asked to provide the example entities which are similar to the target entity. However the low-dimensional vectors generated by the existing embedding methods are distributed representations, thus lack of semantic interpretation. We expect the HIN embedding vectors could still preserve the semantics, to facilitate various HIN based applications. Therefore, we are considering an HIN embedding approach to incorporate the HIN semantics in the embedding model and preserve the semantics in the embedding vectors.

In this paper, we propose an HIN embedding (*HINE*) model to embed an HIN into a low-dimensional semantic vector space. In particular, HINE contains two embedding mechanisms: (1) *local semantic embedding* aims to incorporate entity types in HINs via embedding the local structures and types of entities in a supervised way; and (2) *global semantic embedding* leverages multi-hop relation types among entities to propagate the global semantics of similar entities via a Markov Random Field (MRF) [24] to impact the HIN embedding. Then we carefully design a generative model to encode both local semantics and global semantics. By doing so, HINE is capable to capture both local and global semantic information in the embedding vectors. Notice that each dimension of the embedding vectors is a distribution over entities, thus is able to preserve the user-guided semantics. We demonstrate the effectiveness of HINE over existing state-of-the-art techniques on several multi-label classification tasks in two real world networks. The experimental results show that the HINE is able to leverage semantics for better network embedding while preserving the semantics in the resultant embedding vectors.

The main contributions of this paper can be highlighted as below:

– We study the problem of HIN embedding, which is important and has broad applications (e.g., node classification).
– We propose HINE model to embed HINs into low-dimensional semantic vector spaces by consuming both local and global semantic information in HINs.
– We conduct various multi-label network classification tasks on two HIN datasets. The results show that HINE provides significant improvements over state-of-the-art methods with even less training data.

2 Problem Definition

In this section, we first formally introduce HIN, then define the problem of heterogeneous information network embedding (HINE).

Definition 1. *A* ***heterogeneous information network*** *(HIN) is a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \rho, \psi)$*, where* $\mathcal{V}$ *denotes the node (or entity) set, and* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ *denotes the set of edges (or relations) connecting the nodes in* $\mathcal{V}$*, with entity type mapping function* ρ*:* $\mathcal{V} \rightarrow \mathcal{Y}$ *and relation type mapping function* ψ*:* $\mathcal{E} \rightarrow \mathcal{R}$*.* $\mathcal{Y}$ *denotes the set of node types, and* $\mathcal{R}$ *denotes the set of edge types. The number of entity types* $|\mathcal{Y}| > 1$ *or the number of relation types* $|\mathcal{R}| > 1$*.*

Definition 2 ***Heterogeneous information network embedding.*** *Given a network* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \rho, \psi)$*, the heterogeneous information network embedding aims at incorporating semantic information in* $\mathcal{G}$ *to map the entities into a low-dimensional space* $\mathbb{R}^d$*, where* $d << |\mathcal{V}|$*. The embedding vectors preserve the semantics in* $\mathcal{G}$*.*

3 HINE: HIN Embedding

To enable embedding semantics for HINs, we propose HINE model to embed both local and global semantic information in HINs into low-dimensional vectors. To incorporate local semantics, we design a local semantic embedding layer to embed the local structure of each entity as well as its type information in the embedding vectors. To incorporate global semantics, we design a global semantic embedding layer to propagate multi-hop relation type information via an MRF to impact the embedding vectors.

3.1 Model Description

The graphical model representation of HINE is shown in Fig. 1, which has global semantic embedding layer and local semantic embedding layer. Let $\boldsymbol{\theta}_i$ be the embedding vector of entity v_i ($v_i \in \mathcal{V}$) on HIN $\mathcal{G}$, which is a K dimensional multivariate random variable. Let $\boldsymbol{\theta}$ be $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$, where $N = |\mathcal{V}|$. In global semantic embedding, we construct a Markov Random Field (MRF), referred as G, over the embedding vectors $\boldsymbol{\theta}$ to describe the dependency relationships

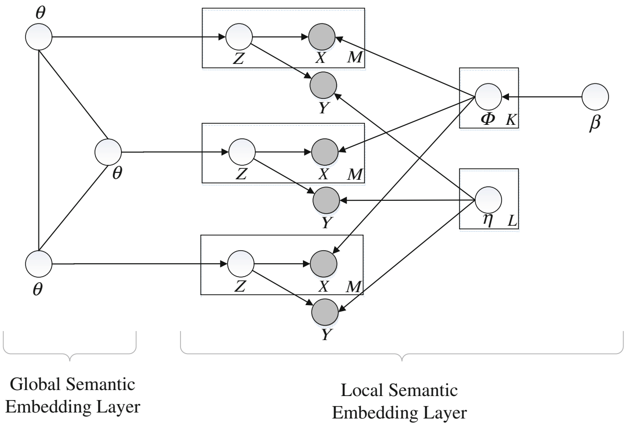

Fig. 1. Model description of HINE. HINE includes global and local semantic embedding layers.

among local semantic embedding, following the topology structure of the HIN. Local semantic embedding consists of generative models for each entity. We assume that each entity can be represented by its local structure in local semantic embedding. Let $\boldsymbol{x}_i$ be the local structure for entity v_i, and $\boldsymbol{z}_i$ be the embedding vectors of its local structure, while y_i is the type of v_i and $y_i \in \mathcal{Y}$. Local semantic embedding is used to embed the local structure $\boldsymbol{x}_i$ for v_i, under the supervision of y_i. The joint embedding probability of both global and local semantic embedding is defined as:

$$\begin{aligned} p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\beta}, \mathcal{G}, \boldsymbol{\eta}) &= p(\boldsymbol{\theta}|\mathcal{G})p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}) \\ &= p(\boldsymbol{\theta}|\mathcal{G}) \int \left(\prod_{i=1}^{N} p(\boldsymbol{x}_i, y_i, \boldsymbol{z}_i|\boldsymbol{\theta}_i, \boldsymbol{\phi}, \boldsymbol{\eta}) \right) p(\boldsymbol{\phi}|\boldsymbol{\beta})\, d\boldsymbol{\phi}, \end{aligned} \tag{1}$$

where it can be decomposed into global semantic embedding $p(\boldsymbol{\theta}|\mathcal{G})$ and local semantic embedding $p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta})$. Once $\boldsymbol{\theta}$ on HIN $\mathcal{G}$ are given in global semantic embedding, the local semantic embedding of entities is conditional independent with each other.

Global Semantic Embedding Layer. By defining an MRF on HIN $\mathcal{G}$, we give the definition of the global semantic embedding $p(\boldsymbol{\theta}|\mathcal{G})$. Inspired by [25] which modeling the document relationships with MRF, we use Markov Random Field [24], a graphical way to represent cyclic dependencies, to model the dependency relationships between entities and propagate global structural and semantic information. Since the links between entities are multi-typed in the HIN, different types of relations may have a broad range of frequencies and weights. Thus we construct the MRF on the HIN, by normalizing multi-typed relation frequencies and weights.

Motivated by community modularity [10] which measures the density comparison between the actual subgraph and random subgraph with the same degree distribution, we build multi-typed relation frequency normalization which measures the frequency comparison between the actual multi-typed relation and the expected relation. The expected relation is what would be expected if the link was randomly placed. The basic idea is that expected relation is viewed as the average relation for those pair nodes with the same type, so the frequency normalized weight is revealed by the difference between the actual relation and the corresponding expected relation. Expected relation w_{ij}^e, the probability of having entity v_i connected to entity v_j with relation type r, is defined as:

$$w_{ij}^e = \frac{\sum_{k \in N_r(v_i)}^{out} w_{ik} \sum_{k \in N_r(v_j)}^{in} w_{kj}}{W_r}, \tag{2}$$

where $N_r(v_i)$ are neighbor entities connected v_i with type r, and W_r is the sum of weights of all relations with type r on HIN $\mathcal{G}$, while $r \in \mathcal{R}$. The frequency normalized weight w_{ij}^f is defined as:

$$w_{ij}^f = \frac{1}{W_r} \left(w_{ij} - w_{ij}^e \right). \tag{3}$$

Then we use Min-Max Normalization [2] to normalize multi-typed relation weights for each relation type. Let W' be the result of W, after normalizing multi-typed relation frequencies and weights.

MRF G is constructed following the topological structure of the HIN with normalized weights W' which considering multi-typed relations. Now we introduce the definition of MRF over $\boldsymbol{\theta}$. Since we assume modeling the entity's $\boldsymbol{\theta}$ by using its neighbors' $\boldsymbol{\theta}$, our MRF satisfy local Markov property. Thus the joint density function can be factorized over the cliques of $\mathcal{G}$:

$$p(\boldsymbol{\theta}|\mathcal{G}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} V_c(\boldsymbol{\theta}_c), \tag{4}$$

where $\mathcal{C}$ is the set of cliques of $\mathcal{G}$, and $Z = \sum_{\boldsymbol{\theta}} \prod_{c \in \mathcal{C}} V_c(\boldsymbol{\theta}_c)$ is the partition function. Since $\boldsymbol{\theta}_i$ is affected by its neighbors $\boldsymbol{\theta}_{N(i)}$, the global semantic embedding $p(\boldsymbol{\theta}|\mathcal{G})$ is defined as:

$$p(\boldsymbol{\theta}|\mathcal{G}) = \frac{1}{Z} \prod_{i=1}^{N} p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{N(i)}), \tag{5}$$

where $p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{N(i)})$ is a Dirichlet distribution as following:

$$p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{N(i)}) \sim Dir \left(\sum_{j \in N(i)} w'_{ij} \boldsymbol{\theta}_j \right). \tag{6}$$

Local Semantic Embedding Layer. Now we define the probability of local semantic embedding $p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta})$ of the joint embedding in Eq.(1). Since there are multi-typed entities in the HIN, motivated by supervised LDA [19] which uses documents' values or labels to supervise topics, we use types of the entities to supervise their local semantic embedding. We assume that each entity in local semantic embedding can be represented by its local structure which is defined as surrounding nodes, such as neighbors, and the corresponding normalized weights from W'. Then $\boldsymbol{x}_i$ is the local structure for entity v_i, which consists of surrounding nodes $x_{i,1}, ..., x_{i,m}, ..., x_{i,M_i}$ with weights $w'_{i1}, ..., w'_{im}, ..., w'_{iM_i}$. Let U be the node set of the HIN, which is used as the set of tokens for local semantic embedding layer. By generating all surrounding nodes for each entity, we produce the local embedding vectors for entities. To generate each surrounding node $x_{i,m}$, we first draw a surrounding node vector $z_{i,m}$ which is a multinomial distribution $Mult(\boldsymbol{\theta}_i)$, then choose a token u from U following multinomial distribution $Mult(\boldsymbol{\phi}_{z_{i,m}})$, where $\boldsymbol{\phi}$ are sampled from $Dir(\boldsymbol{\beta})$. Let L be the number of types of all entities. For entity v_i, we use entity type y_i to supervise its local semantic embedding, by drawing entity type y_i which sampled from multinomial distribution $Mult(\frac{exp(\boldsymbol{\eta}_{y_i}\bar{\boldsymbol{z}_i}^T)}{\sum_{l=1}^{L} exp(\boldsymbol{\eta}_l \bar{\boldsymbol{z}_i}^T)})$, where $\bar{\boldsymbol{z}_i} := \frac{1}{\sum_{m=1}^{M_i} w'_{im}} \sum_{m=1}^{M_i} w'_{im}\, \boldsymbol{z}_{i,m}$. Then the probability of local semantic embedding $p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta})$ is defined as:

$$
\begin{aligned}
&p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}) \\
&= \int \left(\prod_{i=1}^{N} p(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}_i, \boldsymbol{\phi}) p(y_i|\boldsymbol{z}_i, \boldsymbol{\eta})\right) p(\boldsymbol{\phi}|\boldsymbol{\beta})\, d\boldsymbol{\phi} \\
&= \int \left\{\prod_{i=1}^{N} \left[\prod_{m=1}^{M_i} \left(p(z_{i,m}|\boldsymbol{\theta}_i) p(x_{i,m}|z_{i,m}, \boldsymbol{\phi}_{z_{i,m}})\right)^{w'_{im}}\right] p(y_i|\boldsymbol{z}_i, \boldsymbol{\eta})\right\} \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta})\, d\boldsymbol{\phi}\,.
\end{aligned} \tag{7}
$$

3.2 Model Inference

The key inference problem of HINE is to compute the posterior $p(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{G})$ of latent variables $\boldsymbol{\theta}$ and $\boldsymbol{Z}$ with observed data X, Y on HIN $\mathcal{G}$. HINE is an undirected MRF coupled with a directed graphic, which makes the posterior inference tough. Since exact inference is generally intractable, we use Gibbs sampling method to perform approximate inference.

Since $p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{N(i)})$ is a Dirichlet distribution and $p(\boldsymbol{z}_i|\boldsymbol{\theta}_i)$ is a Multinomial distribution, the posterior distribution of $\boldsymbol{\theta}_i$ is a Dirichlet distribution. Then each $\boldsymbol{\theta}_i$ is updated as:

$$
\begin{aligned}
p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{\neg(i)}, \boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\beta}, \boldsymbol{\eta}) \propto\, & p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{N(i)}, \boldsymbol{z}_i) \\
=\, & Dir(\boldsymbol{\theta}_i|\boldsymbol{n}_i + \sum\nolimits_{j\in N(i)} w'_{ij}\boldsymbol{\theta}_j),
\end{aligned} \tag{8}
$$

where $\boldsymbol{n}_i = (n_{i,1}, ..., n_{i,k}, ..., n_{i,K})$ and $n_{i,k}$ is the weighted sum of tokens in entity v_i on k^{th} dimension. Once $\boldsymbol{\theta}$ are given, the local embedding of all entities is conditional independent with each other. Then every $\boldsymbol{z}_{i,m}$ will be updated in turn as:

$$
\begin{aligned}
& p(\boldsymbol{z}_{i,m}|\boldsymbol{z}_{\neg(i,m)}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}) \\
\propto & \frac{p(\boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta})}{p(\boldsymbol{z}_{\neg(i,m)}, X_{\neg(i,m)}, Y_{\neg z_{(i,m)}}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta})} \\
= & \theta_{i,z_{(i,m)}} \frac{n_{z_{(i,m)},x_{(i,m)}}^{\neg(i,m)} + \beta_{x_{(i,m)}}}{\sum_{u=1}^{U}(n_{z_{(i,m)},u}^{\neg(i,m)} + \beta_u)} \frac{exp[\boldsymbol{\eta}_{y_i}(\bar{\boldsymbol{z}}_i - \bar{\boldsymbol{z}}_i^{\neg(i,m)})^T]}{\sum_{l=1}^{L} exp[\boldsymbol{\eta}_l\,(\bar{\boldsymbol{z}}_i - \bar{\boldsymbol{z}}_i^{\neg(i,m)})^T]},
\end{aligned} \tag{9}
$$

where $n_{z_{(i,m)},x_{(i,m)}}^{\neg(i,m)}$ is the weighted sum of tokens $x_{(i,m)}$ which are assigned to $z_{(i,m)}$ except for m^{th} token of i^{th} entity, and $\bar{\boldsymbol{z}}_i^{\neg(i,m)} = \frac{1}{\sum_{j=1}^{M_i} w'_{ij} - w'_{im}} (\sum_{j=1}^{m-1} w'_{ij} \boldsymbol{z}_{i,j} + \sum_{j=m+1}^{M_i} w'_{ij} \boldsymbol{z}_{i,j})$.

After sampling all entities, we update each $\boldsymbol{\eta}_l$ through MLE, where $l \in L$. Since the maximum of likelihood function cannot be solved analytically, we use gradient descent as following:

$$\boldsymbol{\eta}_l := \boldsymbol{\eta}_l - \lambda \left\{ -\frac{1}{N} \sum_{i=1}^{N} \left[\bar{\boldsymbol{z}}_i \left(\mathbf{1}\{y_i = l\} - \frac{exp(\boldsymbol{\eta}_{y_i} \bar{\boldsymbol{z}_i}^T)}{\sum_{l=1}^{L} exp(\boldsymbol{\eta}_l \, \bar{\boldsymbol{z}_i}^T)} \right) \right] \right\}, \tag{10}$$

where $\mathbf{1}\{\}$ is the indicator function and λ is the learning rate. The outer loop will be terminated, once all the parameters $\boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\eta}$ are equilibrium.

4 Experiments

4.1 Data and Evaluation Measures

We use the following two representative HIN datasets to evaluate HINE.

- **DBLP** [14]**:** is the network used most frequently in the study of HINs. It has four node types: Paper, Author, Conference, Term, and four edge types: authorOf, publishedIn, containsTerm, and cites.
- **PubMed:** is the bibliographic network for medicine area, which has the same node and edge types with DBLP.

To promote the comparison between HINE and the comparable methods, we use the same task, multi-label classification, as in [11,23,27]. In research bibliography networks, "research domain" information is critical for many applications. Thus, the aims of our multi-label classification tasks are to classify researchers' fields. We exploit the domain information crawled from Microsoft Academic Search to derive the gold standard. After mapping conferences' and authors' names, about 2 K authors and 1 K conferences are matched. For paper nodes, we use their conferences' domains to be their labels. Since there is no ground truth for terms' domains, we only evaluate three former type nodes in tasks. The statistics of two datasets are represented in Table 1.

Table 1. Statistics of two datasets

Datasets	#(Author)	#(Paper)	#(Conference)	#(Term)	$\|V\|$	$\|E\|$	$\|y\|$
DBLP	885	5,952	921	8,811	16,569	129,186	24
PubMed	530	580	152	4,594	5,856	22,268	23

We use the same metrics (Micro-F1 and Macro-F1) as in [11,23,27] to evaluate the multi-label classification performance for network embedding. Besides, we choose example-based metric Exact Match [31] to show exact match performance. Given a multi-label dataset involving N instances and J category labels, let D be the $(N \times J)$ matrix whose each row is a vector of an instance's ground true labels. P denotes a $(N \times J)$ matrix whose each row is a vector of an instance's predicted labels. We use the following metrics to evaluate multi-label task performance. For those metrics, the bigger the value, the better the performance.

- **Micro-F1** [31]**:** evaluates both micro average of Precision [31] and Recall [31]. It would be more affected by the performance of the categories with more instances.

$$Micro - F1 = \frac{2\sum_{j=1}^{J}\sum_{i=1}^{N} D_{i,j}P_{i,j}}{\sum_{j=1}^{J}\sum_{i=1}^{N} D_{i,j} + \sum_{j=1}^{J}\sum_{i=1}^{N} P_{i,j}}. \tag{11}$$

- **Macro-F1** [31]**:** computes both Precision and Recall on each type of label separately, then evaluates the average of them. It would be more affected by the performance of the categories with fewer instances.

$$Macro - F1 = \frac{1}{J}\sum_{j=1}^{J} \frac{2\sum_{i=1}^{N} D_{i,j}P_{i,j}}{\sum_{i=1}^{N} D_{i,j} + \sum_{i=1}^{N} P_{i,j}}. \tag{12}$$

- **Exact Match** [31]**:** is a very rigorous evaluation measure due to requiring the predicted label set to be an exact match of the true label set.

$$ExactMatch = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{P_i = D_i\}, \tag{13}$$

where $\mathbf{1}\{\}$ is the indicator function.

4.2 Compared Methods

We use the following eight methods as the comparable methods. The first four are the latest representative homogeneous network embedding methods. Since knowledge graphs consist of entity-relation types, they can be regarded as one typical type of heterogeneous information networks. We incorporate the comparison with recently typical knowledge graph embedding methods to show the robustness of the proposed embedding model.

- **DeepWalk** [23]**:** is a network representation method which converts the graph structure to linear sequences though fixed length random walks and learns the sequences with skip-gram.
- **LINE** [27]**:** is a network representation algorithm that maintains the first and second order proximity between the vertexes.
- **GraRep** [5]**:** is a network representation method that captures k-step (with $k > 2$) proximity information, called global structure, between the vertexes.
- **Node2vec** [11]**:** is a semi-supervised network representation method that preserves flexible neighborhood information for vertexes.

- **TransE** [4]**:** is a typical neural-based knowledge base representation learning method which embeds both entities and relations into a low-dimensional space, by treating the relations as translation operations between head and tail entities.
- **TransH** [40]**:** models relations using hyperplanes and translation vectors, which enables entities having different representations in different relationships.
- **TransR** [17]**:** embeds entities and relations into separate spaces and builds translations between entities which projected to the corresponding relation space.
- **PTransE** [16]**:** encodes multiple-step relation paths to learn knowledge base representation, which includes PTransE-ADD, PTransE-MUL, and PTransE-RNN. Since the performance of three models in our tasks is similar, we use PTransE-RNN to represent PTransE.

4.3 Effectiveness Analysis

To compare our method with baselines properly, we use the similar experimental procedure as in [11,23,27]. Different percentages of the vertexes are randomly sampled for training, and the rest are used as the test data for evaluation. We report average performance of Exact Match, Macro-F1 and Micro-F1 over ten different runs. For all models, the multi-label classification problems are decomposed into multiple binary classifications. We use logistic regression implemented by LibLinear [9] for the binary classification. For Node2vec, we search $p, q \in \{0.5, 1, 2, 4\}$. We set p as 1 and q as 4, which makes Node2vec achieving the best performance in tasks generally. We present results for GraRep with $k = 4$, which is enough for DBLP and PubMed.

Table 2 shows the results of training ratio from 1% to 9% for all models with 300 dimensions on DBLP dataset. Numbers in parenthesis represent the percentage improvement, comparing with the highest score of baselines in the column. HINE performs significantly better than all the other methods. As results, with only 4% of the entities used for training, HINE outperforms all the baselines when they are given 9% of the entities. Among all the baselines, knowledge base representation methods, including TransE and its extensions, perform much worse than homogeneous network embedding methods (DeepWalk, LINE, GraRep and Node2vec). That is because the types of relations used in knowledge base representation are very fine-grained, which make models easy to overfit on HINs. Besides, they also ignore the weights of the relations. Although homogeneous network embedding methods achieve better performance among the baselines, the Macro-F1 of HINE achieves 20.42%–72.96% improvement and the Exact Match and Micro-F1 of HINE achieve around 30% increase. It is not surprising because the multi-typed entities and relations encode semantic insights for heterogeneous information network representation learning. This experiment also demonstrates the advantage of joint structural and semantic information for HIN embedding.

Table 2. Results of multi-label classification on DBLP (Numbers in parenthesis represent the percentage improvement, comparing with the highest score of baselines in the column.)

Metric	Algorithm	1%	2%	3%	4%	5%	6%	7%	8%	9%
Exact Match	DeepWalk	6.36	8.57	12.97	13.94	14.01	13.68	15.26	15.38	15.19
	GraRep	11.92	15.97	17.15	17.93	18.19	22.8	22.06	23.85	22.16
	LINE	6.13	8.86	9.98	11.79	13.73	16.33	15.17	18.07	19.45
	Node2vec	8.62	11.12	13.23	13.84	17.21	18.71	21.54	20.91	21.94
	PTransE	4.37	3.96	3.06	2.32	1.81	2.22	2.6	2.15	2.65
	TransE	5.11	2.09	2.61	2.75	2.76	2.73	2.78	3.5	3.04
	TransH	3.36	3.52	3.11	2.88	2.62	2.96	4.05	2.76	3.23
	TransR	4.45	3.29	2.92	3.79	3.28	2.56	3.24	3.5	3.52
	HINE	**14.27**	**17.49**	**21.7**	**26.07**	**30.17**	**29.22**	**32.28**	**34.16**	**34.47**
		(19.71%)	(9.51%)	(26.53%)	(45.40%)	(65.86%)	(28.16%)	(46.33%)	(43.23%)	(55.55%)
Micro-F1	DeepWalk	16.82	16.67	18.63	17.94	19	17.57	17.39	18.56	18.45
	GraRep	17.09	25.71	26.35	30.09	30.31	35.86	33.36	37.5	36.09
	LINE	12.18	16.43	18.13	21.92	23.85	27.85	27.24	31.29	32.84
	Node2vec	14.4	18.44	22.49	25.29	29.05	31.31	35.81	34.1	37.91
	PTransE	11.48	12.29	10.67	9.96	8.22	8.63	9.66	8.77	8.87
	TransE	13.14	10.72	10.5	9.01	10.46	9.28	9.76	10.07	10.13
	TransH	12	12.43	10.64	10.62	10.78	9.86	10.86	9.39	9.84
	TransR	13.22	12.42	11.46	11.96	10.87	9.84	10.11	10.52	10.53
	HINE	**22.63**	**27.69**	**33.99**	**38.46**	**43.31**	**42.25**	**45.3**	**47.42**	**48.64**
		(32.41%)	(7.7%)	(28.99%)	(27.81%)	(42.89%)	(17.81%)	(26.50%)	(26.45%)	(28.3%)
Macro-F1	DeepWalk	5.27	7.39	9.72	10.4	11.06	11.52	11.79	12.26	12.47
	GraRep	5.76	11.21	10.79	14.06	16.31	19	17.83	20.62	19.67
	LINE	5.88	7.92	9.29	12.34	12.27	14.62	16.79	18.74	19.63
	Node2vec	4.82	9.19	11.78	12.75	14.47	17.28	19.34	22.03	23.15
	PTransE	3.88	4.39	4.76	4.74	4.19	4.02	4.76	4.65	4.2
	TransE	4.42	4.67	4.68	4.16	4.76	4.33	4.27	4.33	5.02
	TransH	4.28	5.15	4.58	5.3	5.13	5.15	4.61	4.37	4.07
	TransR	4.52	5.13	5.2	5.03	5.18	5.45	5.35	5.01	4.98
	HINE	**9.25**	**13.5**	**20.1**	**23.49**	**28.21**	**26.9**	**30.48**	**33.16**	**35.49**
		(57.31%)	(20.42%)	(70.62%)	(67.06%)	(72.96%)	(41.57%)	(57.60%)	(50.52%)	(53.30%)

Table 3 presents the results of varying the training ratio from 10% to 90% on PubMed dataset. Since PubMed network is sparser than DBLP, we use 400 dimensions to present the results. The performance of HINE is significantly better than all the baselines, which is consistent with the previous experiment. Comparing to all the baselines, the Exact Match of HINE achieves 24.72%–39.37% improvement and the Micro-F1 and Macro-F1 of HINE achieve around 15–30% increase. Besides, with only 40% of the entities used for training, the performance of all the metrics for HINE exceeds all the baselines even when they have been given 90% of the entities. That is HINE can beat all the baselines with 50% less training data. Comparing to the previous experiment, the performance of knowledge base representation methods remain worse with more training data, while the other methods including HINE achieve significant increase generally. This indicates that the types of relations used in knowledge base representation make models easy to overfit on HINs.

Table 3. Results of multi-label classification on PubMed (Numbers in parenthesis represent the percentage improvement, comparing with the highest score of baselines in the column.)

Metric	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
Exact Match	DeepWalk	27.62	34.09	37.46	40.94	43.98	45.26	48.29	50.08	49.28
	GraRep	29.55	38.94	43.12	48.63	51.63	53.73	55.15	54.26	53.06
	LINE	26.51	36.92	42.29	47.89	51.23	55.88	57.68	58.59	58.94
	Node2vec	30.45	38.48	42.59	45.96	46.47	48.13	52.88	51.13	54.84
	PTransE	2.98	3.96	6.91	11.12	12.3	15.1	17.64	20.03	21.54
	TransE	3.84	4.96	6.36	9.22	11.62	13.92	16.85	18.81	20.89
	TransH	4.41	5.39	8.07	10.74	15.05	16.58	19.26	21.07	25.26
	TransR	4.21	4.48	7.73	10.36	13.51	15.03	16.98	18.58	22.58
	HINE	**41.76**	**49.91**	**60.1**	**64.54**	**69.03**	**70.34**	**71.94**	**73.55**	**78.2**
		(37.14%)	(28.17%)	(39.37%)	(32.71%)	(33.70%)	(25.87%)	(24.72%)	(25.53%)	(32.67%)
Micro-F1	DeepWalk	40.96	43.23	46.82	49.27	52.31	54.57	55.68	57.85	60.75
	GraRep	48.79	60.3	64.72	67.97	71.21	73.42	75.38	75.25	74.88
	LINE	43.52	57.04	62.82	67.75	70.91	73.91	76.48	77.49	77.57
	Node2vec	49.78	60.83	64.62	68.83	69.16	70.13	73.6	75.97	75.52
	PTransE	6.36	8.65	14.84	22.77	25.69	29.44	33.39	36.12	38.76
	TransE	8.18	10.86	13.97	19.92	24.28	29.06	33.46	35.92	39.45
	TransH	9.73	12.15	16.13	21.28	29.18	30.79	35.02	37.29	42.64
	TransR	9.37	10.19	16.14	22.08	26.94	29.82	33.05	34.73	41.4
	HINE	**61.09**	**68.71**	**76.1**	**79.59**	**83.03**	**83.88**	**84.54**	**86.92**	**89.16**
		(22.71%)	(12.95%)	(17.58%)	(15.63%)	(16.59%)	(13.48%)	(10.53%)	(12.16%)	(14.94%)
Macro-F1	DeepWalk	25.2	29.37	32.04	33.92	36.22	37.35	38.6	39.32	39.96
	GraRep	27.26	37.15	45.9	48.68	52.04	53.73	53.89	57.46	39.79
	LINE	24.2	40.65	45.38	54.1	55.32	54.33	60.53	55.78	52.16
	Node2vec	29.21	47.03	53.02	59.83	56.09	56.16	58.32	62.04	57.77
	PTransE	2.44	3.3	5.89	9.95	11.36	12.81	14.49	16.88	17.1
	TransE	3.27	4.45	5.76	8.86	10.94	12.48	14.27	16.18	17.21
	TransH	3.75	5.07	6.98	9.13	12.71	13.9	15.59	16.39	18
	TransR	3.8	4.45	7.01	9.77	11.71	13.2	14.68	15.23	18.15
	HINE	**42.17**	**55.1**	**63.7**	**64.12**	**67.9**	**71.27**	**68.53**	**66.54**	**60.32**
		(44.36%)	(17.15%)	(20.14%)	(7.17%)	(21.05%)	(26.90%)	(13.21%)	(7.25%)	(4.41%)

These experiments indicate that properly using multi-typed entities and relations to embedding HINs is critical.

4.4 HINE Parameter Study

To evaluate how the number of dimensions affects the performance, we test the changes in performance of HINE on multi-label classifications task on PubMed dataset. Figure 2 shows the performance of the HINE model with different dimensions and training rates. In Fig. 2, increasing the number of dimensions improves performance. Then improvements tend to be gentle once the numbers of dimensions reach around 300. It is not surprising since HINE captures network structural and semantic information in a top-down manner. The smaller the number of dimensions is, more generalized information is captured. With more dimensions, more detailed information is added into the embedding vectors. Once the global and local information is enough for the current task, the performance tends

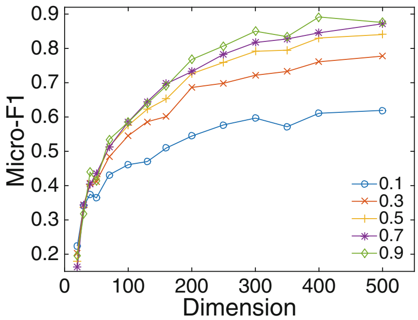

Fig. 2. Performance over dimensions on PubMed

to increase slightly. Besides, results show that the optimal number of dimension which is determined by Elbow criteria grows with training rates. This is mainly because the larger number of dimensions brings more information, which increases the performance with more labeled data. This experiment suggests that HINE captures more and more structural and semantic information (starting from generalized ones to specific ones), with the growing number of dimensions.

4.5 Case Study of HINE Vectors

To provide the readers more insights about the semantics of embedding vectors, Table 4 empirically shows part of dimensions and two entities' vectors on DBLP dataset. Numbers in bold represent the values of top 3 highest dimensions for two vectors. Since each dimension is a distribution on all entities in the network, the last row of results shows the top 15 entities and their weights from corresponding distributions for those dimensions, where the letter before @ is the abbreviation of the entity's type. For example, "a" is the abbreviation of node type *Author*. We can see that dimension #59 and #130 are mainly about information retrieval, while #121 focuses on XML data search and #41 is more concern of feedback and safety information retrieval. Entity "SIGIR" is mainly distributed on dimension #130 and #59 which are highly related to it. Comparing to "SIGIR", the distribution of entity "search" on dimensions is more gentle. It is not surprising since "search" is used on a much broader scale. By using the distributions of entities to represent dimensions, the embedding vectors preserve semantics, which will significantly improve the understanding of the HIN embedding.

5 Related Work

Network embedding technology has been widely studied in these years. The classical methods, belonging to graph embedding, embed graph matrix into a low dimensional space, such as linear methods based on SVD [28,29], IsoMap

Table 4. Demonstration for part of dimensions and the vectors of entity "SIGIR" and "search" on DBLP (Numbers in bold represent the values of top 3 highest dimensions for two vectors.)

	Dimension #41	Dimension #59	Dimension #121	Dimension #130	...
SIGIR	**0.089481**	**0.173554**	0.009164	**0.301765**	...
search	0.003949	**0.117275**	**0.122546**	**0.142530**	...
	p@Fine-grained relevance feedback for XML retrieval:0.209451, p@Warping-Based Offline Signature Recognition:0.171994, a@Suneel Suresh:0.097583, v@IEEE Transactions on Information Forensics and Security:0 .096428, t@feedback:0.079638, t@relev:0.043808, t@structur:0.036836, t@xml:0.030606, t@signatur:0.018158, t@offlin:0.017672, t@grain:0.016550, t@fine:0.016550, t@retriev:0.014943, t@recognit:0.0 14522, t@ir:0.008294	v@SIGIR Forum:0.123274, p@Hierarchical Fuzzy Intelligent Controller for Gymnastic Bar Actions:0.103294, p@Report on INEX 2008:0.096149, p@The first joint international workshop on entity-oriented and semantic search (JIWES):0.076257, p@Temporal index sharding for space-time efficiency in archive search:0.071717, p@A novel hybrid index structure for efficient text retrieval:0.069568, p@Index maintenance for time-travel text search:0.066497, p@Report on INEX 2010:0.059012, p@Report on INEX 2009:0.058984, v@JACIII:0.053933, t@report:0.044425, t@entiti:0.007204, t@joint:0.006976, t@fuzzi:0.006553, t@search:0.004659	p@Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data:0.122698, p@Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks:0.119706, v@Intelligent Search on XML Data:0.118028, p@Classification and Focused Crawling for Semistructured Data:0.102976, p@Ontology-Enabled XML Search:0.097273, v@WebDB:0.041657, a@Dominique A. Winne:0.041657, t@focus:0.031450, t@xml:0.030943, t@ontolog:0.028016, t@data:0.024637, t@classif:0.023185, t@search:0.022258, t@enabl:0.018745, t@crawl:0.015353	v@SIGIR:0.215321, p@Efficient and self-tuning incremental query expansion for top-k query processing:0.159374, p@Making SENSE: socially enhanced search and exploration:0.148130, p@Efficient top-k querying over social-tagging networks:0.131191, t@tag:0.031723, t@search:0.026990, t@user:0.021907, t@recommend:0.019541, t@work:0.013616, t@item:0.012864, t@sens:0.009380, t@make:0.009368, t@enhanc:0.009309, t@effici:0.008464, t@content:0.007294	...

[30], MDS [8], and graph factorization [1]. Due to their high complexity, various neural network embedding methods are proposed. DeepWalk [23] converts the network structure to linear sequences though fixed length random walks and learns the sequences with skip-gram. LINE [27] maintains the first and second order proximity between the nodes, while GraRep [5] and HOPE [21] consider high-order proximities. DNGR [6] and SDNE [39] adopt deep neural network to capture graph structural information. TriDNR [22] and TADW [41] learn network representation with text information. Node2vec [11] proposes a semi-supervise algorithm to learn network representation flexibly. We note that these methods focus on homogeneous networks. Besides, HNE [7] aims at embedding networks consisting of various data sources of nodes (such as text, image, and video). All the above methods discard the semantic information carried by the multi-typed entities and relations during the embedding. Thus they can not be adapt to HINs.

Since knowledge graphs consist of billions of entity-relation types, they can be regarded as one typical type of heterogeneous information networks [36,37]. TransE [4] is a typical neural-based knowledge base representation method which embeds both entities and relations into a low-dimensional space, by treating the relations as translation operations between head and tail entities. There are various methods proposed to expand TransE, such as TransH [40], TransR [17], PTransE [16], TransD [12], TranSparse [13], and so on. However, the types of relations in TransE and its extensions are very fine-grained, which makes models easy to overfit on HINs. In contrast, by properly incorporating the HIN semantics in the embedding model and preserving the semantics in the embedding vectors, HINE can learn the embedding for HINs.

6 Conclusion

We propose HINE, a novel model for learning semantic representations of entities for HINs. Our method incorporates the local and global HIN semantics in the embedding model and preserves the semantics in the embedding vectors. Each dimension of our embedding vectors is a distribution of semantic entities, which will significantly improve the understanding of the HIN embedding and be very useful for later follow-up HIN studies. Extensive experiments over existing state-of-the-art methods exhibit the effectiveness of our method on various real world HINs.

References

1. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J.: Distributed large-scale natural graph factorization. In: WWW, pp. 37–48 (2013)
2. Al Shalabi, L., Shaaban, Z., Kasasbeh, B.: Data mining: a preprocessing engine. J. Comput. Sci. **2**(9), 735–739 (2006)
3. Bhagat, S., Cormode, G., Muthukrishnan, S.: Node classification in social networks. In: Social Network Data Analytics, pp. 115–148. Springer, US (2011)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
5. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: CIKM, pp. 891–900 (2015)
6. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: AAAI, pp. 1145–1152 (2016)
7. Chang, S., Han, W., Tang, J., Qi, G.J., Aggarwal, C.C., Huang, T.S.: Heterogeneous network embedding via deep architectures. In: KDD, pp. 119–128 (2015)
8. Cox, T.F., Cox, M.A.: Multidimensional Scaling. CRC Press, Boca Raton (2000)
9. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. JMLR **9**, 1871–1874 (2008)
10. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
11. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD (2016)
12. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL, pp. 687–696 (2015)
13. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: AAAI, pp. 985–991 (2016)
14. Ley, M.: DBLP: some lessons learned. VLDB **2**, 1493–1500 (2009)
15. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. **58**, 1019–1031 (2007)

16. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: EMNLP, pp. 705–714 (2015)
17. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, pp. 2181–2187 (2015)
18. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**, 2579–2605 (2008)
19. Mcauliffe, J.D., Blei, D.M.: Supervised topic models. In: Advances in Neural Information Processing Systems, pp. 121–128 (2008)
20. Meng, C., Cheng, R., Maniu, S., Senellart, P., Zhang, W.: Discovering meta-paths in large heterogeneous information networks. In: WWW, pp. 754–764 (2015)
21. Ou, M., Cui, P., Pei, J., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: KDD (2016)
22. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. In: IJCAI, pp. 1895–1901 (2016)
23. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: KDD, pp. 701–710 (2014)
24. Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. CRC Press, Boca Raton (2005)
25. Sun, Y., Han, J., Gao, J., Yu, Y.: iTopicModel: information network-integrated topic modeling. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 493–502 (2009)
26. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: meta path-based top-k similarity search in heterogeneous information networks. In: VLDB, 992–1003 (2011)
27. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: WWW, pp. 1067–1077 (2015)
28. Tang, L., Liu, H.: Scalable learning of collective behavior based on sparse social dimensions. In: CIKM, pp. 1107–1116 (2009)
29. Tang, L., Liu, H.: Leveraging social media networks for classification. In: DMKD, pp. 447–478 (2011)
30. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
31. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2009)
32. Wang, C., Duan, N., Zhou, M., Zhang, M.: Paraphrasing adaptation for web search ranking. In: ACL, pp. 41–46 (2013)
33. Wang, C., Song, Y., El-Kishky, A., Roth, D., Zhang, M., Han, J.: Incorporating world knowledge to document clustering via heterogeneous information networks. In: KDD, pp. 1215–1224 (2015)
34. Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Knowsim: a document similarity measure on structured heterogeneous information networks. In: ICDM, pp. 1015–1020 (2015)
35. Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Text classification with heterogeneous information network kernels. In: AAAI, pp. 2130–2136 (2016)
36. Wang, C., Song, Y., Roth, D., Wang, C., Han, J., Ji, H., Zhang, M.: Constrained information-theoretic tripartite graph clustering to identify semantically similar relations. In: IJCAI, pp. 3882–3889 (2015)
37. Wang, C., Song, Y., Roth, D., Zhang, M., Han, J.: World knowledge as indirect supervision for document clustering. TKDD **11**(2), 13:1–13:36 (2016)
38. Wang, C., Sun, Y., Song, Y., Han, J., Song, Y., Wang, L., Zhang, M.: Relsim: relation similarity search in schema-rich heterogeneous information networks. In: SDM (2016)

39. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: KDD (2016)
40. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119 (2014)
41. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: IJCAI, pp. 2111–2117 (2015)
42. Yu, X., Sun, Y., Norick, B., Mao, T., Han, J.: User guided entity similarity search using meta-path selection in heterogeneous information networks. In: CIKM, pp. 2025–2029 (2012)
43. Zhou, Y., Liu, L.: Activity-edge centric multi-label classification for mining heterogeneous information networks. In: KDD, pp. 1276–1285 (2014)