Intelligent Data Analysis 24 (2020) 1207–1233
DOI 10.3233/IDA-194843
IOS Press

W-Com2Vec: A topic-driven meta-path-based intra-community embedding for content-based heterogeneous information network

Phu Pham and Phuc Do*
University of Information Technology, VNU-HCM, Vietnam

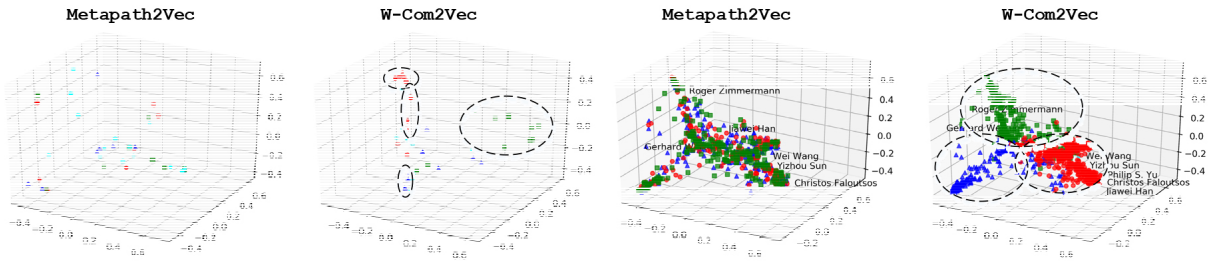
Abstract. Heterogeneous information network (HIN) are becoming popular across multiple applications in forms of complex large-scaled networked data such as social networks, bibliographic networks, biological networks, etc. Recently, information network embedding (INE) has aroused tremendously interests from researchers due to its effectiveness in information network analysis and mining tasks. From recent views of INE, community is considered as the mesoscopic preserving network's structure which can be combined with traditional approach of network's node proximities (microscopic structure preserving) to leverage the quality of network's representation. Most of contemporary INE models, like as: HIN2Vec, Metapath2Vec, HINE, etc. mainly concentrate on microscopic network structure preserving and ignore the mesoscopic (intra-community) structure of HIN. In this paper, we introduce a novel approach of topic-driven meta-path-based embedding, namely W-Com2Vec (Weighted intra-community to vector). Our proposed W-Com2Vec model enables to capture richer semantic of node representation by applying the meta-path-based community-aware, node proximity preserving and topic similarity evaluation at the same time during the process of network embedding. We demonstrate comprehensive empirical studies on our proposed W-Com2Vec model with several real-world HINs. Experimental results show W-Com2Vec outperforms recent state-of-the-art INE models in solving primitive network analysis and mining tasks.

Keywords: Meta-path, community detection, content-based HIN, network embedding

1. Introduction

Recently, INE [1–3] has become the most popular researching area from in information network analysis and mining field. Multiple INE models have been proposed and achieved considerable attentions from many researchers and organizations due to its effectiveness in preserving the complex structure of real-world HINs as well as high performance in a wide range of applications. Up to present, recent INE models have been shifted to the development of effective and scalable representation learning mechanism which are applied for highly complex and large-scaled network like as social networks (Facebook, Twitter, etc.). Most of previous proposed INE models [1,4–8] aim to embed the network's nodes into a latent low-dimensional vector space by preserving the nodes' attributes and their order proximity. Contemporary network embedding models like as DeepWalk [9], LINE [10] and Node2Vec [11] learn

*Corresponding author: Phuc Do, University of Information Technology, VNU-HCM, Vietnam. E-mail: phucdo@uit.edu.vn.



A. Representations of member nodes in Zachary's karate club network

B. Representations of author node in DBLP network

Fig. 1. Illustrations of different networks' node representations in PCA-3D plot through Metapath2Vec and W-Com2Vec approaches.

the node representation by preserving the node first-order and second-order proximities to ensure the similar embedded representations between nodes which are linked or shared same sets of contextual nodes. The sets of contextual nodes for each target node are generated via different approaches like as random walk [9], neighborhood (first/second) order node evaluation [10,11], etc. From the past, most of the previous INE models are mainly designed to work on Homogeneous Information Networks (HoINs). These HoIN-based models treat all nodes and relations as the same type. Extending the idea of using node order proximity of HoIN-based model in preserving network structure, recently there are some emerged models have been proposed to challenge the diversity of nodes and links in HINs. Some HIN-based INE models (HIN2Vec, Metapath2Vec, etc.) use meta-path-based random walk mechanism to obtain the network's structure. In general, all of these HoIN-based and HIN-based methods only focus on the "microscopic" structure (pairwise nodes and links) of the network. Therefore, these models fail to capture higher-level of network's structure such as communities, clusters, network's centroids, etc. These network's structural aspects are called as "mesoscopic" level structure. In fact, community structures are considered important factors for network analysis and representation learning. In specific area of INE, community structure is one of the most important structural descriptions of the networks. Such as in social network analysis, communities naturally reveal the organizational structures as well as principal components of social networks (Facebook, Twitter, Weibo, etc.), like as user's groups, friends, social clubs, etc. The use of preserving community structure in INE can help to improve the quality of node representation. Recently, there are some mesoscopic structural network embedding techniques have been proposed to tackle the problem of capturing global properties of the networks like as ComE [12], M-NMF [13], etc. However, these models can only be applied for homogeneous networks and fail to learn the rich semantics of network's heterogeneity. These models only capture the community proximity of single-typed nodes which are linked by one type of links. Therefore, they are unable to combine different semantic paths in forms of meta-paths between two same-typed nodes in HINs. Moreover, most of recent meta-path-based INE models are mainly concentrated on the node order proximity between pairwise nodes. They largely ignore the topic similarity between text-based nodes within meta-paths in content-based HINs. It is needless to say that most of real-world HINs are composed with large quantity of text-based nodes like as comments, posts in social networks (Facebook, Twitter, etc.), papers in bibliographic networks (DBLP, DBIS, etc.), etc. By evaluating the topic similarity between text-based nodes within meta-paths, we can improve the quality of node order proximity evaluation in INE. To fill gaps between approaches of microscopic and mesoscopic structure preserving in HIN-based INE and topic similarity evaluation in content-based HINs, we clearly define these challenges as well as contributions of our works in the introduction of W-Com2Vec model. The W-Com2Vec model is designed

to enable for capturing both microscopic topic-driven node proximity as well as mesoscopic structure (intra-community) of given HINs by using meta-path.

Figure 1 illustrates the community-aware representations of author's nodes (in DBLP network) (Fig. 1A) and karate-club's members (in Zachary's karate club network) (Fig. 1B) which are obtained by using Metapath2Vec and our proposed W-Com2Vec models. As shown from Fig. 1, our proposed W-Com2Vec model achieves better quality than well-known Metapath2Vec [7] model in preserving the mesoscopic network structure (intra-community). Such as in DBLP network, we applied the W-Com2Vec model to embed author node along with their associated community, so relevant authors who belong to same community will be represented as similar vectors in vector space (as shown in Fig. 1B).

1.1. Challenge definitions

1.1.1. Meta-path-based intra-community network structure preserving

Despite high performance in accuracy and great potential applications of recent intra-community based embedding models, there are remaining challenges which are related to the capability of capturing rich semantics of heterogeneous types of nodes and links. Recent intra-community based embedding models such as ComE [12], M-NMF [13] only focus on evaluating node and intra-community proximity of single-typed nodes in HoINs, such as friendship networks (only contains user's node and friendship relation) or authorship network (only contains author's node and coauthor relation). Contemporary community detection techniques are only applicable for single-typed nodes and links of HoINs. They are definitely invalid for evaluating complex topological features of multi-typed nodes and links of real-world HINs. It is undeniable that using HoIN-based community detection techniques can't fully reflect the real situation of linked nodes. For example, the authorship relation is actually a sequence of relations between two author and a paper node in forms of meta-path: A[author]-P[paper]-A. Or there are several types of friendship relation in complex social networks such as Facebook: mutual friends (U[user]$\xrightarrow{\text{mutual_friend}}$U), close friend (U$\xrightarrow{\text{close_friend}}$U), family relationship (U$\xrightarrow{\text{family_relationship}}$U), etc. Therefore, community detection and intra-community structure preserving in heterogeneous networks have been becoming a hot topic. The major challenges of community detection in HINs is how to effectively organize and combine different semantic paths between same-typed nodes in forms of met-paths for acquiring desired communities. For example, we want to detect groups of similar authors who have both co-worker (A-O[organization]-A) and co-author (A-P-A) relations in DBLP. Or extracting groups of users who both commenting about specific posts (U-C[comment]-P[post]-C-U) and participate in common groups (U-G[group]-U). In fact, community detection and network's structure preserving in HINs might be more flexible than in HoINs due to the changes of used meta-paths will influence the node representations. Depending on the user's expectation, different meta-paths will be applied to produce different results within a same network.

1.1.2. Topic-driven community detection and intra-community network embedding

Beside the challenges of different types of nodes and links, the content and topic similarity preserving of rich-text network's nodes is also a considerable problem which deeply affect the quantity of node's representations. Most of real-world networks contain large number of text-based nodes, such as posts, comments, etc. in social networks (Facebook, Twitter, Weibo, etc.), movie's descriptions, user's comments in movie's networks (IMDB, TMDB, etc.), papers in bibliographic networks (DBIS, DBLP, etc.), etc. called content-based HINs. From the past, the task of community detection and embedding only involve in discovering groups of nodes which are connected densely via evaluating their inter-connected relations.

They concentrated on using graph analysis techniques to extract topological structures of how network's nodes are linked rather than node's attributes. Content and topic attributes of text-based nodes play an important role in attribute-level node proximity measurement. For example, two authors are likely to be cooperating more (A-P-A) if they are working on the same fields/subjects, or two users likely to be familiar if they frequently comment similarly on common posts. In content-based HINs, topic attribute of text-based nodes can help to leverage the network structure preserving and node representation by rendering more information for nodes as well as their relations. Community detection problem in HINs is considered as an approach of using different meta-paths between target same-typed nodes to identify densely connected nodes. Traditional approaches of community detection in HINs do not consider the weights of relations between nodes within the communities. All paths between same-typed nodes are considered as the binary weighting relations, 1 for existed relation and otherwise 0. Therefore, it is necessary to propose a new approach which can help to analyze the topic similarities between two nodes and are used as the weights of relations between them.

1.2. Our contributions

In this paper, we propose a novel approach of topic-driven meta-path-based intra-community network embedding, namely W-Com2Vec. Our proposed model supports to preserve both node order and community structure proximity for INE. In particular, we continue to improve and apply our previous works on topic-driven community detection to detect communities in HINs. To learn the node representation via node order proximity, we apply our previous proposed model on topic-driven meta-path-based network embedding (W-Metapath2Vec) [14,15] to learn the node's presentation. Then, the detected communities are combined with node's representations in previous stage to form the final network representations which enable to exploit the consensus relationships of node order proximity as well as community structure in INE. We conduct extensive experiments on various real-world HINs, including: DBLP and MoviesLen1M to evaluate the performance of our proposed model in principal mining tasks (clustering and multi-classed classification) with recent state-of-the-art INE baselines. To sum up, our contributions in this paper can be summarized as following points:

- First of all, we present the approach of topic similarity measurement between text-based nodes in content-based HINs via applying LDA (Latent Dirichlet Allocation) topic model. These topic similarity values are then used as the weights for meta-paths between target nodes in topic-driven community detection and network's node representation.
- Next, we demonstrate the approach of combining between order proximity based node representation with detected community to produce the final network's representations of W-Com2Vec model. The W-Com2Vec can ensure both microscopic (node order proximity via meta-path-based random walks) and mesoscopic (intra-community) structure are preserved.
- Finally, we perform comprehensive experiments on benchmark datasets to show significant improvements of our proposed W-Com2Vec models with up-to-date INE models in multiple network mining tasks.

The rest of our paper is organized into 4 sections. In Section 2, we give brief reviews and discussions about previous works and our motivations. In Section 3, we demonstrate the background concepts and methodology of our W-Com2Vec model. All algorithms, model's architecture and optimization strategies also introduced in this section. For Section 4, we present extensive experiments of W-Com2Vec model on different benchmark datasets as well as discussions about the experimental outputs. Finally, we conclude our works and provide our future continual improvements in Section 5.

2. Preliminaries and related works

2.1. Preliminaries and backgrounds

An information network is defined as an directed/undirected graph, denoted as: $G = (V, E, A, R)$, where V and E present for sets of nodes ($v, v \in V$) and links ($e, e \in E$) of the network, respectively. We have two mapping functions which are $\phi : V \rightarrow A$ and $\psi : E \rightarrow R$, where A and E denote sets of node's types and link's types. The network representation learning or embedding (Definition 1) supports to transform a set of node V from highly complex dimensional space into a fixed and low dimensional space structure but still preserves the original structure and principal components of the network.

Definition 1: Information network embedding (INE) [2,3]: is defined as a mapping function $f : V \rightarrow \mathbb{R}^{|V| \times d}$, where $\mathbb{R}$ presents for the $|d|$-dimensional latent space of network's nodes (V), with $|d| \ll |V|$. The output of INE is the low-dimensional matrix $\mathbb{R}$ with each row $\mathbb{R}_v$ corresponding to the representation of node (v).

Definition 2: Network node order proximity [2,10]: normally defined as the proximity weight of two pairwise node (a) and (b) in a given network, where $a, b \in V$. For the first order proximity, if a and b have a directed link (1-hop), denoted $e(a, b), e \in E$, the first order of a and b will be assigned a weighting value, denoted as: w_{ab}, normally 1, otherwise it will be assigned as 0. Similar to first-order, the second-order and k-order capture the 2-hop and k-hops relations between a and b.

Traditionally, INE models focuses on capturing the network's featured properties by preserving node order proximity, including first-order, second-order and k-order proximity (Definition 2) to enforce same sharing neighboring nodes (also called contextual nodes) to have similar embedded vectors. There are two main ways for extracting the set of contextual nodes. The first one is node's neighborhood evaluation and the second one is using random walk on graph. LINE is the most well-known model which use first-order (LINE_1) and second-order (LINE_2) node's neighborhood evaluation techniques. Similar to the approach of LINE, DeepWalk and Node2Vec use random walk mechanisms to capture contextual nodes for each target node in a given network. These most well-known HoIN-based INE models like as DeepWalk, LINE, PTE and Node2Vec treat all nodes and links as the same type ($|A| = 1, |R| = 1$). However, the recent main challenges of real-world applications are come from the network heterogeneity which is difficult to apply directly HoIN-based INE models for transforming the different-typed nodes into latent space vectors. Recently, several HIN-based INE models have been proposed to tackle challenge of diversity in types of nodes and links. Inspiring from node order proximity of previous HoIN-based models, HIN-based models like as HIN2Vec, Metapath2Vec use the meta-path-based random walk mechanism to generated sets of contextual nodes which are used as the training set for learning the network's node representations by using single-hidden-layer neural network. These novel proposed models have demonstrated outperformances on multiple HIN analysis and mining tasks in comparing with HoIN-based models.

Definition 3: Microscopic network structure preserving (MiNSP) [2,3]: Microscopic network structure preserving embedding refers to techniques that intend to preserve the overall structure of a given network by capturing the nodes and edges order proximity. MiNSP is mainly used for capturing the local structure and information of a network by evaluating directed and undirected nodes and links.

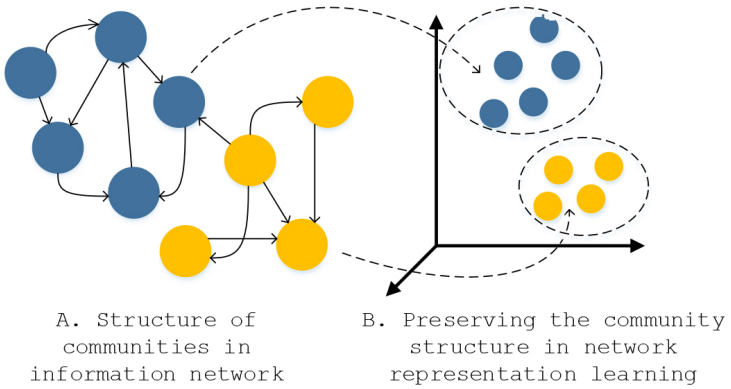


Fig. 2. Illustration of community structure preserving (MeNSP) in network representation learning.

Definition 4: Mesoscopic network structure preserving (MeNSP) [2,3]: different from MiNSP, the MeNSP concentrates on high-level structure of the network like as communities, network's centroids, etc. MeNSP aims to enforces nodes which share common network's structural properties into a similar embedding spaces, such as two nodes in a same community, also called intra-community approach.

However, the views of INE have been recently changed. Most of node order proximity models are considered as the microscopic network structure preserving (Definition 3) approach and they fail to capture higher level of network's structure like as communities (illustrated in Fig. 2), clusters and other network's global properties, called mesoscopic network structure preserving (Definition 4). In fact, community structures are very important in network representations which enable to generate insights for subnetworks analysis, in-community node ranking and similarity measurement. Many researches have demonstrated the effectiveness of combination between nodes order proximity and intra-community evaluations in network representation learning. Two well-known ComE [12] and M-NMF [1] models are designed to be used for capturing both microscopic and mesoscopic network structures. ComE is a model for network and graph analysis which formulate community embedding as tuples of mean vectors. Similar to that, M-NMF combine between node first-order and second-order proximity with community detection to leverage the network's structure preserving. M-NMF uses matrix factorization to learn the node presentation from both node order proximity and global intra-community properties. However, these models are only applied for homogeneous network only which are unable to capture rich semantics of network's heterogeneity.

2.2. Literature reviews and motivations

In information network analysis and mining, node and community embedding have been extensively studied recently. In term of network's node order proximity preserving, almost network embedding techniques such as DeepWalk [9], LINE [10], PTE [16] and Node2Vec [11] obtain the distributions of nodes by capture first and second orders of nodes. Most of recent INE models fall into this approach. DeepWalk is a most well-known model which is inspired from the Skip-grams of Word2Vec model to learn the latent representation of network's nodes from their contextual nodes. Similar to the idea of Word2Vec, nodes that share similar contextual nodes in random walk sequence will be represented similarly in the corresponding embedding spaces. Beside that LINE model is proposed to enable for learning the representations of node by preserving the first-order (LINE_1) and second-order (LINE_2) proximities instead of using random walks to exploit the network's structure. Following the DeepWalk's network

learning architecture, Node2Vec proposes a novel flexible contextual node sampling strategies, including BFS (breadth-first-sampling) and DFS (depth-first-sampling). The Node2Vec's sampling strategies are considered be better in capturing first-order, second-order and high-order node proximities. Naturally, most of real-world networks are composed by communities which their inside nodes are densely linked to each other and sparsely linked to nodes in other communities. Community structure is considered as very important in information network analysis and mining which help to identify groups of nodes which share common attributes. For example, users who share common interests in social networks like as Facebook, Twitter, etc. often form a community. Or in bibliographic networks, like as DBLP, DBIS, etc. authors/researchers who work on similar research topics tend to frequently cooperate to each other. In terms of the target network's structure to embed, most INE methods focus on nodes and links which try to preserve nodes' proximities which are reflected by their inter-connected local structure. Recently, there are intra-community representation learning models which are proposed to learn richer informative node representations. The ComE [12] model combines network's nodes and community embedding by using community detection techniques. In ComE model, the community embedding is defined as a multivariate Gaussian distribution, which help to characterize how network's nodes are distributed inside each community. However, ComE does not support to represent detected communities as vectors which is considered as indirect approach for optimizing the relationships between network's nodes and their communities. Node and community embedding must be incorporated to each other to generate more accurate high-order node as well as community-aware proximity representations. To challenge problems related to the relationship between node and intra-community proximity, M-NMF [13] model is proposed to leverage broader network's structure representation learning. The M-NMF model relies on matrix factorization to learn both microscopic structure (the first-order and second-order proximity) and mesoscopic structure (the intra-community proximity) of given networks. In contrast to recent intra-community network representation learning model, the target to embed multi-typed nodes and links in HIN is not given. Hence, the remained challenge is how to properly generate multi-typed nodes and their communities of HINs into common embedding spaces simultaneously.

3. Methodology

In this section, we demonstrate the heterogeneous network structural microscopic (node embedding) and mesoscopic (intra-community) embedding approaches of the proposed W-Com2Vec model. For the microscopic approach, we apply the meta-path-based random walk mechanism to obtain the node embedding, denoted as: $\mathbb{N}$. Then, depending on the communities which are detected from the network via Louvain algorithm we produce the community-aware network representations which is obtained by maximizing the modularity between node and community relationships, denoted as: $\mathbb{C}$. Then, the representations of node embedding ($\mathbb{N}$) and intra-community embedding ($\mathbb{C}$) is combined to produce the final representation, denoted as: $\mathbb{X}$. The final $\mathbb{X}$ network representation is capable to preserves both the microscopic structure (node similarity) as well as mesoscopic community structure of given HINs.

3.1. Meta-path-based microscopic network (node proximity) embedding

3.1.1. Meta-path-based node proximity embedding by random walk

Consider a given HIN, denoted as: $G = (V, E, A, R)$, with $n = |V|$ is number of nodes, the ultimate goal of node proximity embedding is aimed to learn a proper mapping function $f_v : v \rightarrow r^{1\times d}$ with $v \in V$ and $r \in \mathbb{R}^{|V|\times d}$, where $r^{1\times d}$ is a d-dimensional (row) vector representation of node v, $d \ll n$. For each

node (v) in the network, we obtain a set of neighborhood nodes by using meta-path-based random walk mechanism. These captured neighborhood nodes are considered as the node's contexts which are used to train the node proximity representation models. Same-typed nodes are considered as relevant if they share same sets of contextual nodes. For examples, two similar authors often have same sets of frequent submitted venues/conferences (represented as meta-path A-P-V-P-A) in bibliographic networks (DBLP, DBIS, etc.). Or similar users on social networks (Facebook, Twitter, etc.) often comment on same posts which belong to specific topics (U-C-P-C-U). Formally, given a meta-path $\mathcal{P} : A_s \xrightarrow{R_1} A_2 \ldots \xrightarrow{R_{l-1}} A_t$ which is defined as a path between two same-typed nodes v_s and v_t with: $\phi(v_s) = \phi(v_t)$, $A_s = A_t$. We define the transitional probability of moving from source node (v_s) to a specific target node (v_t) via a path instance of meta-path $\mathcal{P}$, denoted as: $\pi(v_s \rightsquigarrow v_t, \mathcal{P})$, is defined as following equation (Eq. (1)):

$$\pi(v_s \rightsquigarrow v_t, \mathcal{P}) = \begin{cases} \frac{\sum_{\mathcal{P}_{(v_s \rightsquigarrow v_t)}} \frac{1}{|N(v_s)|} + w_{\mathcal{P}}}{\lambda} \\ 0, \text{ with } |\mathcal{P}_{(v_s \rightsquigarrow v_t)}| = 0 \end{cases}, \text{ with } |\mathcal{P}_{(v_s \rightsquigarrow v_t)}| > 0 \quad (1)$$

Where,

- $\mathcal{P}_{(v_s \rightsquigarrow v_t)}$, presents a set of path instances of meta-path ($\mathcal{P}$) between two same-typed nodes of (v_s) and (v_t).
- $N(v_s)$, presents total number of neighborhood nodes of source node (v_s).
- $w_{\mathcal{P}}$, presents for the weight of each path instance between two same-typed nodes of (v_s) and (v_t). The path's weight ($w_{\mathcal{P}}$) is calculated depending on the applied meta-paths. There are two types of path's weight, the first one is binary weight [0, 1], normally assigned 1 if there is an existent path instance between (v_s) and (v_t), otherwise 0. The second type of path's weight is topic similarity weight. Topic similarity weight of path instance is considered as important in evaluating the proximity between nodes in content-based HINs. We will discuss about the topic similarity weight of meta-path in the next part (3.1.2).
- λ, presents for the global constant which helps to normalize the transitional probabilities between nodes from 0 to 1 range, where: $\lambda = \frac{\pi(v_s \rightsquigarrow v_t, \mathcal{P})}{\sum_{v_i \in V} \pi(v_i \rightsquigarrow v_{\neg i}, \mathcal{P})}$, with $v_{\neg i}$ presents for a set of nodes which are not v_i.

3.1.2. Topic similarity weight of meta-path in content-based HIN

Most of real-world HINs contain a large number of text-based nodes like as papers in bibliographic networks (DBLP, DBIS, etc.) or comments, posts, etc. in social network (Facebook, Twitter, etc.) or movies' descriptions in movie networks (IMDB, TMDB, etc.). These text-based nodes are ubiquitous and rich in semantic which can help to leverage the outputs of network analysis and mining tasks. For example, we can effectively group relevant authors in DBLP network depending on the topic similarities in their published papers such as "Jiawei Han", "Christos Faloutsos", "Philip S. Yu", "Rakesh Agrawal", etc. in "data mining" topic or "Christopher D. Manning", "Tomas Mikolov", "Yoshua Bengio", "Quoc V. Le", etc. in "machine learning/natural language processing" areas, etc. Another example to illustrate the importance of text-based nodes in social networks such as Facebook, by evaluating the similarity in the contents of users' posts or comments, we can identify groups of similar users who are interesting on same subjects/areas and these data are very useful for constructing recommendation system. Therefore, in this part, we present an approach of using the topic distributions of text-based nodes for calculating the topic similarity weights of meta-paths in content-based HINs.

There are several methods for extracting distributions of topics from text-based nodes of HINs. One of the most common ways is applying the unsupervised probabilistic distribution evaluation technique

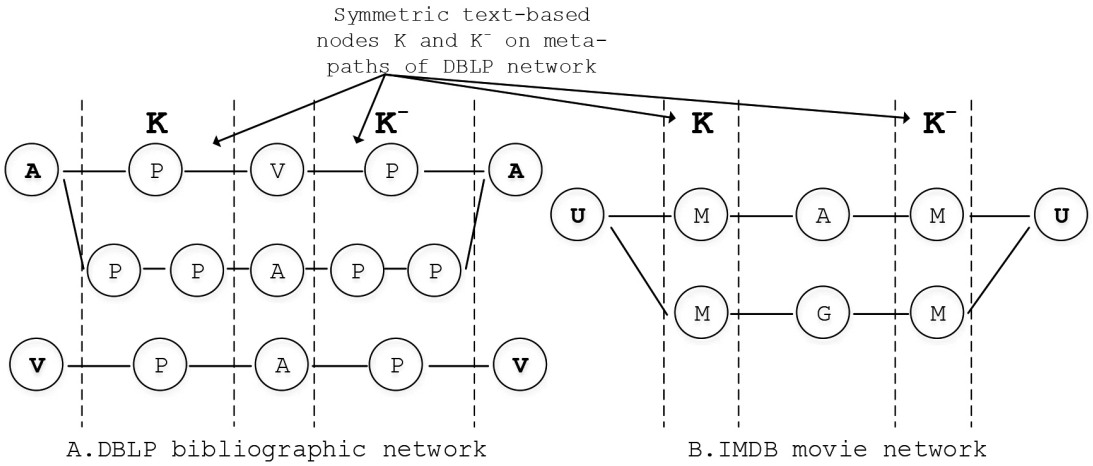


Fig. 3. Examples of symmetric text-based nodes in meta-paths of DBLP and IMDB networks.

to discover the latent topic distributions of given text corpora. Latent Dirichlet Allocation (LDA) topic modelling (Definition 5) is one of the most popular technique for doing this.

Definition 5: Topic modelling: is a statistical approach for discovering latent topics from collection of text documents. Latent Dirichlet Allocation (LDA) [17] is one of the most common approach for obtaining document-topic and topic-word distributions from given text corpus by applying sparse Dirichlet prior distributions.

Consider each text-based node is a document denoted as (d). The LDA help to evaluate the topic distributions in each text document, denoted as: $Prob(z_i|d_j) = \theta^{d_j}_{z_{(i,i\in|Z|)}}$, with $(z_i{:}z_i{\in}Z)$, where z is the latent topic distribution over each document. By using LDA, with a defined number of latent topics $|Z|$, each document is now represented as $|Z|$-dimensional topic distribution vectors, denoted as: $d = [prob(z_1|d), \ldots prob(z_{|Z|}|d)] = [\theta^d_{z_1}, \ldots \theta^d_{z_{|Z|}}]$. Given a meta-path with pattern, $\mathcal{P}{:}A_s \xrightarrow{R_1} \ldots K \ldots \longrightarrow A_c \longrightarrow \ldots K^- \ldots \xrightarrow{R_{l-1}} A_t$, with K and K^- are two sets of text-based nodes which are occurred in meta-path $\mathcal{P}$. Because most of meta-paths are composed in a symmetric way, that is, the sets of left side nodes are the same with the right side ones, therefore, sets of text-based nodes $(k, k{\in}K)$ and $(k^-, k^-{\in}K^-)$ always have a same size, $|K| = |K^-|$. For example, we have a lot of common meta-paths with their symmetric text-based nodes K and K^- in DBLP network, like as A-P-V-P-A, A-P-P-A-P-P-A, V-P-A-P-V, etc. (as illustrated in Fig. 3A) or meta-paths: U-C-P-C-U, U-C-C-C-U, etc. in Facebook social network or meta-paths: U-M-A[actor]-M-U, U-M-G[genre]-M-U, etc. in IMDB movie network (as illustrated in Fig. 3B). In a specific content-based HINs, we consider a meta-path $\mathcal{P}$ with n number of symmetric text-based nodes at each side of meta-path $n = |K| = |K^-|$, the topic similarity weight of given meta-path $\mathcal{P}$, denoted as: $w_\mathcal{P}$ is calculated as following equation (Eq. (2)):

$$w_\mathcal{P} = \frac{\sum_{i=1}^{n} \frac{\overrightarrow{\theta^k}.\overrightarrow{\theta^{k^-}}}{||\theta^k||\cdot||\theta^{k^-}||}}{n}, k \in K \text{ and } k^- \in K^- \tag{2}$$

Where,

- n, is number of text-based nodes at each site of given meta-path $\mathcal{P}$, $n = |K| = |K^-|$.
- θ^k and θ^{k^-}, are the topic distributions which are extracted via LDA model of text-based nodes at k-th and k^--th in given meta-path $\mathcal{P}$.

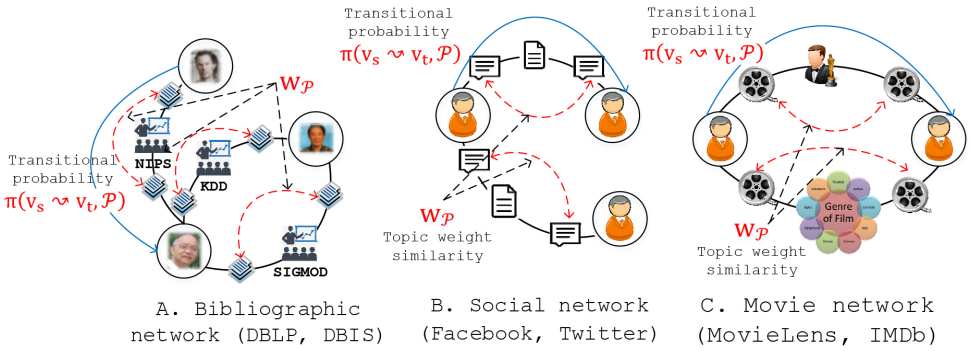


Fig. 4. Illustrations of topic weight similarity measurement $w_{\mathcal{P}}$ in topic-driven meta-path-based random walk between same-typed nodes in C-HINs.

In fact, calculating the topic similarity weight of path instances (see Fig. 4) between nodes can support the process of evaluating the node proximity while learning the microscopic network representations. By adding the topic similarity weight of meta-path to the transitional probabilistic calculation, we can provide a direct incorporation between content-based attributes (text, topics, subjects, etc.) with structure-based attributes (links) of network learning process. Traditional approaches of microscopic node proximity learning only focus on transforming network's nodes into similar low-dimensional vector spaces if they have been linked via direct (first-order) or indirect (second-order, third-order, etc.) relationships. However, evaluating the similarity between nodes via their number of relations is not enough. For example, several authors working on multiple disciplines might share same sets of venues/conferences (A-P-V-P-A) because top authors usually submit/publish their works at same top venues/conferences. However, the differences in their interesting topics might make specific groups of authors who work on same topics be more similar than other groups, such as "Jiawei Han" and "Christos Faloutsos" must be similar to "Philip S. Yu" than "Yoshua Bengio" because they are all interesting on "data mining" topic. By combining the topic similarity weight of paths with transitional probabilities of path between same-typed nodes in content-based HINs, also called topic-driven meta-path-based random walk mechanism, our proposed W-Com2Vec model is promised to effectively perverse both content-based and structure-based attributes of given HINs.

3.1.3. Meta-path-based network representation learning

In this part, we demonstrate the approach of using heterogeneous skip-gram model which are inspired from Metapath2Vec model [7] to learn the microscopic network representation. From topic-driven meta-path-based random mechanism (described in Sections 3.1.1 and 3.1.3), we obtain the set of contextual nodes, denoted as (c_t) for each target node (v) with t is a specific type of target node and contextual nodes. In skip-gram model, from a specific give node (v), we predict the existence of contextual nodes (c_t) which are same-typed with node (v). This is achieved by maximizing the following probability function (as shown in Eq. (3)):

$$argmax_\theta \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} Prob(c_t|v;\theta) \tag{3}$$

Where,

- $N(v)$, presents the set of neighborhood nodes of target node (v).
- $N_t(v)$, presents the set of neighborhood nodes belong to (t) type of target node (v).

We further inspired the approach of heterogeneous negative sampling of Dong et al. in Metapath2Vec model which use the softmax function to normalize the node v to specific contextual nodes belong to a specific type t, denoted as:

$$Prob(c_t|v;\theta) = \frac{e^{\mathbb{N}_v.\mathbb{N}_{c_t}}}{\sum_{neg_t \in V_t} e^{\mathbb{N}_{neg_t}.\mathbb{N}_{c_t}}},$$

with V_t is a set of network's nodes belong to (t) type and neg_t presents a set of sampling nodes of contextual nodes c_t. The ultimate purpose of applying negative sampling in heterogeneous microscopic sampling is to speed up the training process by selecting small sets of same-typed nodes (neg_t), normally 5 nodes, of contextual nodes c_t in each iteration. The objective function is applied to maximize the occurrence of contextual nodes (c_t) to target node (v) as following (see Eq. (4)):

$$\mathcal{O}_{\langle c_t,v\rangle} = \log\sigma(\mathbb{N}_{c_t} \times \mathbb{N}_v) + \sum_{k=1}^{K} \log\sigma(-\mathbb{N}_{neg_t^k} \times \mathbb{N}_v) \tag{4}$$

Where,

- K, is a defined number of same-typed negative sampling nodes which will be randomly selected in each training iteration.
- neg_t^k, presents a specific (k)-th negative node which is sampled for the contextual nodes (c_t).
- $\mathbb{N}_{c_t}$, $\mathbb{N}_v$ and $\mathbb{N}_{neg_t^k}$ present the vector representation of contextual nodes (c_t), target node (v) and negative sampling node neg_t^k.

In order to obtain the microscopic embedding of a heterogeneous network with given objective function (Eq. (4)), we apply the stochastic gradient descent (SGD). The SGD is used to approximate optimized model's parameters by gradually adjusting the $\mathbb{N}_v$ and $\mathbb{N}_{neg_t^k}$ via back propagation with a given defined learning rate (η), as following (as shown in Eqs (5a) and (5b)):

$$\mathbb{N}_v = \mathbb{N}_v - \eta\frac{\partial\mathcal{O}_{\langle c_t,v\rangle}}{\partial\mathbb{N}_v} \tag{5a}$$

$$\mathbb{N}_{neg_t^k} = \mathbb{N}_{neg_t^k} - \eta\frac{\partial\mathcal{O}_{\langle c_t,v\rangle}}{\partial\mathbb{N}_{neg_t^k}} \tag{5b}$$

In general, the embedding representation of $\mathbb{N}_v$ and $\mathbb{N}_{neg_t^k}$ are adjusted via feedforward and back propagation processes via SGD until the learning model is converged or specific of iteration number. The overall steps of microscopic network embedding are illustrated in Algorithm 1.

At first, we generate the sets of walks for each source node (v_s) in a given network G by applying the meta-path-based random walk mechanism (described in Eq. (1)). The meta-path-based walks for each source node are controlled by number of walks (*n_walk*) and walk's length (*l_walk*). These two values are selected depending on the size of networks and will sufficient impacts on the model's outputs as well as performance. After generating sets of meta-path-based walks for each source node in the network, we will apply the SGD to train our microscopic network embedding model. Then, each node (v) and its associated negative sampling node (neg_t) are sampled from generated meta-path-based walks in previous steps. Finally, in each iteration we update the vector representations of $\mathbb{N}_v$ and $\mathbb{N}_{neg_t^k}$ via back propagation following the Eqs (5a) and (5b).

Algorithm 1. Microscopic network embedding of W-Com2Vec model

Input:

- A given heterogeneous network, $G = (V, E, A, R)$.
- A given meta-path, $\mathcal{P}$
- Initialized parameters:
  - d (embedding dimension).
  - *n_walk* (number of walks per node).
  - *l_walk* (length of each walk).
  - K (negative sampling batch size).
  - η (learning rate).
  - *n_iteration* (number of iteration for SGD).

Output:

- Node vector representation for network's nodes ($\mathbb{N} \in \mathbb{R}^{|V| \times d}$).

Create: $\mathbb{N}$ as a matrix $\in \mathbb{R}^{|V| \times d}$
Create: meta-path random walks, $\mathbb{MP}$
For source node (v_s) in node sets (V):
  Loop range (1, *n_walk*):
    Loop range (1, *l_walk*):
      Draw: meta-path-based walk v_t (Eq. (1))
      Update: $\mathbb{MP}[v_s][\ldots] \leftarrow v_t$
    End loop
  End loop
End for
While (i) iteration $<$ *n_iteration*:
  Sampling: v and neg_t^k in $\mathbb{MP}$ with K
  Adjust: $\mathbb{N}_v$ (Eq. (5a)) and $\mathbb{N}_{neg_t^k}$ (Eq. (5b))
  Update: $i+ = 1$
End while

3.2. Meta-path-based mesoscopic network (intra-community) embedding

3.2.1. Louvain based community detection via meta-path

For modeling the intra-community-based structure of given HINs, we apply the maximum modularity technique to preserve the communities and network's nodes relations. In this work, we use Louvain algorithm to detect existent communities via specific meta-paths of a given HIN. Louvain algorithm is considered as a network modularity based community detection technique which depending on calculating the changes of network's modularity, also called "gain of modularity", denoted as: ΔQ by moving nodes to different communities. The model will iterate until a significant maximum of the overall network modularity (Q) is obtained. Louvain is considered more advanced than recent community detection techniques due to its high performance in large-scaled and complex networks. However, like other community detection approaches, Louvain is designed to work on homogeneous networks which only have single type of link between nodes. Our works in this paper focus on improving the Louvain model for detecting communities from HINs with multi-typed links between nodes via meta-paths. Then, the detected communities then are used to preserve the intra-community structure of nodes while learning network representation.

Consider a given HIN, denoted as: $G = (V, E, A, R)$, contain a set of communities, denoted as: C. Each community is also composed as a graph-based structure, denoted as: $G' = (V', E')$, $C = \{G'_1, G'_2 \ldots G'_n\}$. Because there are many types of relations between nodes so we need to use a specific meta-path, denoted as: $\mathcal{P}$ to specify the main relationships between nodes. The overall network modularity (Q) via meta-path $\mathcal{P}$, denoted as: $Q_{\mathcal{P}}$, is calculated as following equation (as shown in Eq. (6)):

$$Q_{\mathcal{P}} = \sum_{c=1}^{C} \frac{|\mathcal{P}_{V' \rightsquigarrow V'}|}{|\mathcal{P}_{V \rightsquigarrow V}|} - \left(\frac{\mathcal{P}_{V \rightsquigarrow V'}}{2 \times \mathcal{P}_{V' \rightsquigarrow V'}} \right)^2 \tag{6}$$

Where,

- C, is a set of community of the given network.
- $|\mathcal{P}_{V' \rightsquigarrow V'}|$, presents total number of path instances via meta-path $\mathcal{P}$ between nodes in each community (c) of a given network.
- $|\mathcal{P}_{V \rightsquigarrow V'}|$, presents total number of path instances between nodes of overall networks with nodes in community (c) via meta-path $\mathcal{P}$.
- $|\mathcal{P}_{V \rightsquigarrow V}|$, presents total number of path instances between all nodes of the given network via meta-path $\mathcal{P}$.

Similar to the approach of Louvain, at the first phase, each node will be assign to each separated community, it means with $|V|$ numbers of nodes we will have $|V|$ number of communities, or $|V| = |C|$. For each iteration, the community detection model will move each node to another community and then calculate the gain of modularity (ΔQ). The gain of modularity (ΔQ) of a specific node (v) which is moved to another community (c), denoted as: $G' = (V', E')$. All the links between nodes of community (c) and network (G) are identified by meta-path $\mathcal{P}$, the meta-path-based gain of modularity value of moving node (v) to community (c), denoted as: $\Delta Q_{\mathcal{P}}^{v \rightsquigarrow c}$, is calculated by following equation (see Eq. (7)).

$$\begin{aligned} \Delta Q_{\mathcal{P}}^{v \rightsquigarrow c} = & \frac{w_{\mathcal{P}}(\mathcal{P}_{V \rightsquigarrow V'}) + w_{\mathcal{P}}(\mathcal{P}_{v \rightsquigarrow V'})}{2W(\mathcal{P}_{V \rightsquigarrow V})} - \left(\frac{w_{\mathcal{P}}(\mathcal{P}_{V \rightsquigarrow V'}) + w_{\mathcal{P}}(\mathcal{P}_{v \rightsquigarrow V})}{2W(\mathcal{P}_{V \rightsquigarrow V})} \right)^2 \\ & - \left[\frac{w_{\mathcal{P}}(\mathcal{P}_{V' \rightsquigarrow V'})}{2w_{\mathcal{P}}(\mathcal{P}_{V \rightsquigarrow V})} - \left(\frac{w_{\mathcal{P}}(\mathcal{P}_{V \rightsquigarrow V'})}{2w_{\mathcal{P}}(\mathcal{P}_{V \rightsquigarrow V})} \right)^2 - \left(\frac{w_{\mathcal{P}}(\mathcal{P}_{v \rightsquigarrow V})}{2w_{\mathcal{P}}(\mathcal{P}_{V \rightsquigarrow V})} \right) \right] \end{aligned} \tag{7}$$

Where,

- $w_{\mathcal{P}}$, is the total weights of all path instances via meta-path $\mathcal{P}$. The weight of meta-path might be binary weight [0, 1] in non-content-based HINs or topic similarity weight in content-based HINs (described in 3.1.2).
- $\mathcal{P}_{v \rightsquigarrow V'}$, is a set of path instances between node (v) and all nodes in community (c) which node (v) is moved into, via meta-path $\mathcal{P}$.
- $\mathcal{P}_{V \rightsquigarrow V}$, $\mathcal{P}_{V' \rightsquigarrow V'}$ and $\mathcal{P}_{V \rightsquigarrow V'}$ present path instances between: nodes of overall network with community (c), nodes of community (c) and nodes of community (c) with overall network, respectively.

In general, the model loops through all communities of a given network and calculates all gain of modularity $\Delta Q_{\mathcal{P}}^{v \rightsquigarrow c}$, and node ($v$) will be assigned to community which has highest value of $\Delta Q_{\mathcal{P}}^{v \rightsquigarrow c}$. This process is repeated until the value of overall network's modularity (Q) (Eq. (6)) has no increase and our model become converged. In the second phase, the given network is restructured with new "super-nodes" which are represented by communities which are detected in the first phase. The weights of relationships between super-nodes are identified by sum of links' weights between nodes between their corresponding super-nodes/community. Then, when a new network is created, the first phase is reapplied. These two phases are repeated until the set of communities of a given network become stable. After finish the process of identifying community for each network's node, we construct a node-community membership indicator, denoted as: $\mathbb{H} \in \mathbb{R}^{|V| \times |C|}$, with each column presents for each network's community. For community representation, we use a non-negative matrix, denoted as: $\mathbb{C} \in \mathbb{R}^{|C| \times d}$ where each row presents for the network's community. For the approach of intra-community structure preserving in next part, we also produce a modularity matrix between network's nodes, denoted as: $\mathbb{Q} \in \mathbb{R}^{|V| \times |V|}$, where each cell of (i)-th

row and (j)-th column, denoted as: $\mathbb{Q}_{ij}$ presents as the modularity between (i)-th and (j)-th nodes for a specific meta-path $\mathcal{P}$, the modularity between two nodes is calculated as following: $\mathbb{Q}_{ij}=\frac{|\mathcal{P}_{i\rightsquigarrow j}|}{2\times|\mathcal{P}_{V\rightsquigarrow V}|}$, where $|\mathcal{P}_{i\rightsquigarrow j}|$ is the number of path instances between (i)-th and (j)-th nodes.

3.2.2. Meta-path-based mesoscopic network embedding

The main idea of intra-community preserving is that these nodes which belong to a specific community will have similar embedding spaces. Inspiring from the approach of modularized non-negative matrix factorization (M-NMF) model, we use the matrix factorization approach to capture the node representations with two non-negative matrices, denoted as: $\mathbb{X}\in\mathbb{R}^{|V|\times d}$ and $\mathbb{M}\in\mathbb{R}^{|V|\times d}$. The objective function is defined as following (as shown Eq. (8a)):

$$\text{argmin}_{\mathbb{M},\mathbb{X},\mathbb{H},\mathbb{C}}||\mathbb{N}-\mathbb{M}\mathbb{X}^T||_F^2+\alpha||\mathbb{H}-\mathbb{X}\mathbb{C}^T||_F^2-\beta tr(\mathbb{H}^T\mathbb{Q}\mathbb{H}) \tag{8a}$$

In this objective function, α and β parameters are model's hyper-parameters and $tr(.)$ is trace of matrix. For this approach, we use the matrix factorization method to optimize the objective function with four separated updates on $\mathbb{M}, \mathbb{X}, \mathbb{H}$ and $\mathbb{C}$. For non-negative matrix $\mathbb{M}$, we update $\mathbb{M}$ with $\mathbb{N}$ and $\mathbb{X}$ parameters, as following (see Eq. (8a)) [13]:

$$\mathbb{M}\leftarrow\mathbb{M}\odot\frac{\mathbb{N}\mathbb{X}}{\mathbb{M}\mathbb{X}^T\mathbb{X}} \tag{8b}$$

For non-negative matrix $\mathbb{X}$, we update $\mathbb{X}$ with $\mathbb{N}$, $\mathbb{M}$ and $\mathbb{C}$ as following (see Eq. (8b)) [13]:

$$\mathbb{X}\leftarrow\mathbb{X}\odot\frac{\mathbb{N}^T\mathbb{M}+\alpha\mathbb{H}\mathbb{C}}{\mathbb{X}(\mathbb{M}^T\mathbb{M}+\alpha\mathbb{C}^T\mathbb{C})} \tag{8c}$$

For estimating the representation of community embedding matrix $\mathbb{C}$, we update this parameter with $\mathbb{H}$ and $\mathbb{X}$ as: (see Eq. (8c)) [13]:

$$\mathbb{C}\leftarrow\mathbb{C}\odot\frac{\mathbb{H}^T\mathbb{X}+\alpha\mathbb{H}\mathbb{C}}{\mathbb{C}\mathbb{X}^T\mathbb{X}} \tag{8d}$$

For the node-community membership indicator matrix $\mathbb{H}$, the update of this parameter is more complex than other parameters due to the constraint of the $tr\left(\mathbb{H}^T\mathbb{H}\right)=|V|$ which makes the optimization of $\mathbb{H}$ becomes a NP-hard problem. In M-NMF model, Xiao Wang et al. proposed an approach of relaxing constraint of $\mathbb{H}^T\mathbb{H}=\mathbb{I}$ and provide an update rule for $\mathbb{H}$ as (as shown in Eq. (8d)), with λ is the model's regularization coefficient:

$$\mathbb{H}\leftarrow\mathbb{H}\odot\sqrt{\frac{-2\beta\mathbb{Q}\mathbb{H}+\sqrt{\Delta}}{8\lambda\mathbb{H}\mathbb{H}^T\mathbb{H}}} \tag{8e}$$

with $\Delta=\ 2\beta\left(\mathbb{Q}\mathbb{H}\right)\odot 2\beta\left(\mathbb{Q}\mathbb{H}\right)+16\lambda(\mathbb{H}\mathbb{H}^T\mathbb{H})\odot(2\beta\mathbb{A}\mathbb{H}+2\alpha\mathbb{C}\mathbb{X}^T+(4\lambda-2\alpha)\mathbb{H})$.

By using the Stochastic Gradient Descent (SGD) approach, the optimization process repeats the adjustments of $\mathbb{M}, \mathbb{X}, \mathbb{H}$ and $\mathbb{C}$ parameters until the model become converged. Finally, we obtain the final mesoscopic representation of network's nodes $\mathbb{X}\in\mathbb{R}^{|V|\times d}$ and community $\mathbb{C}\in\mathbb{R}^{|C|\times d}$.

The overall process of intra-community network preverseing is described in Algorithm 2. At first, we apply the meta-path-based random walk embedding to obtain the microscopic network structure represenation, $\mathbb{N}\in\mathbb{R}^{|\boldsymbol{V}|\times\boldsymbol{d}}$ (illustrated in Algorithm 1), via specific meta-path $\mathcal{P}$. Next, the meta-path-based Louvain community detection method (described in Section 3.2.1) is used to discover latent community inside the given network, then we use these outputs to initialize the community representation matrix: $\mathbb{C}\in\mathbb{R}^{|\boldsymbol{C}|\times\boldsymbol{d}}$. In the first phase, the model loops through all network's node to construct the node's

Algorithm 2. Mesoscopic network embedding of W-Com2Vec model

Input:

- A given heterogeneous network, $G = (V, E, A, R)$.
- A given meta-path, $\mathcal{P}$.
- Microscopic network's nodes representation $\mathbb{N} \in \mathbb{R}^{|V| \times d}$ (by applying Algorithm 1).
- Set of detected community $C = \{c_1, c_2 \dots c_n\}$, where each community (c) is represented as a graph, denoted as: $G' = (V', E')$ (by using meta-path-based Louvain community detection in Section 3.2.1).
- Initialized parameters:
  - d (embedding dimension).
  - Model's hyper-parameters: α, β
  - *n_iteration* (number of iteration)

Output:

- Mesoscopic (intra-community) node's representation: $\mathbb{X} \in \mathbb{R}^{|\boldsymbol{V}| \times \boldsymbol{d}}$ and community's representation: $\mathbb{C} \in \mathbb{R}^{|\boldsymbol{C}| \times \boldsymbol{d}}$.

1: Initialize: community representation matrix: $\mathbb{C} \in \mathbb{R}^{|C| \times d}$
2: Initialize: non-negative matrices: $\mathbb{X} \in \mathbb{R}^{V} \times d$, $\mathbb{M} \in \mathbb{R}^{|V| \times d}$
3: Initialize: node's modularity matrix: $\mathbb{Q} \in \mathbb{R}^{|V| \times |V|}$
4: For source node (i) in node sets (V):
5: For target node (j) in node sets (V):
6: Calculate: $\mathbb{Q}_{ij} = \frac{|\mathcal{P}_{i \rightsquigarrow j}|}{2 \times |\mathcal{P}_{V \rightsquigarrow V}|}$
7: End for
8: End for
9: While (i) iteration $<$ *n_iteration*:
10: Adjust: $\mathbb{M} \leftarrow \mathbb{M} \odot \frac{\mathbb{N}\mathbb{X}}{\mathbb{M}\mathbb{X}^T\mathbb{X}}$ (Eq. (8b))
11: Adjust: $\mathbb{X} \leftarrow \mathbb{X} \odot \frac{\mathbb{N}^T\mathbb{M} + \alpha\mathbb{H}\mathbb{C}}{\mathbb{X}(\mathbb{M}^T\mathbb{M} + \alpha\mathbb{C}^T\mathbb{C})}$ (eq. (8c))
12: Adjust: $\mathbb{C} \leftarrow \mathbb{C} \odot \frac{\mathbb{H}^T\mathbb{X} + \alpha\mathbb{H}\mathbb{C}}{\mathbb{C}\mathbb{X}^T\mathbb{X}}$ (Eq. (8d))
13: Adjust: $\mathbb{H} \leftarrow \mathbb{H} \odot \sqrt{\frac{-2\beta\mathbb{Q}\mathbb{H} + \sqrt{\Delta}}{8\lambda\mathbb{H}\mathbb{H}^t\mathbb{H}}}$ (Eq. (8e))
14: Update: $i+ = 1$
15: End while

modularity matrix: $\mathbb{Q} \in \mathbb{R}^{|\boldsymbol{V}| \times |\boldsymbol{V}|}$ by evaluating the number of path instances of meta-path $\mathcal{P}$ between two nodes (line 6). Next, the model try to adjust four main model's prameters, which are: $\mathbb{M}, \mathbb{X}, \mathbb{H}$ and $\mathbb{C}$ seperately via specific adjustment rules (lines 10–13). After a specific interation number, the model become converged and we can obtain the final mesoscopic representation of network's nodes $\mathbb{X}$ and community $\mathbb{C}$.

4. Experiments and discussions

To demonstrate the effectiveness of our proposed W-Com2Vec model, we conduct thorough experiments on real-world datasets compared to recent state-of-the-art INE models by solving the network's node clustering and classification tasks.

4.1. Experimental settings and dataset usage

4.1.1. Experimental settings

To illustrate the correctness of our studies in this paper, we compare our proposed W-Com2Vec model against the following well-known embedding algorithms for both homogeneous (HoIN) and heterogenous (HIN) networks, which are (as shown in Table 1).

Table 1
Comapred HoIN-based and HIN-based nework embedding algorithms

Network's type	Algorithm	Description
HoIN	DeepWalk [9]	DeepWalk is considered as an primary HoIN-based network embedding algorithm which uses uniform random walk mechanism to capture w-hop contextual neighbors of each network's node. Then, these contextual neighbors are used to learn the $\|d\|$-dimensional node's representation.
	LINE (LINE_1 and LINE_2) [10]	LINE has two main version, first-order node proximity (LINE_1) and second-order node proximity (LINE_2) which aims to learning $\|d\|$-dimensional node's representation by exploring the common neighbors of evaluted pairwise nodes.
	Node2Vec [11]	Similar to the approach of using random walk mechanism in DeepWalk model, Node2Vec introduces two techiques for exploiting contextual nodes, which are: BFS (breath-first-sampling) and DFS (deep-first-sampling) with two initialized parameters (p, q).
	ComE [12]	ComE is considered as a first community-based embedding approach. In this model, Vincent W. Zheng et al. propose a combintation of node embedding and community detection to preserve the community-aware in network's structure representation. The intra-community embedding is obtained by using multivariate Gaussian distribution to evaluate the distributions of network's nodes inside each community.
	M-NMF [13]	The M-NMF is also considered as a community-aware network representation learning approach. In this model, Xiao Wang et al. introduce an approach of learning both node proximity (first and second orders) and intra-community representation together via non-negative matrix factorization.
HIN	HIN2Vec [6]	This model proposed a noval approach of meta-path-based node proximity to learn the reprensentations of same-typed nodes and their corresponding meta-paths. The HIN2Vec model uses Hadamard multiplication of nodes and meta-paths for acheiving heterogeneous semantic features of the given HIN.
	Metapath2Vec [7]	Considering as the most well-known meta-path-based network representation learning approach. In Metapath2Vec model, Dong et al. uses meta-path-based random walk mechanism and heterogeneous Skip-gram technique to maximize the occurence probability of contextual nodes for given evaluated nodes. Metapath2Vec has two versions: Metapath2Vec (for homogeneous network) and Metapath2Vec++ (for heterogeneous network). For experiments in this paper, we only use Metapath2Vec++ version.

Table 2
Initalized model's paramters

Parameter	Value
Embedding dimension (d)	128
Number of walks per node (*n_walk*)	1000
Length of each walk (*l_walk*)	100

For all experiments, we use the same initialized model's parameters for all network embedding algorithms, including embedding dimension (d), number of walks per node (*n_walk*) and lengh of each walk (*l_walk*) (as shown in Table 2). For experiments with meta-path-based algorithms, includes: HIN2Vec and MetapathVec we use the same negative sampling batch size *neg_batch_size* $= 5$.

In order to evaluate the performance of listed network embedding algorithms (Table 1), we apply these baselines for solving two principal network analysis and mining tasks, which are:

– Node clustering task experiment. Node clustering is considered as a primative task in network analysis and mining tasks. For the outputs of node's reprensetations, we use k-means algorithm to group network's nodes into defined k clusters. The number of groups (k) is already identified in the used dataset. To evaluate the output's quality of network's node clustering tasks via different

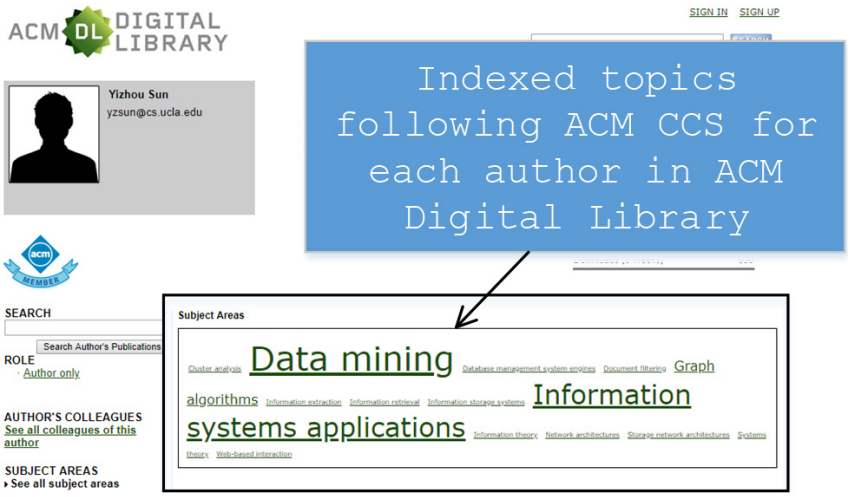


Fig. 5. Example of indexed topics for author "Yizhou Sun" in ACM digital library.

embedding algorithms, we use three evaluation metrics: F1 measure, Purity and NMI (normalized mutual information). The inital centroids of k-means algorithm for each models are randomly selected and we repeat each test 10 times and take the average value of all outputs as the final result.

– Node classification task experiment. For node classification task, we use the defined dataset's labels for determining the class of each network's node. The outputs of network's representation learning in each model are divided into two main parts, which are: training and testing. For each test case, we vary the training set from 10% to 90% and the remaining nodes are used for the testing set. Then the training set is used to train the Logistic Regression (LR) classifier to predict the classes of nodes in testing set. The class prediction outputs are then evaluated by Marco-F1 and Micro-F1 metrics. We repeat the node classification task 20 times for each network embedding model and report the average performance.

4.1.2. Experimental metric usage

To evaluate the performance of embedding approaches, in this paper we use different model evaluation metrics, includes:

– F-1 measure [18]: is calculated as: $F1 = 2\frac{PR}{P+R}$, where P (precision) is the precision of a given class/cluster in classification/clustering task, denoted as: $P = \frac{TP}{TP+FP}$ and R (recall) is the measurement of the true positive rate (TPR) or sensitivity of a given class/cluster in classification/clustering task, denoted as: $R = \frac{TP}{TP+FN}$.

– Purity [18]: is a traditional metric for evaluate the quality of clustering/classification model. It calculates the percentage of total corrected categorized data points within range [0, 1], denoted as: $Purity = \frac{1}{N}\sum_{c\in C}\max_{y\in Y}|c\cap y|$, where Y (clusters/classes) and C (labels).

– NMI (Normalized Mutual Information) [18]: is the most common metrics for evaluating the quality of classification/clustering models, denoted as: $NMI(Y,C) = \frac{2\times I(Y;C)}{H(Y)+H(C)}$, where $I(Y;C)$ presents for mutual information between Y (clusters/classes) and C (labels), and $H(.)$ is the entropy value.

– Cohen's Kappa Index ($\mathcal{K}$) [18]: (also called Cohen's Kappa coefficient) helps to measure the agreement between the ground truth (labels) and clustering/classification outputs. $\mathcal{K}$ is denoted as: $\frac{p_o-p_e}{1-p_e}$, where:

Table 3
List of used datasets for experimental studies

Dataset's name	Dataset's description
DBLP bibliographic network[1]	This is the most well-known networked dataset which is used for conducting empirical studies in almost network analysis and representation learning algorithms. DBLP is considered as a content-based HIN with over 4.7M paper nodes and the corresponding abstract's contents which are collected from Aminer dataset.[2] DBLP dataset also has 2.3M author nodes and over 7K venue/journal nodes. – Author's label/class: For labels/classes of author nodes we use the external third-party labels from ACM Digital Library resource. In ACM resource, there are authors' profiles which are indexed by topics which are organized following the ACM Computing Classification System 2012.[3] And we use these indexed topics as the labels/classes for each other. One author might be indexed by multiple ACM's topics but we only select the topic with highest ranking as the main author's label/class. For example, author "Yizhou Sun" is assigned to "data mining" (highest indexing topic) class in ACM digital library (as illustrated in Fig. 5). We have collected 300K author's profiles from ACM digital library and categorized these 300K authors into 8 main topics, which are: "hardware", "security & privacy" "data mining", "artificial intelligence", "computer vision", "networks", "database & information systems", "software & and its engineering". The set of 300K labelled authors are then used for node classification task. – Venue's and journal's label/class: We use the indexed topics of Google Scholar Metrics (GSM)[4] as the third-party ground truth classes for venues and journals in DBLP network (as illustrated in Fig. 6). We have collected five main indexed topics, which are: "database/data mining", "artificial intelligence", "computer graphics", "computer hardware design" and "computer networks & wireless communication" from GSM and assigned to 280 venues/journals in DBLP network. The set of 280 labelled venues/journals are used for further experiments on node classification task.
MovieLens1M movie network dataset[5]	This is a movie network dataset contains about 1M ratings of 6K users to 4K movies. In our experiments, we also consider MoveLens1M as a context-based HIN with 4K movie's descriptions which are collected from three movie networks: MovieLens,[6] IMDM[7] and TMDB[8] (as illustrated in Fig. 7). For labels/classes of user node in MovieLens1M, we used movie's genre which user mostly rated highest scores to. For example, a user mostly rates 5/5 score to movies which belongs to "Animation" genre should be classified as "Animation" class. Depending on the ratings data of MovieLens1M dataset, we assigned 6K users to the appropriate 15 movie's genres. Then, these labelled user nodes are used for evaluating the node classification task of network embedding models.

* p_o is the experimental observed agreement between the ground truth and the (clustering/classification) model's outputs
* p_e is the expected agreement between the ground truth and the (clustering/classification) model's outputs

4.1.3. Experimental dataset usage

In this section, we describe about the datasets which are used for conducting experiments with different network embedding baselines. For all tests of node clustering and classification, we use two real-world datasets, which are (as listed in Table 3).

For DBLP dataset, we apply network embedding models to learn the representations of author and venue nodes. For author nodes representation learning, we use the meta-path: A-P-V-P-A, which indicates the

[1] DBLP dataset: https://dblp.org/db/journals/network/.
[2] Aminer dataset: https://aminer.org/.
[3] ACM CCS-2012: https://www.acm.org/publications/class-2012.
[4] Google Scholar Metric (GSM) for top venues/journals in "Artificial Intelligence" topic: https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_artificialintelligence.
[5] MovieLens1M dataset: https://grouplens.org/datasets/movielens/1m/.
[6] MovieLens website: https://movielens.org/.
[7] IMDB website: https://www.imdb.com/.
[8] TMDB website: https://www.themoviedb.org/.

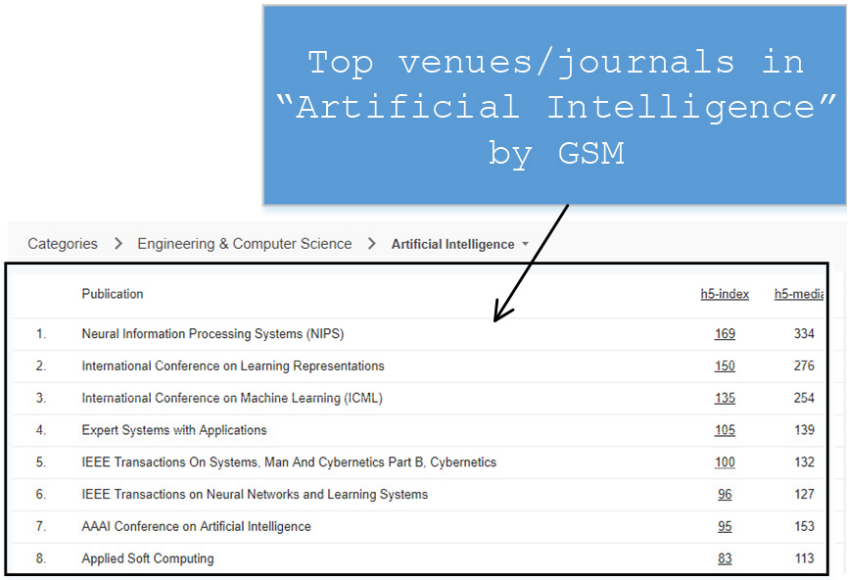


Fig. 6. Example of top venues/journals in "Artificial Intelligence" topic by GSM.

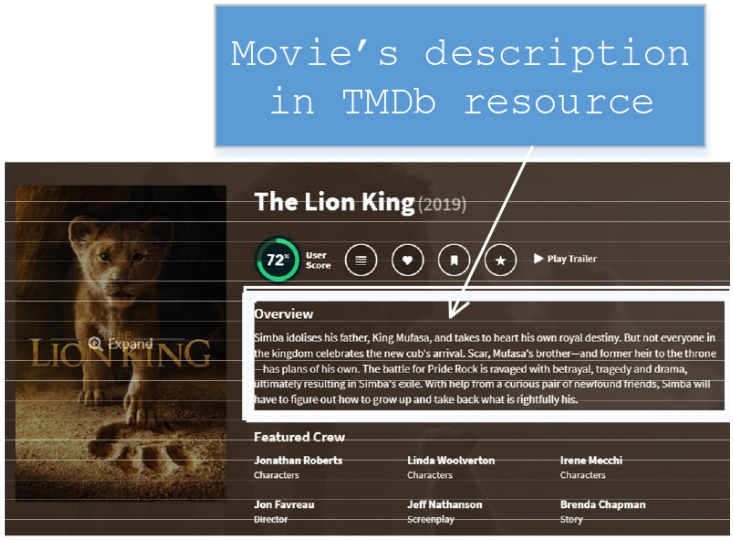


Fig. 7. Example of a movie's description which are collected from TMDb.

relationships of two authors who frequently submit their works at common venues/journals, as the main meta-path for HIN-based models (HIN2Vec, Metapath2Vec and W-Com2Vec). For venue/journal nodes representation learning, we use the meta-path: V-P-A-P-V. This meta-path indicates the relationships of venues/journals which share common sets of authors (who frequent published their works to these venues/journals). For MovieLens1M dataset, we use the meta-path: U-M-G-M-U which indicates two users who frequently rate for movies which belong to a same genre. For HoIN-based models, such as: DeepWalk, LINE, Node2Vec, ComE and M-NMF, we transform the two given heterogeneous datasets into homogeneous datasets (Hin2HoIN) by create new links between target-typed nodes (authors, venues

Table 4
Accuracy of author node clustering in DBLP dataset in terms of F1, Purity, NMI and Kappa Index ($\mathcal{K}$) ($k = 8$)

	F1	Purity	NMI	$\mathcal{K}$
DeepWalk	0.71623	0.81723	0.57281	0.61603
LINE_1	0.68291	0.79825	0.55921	0.60145
LINE_2	0.72812	0.82082	0.56271	0.62829
Node2Vec	0.74821	0.82776	0.57281	0.65998
ComE	0.76721	0.83927	0.58176	0.67571
M-NMF	0.79031	0.84793	0.63281	0.70639
HIN2Vec	0.78271	0.83721	0.61823	0.68916
Metapath2Vec	0.81723	0.85829	0.65212	0.73478
W-Com2Vec	**0.83721**	**0.86052**	**0.67823**	**0.76563**

Table 5
Accuracy of venue/journal node clustering in DBLP dataset in terms of F1, Purity, NMI and Kappa Index ($\mathcal{K}$) ($k = 5$)

	F1	Purity	NMI	$\mathcal{K}$
DeepWalk	0.83721	0.92823	0.63128	0.77682
LINE_1	0.79281	0.90281	0.60217	0.69283
LINE_2	0.81723	0.91872	0.61879	0.76271
Node2Vec	0.85627	0.92721	0.64823	0.79286
ComE	0.86271	0.93712	0.66584	0.80584
M-NMF	0.88682	0.94872	0.68685	0.82286
HIN2Vec	0.87261	0.93687	0.67271	0.81577
Metapath2Vec	0.91271	0.94153	0.69032	0.86996
W-Com2Vec	**0.92287**	**0.95026**	**0.69873**	**0.88228**

in DBLP, users in MovieLens1M) following the used meta-path and removing all other node's types. This make easier for HoIN-based approaches to directly learn target-typed nodes and their corresponding semantic relations which are represented via evaluated meta-paths (A-P-V-P-A and V-P-A-P-V in DBLP, U-M-G-M-U in MovieLens1M).

4.2. Experimental outputs and discussions

4.2.1. Node clustering

In this section, we demonstrate the experiments of using different network embedding model for solving node clustering task. The node representation outputs of embedding models are used as the input for k-mean clustering algorithm. The number of cluster (k) for each node's type in each dataset is set equal to the number of labels/classes of these nodes' types in the given datasets (described in Section 4.1.3). For author and venue node clustering tasks in DBLP network, the initial number of clusters are 8 ($k = 8$) and 5 ($k = 5$), respectively. For MovieLens1M dataset, we use the number of movie's genres ($k = 15$) which each user is assigned to. To measure the distance between two nodes in k-means algorithms we use Euclidean distance metric. We use F1, Purity and NMI metrics for evaluating the outputs of each network embedding models. All node clustering experiments in two datasets are conducted 10 times and report the average value as the final result.

For DBLP bibliographic network, Tables 4 and 5 respectively report the experimental outputs in terms of F1, Purity and NMI for author and venue/journal nodes clustering tasks. In overall, our proposed W-Com2Vec model outperforms all state-off-the-art network embedding approaches. In comparing with HoIN-based approaches, W-Com2Vec model improve the performance of node clustering task about

Table 6
Accuracy of user node clustering in MovieLens1M dataset in terms of F1, Purity, NMI and Kappa Index ($\mathcal{K}$) ($k = 15$)

	F1	Purity	NMI	$\mathcal{K}$
DeepWalk	0.74621	0.83721	0.59253	0.63838
LINE_1	0.73528	0.81998	0.57928	0.61879
LINE_2	0.74678	0.82675	0.58078	0.65978
Node2Vec	0.76281	0.85621	0.60291	0.67809
ComE	0.77261	0.86775	0.62677	0.68797
M-NMF	0.78695	0.87986	0.64085	0.70378
HIN2Vec	0.79827	0.86218	0.63956	0.73514
Metapath2Vec	0.83728	0.87828	0.65227	0.74602
W-Com2Vec	**0.84078**	**0.88472**	**0.66083**	**0.75699**

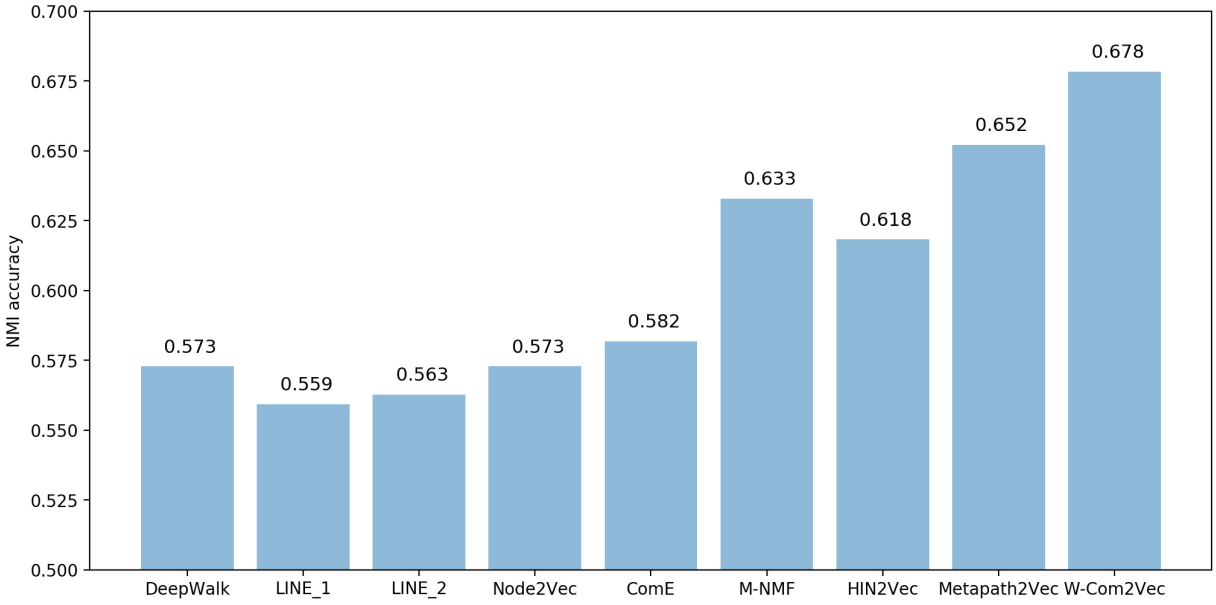


Fig. 8. Performance of author node clustering in terms of NMI evaluation metric (DBLP).

14.35% (DeepWalk), 17.55% (average for LINE_1 & LINE_2), 12.76% (Node2Vec), 10.36% (ComE) and 4.34% (M-NMF) (Figs 8 and 9). For HIN-based approaches, such as: HIN2Vec and Metapath2Vec, W-Com2Vec also gain better performance approximately 6.67% and 2.57% in comparing with HIN2Vec and Metapath2Vec, respectively. The experimental outputs also demonstrate the author nodes clustering is more challenging than venue/journal nodes clustering task due to the larger number of nodes and clusters. With MovieLens1M dataset (see Table 6), the experimental results also indicate our proposed W-Com2Vec model produces better quality of node representations than recent network embedding baselines. The W-Com2Vec model achieves averagely 9.59% and 2.31% improvements in comparing with HoIN-based models (DeepWalk, LINE_1 & LINE_2, Node2Vec, ComE and M-NMF) and HIN-based models (HIN2Vec and Metapath2Vec), respectively (Fig. 10). To further observe the behaviors and performance of W-Com2Vec model in node clustering task, we calculated the Cohen's Kappa index ($\mathcal{K}$) and NMI of different node clustering tasks. Figure 11 illustrates the comparisons of node clustering tasks by W-Com2Vec model in different datasets (DBLP, MovieLens) in terms of $\mathcal{K}$ and NMI metrics. As shown in the first chart (for author node clustering task in DBLP network), the both $\mathcal{K}$ and NMI

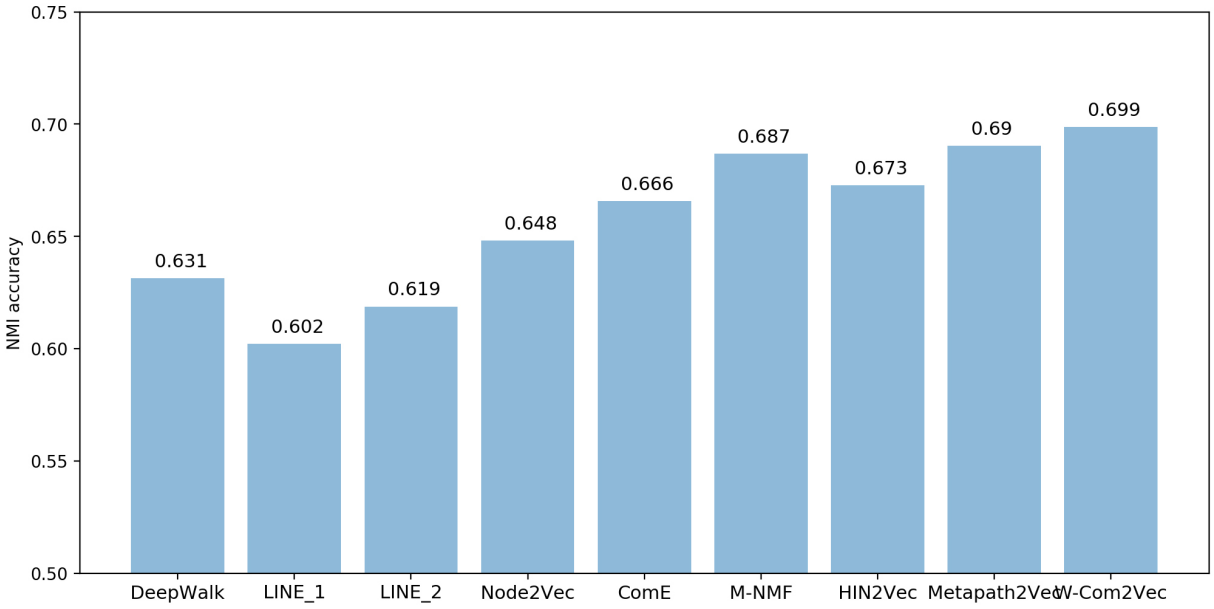


Fig. 9. Performance of venue node clustering in terms of NMI evaluation metric (DBLP).

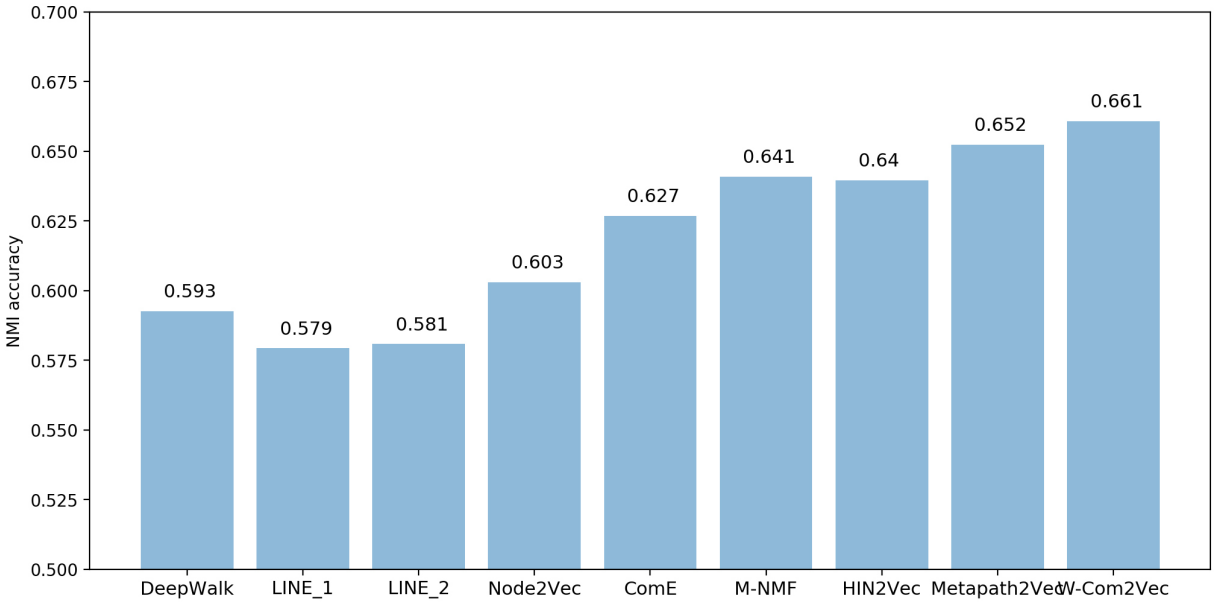


Fig. 10. Performance of user node clustering in terms of NMI evaluation metric (MovieLens).

increase gradually with growing node numbers (from 10% to 80%). Within range of $>$ 80% to 100%, the $\mathcal{K}$ increases rapidly to the maximum value at around 76.56% which are similar to NMI also raises tremendously from around 55% to 67.82%. The behaviors of $\mathcal{K}$ and NMI metrics in venue (DBLP) and user (MovieLens) nodes clustering are similar with author node (DBLP) clustering. As shown in the second and third line charts (Fig. 11), at first, both $\mathcal{K}$ and NMI increase gradually within 10% to 80%

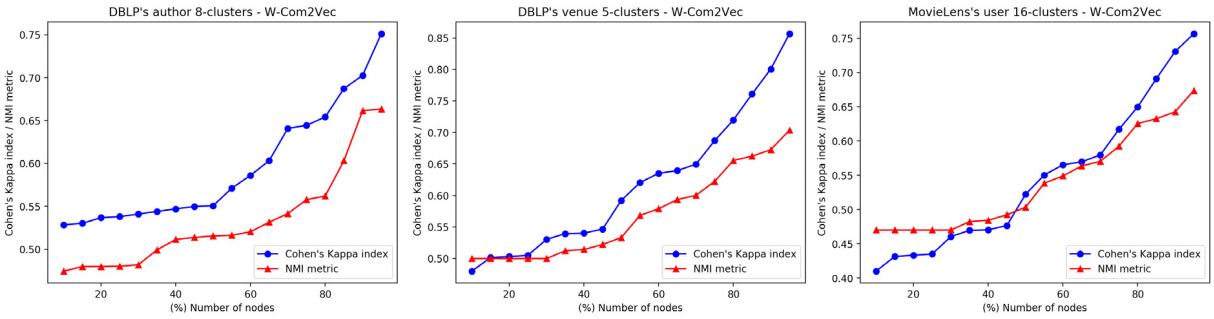
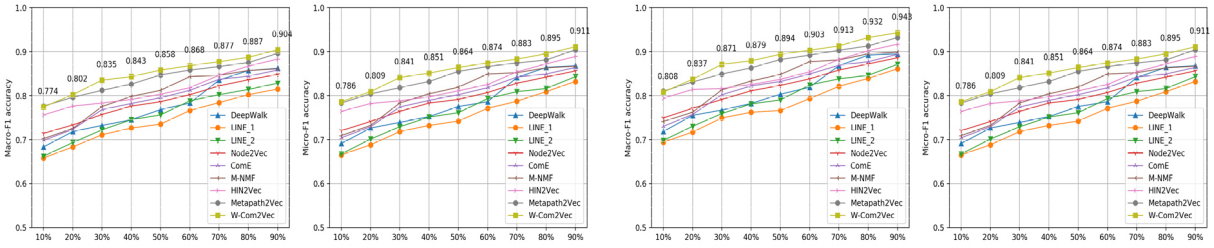


Fig. 11. Cohen's Kappa index vs. NMI metric for node clustering task of W-Com2Vec model in DBLP and Movielens datasets.



A. Author node (DBLP) B. Venue/journal node (DBLP)

Fig. 12. Author and venue/journal node classification tasks in DBLP with different sizes of training set (%) in terms of Macro-F1& Micro-F1 evaluation metrics.

range of dataset, then rapidly increase when the dataset is over 80%. For venue and user nodes clustering tasks by W-Com2Vec, the $\mathcal{K}$ metric gains maximum value at 88.2% and 75.7%, respectively.

4.2.2. Node classification

In this section, we demonstrate experimental studies on using multiple network embedding baselines for solving network's node classification task. For DBLP network, we use the set of 300K author nodes which are assigned to 8 topics/classes by ACM digital library and the set of 280 venues which are assigned to 5 classes by GSM. For MovieLens1M network, we use the set of 6K users which are assigned to 15 movie's genres/classes. At first, each embedding model is applied to learn representations of all network's nodes. Next, we vary the size of the node embedding outputs from 10% to 90% which are used as the training set and the remaining part are used as the testing set. The training set is used to feed the Logistic Regression (LR) classifier and then is used to predict the class of nodes in testing set. For each embedding approach, we repeat the node classification experiments 20 times and report the average value as the final result. In this experiment, we use Macro-F1 and Micro-F1 metrics for evaluating the node classification outputs.

For DBLP dataset, Tables 7 and 8 list the experimental outputs for author and venue/journal nodes classification in terms of Macro-F1 and Micro-F1. The experimental results demonstrate that our proposed W-Com2Vec model outperforms all recent embedding baselines. For author node classification task (Fig. 12A), W-Com2Vec significantly achieves better performance than HoIN-based approaches about 9.54% (DeepWalk), 13.36% (LINE_1 & LINE_2), 8.13% (Node2Vec), 7.52% (ComE) and 5.96% (M-NMF). With HIN-based approaches, W-Com2Vec model also slightly increases the performance of node classification task approximately 4.45% and 1.27% in comparing with HIN2Vec and Metapath2Vec

Table 7
Accuracy for author nodes classification in DBLP dataset in terms of Macro-F1 and Micro-F1

		10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk	0.683	0.718	0.732	0.745	0.768	0.783	0.835	0.857	0.862
	LINE_1	0.658	0.683	0.711	0.726	0.735	0.766	0.784	0.802	0.815
	LINE_2	0.662	0.693	0.722	0.745	0.756	0.788	0.802	0.814	0.828
	Node2Vec	0.714	0.733	0.757	0.776	0.786	0.802	0.822	0.836	0.848
	ComE	0.698	0.723	0.768	0.782	0.795	0.812	0.837	0.844	0.858
	M-NMF	0.702	0.725	0.775	0.798	0.812	0.843	0.846	0.857	0.861
	HIN2Vec	0.756	0.776	0.783	0.791	0.803	0.818	0.846	0.867	0.883
	Metapath2Vec	**0.776**	0.796	0.812	0.826	0.847	0.858	0.866	0.875	0.896
	W-Com2Vec	0.774	**0.802**	**0.835**	**0.843**	**0.858**	**0.868**	**0.877**	**0.887**	**0.904**
Micro-F1	DeepWalk	0.691	0.726	0.739	0.752	0.775	0.786	0.841	0.864	0.868
	LINE_1	0.665	0.688	0.718	0.732	0.742	0.771	0.787	0.809	0.832
	LINE_2	0.667	0.701	0.729	0.751	0.761	0.793	0.809	0.816	0.843
	Node2Vec	0.720	0.741	0.764	0.783	0.791	0.807	0.828	0.842	0.856
	ComE	0.704	0.729	0.774	0.789	0.802	0.818	0.842	0.849	0.864
	M-NMF	0.709	0.732	0.783	0.804	0.819	0.849	0.852	0.863	0.867
	HIN2Vec	0.764	0.782	0.788	0.797	0.811	0.825	0.854	0.872	0.889
	Metapath2Vec	0.783	0.804	0.818	0.832	0.855	0.866	0.874	0.881	0.904
	W-Com2Vec	**0.786**	**0.809**	**0.841**	**0.851**	**0.864**	**0.874**	**0.883**	**0.895**	**0.911**

Table 8
Accuracy for venue/journal nodes classification in DBLP dataset in terms of Macro-F1 and Micro-F1

		10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk	0.718	0.755	0.767	0.782	0.803	0.819	0.868	0.892	0.895
	LINE_1	0.694	0.717	0.749	0.762	0.766	0.793	0.821	0.839	0.861
	LINE_2	0.698	0.730	0.759	0.781	0.790	0.824	0.838	0.846	0.871
	Node2Vec	0.749	0.771	0.791	0.811	0.823	0.838	0.858	0.872	0.886
	ComE	0.730	0.757	0.802	0.822	0.832	0.849	0.869	0.877	0.894
	M-NMF	0.741	0.762	0.813	0.833	0.848	0.877	0.881	0.895	0.899
	HIN2Vec	0.794	0.814	0.816	0.825	0.837	0.854	0.882	0.902	0.917
	Metapath2Vec	0.810	0.831	0.849	0.863	0.882	0.892	0.903	0.913	0.932
	W-Com2Vec	**0.818**	**0.837**	**0.871**	**0.879**	**0.894**	**0.903**	**0.913**	**0.932**	**0.943**
Micro-F1	DeepWalk	0.728	0.765	0.777	0.792	0.813	0.829	0.878	0.901	0.905
	LINE_1	0.703	0.727	0.759	0.771	0.779	0.819	0.827	0.849	0.869
	LINE_2	0.707	0.737	0.769	0.791	0.798	0.833	0.848	0.855	0.881
	Node2Vec	0.758	0.780	0.800	0.821	0.832	0.848	0.868	0.882	0.896
	ComE	0.736	0.766	0.811	0.831	0.841	0.859	0.878	0.886	0.903
	M-NMF	0.750	0.772	0.822	0.843	0.857	0.887	0.888	0.905	0.908
	HIN2Vec	0.804	0.824	0.826	0.834	0.847	0.863	0.892	0.911	0.929
	Metapath2Vec	0.819	0.841	0.859	0.872	0.892	0.902	0.913	0.922	0.941
	W-Com2Vec	**0.828**	**0.847**	**0.881**	**0.888**	**0.903**	**0.912**	**0.922**	**0.941**	**0.952**

models, respectively. In venue/journal node classification task (Fig. 12B), W-Com2Vec clearly outperforms averagely 10.05% in comparing with overall HoIN-based methods (DeepWalk, LINE_1 & LINE_2, Node2Vec, ComE and M-NMF) and 2.9% in comparing with two HIN-based methods (HIN2Vec and Metapath2Vec). For MovieLens1M dataset (as shown in Table 9), W-Com2Vec model also consistently shows better performance than both HoIN-based (9.69%) and HIN-based (3.1%) approaches (Fig. 13). In overall, the experiments on node clustering and classification show the superiority of our proposed W-Com2Vec model in heterogeneous network's node representation learning. To evaluate the capability of distinguishing classes between nodes, we used the AUC/ROC metric to demonstrate the performance of node embedding quality by using our proposed W-Com2Vec model in node classification task with LR classifier (see Fig. 14). As shown in the second chart (Fig. 14), the venue (DBLP) node classification gain

Table 9
Accuracy for user nodes classification in MovieLen1M dataset in terms of Macro-F1 and Micro-F1

		10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk	0.704	0.737	0.750	0.766	0.787	0.804	0.853	0.877	0.879
	LINE_1	0.678	0.701	0.732	0.745	0.752	0.783	0.804	0.821	0.846
	LINE_2	0.680	0.713	0.741	0.765	0.773	0.807	0.821	0.832	0.857
	Node2Vec	0.732	0.754	0.776	0.796	0.806	0.822	0.844	0.856	0.869
	ComE	0.715	0.743	0.786	0.804	0.815	0.833	0.854	0.861	0.877
	M-NMF	0.724	0.745	0.796	0.818	0.832	0.860	0.866	0.877	0.882
	HIN2Vec	0.777	0.797	0.801	0.809	0.823	0.839	0.865	0.886	0.902
	Metapath2Vec	0.795	0.816	0.832	0.847	0.868	0.874	0.886	0.895	0.916
	W-Com2Vec	**0.802**	**0.823**	**0.854**	**0.862**	**0.879**	**0.889**	**0.897**	**0.916**	**0.925**
Micro-F1	DeepWalk	0.714	0.746	0.756	0.774	0.795	0.812	0.863	0.885	0.889
	LINE_1	0.687	0.710	0.742	0.754	0.762	0.793	0.813	0.824	0.855
	LINE_2	0.689	0.723	0.751	0.773	0.782	0.816	0.829	0.841	0.866
	Node2Vec	0.741	0.762	0.786	0.806	0.815	0.832	0.853	0.865	0.879
	ComE	0.724	0.752	0.794	0.813	0.824	0.841	0.864	0.869	0.887
	M-NMF	0.734	0.753	0.804	0.827	0.840	0.868	0.875	0.886	0.892
	HIN2Vec	0.785	0.805	0.811	0.818	0.832	0.847	0.874	0.895	0.911
	Metapath2Vec	0.805	0.825	0.841	0.857	0.877	0.883	0.896	0.904	0.924
	W-Com2Vec	**0.816**	**0.832**	**0.864**	**0.871**	**0.889**	**0.898**	**0.906**	**0.925**	**0.934**

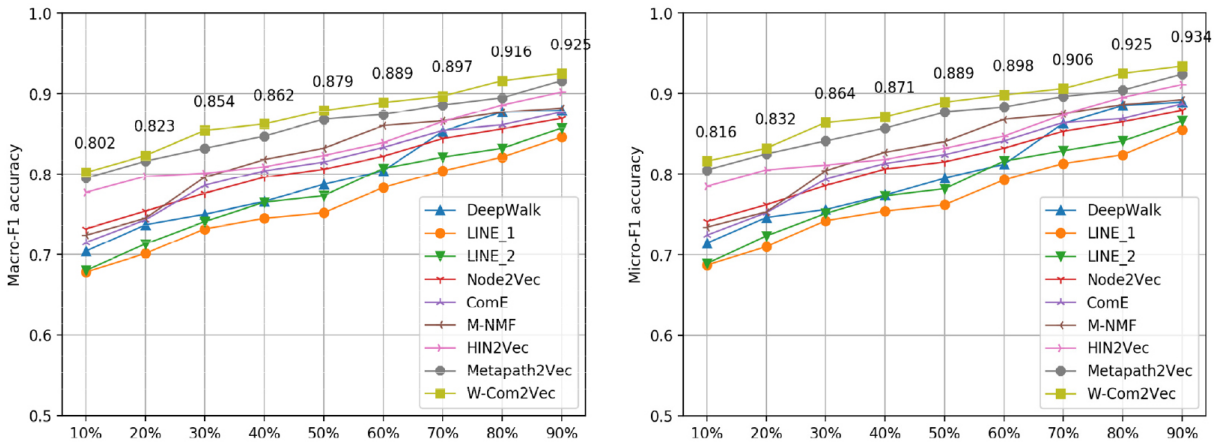


Fig. 13. User node classification task in MovieLens1M with different sizes of training set (%) in terms of Macro-F1& Micro-F1 evaluation metrics.

the highest AUC/ROC scores than author (DBLP) and user (MovieLens) nodes due to the small amount of nodes (about 280) and number of classes (only 5).

4.2.3. Parameter sensitivity analysis

In this part, we conduct extensive experiments for studying the influences of W-Com2Vec model's parameters on the quality of network representation output. We use the accuracy outputs of author and venue node clustering tasks which are evaluated by F1 metric, in DBLP network. The ultimate purpose of this experimental study is to show the effects of three parameters on overall model's performance, include: embedding dimension (d) (varying from 40 to 160), number of walks per node (n_walk) (varying from 100 to 1300) and length of each walk (l_walk) (varying from 60 to 180) walks per node (w) (from 100 to 1300), walk length (l) (from 60 to 180). As shown from experimental outputs on author and venue node clustering tasks (Fig. 15) in DBLP via F1 evaluation metric, the W-Com2Vec gain the highest

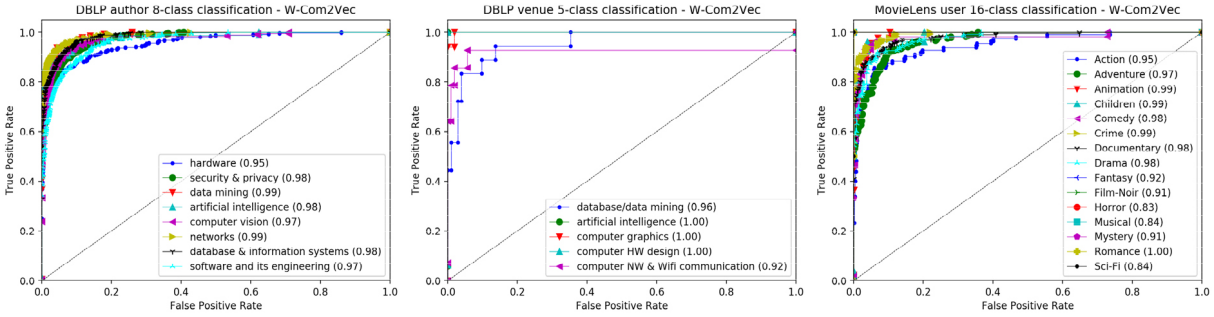


Fig. 14. AUC/ROC for node classification tasks in different datasets.

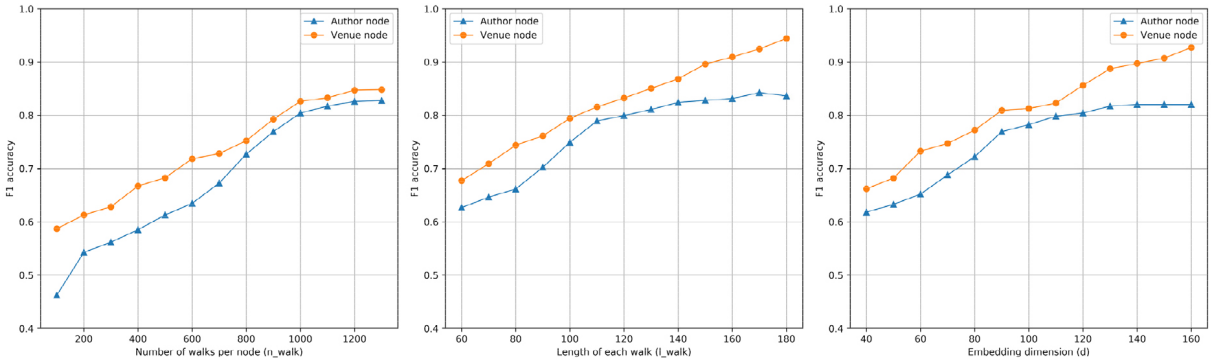


Fig. 15. Experiments on parameters sensitivity of W-Com2Vec model.

performance and become stable with embedding dimension (d) at about $128 \sim 160$, walks per node (n_walk) at around $1000 \sim 1200$ and length of each walk (l_walk) about $125 \sim 170$.

5. Conclusions and future works

In this paper, we formally present a novel approach of network embedding, namely W-Com2Vec which enables to preserve both microscopic (node proximity via topic driven meta-path-based random walk) and mesoscopic (intra-community) structures of HINs. At first, we propose a topic-driven meta-path-based random walk for capturing sematic context of each given nodes via meta-path. The semantic contextual nodes are then used to learn the heterogeneous node representations. This representation is known as microscopic structure preserving. Next, we introduce a novel approach of meta-path-based Louvain community detection for discovering community in HINs which are used to combine with previous node (microscopic) representations to exploit the consensus relationship between network's nodes and detected communities (mesoscopic). The matrix factorization technique is applied to jointly optimize both two representations in order to reproduce a final network's node representations where each community has similar embedding vectors with their node members. Extensive experimental studies on two real-world networked datasets demonstrate the effectiveness of our proposed W-Com2Vec model in comparing with recent state-of-the-art INE baselines. For future improvements, we intend to implement our W-Com2Vec model under the distributed processing environment of Apache Spark in order to enable the capability of handling large-scaled networks for our proposed model.

References

[1] C. Li and Y. Tang, Efficient Heterogeneous Proximity Preserving Network Embedding Model, *Expert Systems with Applications*, 2019.

[2] D. Zhang, J. Yin, X. Zhu and C. Zhang, Network representation learning: A survey, *IEEE transactions on Big Data*, 2018.

[3] P. Cui, X. Wang, J. Pei and W. Zhu, A survey on network embedding, *IEEE Transactions on Knowledge and Data Engineering* **31**(5) (2018), 833–852.

[4] Y. Du, W. Guo, J. Liu and C. Yao, Classification by multi-semantic meta path and active weight learning in heterogeneous information networks, *Expert Systems with Applications* **123** (2019), 227–236.

[5] M. Gupta, P. Kumar and B. Bhasker, HeteClass: A meta-path based framework for transductive classification of objects in heterogeneous information networks, *Expert Systems with Applications* **68** (2017), 106–122.

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

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

[8] Y. Chen and C. Wang, HINE: heterogeneous information network embedding, in: *International Conference on Database Systems for Advanced Applications*, 2017, pp. 180–195.

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

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

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

[12] V.W. Zheng, S. Cavallari, H. Cai, K.C.C. Chang and E. Cambria, From node embedding to community embedding, *arXiv preprint arXiv:1610.09950*, 2016.

[13] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu and S. Yang, Community preserving network embedding, in: *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] P. Pham, P. Do and C.D.C. Ta, W-PathSim: Novel approach of weighted similarity measure in content-based heterogeneous information networks by applying LDA topic modeling, in: *Asian Conference on Intelligent Information and Database Systems*, Springer, Cham, 2018, pp. 539–549.

[15] P. Pham and P. Do, W-MetaPath2Vec: The topic-driven meta-path-based model for large-scaled content-based heterogeneous information network representation learning, *Expert Systems with Applications* **123** (2019), 328–344.

[16] J. Tang, M. Qu and 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*, 2015, pp. 1165–1174.

[17] D.M. Blei, A.Y. Ng and M.I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research* **3**(Jan) (2003), 993–1022.

[18] X. Liu, H.-M. Cheng and Z.-Y. Zhang, Evaluation of Community Detection Methods, *IEEE Transactions on Knowledge and Data Engineering*, 2019.