MARU: Meta-context Aware Random Walks for Heterogeneous Network Representation Learning

Jyun-Yu Jiang, Zeyu Li, Chelsea J.-T. Ju and Wei Wang
Department of Computer Science, University of California, Los Angeles, CA, USA
{jyunyu,zyli,chelseaju,weiwang}@cs.ucla.edu

ABSTRACT

Information networks, such as social and citation networks, are ubiquitous in the real world so that network analysis plays an important role in data mining and knowledge discovery. To alleviate the sparsity problem of network analysis, it is common to capture the network semantics by projecting nodes onto a vector space as network embeddings. Moreover, random walks are usually exploited to efficiently learn node embeddings and preserve network proximity. In addition to proximity structure, heterogeneous networks have more knowledge about the types of nodes. However, to profit from heterogeneous knowledge, most of the existing approaches guide the random walks through predefined meta-paths or specific strategies, which can distort the understanding of network structures. Furthermore, traditional random walk-based approaches much favor the nodes with higher degrees while other nodes are equivalently important for the downstream applications. In this paper, we propose Meta-context Aware Random Walks (MARU) to overcome these challenges, thereby learning richer and more unbiased representations for heterogeneous networks. To reduce the bias in classical random walks, the algorithm of bidirectional extended random walks is introduced to improve the fairness of representation learning. Based on the enhanced random walks, the meta-context aware skip-gram model is then presented to learn robust network embeddings with dynamic meta-contexts. Therefore, MARU can not only fairly understand the overall network structures but also leverage the sophisticated heterogeneous knowledge in the networks. Extensive experiments have been conducted on three real-world large-scale publicly available datasets. The experimental results demonstrate that MARU significantly outperforms state-of-the-art heterogeneous network embedding methods across three general machine learning tasks, including multi-label node classification, node clustering, and link prediction.

KEYWORDS

Network Embedding; Heterogeneous Information Networks; Representation Learning; Feature Learning

ACM Reference Format:

Jyun-Yu Jiang, Zeyu Li, Chelsea J.-T. Ju and Wei Wang. 2020. MARU: Meta-context Aware Random Walks for Heterogeneous Network Representation Learning. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412040

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412040

1 INTRODUCTION

Network analysis has already been a prevalent research topic because of its enormous potential in many downstream applications, such as node classification [33], node clustering [22], and link prediction [20]. More specifically, most of the important tasks in network analysis involve predictions over nodes and edges. However, the sparsity of networks usually results in significant difficulty of generalization for machine learning models. To resolve this issue, one of the most popular approaches is to map nodes to continuous low-dimensional representations as embeddings that preserve the structural information and semantics of nodes [2].

To efficiently learn node representations, random walks have been widely exploited to preserve the proximity between node pairs [14, 24]. More precisely, the embedded representations of nodes are optimized to infer the nearby nodes on random walks [24] with a skip-gram model [21] inspired by word embedding in the field of natural language processing [21]. Moreover, the complicated proximity structures of networks can be also gained by sampling biased random walks [14]. Practically, each of the generated random walks can be treated as a word sequence so that the task of network embedding is equivalent to the setting of word embedding [9, 14, 18, 24]. More specifically, a sliding window is applied to capture the nearby nodes as the context for each node over random walks. To ensure the coverage of the nodes for learning representations, most of the existing approaches simply sample a few random walks starting from each of the nodes. However, there are a few shortcomings for the existing sampling approaches. First, one-directional random walks that evenly start from all of the nodes would favor nodes with higher degree and betweenness scores when nodes in the network should be equally important for the downstream applications. Second, tail nodes tend to be visited at the very beginning of random walks, especially for the random walks starting from them. As a result, the number of context nodes in the sliding window will be much underestimated for the tail nodes. In addition, the tail nodes will have fewer chances to be observed as the context of other nodes during optimization.

Compared to homogeneous networks with a singular type of node, heterogeneous networks with various types of nodes are more common in real-world applications. Although the homogeneous network embedding methods can still learn the representations for heterogeneous networks, the information of node types can be significantly neglected. As a result, the semantics of the heterogeneous knowledge in networks is totally lost in the embeddings. To leverage the heterogeneous knowledge in networks for representation learning, existing methods usually rely on meta-paths [29], which are predefined sequences of node types. In other words, different meta-paths indicate distinct human-explainable semantics. For example, the meta-paths APA and APVPA are used to indicate that two authors had co-authorship and published papers in the same venue respectively, where A, P, and V are the node types referring to author, paper, and venue in a heterogeneous bibliographic network. To exploit the meta-paths, most of the existing heterogeneous network embedding methods guide the generated random walks through a predefined set of meta-paths so that the prior knowledge can be incorporated into the produced node sequences [6, 9, 11, 27]. For instance, each meta-path can be solely applied to measure the relationship between two nodes with a short random walk [11, 27]; different meta-paths may also overlap to approximate longer random walks as a mixture of prior knowledge [6, 9]. However, the choices of random walk significantly affect the quality of network representations [18]. Accordingly, the requirement of high-quality meta-paths that are hand-picked by domain experts leads to reduced robustness for general tasks. In addition, the usage of meta-paths can limit and distort the understanding of the network structures. More precisely, given a limited set of meta-paths, a new path in a network is less likely to be induced. Even though some works [18] have proposed to employ specific strategies to guide random walks instead of using meta-paths, adjusted random walks can still be biased and overlook some vital network structures.

To learn network representations with random walks, one of the most popular optimized approaches is the skip-gram model inspired by word embedding in the field of natural language processing [21]. Each of the generated random walks can be treated as a word sequence so that the task of network embedding is equivalent to the setting of word embedding [9, 14, 18, 24]. More specifically, a sliding window is applied to capture the nearby nodes as the context for each node over random walks. To ensure the coverage of the nodes for learning representations, most of the existing approaches simply sample a few random walks starting from each of the nodes. However, there are a few shortcomings for the existing sampling approaches. First, one-directional random walks that evenly start from all of the nodes would favor nodes with higher degree and betweenness scores when nodes in the network should be equally important for the downstream applications. Second, tail nodes tend to be visited at the very beginning of random walks, especially for the random walks starting from them. As a result, the number of context nodes in the sliding window will be much underestimated for the tail nodes. In addition, the tail nodes will have fewer chances to be observed as the context of other nodes during optimization.

In this paper, Meta-context Aware Random Walk (MARU) is proposed to address the limitations of the existing heterogeneous network embedding approaches. More specifically, we focus on deriving robust embeddings that are more comprehensive and fair to represent the heterogeneous networks. The algorithm of bidirectional extended random walks is first introduced to alleviate the bias caused by classical random walks. Instead of manipulating random walks [9, 14, 18], we employ general random walks for a more comprehensive understanding of network structures and encode the types of surrounding nodes as meta-contexts to incorporate heterogeneous knowledge. Given a node and its meta-contexts in the random walk, we extend the skip-gram model to infer not only the nearby nodes but also their corresponding meta-contexts. In other words, the learned representations can reflect various situations in terms of different meta-contexts, thereby describing the nature of heterogeneous networks more precisely. Here, we summarize our contributions in the following.

- To the best of our knowledge, this paper is the first work to address the bias of classical random walks for network representation learning. For the tail nodes with lower degree and betweenness scores, the proposed bidirectional extended random walks can capture the context and optimize the representations more fairly and comprehensively.
- We propose the framework MARU, generating network representations that simultaneously capture general network structures and local heterogeneous knowledge. More specifically, leveraging the types of surrounding nodes as meta-contexts enable the model to represent different semantics according to local contexts in random walks. Hence, the learned network representations are more robust to preserve the properties of heterogeneous networks.
- Extensive experiments conducted on three large-scale real-world datasets indicate that MARU significantly outperforms existing heterogeneous network embedding methods. A study of parameter sensitivity then demonstrates the robustness of the proposed framework across different situations. In addition, we will release our implementations to facilitate future research.

2 RELATED WORK

Networks can be categorized into two types, including *homogeneous* and *heterogeneous* networks. Homogeneous networks contain a single node type, e.g., social networks of users, whereas heterogeneous networks involve multiple types of nodes, such as citation networks of authors, papers, and venues. Network representation learning for both categories aims at mapping nodes in graphs to low-dimensional continuous vectors. These low-dimensional vectors are learned to capture the essential information of the nodes, and consequently, better preserve the structure and semantic similarity among nodes.

A range of network representation learning algorithms has been proposed for homogeneous network embedding learning [5, 14, 24, 32, 36] and heterogeneous network embedding learning [4, 9, 11, 18, 31]. In this section, we briefly summarize these algorithms below.

2.1 Homogeneous Network Embedding Models

DeepWalk [24] is a pioneering representation learning approach for homogeneous networks. It explores the network structure through the *random walks* sampled from the network. Mapping to the concepts in work2vec [21], nodes and random walks are treated as words and sentences, respectively. The node representations can be learned by using the vanilla skip-gram model [21] on the random walks. The paradigm of DeepWalk has inspired many studies [9, 14, 23, 37] that are applied to diverse types of networks. *node2vec* [14] is one of the examples that extend DeepWalk by relaxing the definition of network neighborhood and designing a biased random walk procedure to explore more diverse node representations. However, previous literature has demonstrated that such walk generation methods introduce a bias towards the nodes with higher degrees [29]. Therefore, the structural and semantic information of the isolated or less connected nodes becomes difficult to be captured by the model, which eventually leads to the inefficiency of the training procedure and poor accuracy of the trained representations of nodes with lower degree numbers. Most importantly, the model is prone to preserve only the global structure [36], assuming that nodes with more common neighbors yield similar representations.

To better capturing the complicated underlying network structure, *LINE* [32] and *SDNE* [36] use edge-sampling algorithms to preserve both the local and global network structure. They model both the first-order proximity, defined as the proximity between directly connected nodes, and the second-order proximity, defined as the proximity between nodes that share common neighbors.

All of the aforementioned algorithms are specifically designed for homogeneous networks. In other words, they fail to take advantage of the diverse semantic relations encompassed in heterogeneous networks.

2.2 Heterogeneous Network Embedding Models

In order to comprehensively capture the rich semantics in edges and to better understand the different interactions between multi-typed nodes, heterogeneous information network embedding models are proposed. These methods either construct the embeddings for each modality defined beforehand, or learn all node embeddings together in the same latent space.

Most of the approaches that use predefined modality learn the node embeddings by minimizing the loss over each modality. HNE [4] presents a deep embedding framework that leverages a highly nonlinear multi-layered embedding function to capture the complex interactions. Each modality, such as image and text, is constructed separately. The embeddings of different modalities are then mapped to the same embedding space. Zhao et al. [38] specifically model the network structure of Wikipedia data that consists of three types of nodes: entities, words, and categories. It uses the coordinate matrix factorization technique to jointly learn the representations of these three types of nodes. PTE [31] is a semi-supervised representation learning method designed for text data. Based on the edge types, it decomposes the heterogeneous network into a set of bipartite networks. The method learns the embeddings of each node according to its one-hop neighbors, i.e. directly connected nodes, of the resulting bipartite networks. These approaches have demonstrated satisfactory performance in specific applications. Nevertheless, they can only capture limited types of relationships between nodes or miss the different semantics of relationships between nodes.

To address the caveat of explicit node types, several approaches have been proposed to incorporate meta-paths, which are sequences of node types, for heterogeneous graph embeddings. For instance, metapath2vec [9] is another extension of DeepWalk that uses meta-paths to capture the relationships between different node types. More specifically, a strategy for sampling random walks from heterogeneous networks is proposed to restrict random walks to follow particularly predefined transitions of node types. However, the set of meta-paths needs to be predefined while the selection of meta-paths significantly affects the performance. To avoid the requirement of meta-paths, Fu et al. [11] propose HIN2Vec to learn node representations by predicting the meta-paths as relations between nodes while Hussein et al. [18] manipulate the procedure of sampling random walks. Nevertheless, HIN2Vec suffers from the capability of solely inferring contexts of a node while manipulating random walks can lead to significant sampling bias. To address these problems, MARU does not require predefined meta-paths while the bidirectional extended random walk algorithm can theoretically reduce sampling bias. Moreover, any learned node representation is capable of predicting the contextual information on the network.

3 PROBLEM STATEMENT

In this section, we first introduce the notations of heterogeneous networks and then formally define the objective of learning heterogeneous network representations.

3.1 Heterogeneous Network

We first formally define the notations to represent heterogeneous networks. Note that the definition is consistent with previous studies [9, 28, 30].

Definition 3.1 (Heterogeneous Network). A heterogeneous network is defined as a graph $G = (V, E, T)$, where V is the set of nodes; $E \subseteq V \times V$ is the set of edges connecting nodes; T represents the set of node types. For each node $v \in V$, a mapping function $\psi(v) \in T$ indicates the corresponding type of the node.

To simplify the representation and implementation, for each node v, we denote the neighbors in the graph as

$$N(v) = \{v_i \mid \forall(v, v_i) \in E\},$$

which can be treated as an adjacency list [7] generated by the edge set E.

3.2 Problem Definition

We formalize the problem of learning heterogeneous network representations based on the aforementioned notations.

Problem 1 (Representation Learning for Heterogeneous Networks). *Given a heterogeneous network $G = (V, E, T)$, for each node $v \in V$, the task aims to learn a d-dimensional embedding vector $\Phi(v) : V \rightarrow \mathbb{R}^d$, where $d \ll |V|$, so that $\Phi(v)$ can capture the structural information and semantic knowledge of the node.*

More specifically, the network representations project nodes onto a d-dimensional continuous latent feature space. Note that although nodes can belong to different types, all of the nodes are projected on the identical feature space for the convenience of representing relationships among different nodes. As a result, the learned node representations can further benefit various data mining tasks for heterogeneous networks, such as node classification, node clustering, and link prediction. Moreover, heterogeneous network representation learning is an unsupervised machine learning task. In other words, the representations can be acquired with only the network and then directly applied to various downstream applications for heterogeneous network data mining. Therefore, the problem of heterogeneous network representation learning is important and beneficial.

Table 1: Summary of notations and their descriptions.

Notation	Descriptions
G	the heterogeneous network for learning representations
V	the set of nodes
E	the set of edges connecting nodes
T	the set of node types
$\psi(v)$	the function mapping a node v to the corresponding type
$N(v)$	the set of neighbors of the node v in the graph
d	the embedding dimension
l	the walk length for bidirectional extended random walk
k	the neighborhood size in the skip-gram model
w	the number of generated random walks per node
t	the meta-context size
r	the number of negative samples per neighbor
M	the set of available meta-contexts
$C(v, m, m_c)$	the context nodes with $m_c \in M$ for the node v with $m \in M$
$\Phi(v, m)$	the embedding of the node v with the meta-context m
$\Phi(v)$	the ultimate embedding of the node v

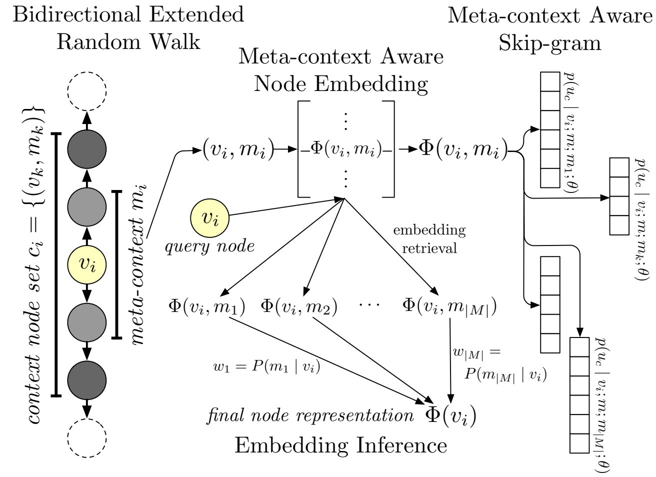


Figure 1: The schema of the proposed framework Meta-context Aware Random Walks (MARU).

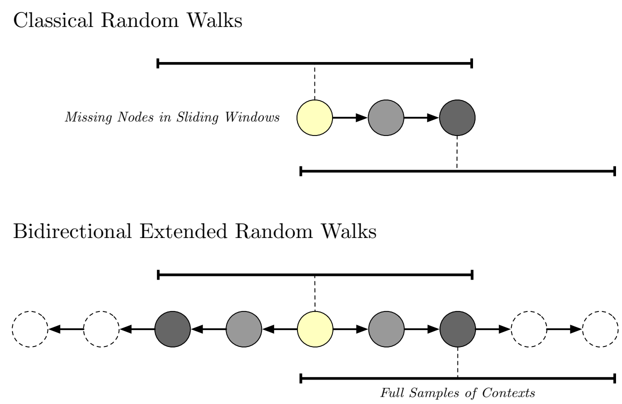


Figure 2: The illustrations of classical random walks and our proposed bidirectional extended random walks for learning network representations. The yellow nodes are the starting nodes of random walks while the white nodes with dotted strokes are the extended nodes. The lines are the sliding windows for the corresponding nodes for optimization.

4 THE *MARU* FRAMEWORK FOR HETEROGENEOUS NETWORK EMBEDDING

In this section, we present the proposed framework, Meta-context Aware Random Walks (MARU), for learning heterogeneous network representations.

4.1 Framework Overview

Figure 1 depicts the general schema of MARU. More specifically, the model mainly consists of four stages, including bidirectional extended random walks, meta-context aware node embedding, meta-context aware skip-gram, and embedding inference. To efficiently and adequately capture the structural information, bidirectional extended random walks guarantee the generality of sampled structures and the fairness of context information for each node in random walks. To properly encode the heterogeneous knowledge, the stage of meta-context aware node embedding represents a node with different embedding vectors for distinct meta-contexts, which are the types of surrounding nodes on random walks. Based on the meta-context aware embeddings, the meta-context aware skip-gram model optimizes the representations by inferring not only the context nodes but also their meta-contexts. Finally, the ultimate representation of a node can be computed as an aggregation of meta-context aware embeddings over the estimated distribution of meta-contexts for the node in the stage of embedding inference. In sum, Table 1 summarizes the major notations in this paper and the corresponding descriptions.

4.2 Bidirectional Extended Random Walks

One of the most efficient approaches of capturing the network structures is to sample a few random walks that cover the network and then optimize the proximity between nodes within a sliding window on the random walks. However, classical random walks result in significant biases. More precisely, simple random walks would favor the nodes with high degree and betweenness scores, especially for the walks with longer lengths [8]. In addition, conventional random walks also lead to the bias of underestimating the contextual information of tail nodes while learning network representations. Figure 2 shows how classical random walks are applied to network representation learning. For the endpoints of random walks, there can be at most half of nodes that are missing in the sliding windows for deriving the contexts. Moreover, the most typical approach to optimize tail nodes is to start a number of random walks from them. In other words, the contexts for the tail nodes can be highly underestimated, and thus reveal incorrect structural information.

To address this problem, we propose the algorithm of bidirectional extended random walks as presented in Algorithm 1. Instead of walking through only a single direction, the starting node is treated as the center of the walk that grows from both sides simultaneously. Furthermore, to secure the fairness of the observed contexts, the number of actual walking steps is extended according to the size of sliding windows in optimization. As shown in Figure 2,

Algorithm 1: BidirectionalExtendedRandomWalk(G, u, l, k)

Input: the graph G, the starting node u, the walk length l, the neighborhood size k
Output: the bidirectional extended random walk L

W = [u]
$v_f = v_b = u$
for *iter = 1 to* $\lceil \frac{l-1}{2} \rceil + k$ **do**
  v_f = RandomlySample($N(v_f)$) // forward step.
  v_b = RandomlySample($N(v_b)$) // backward step.
  W = $[v_b] + W + [v_f]$
return W

all of the nodes in the random walks can fairly have full samples of contexts for optimization. Moreover, bidirectional random walks can theoretically retrieve more tail nodes than one-directional random walks as shown in Corollary 1.

Corollary 1. *Assume head nodes are never transitioning to tail nodes in random walks, and the probability of transitioning between tail nodes is* $0 < p < 1$. *Given a tail starting node u and the walking length* $2n + 1$, *the expected number of tail nodes in a bidirectional random walk is greater than the expected number in a one-directional random walk.*

Note that we show the proof of Corollary 1 in Appendix A. As a result, the algorithm of bidirectional extended random walks is able to efficiently and fairly capture the structural information and provide enough knowledge for the optimization of network representation learning.

4.3 Meta-context Aware Skip-gram Model

To exploit the node types as heterogeneous information, most of the existing approaches guide the random walk through predefined meta-paths [9, 11] or specific strategies [18] before optimizing the proximity between nodes on random walks. However, these manipulations of random walks can distort the understanding of network structures. More specifically, a portion of network structures can be ignored or inadequately covered by manipulated random walks. Hence, we do not guide random walks with any external knowledge. Instead, *meta-contexts* are taken into account to exploit heterogeneous knowledge.

Meta-contexts on Random Walks. In this paper, meta-contexts are defined as the node types within a sliding window. The motivation is that a node should have different contexts of nodes for different local meta-contexts. For example, if the meta-contexts for the node of an author in a bibliographic network are APAPA, the corresponding contexts should be the authored papers and the co-authors instead of the published venues. To some degree, meta-contexts can be treated as the conditions of the particular segments in random walks. The idea is beneficial for the model to learn the dynamic structures in the networks. Formally, given a random walk as $L = \left[v_1, v_2, \cdots, v_{|L|}\right]$, the meta-contexts of the node v_i can be defined as:

$$m_i = \left(\psi\left(v_{i-t}\right), \cdots, \psi\left(v_i\right), \cdots, \psi\left(v_{i+t}\right)\right),$$

where t is the window size for meta-contexts. For simplicity, we denote M as the set of all possible meta-contexts that can be found in the sampled random walks.

Meta-context Aware Node Embedding. To incorporate the knowledge of meta-contexts into the model, we propose the meta-context aware node embedding, which considers a node with different meta-contexts separately. More precisely, instead of learning a stationary representation $\Phi(v)$ for a node v, the node can have distinct representations $\Phi(v, m)$ for different meta-contexts $m \in M$. Note that although meta-contexts can be encoded independently with conditional bits [13] or individual embeddings [34], both of the methods perform unsatisfactorily in our experiments. This observation is mainly due to the sophisticated network structures of our framework. Independently learning representations with different meta-contexts for a node can better model heterogeneous networks.

Meta-context Aware Skip-gram. Similar to the previous studies [9, 14, 18, 24], we extend the skip-gram model originally proposed in the field of natural language processing [21] to learn network representations with the concept of meta-contexts. In addition to the nearby nodes in random walks, we also optimize the likelihood of the corresponding meta-contexts for the context nodes. Given a heterogeneous network $G = (V, E, T)$, the objective of meta-context aware skip-gram model is to maximize the proximity between nodes in terms of local structures and meta-contexts as:

$$\underset{\theta}{\operatorname{argmax}} \sum_{v \in V} \sum_{m \in M} \sum_{u_c \in C(v,m,m_c)} \log p(u_c \mid v; m; m_c; \theta),$$

where θ is the set of model parameters; $u_c \in C(v, m, m_c)$ denotes the context nodes u_c with specific meta-contexts m_c for the node v with the meta-contexts m. Different from conventional skip-gram models that output a single multinomial distribution of all available nodes, the meta-context aware skip-gram model learns multiple multinomial distributions for different meta-contexts. More specifically, as illustrated in Figure 1, the likelihood $p(u_c \mid v; m; m_c; \theta)$ can be estimated by the learned meta-context aware node embeddings and the softmax function [12] as:

$$p(u_c \mid v; m; m_c; \theta) = \frac{\Phi(v, m) \cdot \Phi(u_c, m_c)}{\sum_{\forall u_i \in V_{m_c}} \Phi(v, m) \cdot \Phi(u_i, m_c)},$$

where V_{m_c} is the set of nodes that have been associated with the meta-context m_c. During the training process, positive samples are generated by retrieving neighbors in the random walks with a length-k sliding window while a negative sample u_n can be randomly drawn from the distribution $P(u_n \mid m_c)$ for each neighbor. Therefore, the model can be optimized by using the stochastic gradient descent algorithm [26].

Embedding Inference. To generate the representations of individual nodes, the ultimate node embeddings can be further computed by aggregating the meta-context aware node embeddings as:

$$\Phi(v) = \sum_{m} P(m \mid v) \cdot \Phi(v, m), \text{and } P(m \mid v) = \frac{\#(v, m)}{\sum_{m'} \#(v, m')},$$

where $\#(v, m)$ denotes the number of occurrences for the association of the node v and the meta-context m in the training random walks. Finally, Algorithm 2 gives the pseudocode of the whole meta-context aware skip-gram model.

Algorithm 2: MetaContextAwareSkipGram(G, w, l, k, t, r, M)

Input: the graph $G = (V, E, T)$, the number of walks per node w, the walk length l, the neighborhood size k, the meta-context size t, the number of negative samples per neighbor r, the set of available meta-contexts M.
Output: the node representations $\Phi(v) : V \to \mathbb{R}^d$

$\Phi^{\text{meta}} = \Phi^{\text{node}} = \emptyset$
for $iter\_w$ = 1 *to* w **do**
  for $u \in V$ **do**
    W = BidirectionalExtendedRandomWalk(G, u, l, k)
    for $i = k + 1$ *to* $k + l$ **do**
      for $j = i - k$ *to* $i + k$ & $i \neq j$ **do**
        Φ^{meta} = SGD(Φ^{meta}, $P(W_j \mid W_i; m_i; m_j; \theta) = 1$)
        for $iter\_n$ = 1 *to* r **do**
          Draw a negative sample $u_n \sim P(u_n \mid m_j)$
          Φ^{meta} =
          SGD(Φ^{meta}, $P(u_n \mid W_i; m_i; m_j; \theta) = 0$)
for $v \in V$ **do**
  $\Phi^{\text{node}}(v) = \mathbf{0}$
  for $m \in M$ **do**
    $\Phi^{\text{node}}(v) = \Phi^{\text{node}}(v) + P(m \mid v) \cdot \Phi^{\text{meta}}(v, m)$
return Φ^{node}

4.4 Complexity Analysis

Here we analyze the complexity of MARU.

For the time complexity, the bidirectional extended random walk algorithm spends $O(l + k)$ time to generate each random walk so that the overall time complexity for random walk generation is $O(w|V|(l+k))$. For each random walk, it costs $O\left(lkd \log\left(|V||M|\right)\right)$ time to update the skip-gram model with negative sampling for learning meta-context aware node embeddings. Finally, the embedding inference takes $O(|V||M|)$ to derive the ultimate node embeddings. Therefore, the overall time complexity of MARU is $O\left(wlkd|V|\left(\log|V| + \log|M|\right) + |V||M|\right)$.

For the space complexity, random walk generation requires $O(l+k)$ space as a buffer for the generated random walks. The meta-context aware node embeddings and ultimate node embeddings occupy $O(d|V||M|)$ and $O(d|V|)$ memory space while the skip-gram model has $O(d|M||V|)$ additional parameters. Hence, the overall space complexity of MARU is $O(l + k + d|M||V|)$.

5 EXPERIMENTS

In this section, we conduct extensive experiments and in-depth analysis to verify the quality of learned heterogeneous network representations and the robustness of MARU in three general machine learning tasks.

5.1 Datasets and General Experimental Settings

Dataset. In the experiments, we adopt three large-scale publicly available heterogeneous network datasets, including DBIS [29], MovieLens [16], and Yelp [3]. Table 2 further shows the statistics of three datasets with more details as follows.

Table 2: The statistics of three experimental datasets of heterogeneous networks.

Dataset	Node Types and Number of Nodes			
DBIS [29]	Author (A)	Paper (P)	Venue (V)	
(264,323 edges)	60,694	72,902	464	
MovieLens [16]	Movie (M)	Actor (A)	Director (D)	User (U)
(1,097,495 edges)	10,197	95,321	4,060	2,113
Yelp [3]	User (U)	Business (B)	Category (C)	Location (L)
(411,263 edges)	16,239	14,284	511	47

- **DBIS** [29] is a bibliographic network dataset in the field of database and information system. The network consists of papers (P), authors (A), and venues (V) as nodes while the relationships of authorship (P-A) and published venues (P-V) are edges.
- **MovieLens** [16] is a network dataset of a movie recommendation system. The nodes of the network include movies (M), actors (A), directors (D), and users (U) while the edges comprise of actorship (M-A), directorship (M-D), and user ratings (M-U).
- **Yelp** [3] is a dataset extracted from the social media released in the competition of *Yelp Dataset Challenge* [3]. The nodes in the network involve users (U), businesses (B), categories (C), and locations (L) while the edges represent the relationships of friendships (U-U), user reviews (B-U), business locations (B-L), and business categories (B-C).

Baseline Methods. To evaluate the performance of MARU and the quality of learned representations, we compare MARU with five state-of-the-art homogeneous and heterogeneous network embedding methods as follows.

- **DeepWalk (DW)** [24] and **node2vec (N2V)** [14] represent random walk based homogeneous network embedding methods. DeepWalk generates a number of fixed-length plain random walks starting from each node while node2vec employs alias-sampling to mimic the process of breadth-first search and manipulate random walks. Both of the methods are based on the vanilla skip-gram model [21].
- **LINE** [32] represents an edge-sampling based homogeneous network embedding method. Based on the edge-sampling algorithm, LINE is able to efficiently capture both the first-order and second-order proximity in the networks.
- **HIN2Vec (H2V)** [11] learns node embeddings by predicting the existence of particular meta-paths between nodes with a meta-path conditioned binary classifier.
- **metapath2vec (M2V)** [9] stands for meta-path based heterogeneous network embedding methods. With a predefined set of meta-paths, metapath2vec guides the random walks through meta-paths so that the prior heterogeneous knowledge can be leveraged to the learned embeddings.
- **JUST** [18] is a heterogeneous network embedding method that manipulates random walks by specific strategies. JUST introduces a tactic for random walks to either jump to other nodes of particular types or to stay on the current paths.
- **HeGAN** [17] enhances HIN by adversarial learning that provides effective negative examples for more robust representations.

Table 3: The statistics of three datasets for the task of multi-label node classification.

Dataset	DBIS	MovieLens	Yelp
Node Type	Author (A)	Movie (M)	User (U)
Semantics	Domains	Genres	Compliments
$\lvert\mathcal{L}\rvert$	8	19	11
Avg. #(labels)	1.00	2.04	5.33

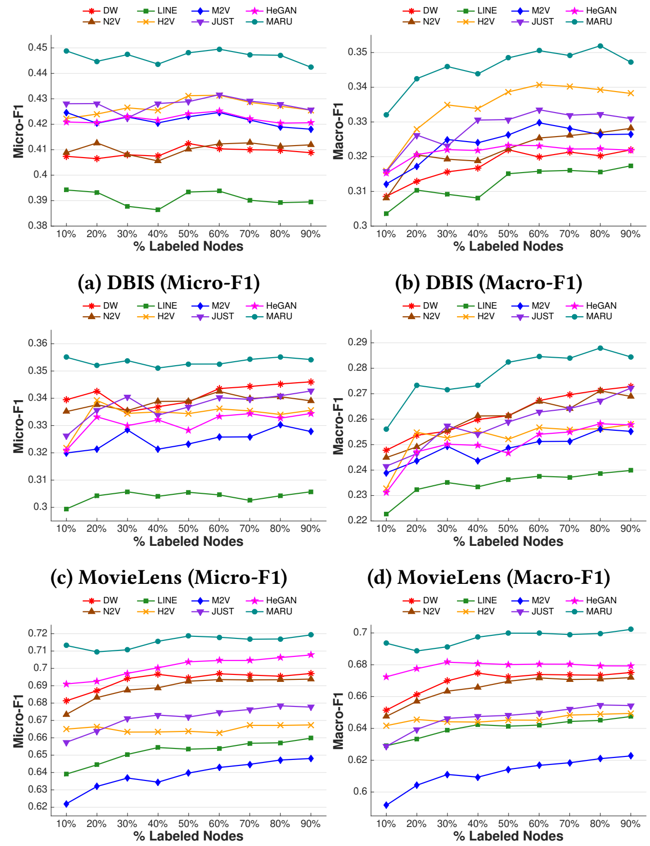
Note that we do not compare with GCN-based approaches because most of those methods cannot tackle unsupervised representation learning. Although some methods like GraphSAGE [15] and GAE [19] are applicable, they heavily rely on node features are not in major comparisons as shown in previous studies [10] For instance, the macro-F1 scores of both GraphSAGE and GAE are less than 23% on the Yelp dataset when all of the other baseline methods can reach over 30% with an arbitrary amount of training data.

Implementation Details. MARU is implemented by C and C++. The size of sliding windows for meta-contexts t is set as 6. The walk length l in the algorithms is 40 while the length of each generated random walks is 81. For all of the methods, the dimension of node embeddings is set to 128; the neighborhood size k is set as 7; the initial learning rate of stochastic gradient descent is set as 0.025; the number of negative samples for each neighbor r is 5.

5.2 Task 1: Multi-label Node Classification

Experimental Setup. In the task of multi-label node classification, every node is associated with one or more labels from a finite label set $\mathbb{L}$. We adopt the author domains, movie genres, and user compliments respectively for the DBIS, MovieLens, and Yelp datasets. The statistics of these datasets are shown in Table 3. Moreover, the labels are encoded in the networks so that the task is challenging because the node embeddings need to reflect the semantics that is not explicitly presented in the networks. To evaluate the performance, we randomly sample 10% of the nodes as testing data while the remaining nodes are treated as labeled data for training. In addition, we also adjust the percentage of labeled data used in the training process to demonstrate the robustness of methods. The node representations of each method are treated as the input of a one-vs-rest logistic regression model with L2 regularization. Macro-F1 and Micro-F1 scores [25] are adopted as the evaluation metrics for multi-label classification, thereby indicating the quality of different representations.

Experimental Results. Figure 3 demonstrates the performance of six methods on the task of multi-label node classification with three datasets. Among all of the baseline methods, most of the heterogeneous network embedding methods, including H2V, M2V, and JUST, outperform the other baselines in DBIS but perform worse than others in Yelp. It can be because the structural information is more important than the heterogeneous knowledge in Yelp. To be more precise, existing heterogeneous network embedding methods sacrifice the comprehensive understanding of network structures to encode the heterogeneous knowledge and obtain unsatisfactory performance when the structural information is imperative. Although HeGAN applies adversarial learning to obtain better robustness in Yelp, it performs worse in both DBIS and MovieLens due to more



(a) DBIS (Micro-F1) (b) DBIS (Macro-F1)

(c) MovieLens (Micro-F1) (d) MovieLens (Macro-F1)

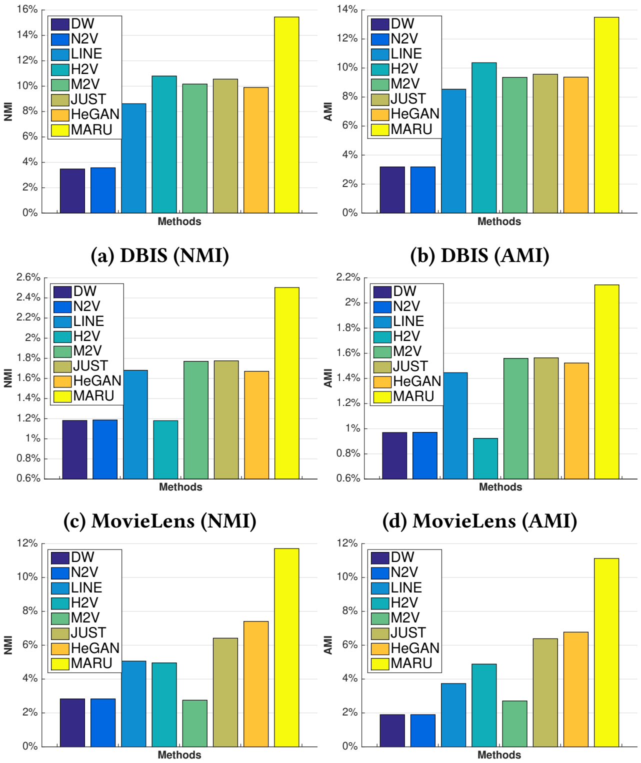
(e) Yelp (Micro-F1) (f) Yelp (Macro-F1)

Figure 3: Performance of different methods for the multi-label node classification task in three datasets. All improvements of our approach over baseline methods are statistically significant at the 95% confidence level in a paired t-test. Note that the Micro-F1 scores do not increase with more labeled nodes in some cases because of the imbalance of class distribution.

parameters and overfitting. The proposed approach in this paper, MARU significantly outperforms all of the baselines across different percentages of training labeled nodes in three datasets. MARU does not distort the generated random walks while incorporating heterogeneous knowledge. At the same time, meta-contexts are also beneficial for MARU as it picks up the tiny differences in local heterogeneous contexts.

5.3 Taks 2: Node Clustering

Experimental Setup. The problem of node clustering is an unsupervised machine learning task. We aim to cluster the nodes so that the generated groups are as close to the true clusters as possible. In each dataset, we modify the classes in multi-label classification to construct the ground truth. For the DBIS dataset, the authors can be categorized into different research domains. Each research domain represents one type of cluster. For the MovieLens dataset, five genres, including `Adventure`, `Action`, `Crime`, `Horror`, and `Sci-Fi`,



(a) DBIS (NMI) (b) DBIS (AMI)

(c) MovieLens (NMI) (d) MovieLens (AMI)

(e) Yelp (NMI) (f) Yelp (AMI)

Figure 4: Performance of different methods for the node clustering task in three datasets. All improvements of our approach over baseline methods are statistically significant at the 95% confidence level in a paired t-test.

represent five clusters. For the Yelp dataset, we separate users into two groups. One group represents those users who have received at least one compliment. The rest of the users are labeled otherwise. For simplicity, the nodes in multiple clusters are removed. In total, DBIS, MovieLens, and Yelp datasets have 8, 5, and 2 clusters, respectively. For evaluation, the node representations of each method are treated as the input of the K-Means++ algorithm [1] to derive clusters. Finally, normalized mutual information (NMI) and Adjusted Mutual Information (AMI) [35] are the evaluation metrics that reveal the quality of node representations.

Experimental Results. Figure 4 illustrates the performance of different methods for the task of node clustering in three datasets. Similar to the results in the multi-label classification task, H2V and M2V perform the best among all of the baselines in DBIS but obtain worse performance than others in Yelp. Differently, JUST and HeGAN perform reasonably well on all datasets. On the other hand, the homogeneous network embedding methods perform poorly in all of the datasets. One explanation is that the heterogeneous knowledge is important for the task of clustering. Interestingly, even though M2V exploits the heterogeneous knowledge by using the meta-paths, the clustering performance significantly drops in Yelp compared to other datasets. A possible reason could be the lack of meaningful meta-paths for clustering in the Yelp network. On the other hand, JUST does not need meta-paths and still performs well. Compared to all of the baseline methods, our proposed MARU consistently presents significant improvements against all baseline methods across all datasets. As a result, it demonstrates that meta-contexts and the algorithm of bidirectional extended random walks are valuable for the node clustering task.

Table 4: The *AUC* scores of different methods with four operators for link prediction in three datasets.

Method	Operator	DBIS	MOVIE	YELP
DeepWalk [24]	Hadamard	0.6367	0.9110	0.7330
	Weighted-L2	0.6094	0.7904	0.6872
node2vec [14]	Hadamard	0.6362	0.9060	0.6622
	Weighted-L2	0.6292	0.7968	0.6848
LINE [32]	Hadamard	0.5001	0.8631	0.5689
	Weighted-L2	0.5751	0.7611	0.6229
HIN2Vec [11]	Hadamard	0.8028	0.9651	0.8117
	Weighted-L2	0.7240	0.7885	0.7137
metapath2vec [9]	Hadamard	0.6778	0.9151	0.7372
	Weighted-L2	0.7363	0.6996	0.8240
JUST [18]	Hadamard	0.6463	0.9119	0.7453
	Weighted-L2	0.6260	0.7845	0.6009
HeGAN [17]	Hadamard	0.9597	0.9207	0.6361
	Weighted-L2	0.6714	0.7970	0.7289
MARU	Hadamard	**0.9979**	**0.9963**	0.7241
	Weighted-L2	0.7468	0.7979	**0.8315**

5.4 Task 3: Link Prediction

Experimental Setup. In the task of link prediction, we predict the missing edges in the given network datasets. Here we randomly remove 50% of edges from the networks for obtaining positive examples while generating an equal number of node pairs as negative examples. To generate the edge features, we follow the previous study [14] to exploit two binary operators to represent edges by aggregating two node representations over all dimensions, including the Hadamard product and weighted L2-distance. The features of example edges are treated as the input of a logistic regression model to learn their existence. Finally, the scores of Area Under Curve (AUC) can be applied to evaluate the performance of link prediction and the quality of representations.

Experimental Results. Table 4 shows the performance of different methods for the task of link prediction in three datasets. In the task of link prediction, our proposed approach MARU significantly surpasses all of the baseline methods. Among the baseline methods, HIN2Vec and metapath2vec perform the best as heterogeneous network embedding methods. Interestingly, although LINE does not have outstanding performances in the tasks of multi-label node classification and node clustering, it has a satisfactory performance for link prediction. It can be because LINE is an edge-sampling based method so that it has more advantage in link prediction to model the edge distributions. Interestingly, Grover and Leskovec [14] report that the Hadamard operator always performs the best in their study while only the datasets with homogeneous networks are evaluated. This is partially inconsistent with the experimental results of heterogeneous networks. The reason can be that the

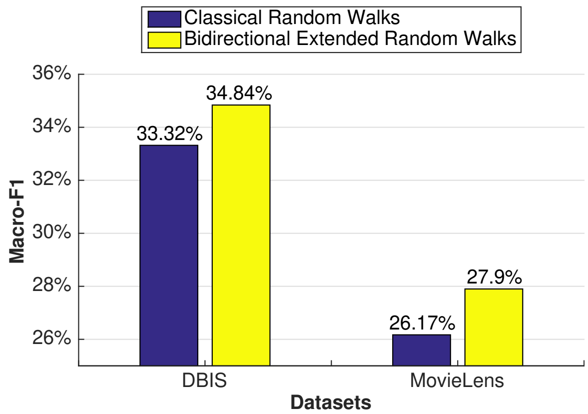


Figure 5: The macro-F1 scores of MARU with classical random walks and our proposed bidirectional extended random walks with 50% of training labeled nodes in the task of multi-label node classification in DBIS and MovieLens.

embeddings become too sophisticated to estimate the relationship between nodes by a simple dot-product when the types of nodes are heterogeneous. The results also show the difference between homogeneous and heterogeneous network and emphasize the importance of designing satisfactory algorithms to derive heterogeneous network representations.

5.5 Analysis and Discussions

In this section, we first analyze the effectiveness of the proposed algorithm of bidirectional extended random walks and then discuss the sensitivity of the window size for observing meta-contexts.

Effectiveness of Bidirectional Extended Random Walks. To verify the contribution of our proposed bidirectional extended random walks, we first investigate the effectiveness of the algorithm. Figure 5 shows the macro-F1 scores of MARU with classical random walks and the proposed bidirectional extended random walks with 50% of training labeled nodes in the task of multi-label node classification in DBIS and MovieLens. After replacing the classical random walks with the bidirectional extended random walks, the classification performances are significantly improved by 2.04% and 4.07% in DBIS and MovieLens, respectively. It shows that the proposed algorithm to generate bidirectional extended random walks is actually beneficial to alleviate the insensitivity of classical random walks to the tail nodes, thereby improving the performance of downstream applications.

Window Size of Meta-contexts. Here we study how the size of the sliding windows for meta-contexts affects the performance. Figure 6 shows the macro-F1 scores of MARU over different percentages of labeled training data with different window sizes for meta-contexts in the Yelp dataset. It is obvious that greater window sizes lead to a better classification performance because the observed contexts are more flexible and informative. However, larger window sizes also lead to larger body of meta-contexts M. For example, in the Yelp dataset with $t = 6$, the size of M is greater than 10,000, which can significantly increase the memory or disk space consumption. On the other hand, the size of M is less than 1,000 with $t = 4$, rendering memory footprints more manageable. Therefore, we set the window size t as 4 in the parameter settings.

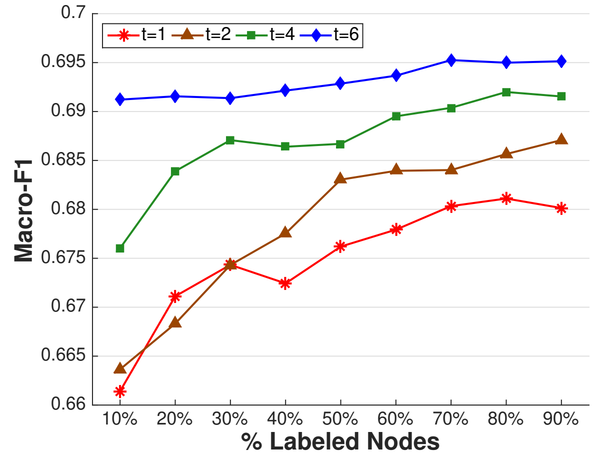


Figure 6: The Macro-F1 scores of MARU as a function of percentage of labeled training data and sliding window size for meta-contexts in Yelp.

Table 5: The classification performance of MARU over different walk lengths l of bidirectional extended random walks with 50% of training labeled nodes in Yelp. Note that the length of generated random walks is $2 \times l + 1$ because MARU conducts random walks bidirectionally.

Metric	$l = 10$	$l = 20$	$l = 40$	$l = 80$	$l = 100$
Macro-F1	0.6842	0.6961	**0.6999**	0.6958	0.6969
Micro-F1	0.6998	0.7139	**0.7186**	0.7150	0.7151

Walk Length l of Bidirectional Extended Random Walks. Here we study how the length of random walks affects the performance. Table 5 presents the classification performance of MARU over different walk lengths l of bidirectional extended random walks (See Algorithm 1 and 2) with 50% training labeled nodes in the Yelp dataset. While the length of random walks increases, both micro-F1 and macro-F1 scores improve because of more prevalent information. However, the performance peaks at $l = 40$ and then drops with longer random walks. This can be because longer random walks cover more nodes with high scores of degree and betweenness so that the contexts with tail nodes are less observed in the generated random walks. The results also demonstrate that it is important to design a good algorithm, such as the proposed bidirectional extended random walk, to alleviate the bias of conventional random walk algorithms.

Size of Embedding Dimensions. We also discuss how the size of embedding dimensions affects the performance. Table 6 shows the classification performance of MARU over different sizes of embedding dimensions d with 50% of training labeled nodes in the Yelp dataset. When the dimension increases, the performance improves and peaks at 128. With a larger size of embedding dimensions, the classification model becomes overfitted. As a result, we apply $d = 128$ as the experimental setting across all experiments.

Table 6: The classification performance of MARU over different sizes of embedding dimensions d with 50% of training labeled nodes in Yelp.

Metric	$d = 16$	$d = 32$	$d = 64$	$d = 128$	$d = 256$
Macro-F1	0.6931	0.6933	0.6982	**0.6999**	0.6974
Micro-F1	0.7112	0.7114	0.7154	**0.7186**	0.7156

6 CONCLUSIONS

In this paper, we propose MARU, a novel approach for heterogeneous network embedding by exploiting meta-contexts in random walks. To address the bias caused by conventional random walks, the algorithm of bidirectional extended random walks is proposed to efficiently and fairly capture the comprehensive structural information in the networks. The meta-context aware node embeddings are then designed and optimized to represent properties of the nodes for different local heterogeneous contexts, thereby inferring the node representations based on aggregations over the meta-context distributions. Extensive experiments demonstrate that our proposed approach significantly outperforms state-of-the-art heterogeneous network embedding methods across three general network mining tasks, including multi-label node classification, node clustering, and link prediction. The reasons and insights can be concluded as follows: (1) the algorithm of bidirectional extended random walks effectively alleviates the bias for tail nodes with a theoretical guarantee; (2) the effectiveness of meta-contexts and meta-context aware node embeddings implies that a node can have distinct properties with different local heterogeneous contexts, which benefit the network representation learning; (3) the nature of heterogeneous networks can be much different from the traits of homogeneous networks, so it is crucial to tackle the problems of heterogeneous networks with specific and appropriate technologies.

APPENDIX

A THE PROOF OF COROLLARY 1

Proof. For the one-directional random walk, the expected number of visited tail nodes is $E_o = 1 + \sum_{i=0}^{2n} i \cdot p^i \cdot (1-p)$. For the bidirectional random walk, the expected number of visited tail nodes is $E_b = 1 + 2 \cdot \sum_{i=0}^{n} i \cdot p^i \cdot (1-p)$. Therefore, we have

$$\begin{aligned}\lim_{n\to\infty} E_b - E_o &= \lim_{n\to\infty} (1-p) \cdot \left(\sum_{i=1}^{n} i \cdot p^i - (i+n) \cdot p^{i+n} \right) \\ &= \lim_{n\to\infty} \frac{p \cdot (1-p^n) \cdot (2 \cdot n \cdot p^{n+1} - (2n+1) \cdot p^n + 1)}{1-p} \\ &= \frac{p}{1-p} > 0\end{aligned}$$

□

ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their helpful comments. The work was partially supported by NSF DGE-1829071 and NSF IIS-2031187.

REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *SODA*. 1027–1035.
[2] Hongyun Cai, Vincent W Zheng, and Kevin Chang. 2018. A comprehensive survey of graph embedding: problems, techniques and applications. *TKDE* (2018).
[3] Yelp Dataset Challenge. 2013. Yelp dataset challenge. (2013).
[4] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *KDD*. ACM, 119–128.
[5] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. 2018. Harp: Hierarchical representation learning for networks. In *AAAI*.
[6] Ting Chen and Yizhou Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *WSDM*. ACM, 295–304.
[7] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.
[8] Hongbo Deng, Jiawei Han, Bo Zhao, Yintao Yu, and Cindy Xide Lin. 2011. Probabilistic topic models with biased propagation on heterogeneous information networks. In *KDD*. ACM, 1271–1279.
[9] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*. ACM, 135–144.
[10] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. 2019. A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893* (2019).
[11] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*. ACM, 1797–1806.
[12] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*. 2672–2680.
[14] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. ACM, 855–864.
[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*. 1024–1034.
[16] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *TIIS* 5, 4 (2016), 19.
[17] Binbin Hu, Yuan Fang, and Chuan Shi. 2019. Adversarial Learning on Heterogeneous Information Networks. In *KDD*. ACM.
[18] Rana Hussein, Dingqi Yang, and Philippe Cudré-Mauroux. 2018. Are Meta-Paths Necessary?: Revisiting Heterogeneous Graph Embeddings. In *CIKM*. ACM, 437–446.
[19] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
[20] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *JASIST* 58, 7 (2007), 1019–1031.
[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.
[22] Tore Opsahl and Pietro Panzarasa. 2009. Clustering in weighted networks. *Social networks* 31, 2 (2009), 155–163.
[23] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *KDD*. ACM, 1105–1114.
[24] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. ACM, 701–710.
[25] David Martin Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).
[26] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*. 693–701.
[27] Chuan Shi, Binbin Hu, Xin Zhao, and Philip Yu. 2018. Heterogeneous Information Network Embedding for Recommendation. *TKDE* (2018).
[28] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.
[29] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB* 4, 11 (2011), 992–1003.
[30] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S Yu, and Xiao Yu. 2013. Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *TKDD* 7, 3 (2013), 11.
[31] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*. ACM, 1165–1174.
[32] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.
[33] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *KDD*. ACM, 817–826.
[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.
[35] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *JMLR* 11, Oct (2010), 2837–2854.
[36] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *KDD*. ACM, 1225–1234.
[37] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. 2017. Signed network embedding in social media. In *SDM*. SIAM, 327–335.
[38] Yu Zhao, Zhiyuan Liu, and Maosong Sun. 2015. Representation learning for measuring entity relatedness with rich information. In *IJCAI*.