Information Systems 81 (2019) 82–91

Contents lists available at ScienceDirect

Information Systems

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

HiWalk: Learning node embeddings from heterogeneous networks☆

Jie Bai [a,b,*], Linjing Li [a,**], Daniel Zeng [a,c]

[a] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[b] The School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, 100049, China
[c] The Department of Management of Information Systems, University of Arizona, Tucson, AZ 85721, United States

HIGHLIGHTS

- A method of learning embeddings for given type of nodes in heterogeneous networks.
- A hierarchical method, leveraging the other part of nodes as "background knowledge".
- More effective, efficient and concentrated in heterogeneous network learning.
- It can be applied to many real-world analysis tasks.

ARTICLE INFO

Article history:
Received 4 February 2018
Received in revised form 15 October 2018
Accepted 20 November 2018
Available online 26 November 2018

Keywords:
Network analysis
Representation learning
Behavioral analysis
Random walk
Heterogeneous network

ABSTRACT

Heterogeneous networks, such as bibliographical networks and online business networks, are ubiquitous in everyday life. Nevertheless, analyzing them for high-level semantic understanding still poses a great challenge for modern information systems. In this paper, we propose HiWalk to learn distributed vector representations of the nodes in heterogeneous networks. HiWalk is inspired by the state-of-the-art representation learning algorithms employed in the context of both homogeneous networks and heterogeneous networks, based on word embedding learning models. Different from existing methods in the literature, the purpose of HiWalk is to learn vector representations of the targeted set of nodes by leveraging the other nodes as "background knowledge", which maximizes the structural correlations of contiguous nodes. HiWalk decomposes the adjacent probabilities of the nodes and adopts a hierarchical random walk strategy, which makes it more effective, efficient and concentrated when applied to practical large-scale heterogeneous networks. HiWalk can be widely applied in heterogeneous networks environments to analyze targeted types of nodes. We further validate the effectiveness of the proposed HiWalk through multiple tasks conducted on two real-world datasets.

1. Introduction

Network analysis has attracted a growing interest with the explosion of Internet-based applications, such as instant messaging, social media, e-commerce sites, etc. Early network analysis tasks mainly focused on homogeneous networks where nodes and edges in the network are of the same type. However, heterogeneous networks are more widespread in everyday life and have also attracted extensive attention from academia [1]. Different from homogeneous networks, there are usually more than one type of nodes and/or edges in heterogeneous networks.

A number of works have studied heterogeneous networks from various perspectives [2]. Existing studies include clustering [3], classification [4], link analysis [5], community analysis [6], as well as network representation learning (or named as network embedding) [7–10]. Network representation learning usually aims at transforming the latent structure of nodes in a network into distributed vectors, which is the foundation of many machine learning approaches in the network context. Currently, the state-of-the-art network representation learning methods for heterogeneous networks mostly consider networks that have several types of nodes and edges, and learn embeddings for all of the nodes simultaneously [8,10,11].

Inspired by the recent advances in network representation learning methods [11–14], we believe that the low-dimensional vector representations of heterogeneous networks can be learned by taking advantage of word embedding learning methods, such as skip-gram, which maximizes classification of context words based on a given word [15]. However, different from the studies above,

☆ Declarations of interest: none.

* Corresponding author at: Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China.
** Corresponding author.
E-mail addresses: baijie2013@ia.ac.cn (J. Bai), linjing.li@ia.ac.cn (L. Li), dajun.zeng@ia.ac.cn (D. Zeng).

https://doi.org/10.1016/j.is.2018.11.008
0306-4379/

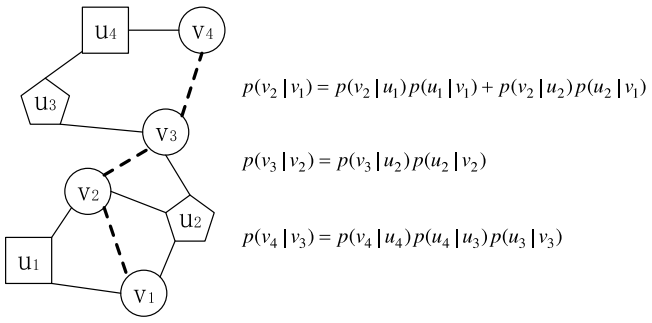


Fig. 1. Entity sequence generation through hierarchical random walk. The algorithm scans the network and constructs the entity sequence based on the probability distribution of the node correlations. "v" denotes the entities and "u" denotes the items. Different shapes represent different node types.

we found several interesting characteristics contained in many heterogeneous networks.

- For many network analysis tasks, one usually pays more attention to some specific type of nodes instead of treating all nodes equally, such as user behavior analysis.
- Usually in heterogeneous networks the edges exist among different types of nodes. For example, in a bibliographical network the "co-authorship" means two or more authors have connections with one particular literature.
- The interactions among different types of nodes usually contain abundant information, which can be used in the network structural learning.

Based on the characteristics above, this paper deals with the problem of learning vector representations for the targeted type of nodes in heterogeneous networks. More specifically, we aim to learn embeddings for the given type of nodes (based on characteristics 1), with the help of the other nodes as "background knowledge" (based on characteristics 3). We differentiate the nodes in a heterogeneous network as "entities" and "items", and learn vector representations for the "entities" by taking advantage of the background knowledge contained in the "items". Moreover, the interactions among different types of nodes (characteristics 2) is the key factor that we use to perform network representation learning.

We propose HiWalk, a hierarchical-random-walk-based method that can learn embeddings for the specific "entities" in heterogeneous networks. HiWalk constructs entity sequences from heterogeneous networks through hierarchical random walk, which maximizing structural correlations of contiguous entities in the sequence even if their correlations are not presented explicitly in the network. Hierarchical random walk is an extension of the uniform random walk methods where each walk step has multiple stages: at each stage, it starts from the current node and randomly walks to an adjacent node. Each walk step starts and ends with a pair of entities. Fig. 1 shows an illustration of the entity sequence generation process through hierarchical random walk, from which we can see that it captures denser semantics compared with general random walk process. By applying hierarchical random walk, HiWalk combines the advantages of both homogeneous-network-oriented and heterogeneous-network-oriented representation learning methods. While learning embeddings for the single type of nodes, HiWalk leverage abundant semantic information contained in the heterogeneous networks. We also demonstrate that HiWalk decomposes the adjacent probabilities of the targeted entities by taking advantage of the structural correlations among multiple types of nodes; thus it actually runs through a high-efficiency sampling process.

Overall, aiming at obtaining better understanding of the network structures in real-world applications and systems, we propose a hierarchical-random-walk-based method for learning embeddings of given type of nodes in heterogeneous networks, which can be seen as a combination of both homogeneous-network-oriented and heterogeneous-network-oriented representation learning methods. Compared with existing methods, our proposed method is superior in the following perspectives.

- Concentration: Instead of learning embeddings for all of the nodes in a heterogeneous network, HiWalk concentrates on the specific type of nodes, which is more suitable for the network analysis tasks where part of nodes is cared about, like user behavior analysis.
- Effectiveness: HiWalk learns embeddings for a targeted set of nodes by leveraging the other nodes as the background knowledge. Compared with general random walk, HiWalk's hierarchical random walk strategy captures denser semantics during the node sequence generation, which makes it more effective than peer methods.
- Efficiency: HiWalk is a fast and memory-saving method compared with existing heterogeneous network representation learning methods. Based on the structure of the algorithm, it can be easily deployed on a distributed architecture.

2. Related works

The homogeneous network representation learning problem has been extensively investigated. Early works usually learned low-dimensional network representations by linear or nonlinear dimensionality reduction from the adjacent matrix of the networks [16], like IsoMap [17], LLE [18], SVD [19], and Graph Factorization [20]. Limited by high computational complexity, those methods cannot perform ideally on realistic large-scale networks.

Recently, with the advances of word embeddings applied to natural language processing [21], a branch of studies is working on learning network embeddings following the ideas of constructing word embeddings [12–14]. More specifically, by generating appropriate node sequences from the homogeneous network, one could learn vector representations for the nodes from a collection of node sequences through models like skip-gram, just like learning vector representations for the words from a corpus. These neural-network-based methods are more efficient than the aforementioned dimensionality-reduction-based methods. However, the main challenge is how to construct node sequences from a given network. DeepWalk [12] used local information obtained from uniform random walks of a homogeneous network to get the word sequences. Later, more sophisticated assumptions were made based on this idea. Line [13] grasped the first order proximity (immediate neighbors information) as well as the second order proximity of nodes. Node2vec [14] integrated the BFS and DFS search strategies and adopted a biased random walk procedure for scalable feature learning in networks, which can also be seen as a generalization of DeepWalk.

Heterogeneous network representation learning is an emerging research area. Some current existing works concentrate on learning knowledge graphs, which have unlimited types of entities and relations in most cases. TransE [7] learns embeddings for both entities and relations of a knowledge graph by interpreting the relations as translation operations between the head and tail entities. Many studies have been proposed to extend TransE, such as TransD [22], which considers multiple types of entities and relations simultaneously, TransR [23], which models entity and relations in different vector spaces, and TranSparse [24], which deals with heterogeneous and unbalanced problems in the knowledge graph.

There are also some network representation learning works that study heterogeneous networks that have limited types of nodes and edges. HNE [25] was proposed to learn multi-type of contents and relations with deep architectures. In HNE, contents can be modeled jointly, including images, entities, and texts. PTE [10] constructed heterogeneous networks based on text, document, and label, and proposed a semi-supervised representation learning method to learn the text embedding by taking text as the nodes in the networks. HINE [8] consists of local and global semantic embeddings, which can explicitly capture the semantics in heterogeneous networks and build models under user guidance. ProxEmbed [9] learned proximity embeddings for node pairs in heterogeneous networks, which can be applied directly to semantic proximity search. Metapath2vec and metapath2vec++ [11] construct the heterogeneous neighborhood nodes in heterogeneous networks by formalizing metapath-based random walks and use the skip-gram model to learn node embeddings.

Different from the above studies, in this paper we develop a hierarchical-sampling-based method, HiWalk, to learn the embeddings for given types of nodes in heterogeneous networks, which significantly improves the effectiveness and efficiency of node embedding learning. HiWalk combines the characteristics from both homogeneous-network-oriented and heterogeneous-network-oriented representation learning methods. It concentrates specifically on the single type of nodes, while leveraging the complex network structure and abundant information contained in heterogeneous networks, which is applicable for scenarios like user behavior analysis in a heterogeneous environment.

3. Problem definition and deduction

Following the convention [1], a network is defined as $G = (V, E)$, in which V is the node collection and is the edge collection in the network. Meanwhile, there is a node type projection $\phi : V \to T$ and an edge type projection $\psi : E \to R$, such that for any $v \in V$, $\phi(v) \in T$; $e \in E$, $\psi(e) \in R$. Here T and R are the corresponding node-type collection and edge-type collection. If $|T| > 1$ or $|R| > 1$, the network G is usually regarded as a heterogeneous network; otherwise G is a homogeneous network.

In this paper, nodes are divided into two collections: the nodes we will learn the vector representations for (entities) and the nodes we leverage as the background knowledge (items). We define these two collections as V and U, respectively. Thus the network can be defined as $G = (U, V, E)$. In the rest of this paper, we will focus on learning the embeddings for $v \in V$ with the help of U as well as the edges E.

3.1. Network representation learning

Skip-gram [21] has been proved to be an effective model for learning node embeddings in both heterogeneous networks [11] and homogeneous networks [12–14]. The original intention of skip-gram models is to learn word embeddings that are useful for predicting the surrounding words.

When applying skip-gram to network representation learning, the goal changes to learning node embeddings which aims to predict the surrounding nodes in a node sequence. So the objective function of a local node sequence can be defined as

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p(v_{t+j}|v_t), \tag{1}$$

where T is the length of the sequence, and c is the width of the window. This objective function implies that in the sequences constructed for learning node embeddings, neighboring nodes should have higher correlations. Assume that the structural correlations of entities are transferable, i.e.,

$$p(v_{t+j}|v_t) = \prod_{k=1}^{j} p(v_{t+k}|v_{t+k-1}). \tag{2}$$

Substitute (2) into (1), we can get

$$\begin{aligned}\mathcal{L} &= \frac{1}{T}\sum_{t=1}^{T}\sum_{j=-c}^{c}\sum_{k=1}^{j}\log p(v_{t+k}|v_{t+(k-1)})\\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{j=-c}^{c}(c+1-j)\log p(v_{t+j}|v_{t+(j-1)})\\ &\propto \frac{1}{T}\sum_{t=1}^{T}\log p(v_{t+1}|v_t).\end{aligned} \tag{3}$$

(3) implies that the problem is to maximize the correlations of adjacent entities. In general cases the correlation of v_t and v_{t+1} is represented explicitly in the network (For example, by whether they are connected by an edge.) Below we need to deal with the problem where the correlation of v_t and v_{t+1} cannot be observed explicitly in the network.

3.2. Representation learning of hiwalk

In the heterogeneous network context, it is natural to employ (1) as the initial objective function from which we learn the embeddings of entities, thus (3) still holds. However, the correlations between entities are usually implemented by their interactions with the items, i.e., the edges usually exist between entities and different items. Then the correlations $p(v_{t+1}|t_i)$ of the entity pair v_t and v_{t+1} should take the items U into consideration.

In the following discussion, we consider a typical case as the example. In a network where edges exist only between U and V, then for the given v_t and v_{t+1}, they may have no common neighbor, or there will be several items that act as their common neighbors. According to the Bayesian formula, their structural correlation $p(v_{t+1}|v_t)$ can be formulated as

$$p(v_{t+1}|v_t) = \sum_{u_k\in U} p(u_k|v_t)p(v_{t+1}|u_k). \tag{4}$$

Assuming that there are vector representations $\overrightarrow{v}$ and $\overrightarrow{u}$, which reflect v and u respectively, the structural correlations can be represented by the softmax function

$$p(v_{t+1}|u_k) = \frac{exp\left(\overrightarrow{v_{t+1}}^T\overrightarrow{u_k}\right)}{\sum_{v_{i'}\in V} exp\left(\overrightarrow{v_{i'}}^T\overrightarrow{u_k}\right)} \tag{5}$$

and

$$p(u_k|v_t) = \frac{exp\left(\overrightarrow{u_k}^T\overrightarrow{v_t}\right)}{\sum_{u_{k'}\in U} exp\left(\overrightarrow{u_{k'}}^T\overrightarrow{v_t}\right)} \tag{6}$$

separately. Then the objective function can be represented as

$$\begin{aligned}\mathcal{L} &= \frac{1}{T}\sum_{t=1}^{T}\log p(v_{t+1}|v_t)\\ &= \frac{1}{T}\sum_{t=1}^{T}\log\left[\sum_{u_k\in U} p(v_{t+1}|u_k)p(u_k|v_t)\right]\\ &\propto \frac{1}{T}\sum_{t=1}^{T}\log\left[\sum_{u_k\in U}\frac{exp\left(\overrightarrow{v_{t+1}}^T\overrightarrow{u_k}\right)}{\sum_{v_{i'}\in V} exp\left(\overrightarrow{v_{i'}}^T\overrightarrow{u_k}\right)}\frac{exp\left(\overrightarrow{u_k}^T\overrightarrow{v_t}\right)}{\sum_{u_{k'}\in U} exp\left(\overrightarrow{u_{k'}}^T\overrightarrow{v_t}\right)}\right].\end{aligned} \tag{7}$$

Our goal is to learn the vector representations for entities ($\overrightarrow{v_t}$ and $\overrightarrow{v_{t+1}}$), whose gradients are

$$\frac{\partial \mathcal{L}}{\partial v_t} = \frac{1}{T} \sum_{u_k \in U} \overrightarrow{u_k} \left[I_{u_k, v_t} - p\left(u_k | v_t\right) \right] \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial v_{t+1}} = \frac{1}{T} \sum_{u_k \in U} \overrightarrow{u_k} \left[I_{v_{t+1}, u_k} - p\left(v_{t+1} | u_k\right) \right] \tag{9}$$

Here I is the indicator of connections between node pairs:

$$I_{v,u} = \begin{cases} 1, & v \text{ and } u \text{ are connected by an edge} \\ 0, & otherwise \end{cases} \tag{10}$$

The Eqs. (8) and (9) implies that the representation learning for the v_t and v_{t+1} are mostly decided by their neighboring items. (Actually all of the items are involved. However, the critical factors are composed by those whose corresponding indicator I is not zero.) Besides, according to (8) and (9), if $\overrightarrow{v_t}$ and $\overrightarrow{v_{t+1}}$ are learned through the Stochastic Gradient Descent algorithm, they will move gradually toward their common neighbors. These conclusions can also be generalized to the common heterogeneous networks.

However, here the representations $\overrightarrow{u}$ will not be learned explicitly because it is resource-wasted if we only care about V in the network. Instead, we use the deductions above as an inspiration to learn the structural correlations among entities, i.e., in a heterogeneous network where entities have no direct connections, the structural correlations among entities can be evaluated through the heterogeneous items. This is the "background knowledge" that we leverage to learn the network structure.

4. Method introduction

Based on the deductions above, we propose HiWalk, a hierarchical-random-walk based method to learn embeddings for entities in heterogeneous networks. We first propose a hierarchical random walk method to construct entity sequences, which aims to maximize the structural correlations of adjacent entity pairs. Then the embeddings of entities are learned through the skip-gram model [21] by adopting the negative-sampling optimization strategy.

4.1. Hierarchical random walk

Random walk can be considered as a sampling method used on networks. Each time it starts from a node and samples a new node through a probabilistic distribution $p(v_{i+1}|v_i)$ constructed over the edges. The larger the weight of an edge, the higher the probability the corresponding node will be sampled. Different from the general principles of random walk, here we aim at transforming a given set of entities {v} into a specific sequence [v], where there is no necessary edge existing between any v_i and v_{i+1}. In the resulting sequence, the entities next to each other should be more structurally correlated compared with the other entities, i.e., $p(v_{i\pm1}|v_i) \geq p(v_{i\pm k}|v_i)$, $k \geq 1$.

Following the previous analysis, the structural correlations of v_i and v_{i+1} are reflected through the items, such as the length of their shortest path or how many common neighbors they have. With the random walk method, the path from v_i to v_{i+1} may contain one or more items, and each walking step corresponds to a probability transition. Finally, the probability of walking from v_i to v_{i+1} is in direct proportion to their correlations.

Thus, the hierarchical random walk is implemented by taking advantage of $u \in U$ as intermediate variables. When sampling a new entity v_{i+1} from the current v_i, it follows a hierarchical sampling procedure constructed by multiple stages. Assuming that the length of the shortest path from v_i to v_{i+1} is S, then there are S stages in each hierarchical random walk step (The illustration of a hierarchical random walk step can be find in Fig. 2):

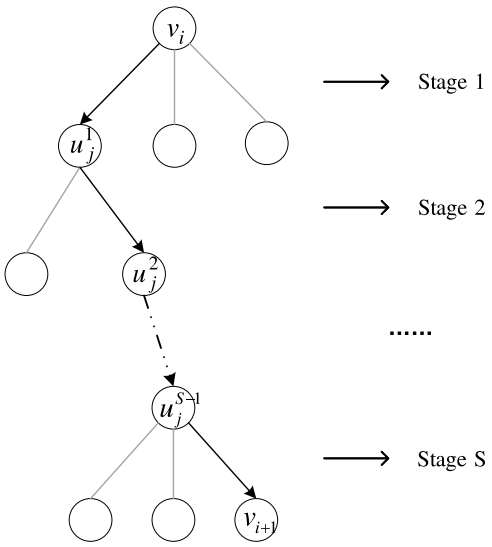


Fig. 2. An illustration of a hierarchical random walk step.

Stage 1: starts from the current entity v_i and randomly walks to an adjacent item u_j^1, according to the distribution $p(u_j^1|v_i)$.

Stage 2: starts from the current selected item u_j^1 and randomly walks to an adjacent item u_j^2, according to the distribution $p(u_j^2|u_j^1)$.

...

Stage S : starts from the current selected item u_j^{S-1} and randomly walks to an adjacent entity v_{i+1} that must belong to the pre-given set v, according to the distribution $p(v_{i+1}|u_j^{S-1})$.

Intuitively, the hierarchical random walk method decomposes the distribution $p(v_{i+1}|v_i)$ with the help of U, i.e., the specific distribution $p(v_{i+1}|v_i)$ is re-represented as a distribution $p(u_j|v_i)$ on top of a set of sub-distributions. Each time it samples a new entity v_{i+1} starting from v_i, it actually hierarchically samples a sub-distribution $p(v_{i+1}|u_j^1)$ or $p(u_j^2|u_j^1)$ first according to $p(u_j^1|v_i)$, and then samples the sequential sub-distributions hierarchically. The sampling continues until it reaches v_{i+1} according to $p(v_{i+1}|u_j^{S-1})$.

4.2. HiWalk: hierarchical random walk based representation learning method

For each item u in the network, HiWalk constructs an entity sequence with its neighboring entities. The entity sequence is constructed by maximizing $p(v_{i+1}|v_i)$ step by step through hierarchical random walks. To ensure the robustness of the algorithm, we also consider the following strategies for the hierarchical random walk procedure. First, for each entity set $\{v\}_k$, the above sequence construction procedure terminates when all entities in the given set have appeared in the resulting sequence. However, imagine there are some entities that interact with some items far more than other nodes. In such situations, these entities will preempt the appearing opportunities and the termination of the procedure will be uncertain. To avoid this situation, we further limit that the appearance time of an entity is identical with its weight in the given set. Second, it is possible that for a specific entity there is no other item connecting to it. Thus, a certain amount of dropout

is permitted in each stage to make sure that the random walk is sustainable.

If we denote the set of nodes that have edges with an arbitrary node x as $N(x)$ and denote the set of entities that have edges with item u_k as $\{v\}_k$, the hierarchical random walk procedure is as seen in Table 1.

With the above hierarchical random walk procedure, for a specific heterogeneous network $G = (U, V, E)$, a collection of entity sequences can be generated by taking the network structure as the background knowledge. Then these sequences can be used as the input of the skip-gram model to learn entity embeddings. Specifically, Eq. (1) is used as the original objective function, where the conditional probability $p(v_{t+j}|v_t)$ is defined by the softmax function over the inner product of the corresponding entity embeddings. For the sake of computing, the negative sampling strategy is adopted, so the objective function can be transformed to

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\left\{\log\sigma(\overrightarrow{v_{t+j}}\cdot\overrightarrow{v_t}) + \sum_{z=1}^{Z}\mathbf{E}_{v_t^z\sim V_{\neg N(v_t)}} \times\left[\log\sigma(\overrightarrow{v_t^z}\cdot\overrightarrow{v_t})\right]\right\}, \tag{11}$$

where σ denotes the sigmoid function, and $V_{\neg N(v_t)}$ denotes the entity collections who are not within the same window with v_t. Then the Stochastic Gradient Descent method is used to optimize the objective function.

HiWalk runs faster than the general random walk methods employed for learning node embeddings in heterogeneous networks because some unnecessary computations are avoided through this hierarchical sampling strategy. At the same time, the original probabilistic distribution is still maintained during the sampling, thus it is more effective than homogeneous network representation learning methods. Moreover, in some specific situations, such as when edge weights are not given, HiWalk is able to leverage more information compared with the peer methods.

4.3. Complexity analysis and comparison

We start with the efficiency analysis of different representations for the networks. For the sake of convenience, we discuss the circumstance of bipartite graphs where items are the same type. However, in the more complicated circumstances established by general heterogeneous networks, the analysis should be different but the conclusions are consistent.

Assume that there are M items and N entities in a specific bipartite graph, and for each node there are k edges on average. Thus, in this network, there are $(M + N)$ nodes and Nk edges in total, and each item has Nk/M edges on average.

Following the research on homogeneous networks, this bipartite graph can be transformed into a homogeneous network by connecting entities that have common items. Therefore, in the resulting homogeneous network, each entity has Nk^2/M edges. There will be N nodes and $N^2k^2/2M$ edges in total. In real-world networks, there are usually no big differences between M and N, while the average degree k is usually very large. Thus, the space consumption of the homogeneous network is larger compared with the original bipartite graph under sparse representations.

Next, we compare the complexity of HiWalk and other typical algorithms. The differences mainly come from the random walk procedures. To achieve the best performance, most random walk algorithms adopt the Alias method [26], which can sample from a probability distribution in $O(1)$ time. However, the Alias method needs to compute and maintain an assignment table for each probability distribution. The time and space consumption of the assignment table is $O(n)$, where n is the dimension of the distribution. For the proposed HiWalk, the time and space consumption of pre-construction is $O(Nk + M \times Nk/M)$, which is identical with $O(Nk)$. For DeepWalk, which performs uniform random walks on homogeneous networks, the time and space consumption of pre-construction is $O(N \times Nk^2/M)$, i.e., $O(N^2k^2/M)$. For more sophisticated methods like Node2vec, the probability distribution varies according to the number of nodes and edges; more specifically, the time and space consumption of preconstruction is $O(N * (N^2k^2/M) \times (Nk/M))$, i.e., $O(N^4k^3/M^2)$.

Table 1
HiWalk entity sequences generation workflow.

Input: A heterogeneous network $G = (U, V, E)$, dropout rate β
Output: A collection of node sequences **s**

```
HiWalk:
  Initialize s = ∅
  for u_k in U, do
    {v_k} ← N(u_k) ∩ V
    v_k1 ← entity sampled from {v}_k
    [v]_k ← [v_k1]
    i = 1
    while {v}_k ≠ ∅, do
      x ← node sampled from N(v_k1)
      while x ∈ U, do
        if dropout according to β
          x' ← node sampled from N(x)
        else
          x' ← node sampled from {v}_k
        x ← x'
      v_k(i+1) ← x
      add v_k(i+1) to [v]_k
      remove v_k(i+1) from {v}_k
      i = i + 1
    add [v]_k to s
  return s
```

5. Experiments

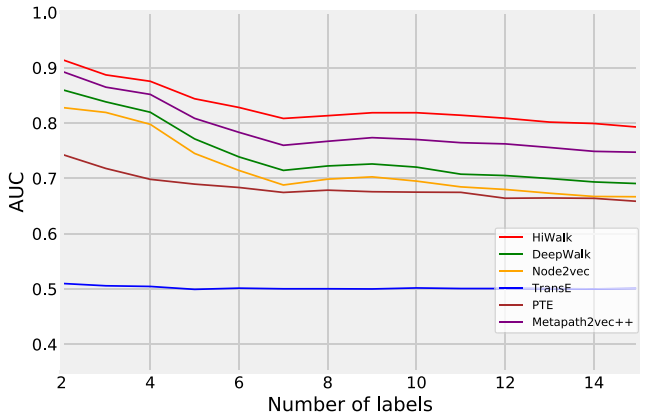
We evaluated the proposed HiWalk by comparing it with some baseline methods using multiple datasets and tasks. In two heterogeneous networks constructed by the datasets, we conducted the multi-label classification experiment for the entities and the classification experiment for items, as well as the link prediction experiment among entity–item pairs. Furthermore, we analyzed the performance of HiWalk under different parameter settings.

5.1. Setups

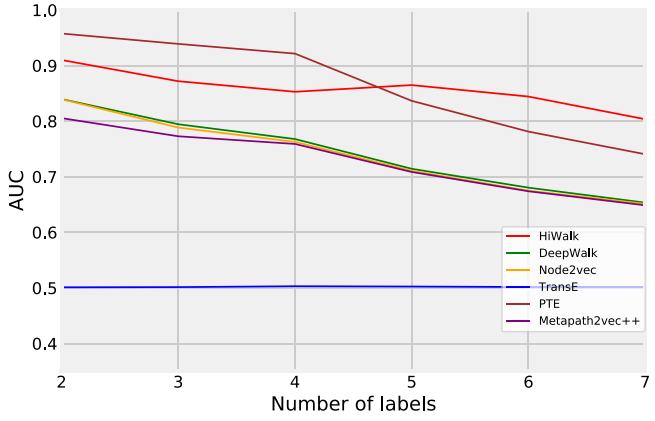
Two real-word datasets, DBLP and Amazon, are used in the following experiments. Both datasets contain more than one type of nodes, which is convenient for constructing heterogeneous networks. To compare different methods, we built both heterogeneous networks and homogeneous networks. For heterogeneous networks, we chose "people" as the entities that we analyze with the HiWalk method and another type of nodes as the items; for homogeneous networks, the nodes that represent "people" are connected when they have common neighbors in the aforementioned heterogeneous networks. Both datasets have group information that can be used as the ground truth in classification-related tasks. The details of the datasets are as follows.

- **DBLP** is a bibliographical dataset collected by the DBLP website.[1] This dynamic increasing dataset has 619,626 authors and 216,636 effective publications in our downloaded version. We constructed a heterogeneous network by treating the authors as entities and the publications as items. DBLP also contains the publisher information for each publication

[1] http://dblp.uni-trier.de/xml/.



(a) results on DBLP



(b) results on Amazon

Fig. 3. Multi-label entity classification results.

(conference name, journal name or book title), which can be used as the ground truth in the experiments.

- **Amazon** is a product review dataset collected from Amazon [27], which has 1,409,103 users and 193,387 effective products. This dataset has user ID, product ID, and product group information in each review. Knowing that each product review is related to a purchasing behavior, a heterogeneous network can be constructed based on this dataset. However, the single computer (with 64G memory) was not affordable for constructing the Alias table when running Node2vec on the whole network. So we reduced the dataset by using the products with between 5–50 related users. This produced a refined heterogeneous network with 820,001 entities and 166,933 items, as well as a homogeneous network with 820,001 nodes.

The baselines adopted for comparison are the state-of-the-art network representation learning algorithms that have been proved to be effective in their own fields.

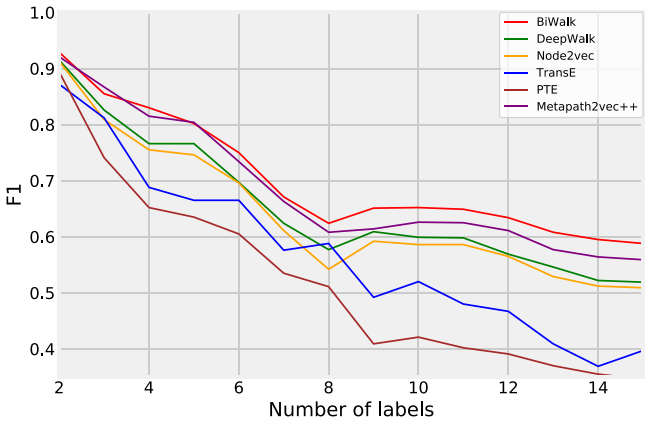
- **HiWalk** is our purposed method, which learns embeddings for the given type of nodes in heterogeneous networks. In the following experiments, we set the dropout rate as 0.2, the window size of the skip-gram model as 5, the number of entity sequences as 10 times the item number, and the length of the entity sequences varies according to the items: for each item, the length of the corresponding sequence is the doubled number of its neighboring entities.
- **DeepWalk** [12] is a well-known node embedding learning method that performs uniform random walks on homogeneous networks. The basic settings of DeepWalk were identical with that of HiWalk, i.e., the number of node sequences was set the same as HiWalk, and the length of node sequences equals to the average length of the sequences in the HiWalk model to make the comparison sensible.
- **Node2vec** [14] learns homogeneous network node embeddings by combining both Breadth-first Sampling and Depth-first Sampling. The basic settings of Node2vec were the same as that of DeepWalk and HiWalk, while the hyper parameters p and q were set as 2 and 0.5 respectively, which made it behave more in a breadth-first sampling style.
- **PTE** [10] learns embeddings for all types of nodes in heterogeneous networks through a semi-supervised training model. The window size of the skip-gram model is 5, and the number of samples in a training batch is 300.
- **TransE** [7] learns embeddings for both entities and relations in general heterogeneous networks. In the experiment we let TransE sample 10,000 times, using 200 triples each time.
- **Metapath2vec++** [11] learns embeddings for all types of nodes in heterogeneous networks by following a predefined metapath scheme. The basic settings of Metapath2vec++ were the same as that of DeepWalk. The metapath used in the experiments is "A-P-A" (which can be translated as "entity-item-entity" in our method).

In the following experiments, we set the edge weights as 1 for all the networks. The dimension of node embeddings was fixed as 128 for all methods. When training the classifiers, we split datasets into 70% training set and 30% test set.

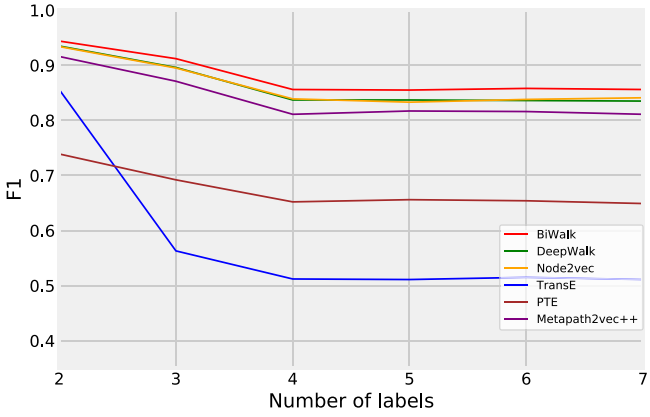
5.2. Multi-label entity classification

In this experiment, we classified the entities according to the groups that their items belonged to, i.e., for each entity we used its embedding as input and predicted a binary vector, where each dimension of this vector corresponded to a specific group. We compared different methods by using the entity embeddings learned through these methods as input and evaluated the multi-label classification results. For DBLP, we chose the first 2 to 15 groups according to the item numbers to run the multi-label classification; for Amazon, we chose the first 2 to 7 groups because only a small number of items existed in the last few groups. Due to data imbalance, we chose Decision Tree as the classifier and AUC score as the metric. AUC score denotes the area under the ROC curve, which is commonly adopted in class imbalance situations [28].

The experimental results are shown in Fig. 3. In most cases, HiWalk outperforms the state-of-the-art baseline methods on the multi-label entity classification task. For all of the labels we tested on the DBLP dataset, the improvements of HiWalk compared with the best results of the baselines are 2.5% to 7.2%; on the Amazon dataset, PTE performs better than HiWalk when the label size is less than 5. Generally speaking, DeepWalk and Node2vec learns node embeddings in homogeneous networks, PTE and Metapath2vec++ are good at learning multiple types of nodes, and TransE is good at learning multiple relationships. However, HiWalk learns targeted node embeddings by taking full advantage of the information in the heterogeneous networks. This is why HiWalk excels in this experiment in most cases. We still noticed that PTE performs better than HiWalk sometimes; this may be because PTE is a semi-supervised method that use the label information during the training. It can also be seen from the figures that there are tiny fluctuations in the curves sometimes, which may be because more labeled data made



(a) results on DBLP



(b) results on Amazon

Fig. 4. Item classification results.

the classifiers better trained, but the growing number of labels raised the difficulty of classification. Generally speaking, the AUC scores decrease as the label number increases. However, we can see that the decrement of the curves corresponding to HiWalk is slighter than the other methods.

5.3. Item classification

We conducted classification experiments for the items by taking their group information as the ground truth. For HiWalk, DeepWalk, and Node2vec, we constructed vector representations for each specific item by computing the center of its corresponding neighboring entities. For TransE, PTE and Metapath2vec++, which learn embeddings for all types of nodes in the heterogeneous network, we extract the embeddings for the items directly. Following the entity classification experiment, we still chose the first 2 to 15 groups in the DBLP dataset and the first 2 to 7 groups in the Amazon dataset. Since there is no multi-label output in this task, we chose Logistic Regression with L2 regularization as the classifier and F1 score as the metric. The experimental results are shown in Fig. 4.

Fig. 4 shows that HiWalk outperforms the other methods in the item classification task. However, HiWalk's improvement in this experiment is not as significant as that in the multi-label entity classification experiment—the improvements of HiWalk compared with the best results of the baselines are only 1.8% to 2.4% for the Amazon dataset, and sometimes HiWalk performs similarly to Metapath2vec++ for the DBLP dataset. This may be because, on the one hand, the classifiers and metrics used in the two experiments are different (due to the nature of different tasks); on the other hand, the item information is represented by simply adding the embeddings of corresponding entities in HiWalk, and part of the information may be lost during the compression.

Table 2
Link prediction experimental results.

Methods	Datasets	
	DBLP	Amazon
HiWalk	**0.945**	**0.953**
DeepWalk	0.664	0.526
Node2vec	0.667	0.524
TransE	0.501	0.557
PTE	0.647	0.708
Metapath2vec++	0.676	0.670

5.4. Link prediction

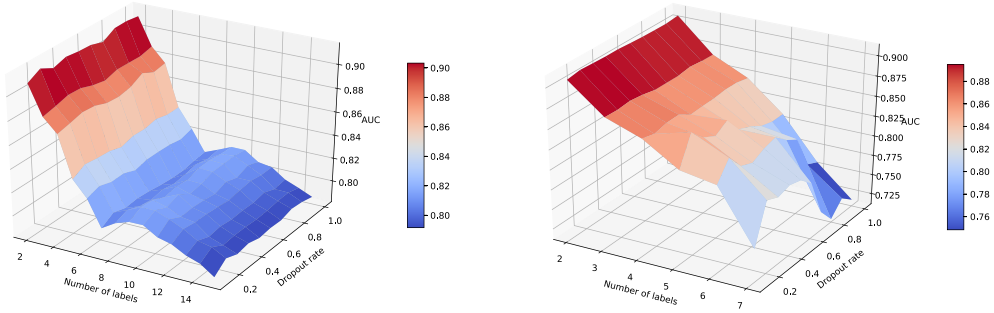
We conducted a link prediction experiment on both datasets. Different from homogeneous network-oriented link predictions, which predict links between node pairs that belongs to the same kind, here we conducted a more precise prediction on the "entity-item" pairs. For each item in the network, we randomly selected one of its neighboring entities and removed the corresponding edges to make the positive cases; to construct the negative cases, we randomly selected an uncorrelated entity for each item. The goal was to predict whether the edge existed between the specific entity–item pairs, so we regarded this task as a binary classification problem.

We assumed that if there is an edge between an entity and an item, their locations in the low-dimensional vector space should be close to each other. To test the quality of the node embeddings learned by different models, we simply computed the difference value of the corresponding entity embeddings and the item vectors (the method of getting item vectors here was mentioned in the item classification experiment) as the classifier input. We used SVM with an RBF kernel as the classifier, with prediction accuracy as the metric. Table 2 lists the experimental results.

From Table 2 we find that the prediction accuracy of HiWalk significantly outperforms the other methods (39.8% and 34.6% for the DBLP and Amazon datasets, respectively, compared with the best baseline results). The performance improvement of HiWalk is much larger than that in the previous experiments. Unlike previous tasks that judged the nodes' groups, link prediction further analyzes the relationship between different types of nodes. In heterogeneous networks this relationship is presented explicitly, but in homogeneous networks this relationship is masked and confused by the connections among nodes. This is the reason why HiWalk performed much better than DeepWalk and Node2vec in the link prediction experiment. However, it is surprising that HiWalk also outperforms PTE, Metapath2vec++, and TransE. This shows that although HiWalk only learns the entities embeddings explicitly, the structural semantic information between entities and items can be well represented through the entity embeddings.

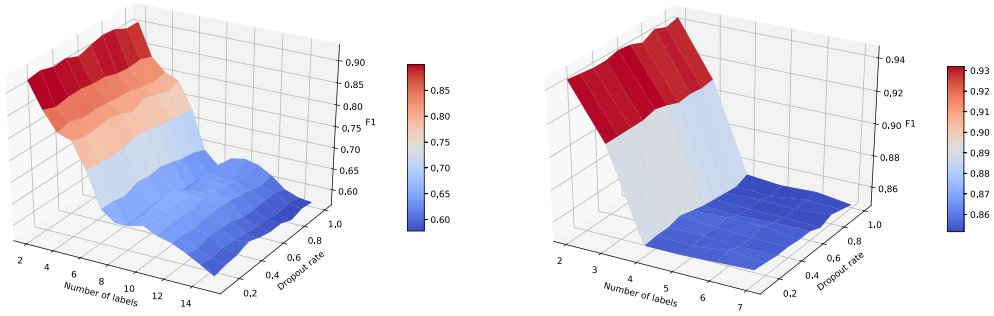
5.5. Parameter evaluation

To evaluate the influences that different parameters have on the performance of HiWalk, we conduct multi-label entity classification and item classification on the DBLP and Amazon datasets under different parameter settings. Three parameters are evaluated during the experiments, which are (1) the dropout rate, (2) the entity sequence length, and (3) the total number of entity sequences.



(a) entity classification results on DBLP

(b) entity classification results on Amazon



(c) item classification results on DBLP

(d) item classification results on Amazon

Fig. 5. Evaluation on the dropout rate.

5.5.1. Dropout rate

We let the dropout rate of the HiWalk models range from 0.1 to 1 (step length is 0.1) and compare the experimental results of these HiWalk models. Here a higher dropout rate means more randomness in the entity sequence construction, i.e., "dropout = 0.0" means that the entity sequence is constructed following the HiWalk steps strictly, while "dropout = 1.0" means that the entity sequence is constructed through completely random choices. To ensure the robustness of the models, we let the dropout rate start from 0.1, and the dropout is triggered automatically when there was no entity selected by HiWalk. The other settings were the same as the previous experiments. The experimental results of dropout rate tests on different datasets and different tasks can be seen in Fig. 5.

From Fig. 5 we can see that when the dropout rate increases, the performance of HiWalk declines to some extent. The best experimental results are achieved when the dropout rate ranges from 0.1 to 0.5, while the worst results occur when the dropout rate ranges from 0.8 to 1.0, which is identical with our analysis. Moreover, by comparing Fig. 5(b) and Fig. 5(d), we discover that the performance gap in the item classification task among dropout rates is not as significant as that in the node classification task. This can also be explained by information loss during the compression of node embeddings.

5.5.2. Entity sequence number

To test how the entity sequence number influences the model performance, we let it range from 1 time to 10 times the item number in the corresponding networks. More specifically, if the entity sequence number parameter is set as i, then for each item in the network, there will be i entity sequences generated. The other settings were the same as the previous experiments. The experimental results are shown in Fig. 6.

Fig. 6 shows that the number of entity sequences has a significant influence on the model performances. With increased number of sequences, the classification results become better, and then stabilize usually when the parameter is around 5. This is because more training data helps the skip-gram model train better, but the marginal effect decreases as the number of training samples increases. We can also find that in the item classification experiments, the influence of the sequence number is not as significant as that in the entity classification experiments, which is similar to the previous experiments.

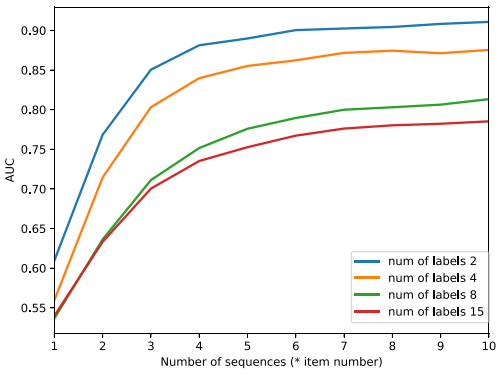
5.5.3. Entity sequence length

In this experiment we test the influences that the entity sequence length has on the model performance. We let the parameter of sequence length range from 1 to 10. More specifically, if the parameter is set as i, then for each sequence its length will be i times the entity number that the corresponding item has. The other settings were the same as the previous experiments. Fig. 7 shows the experimental results.

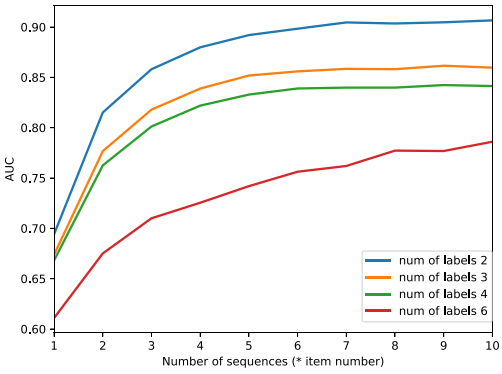
Fig. 7 implies that in most cases the entity sequence length parameter has no significant influence on the model performance. From the experiment on entity sequence number we know that more training data will help the model training, however this effect is not obvious here. This may because the single time of the sequence length is enough for model training, and additional length cannot bring prominent marginal effect for the model performance.

6. Conclusion

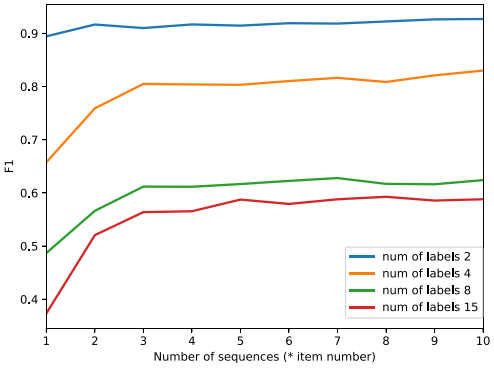
In this paper, we proposed a hierarchical-random-walk-based network representation learning method named HiWalk. HiWalk learns the embeddings for given type of nodes in heterogeneous networks by combining hierarchical random walk adopted on the



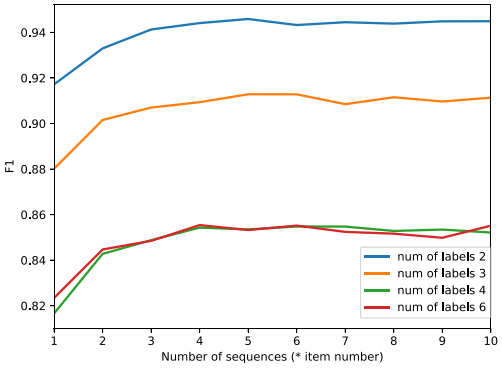
(a) entity classification results on DBLP



(b) entity classification results on Amazon

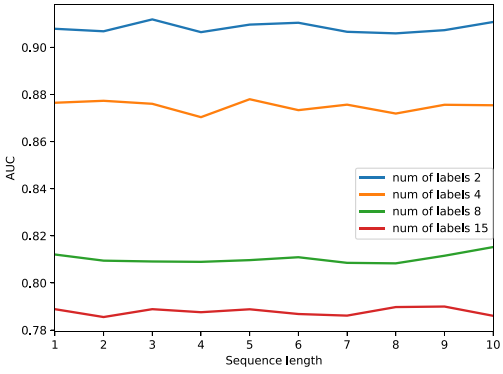


(c) item classification results on DBLP

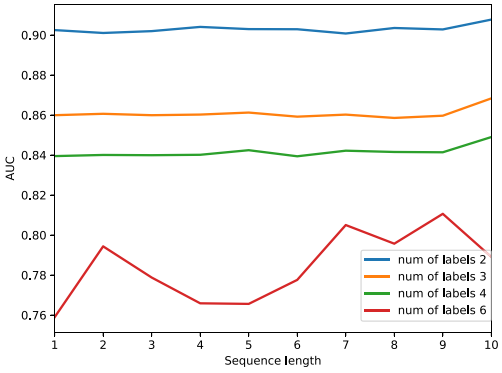


(d) item classification results on Amazon

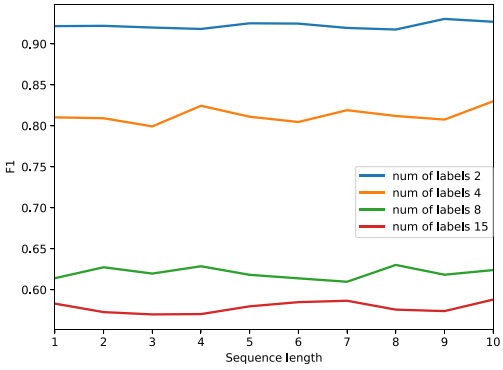
Fig. 6. Evaluation on the entity sequence number.



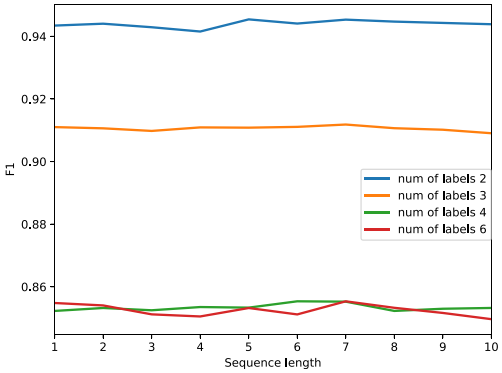
(a) entity classification results on DBLP



(b) entity classification results on Amazon



(c) item classification results on DBLP



(d) item classification results on Amazon

Fig. 7. Evaluation on the entity sequence length.

heterogeneous networks and the skip-gram model that learns embeddings for continues node sequences. Compared with existing node embedding learning algorithms, HiWalk absorbs the advantages from both homogeneous-network-oriented and heterogeneous-network-oriented representation learning methods, which is more effective, efficient and concentrated in heterogeneous network representation learning. Thus, for many real-world applications and information systems, HiWalk can perform superior heterogeneous network analysis works. Our contributions are as follows.

- We discussed the problem of learning embeddings for a specific type of nodes in heterogeneous networks. By doing so, both the effectiveness and efficiency can be improved.
- We proposed HiWalk, a node embedding learning method that takes advantage of both the high-efficiency representation learning models applied in homogeneous networks and the rich information contained in heterogeneous networks.
- We validated the effectiveness of HiWalk by comparisons with the state-of-the-art algorithms through comprehensive experiments.

Despite the above advantages of HiWalk, there are still some problems. First, meaningful information may be lost when constructing the item representations via node embeddings indirectly, which has been found in the experiments. Thus, it is necessary to develop a method to take full advantage of heterogeneous network information. Second, many details of the HiWalk design need to be improved. Third, the effectiveness of HiWalk needs to be evaluated through more practical datasets and applications. We will focus on these directions in our future research.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant No. 2016QY02D0305, the National Natural Science Foundation of China under Grant Nos. 71621002, 71602184, 71202169, 61671450, U1435221, as well as the Key Research Program of the Chinese Academy of Sciences under Grant No. ZDRW-XH-2017-3. Specifically, we would like to thank the anonymous reviewers who helped us to improve this work.

References

[1] J. Han, Y. Sun, X. Yan, P.S. Yu, Mining heterogeneous information networks, in: Tutorial at the 2010 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM.
[2] R.W. Heath, M. Kountouris, T. Bai, Modeling heterogeneous network interference using poisson point processes, IEEE Trans. Signal Process. 61 (16) (2013) 4114–4126.
[3] Y. Sun, B. Norick, J. Han, X. Yan, P.S. Yu, X. Yu, Integrating meta-path selection with user-guided object clustering in heterogeneous information networks, ACM Trans. Knowl. Discov. Data 7 (3) (2013) 11:1–11:23.
[4] C. Luo, R. Guan, Z. Wang, C. Lin, Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks, in: European Conference on Information Retrieval, Springer, pp. 210–221.
[5] Y. Cui, L. Zhang, Q. Wang, P. Chen, C. Xie, Heterogeneous network linkage-weight based link prediction in bipartite graph for personalized recommendation, Procedia Comput. Sci. 91 (2016) 953–958.
[6] J. Bai, L. Li, D. Zeng, J. Lin, Social role clustering with topic model, in: Proceedings of the 14th IEEE Conference on Intelligence and Security Informatics, IEEE, pp. 211–213.
[7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, nips, pp. 2787–2795.
[8] Y. Chen, C. Wang, Hine: Heterogeneous information network embedding, in: International Conference on Database Systems for Advanced Applications, Springer, pp. 180–195.
[9] Z. Liu, V.W. Zheng, Z. Zhao, F. Zhu, K.C.-C. Chang, M. Wu, J. Ying, Semantic proximity search on heterogeneous graph by proximity embedding, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence, aaai, pp. 154–160.
[10] J. Tang, M. Qu, Q. Mei, Pte: Predictive text embedding through large-scale heterogeneous text networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1165–1174.
[11] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 135–144.
[12] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 701–710.
[13] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, ACM, pp. 1067–1077.
[14] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 855–864.
[15] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, Curran Associates Inc, pp. 3111–3119.
[16] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.
[17] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[18] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[19] L. Tang, H. Liu, Scalable learning of collective behavior based on sparse social dimensions, in: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, pp. 1107–1116.
[20] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, A.J. Smola, Distributed large-scale natural graph factorization, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 37–48.
[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013.
[22] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, Association for Computational Linguistics,
[23] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, aaai, pp. 2181–2187.
[24] G. Ji, K. Liu, S. He, J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, aaai, pp. 985–991.
[25] S. Chang, W. Han, J. Tang, G.-J. Qi, C.C. Aggarwal, T.S. Huang, Heterogeneous network embedding via deep architectures, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 119–128.
[26] A.J. Walker, New fast method for generating discrete random numbers with arbitrary frequency distributions, Electron. Lett. 10 (8) (1974) 127–128.
[27] J. Leskovec, L.A. Adamic, B.A. Huberman, The dynamics of viral marketing, ACM Trans. Web 1 (1) (2007) 5.
[28] T. Fawcett, An introduction to roc analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.