Effective Attributed Network Embedding with Information Behavior Extraction

Ganglin Hu (huganglin88@outlook.com)
Southwest University https://orcid.org/0000-0002-9539-7818

Jun Pang
Faculty of Science, Technology and Medicine & Interdisciplinary Centre for Security, Reliability and Trust University of Luxembourg, L-4364 Esch-sur-Alzette, Luxembourg

Xian Mo
College of Computer & Information Science, Southwest University

---

Research Article

Keywords: Attributed networks , Network embedding , Information behavior features , Topic-based community features

Posted Date: April 26th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-369076/v1

Soft Computing manuscript No.
(will be inserted by the editor)

Effective Attributed Network Embedding with Information Behavior Extraction

Ganglin Hu · Jun Pang · Xian Mo

Received: date / Accepted: date

Abstract Network embedding has shown its effectiveness in many tasks such as link prediction, node classification, and community detection. Most attributed network embedding methods consider topological features and attributed features to obtain a node embedding, but ignore its implicit information behavior features, including information inquiry, interaction, and sharing. This can potentially lead to ineffective performance for downstream applications. In this paper, we propose a novel network embedding framework named information behavior extraction (IBE), that incorporates nodes' topological features, attributed features, and information behavior features into a joint embedding framework. To design IBE, we use an existing embedding method (e.g., SDNE, CANE, or CENE) to extract a node's topological features and attributed features into a basic vector. Then, we propose a topic-sensitive network embedding (TNE) model to extract node information behavior features and eventually generate information behavior feature vectors. In our TNE model, we propose an importance score rating algorithm (ISR), which considers both effects of the topic-based community of a node and its interaction with adjacent nodes to capture a node information behavior features. Eventually, we concatenate a node information behavior feature vector with its basic vector to get its ultimate joint embedding vector. Extensive experiments demonstrate that our method achieves significant and consistent improvements, compared to several state-of-the-art embedding methods on link prediction.

Keywords Attributed networks · Network embedding · Information behavior features · Topic-based community features

✉ Jun Pang
E-mail: jun.pang@uni.lu

· Ganglin Hu · Xian Mo
College of Computer & Information Science, Southwest University, 400715 Chongqing, China

Jun Pang
Faculty of Science, Technology and Medicine & Interdisciplinary Centre for Security, Reliability and Trust University of Luxembourg, L-4364 Esch-sur-Alzette, Luxembourg
E-mail: jun.pang@uni.lu

1 Introduction

Network embedding (NE) aiming to map nodes of networks into a low-dimensional vector space, has been proved extremely useful in many applications, such as node classification [15, 17, 22], node clustering [3], link prediction [7]. A number of network embedding models have been proposed to learn low-dimensional vectors for nodes via leveraging their structure and attribute information in the network. For example, spectral clustering is an early method for learning node embeddings, including DGE [18], LE [26], and LLE [20]. Matrix decomposition is another important method for learning node embedding, for example, GraRep [3] and TADW [27]. Spectral clustering and matrix decomposition embedding methods usually have high computational complexity with the increasing network size. In recent years, various network embedding methods have been proposed using random walk-based methods, which are faster and more effective, including DeepWalk [17], Node2Vec [7], LINE [22], and SDNE [25]. More recently, deep learning and attention mechanisms are used to generate network embeddings [23, 24], which extract structure, text, topic, and other heterogeneous feature information more effectively. Nevertheless, these methods are all limited by focusing on the network topological structure and attribute information while ignoring the implicit interactive relationship between node information. In reality, information has interactive behavior in some real-world networks such as social networks and

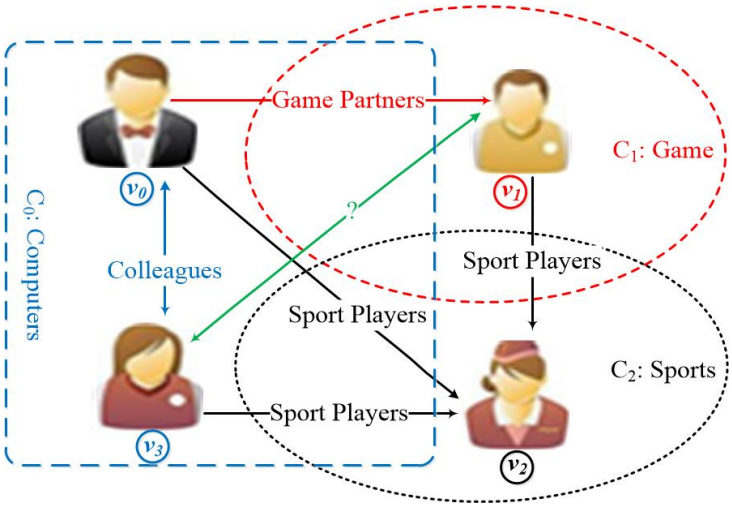


Fig. 1: Node information behavior of multiple topic-based communities (C_0, C_1, and C_2 are topic-based communities, and v_0, v_1, v_2, v_3 are nodes, and in the meantime $(v_0, v_3) \in C_0$, $v_1 \in C_1$, $v_2 \in C_2$). Nodes interact in intra-community, such as nodes v_0 and v_3. Nodes in different communities may interact with each other, such as nodes v_0 and v_1, v_1 and v_2, v_2 and v_3. Meantime, due to the existence of bridge nodes v_0 or v_2, nodes v_1 and v_3 may have a link which is not represented in the current network.

citation networks, where the nodes have information behavior [19], including information inquiry, information access, information interaction, and information sharing. In real-world social networks and citation networks, it is intuitive that all of the nodes naturally prefer to interact with similar nodes. In this way, the exchange and sharing of information between nodes are more efficient, which is also the reason for the formation of various topic network communities. For example, due to different majors, three topic-based communities, C_0, C_1, and C_2, have been formed (as illustrated in Figure 1). In these communities, nodes interact within both intra-community (such as nodes v_0 and v_3) and inter-community (such as nodes v_0 and v_1, v_1 and v_2, v_2 and v_3). This means that one node may communicate and share information on various topics when interacting with neighboring nodes of different communities and build bridges among nodes that are not directly connected, such as nodes v_1 and v_3 may have a link because node v_0 or v_2 acts as a bridge, but this link is not observed. It can be seen that these information behaviors are very important features, and the representation vectors for nodes without information behavior features are incomplete. However, these existing embedding methods are not able to cope with the information behavior of nodes.

To tackle the above-identified problems, we make the following contributions: (1) We demonstrate the importance of integrating structure features, attributed features, and node information behavior features in attribute networks. (2) We propose a joint embedding framework IBE to add the information behavior feature vector to a basic vector generated by one of the existing embedding methods to obtain a final joint embedding vector. Within the framework, we also design an algorithm ISR to generate a topic-sensitive vector for a given topic, and then we get information behavior feature vectors by matrix transposing a topic-sensitive embedding matrix composed of all topic-sensitive vectors. (3) We conduct extensive experiments in real-world information networks. Experimental results prove the effectiveness and efficiency of the proposed ISR algorithm and IBE framework.

The rest of the paper is organized as follows. Section 2 discusses several related works. We provide some definitions and problem formulation in Section 3. Section 4 presents in detail our proposed IBE framework and ISR algorithm. We then show experimental results in Section 5 before concluding the paper in Section 6.

2 Related Work

In the last few years, a large number of NE models have been proposed to learn node network embedding efficiently. These methods can be classified into two categories based on structural information and attributes: 1) SNE methods by considering purely structural information; and 2) ANE methods by considering both structural information and attributes. In this section, we briefly review related work in these two categories.

SNE methods: DeepWalk [17] employs Skip-Gram [13] to learn the representations of nodes in the network. It uses a random selection of nodes and truncated random walk to generate random walk sequences of fixed length. Subsequently, these sequences are transported to the Skip-Gram model to learn the distributed node representations. LINE [22] studies

the problem of embedding very large information networks into low-dimensional vector spaces. Node2vec [7] improves the strategy of random walk and achieves a balance between BFS and DFS. SDNE [25] proposes a semi-supervised deep model, which can learn a highly nonlinear network structure. It combines the advantages of first-order and second-order estimation to represent the global and local structure attributes of the network. Besides, there are many other SNE methods [6], which systematic analysis of various structural graph embedding models, and explain their differences. Nevertheless, these methods fully utilize structural information but do not consider attribute information.

ANE methods: CANE [23] proposed an approach of network embedding considering both node text information of context-free and context-aware. CENE (context-enhanced network embedding) [21] regards text content as a special kind of nodes and leverages both structural and textural information to learn network embeddings. TopicVec [10] proposes to combine the word embedding pattern and document topic model. JMTS [1] proposes a domain-independent topic sentiment model to integrate topic semantic information into embedding. ASNE [11] adopts a deep neural network framework to model the complex interrelations between structural information and attributes. It learns node representations from social network data by leveraging both structural and attribute information. ABRW [9] reconstructs a unified denser network by fusing structural information and attributes for information enhancement. It employs weighted random walks based network embedding method for learning node embedding and addresses the challenges of embedding incomplete attributed networks. There are quite a few survey papers [4, 5, 16, 28], which provide a comprehensive up-to-date review of the state-of-the-art network representation learning techniques. They cover not only early work on preserving network structure but also a new surge of incorporating node content and node labels.

It is challenging to get network embedding considering attributes of local context and topic due to its complexity. Quite a few works have been carried out on this issue. However, none of them consider node information behavior features in attributed networks.

3 Problem Definition

In this section, we present the necessary definitions and formulate the problem of link prediction in attributed networks.

Definition 1 (Networks) A network can be represented graphically: $G = (V, E, \Delta, A)$, where $V = \{v_0, v_1, \ldots, v_{(|V|-1)}\}$ represents the set of nodes, and $|V|$ is the total number of nodes in G. $E \subseteq V \times V$ is the set of edges between the nodes. $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{(\tau-1)}\}$ is a set, where δ represents a topic and it is also a topic-based node label identified the topic of the node, τ is the total number of topics. A is a function which associates each node in the network with a set of attributes, denoted as $A(v)$.

Definition 2 (Adjacent-Node and Node degree) An adjacent-node set of node $v \in V$ is defined as $N_v = \{v' : (v, v') \in E\}$. v^{degree} is the number of nodes in the adjacent-node set of v, called the degree of node v.

Definition 3 (Topic-based community) Each node v has topic-based labels to identify the topics it belongs to. A topic-based community is a node-set that consists of the nodes with same topic-based label. Here, we define a topic-based community as $C_\delta = \{v : v \in V, \delta \in \Delta\}$, and the node number in C_δ is defined as $|C_\delta|$. The topic-based community set is represented as $C^\Delta = \{C_{\delta_0}, C_{\delta_1}, \ldots, C_{\delta_{(\tau-1)}}\}$ ($\bigcup_{j=0}^{(\tau-1)} C_{\delta_j} = V$), where C^Δ is a set of all topic-based communities. Note that we assume that each node has at least one topic-based label, and one node can belong to several topic-based communities.

Definition 4 (Importance Score) Given a topic δ, the importance score x_i ($0 \leq i < |V|$) of node v_i is computed as follows:

$$x_i = \beta * m_i + (1 - \beta) * s_i \tag{1}$$

where $0 \leq \beta \leq 1$ is a hyper-parameter, m_i and s_i are the adjacent score and community score of node v_i, respectively. v_i's adjacent score m_i is defined as the weighted importance score of its adjacent nodes:

$$m_i = \sum_{v_k \in N_i} \frac{x_k}{(v_k^{degree})},$$

where x_k is the importance score of v_k, v_k^{degree} is the degree of node v_k, and N_i is the adjacent-node set of node v_i. Moreover, v_i's community score s_i with respect to the topic δ is defined as:

$$s_i = \begin{cases} \frac{1}{|C_\delta|}, & if\ v_i \in C_\delta \\ 0, & otherwise \end{cases}$$

where C_δ is a topic-based community and $|C_\delta|$ is the number of nodes in C_δ.

The importance score x_i of node v_i reflects the interaction between v_i and its adjacent nodes N_i, as well as the level of correlation between v_i and its topic-based community C_δ ($\delta \in \Delta$).

Definition 5 (Topic-sensitive vector) Given a topic δ, the importance scores of all nodes can be used to form a topic-sensitive vector $\overrightarrow{\gamma^\delta} = (x_0^\delta, x_1^\delta, \ldots, x_{(|V|-1)}^\delta)$ ($\delta \in \Delta, \overrightarrow{\gamma^\delta} \in \mathbb{R}^{|V|}$).

We take each topic δ as the number line δ, then τ topics constitute τ-dimensional topic coordinate system $O_{(\delta_0,\delta_1,\ldots,\delta_{(\tau-1)})}$. We project each element of the topic-sensitive vector $\overrightarrow{\gamma^{\delta}}$ onto the number line δ, so the importance score of each node corresponds to a point of the number line δ. Similarly, we also project each element of the other topic-sensitive vector $\overrightarrow{\gamma^{\delta'}} = (x_0^{\delta'}, x_1^{\delta'}, \ldots, x_{(|V|-1)}^{\delta'})$ on the number line δ', so the importance score $x^{\delta'}$ of each node corresponds to a point of the number line δ'. When we take the importance score of node as the coordinate value of node, the coordinate of node v_i in the coordinate system $O_{(\delta_0,\delta_1,\ldots,\delta_{(\tau-1)})}$ is $(x_i^{\delta_0}, x_i^{\delta_1}, \ldots, x_i^{\delta_{(\tau-1)}})$. So, using the coordinates of nodes, node v_i can be presented as an information behavior feature vector

$$\overrightarrow{z_{v_i}^{I}} = (x_i^{\delta_0}, x_i^{\delta_1}, x_i^{\delta_2}, \ldots, x_i^{\delta_{(\tau-1)}}) \ \ (0 \leq i < |V|).$$

For example, given a topic δ_0, we have $x_0 = 0.10$, $x_1 = 0.15$ and $x_2 = 0.05$ for nodes of v_0, v_1, v_2 on the the number line δ_0, and topic-sensitive vector is $\overrightarrow{\gamma^{\delta_0}} = (x_0^{\delta_0}, x_1^{\delta_0}, x_2^{\delta_0}) = (0.01, 0.02, 0.03)$. Similarly, we have $\overrightarrow{\gamma^{\delta_1}} = (x_0^{\delta_1}, x_1^{\delta_1}, x_2^{\delta_1}) = (0.11, 0.12, 0.13)$ $\overrightarrow{\gamma^{\delta_2}} = (x_0^{\delta_2}, x_1^{\delta_2}, x_2^{\delta_2}) = (0.21, 0.22, 0.23)$ for topics δ_1 and δ_2, respectively. Hence, the coordinate of node v_0 in the coordinate system $O_{\delta_0,\delta_1,\delta_2}$ is $(0.01, 0.11, 0.21)$, which contains the values of information behavior feature vector $\overrightarrow{z_{v_0}^{I}} = (x_0^{\delta_0}, x_0^{\delta_1}, x_0^{\delta_2})$ for node v_0. Similarly, we have $\overrightarrow{z_{v_1}^{I}} = (0.02, 0.12, 0.22)$, $\overrightarrow{z_{v_2}^{I}} = (0.03, 0.13, 0.23)$ for nodes v_1 and v_2, respectively.

The learning process of the topic-sensitive vector is a repetitive iteration process to compute the importance scores of all nodes. The initial importance scores is $x_i = \beta * \frac{1}{|V|} + (1-\beta) * s_i$ (β = 0.85 is a hyper-parameter; $s_i = \frac{1}{|C_\delta|}$ if $v_i \in C_\delta$, otherwise $s_i = 0$.). Illustrated by Equation 1, each iteration is an one-order aggregate operation of adjacent scores and community score and a new value $\frac{1}{|C_\delta|}$ is added to x_i for node v_i. After a number of iterations (i.e., higher-order aggregations), the ratio between x_i of each node v_i will stabilize, but the x_i for each node v_i will continue to grow due to the continuous addition of the v_i's community score s_i. So, after each iteration, we normalize the x_i for every node by

$$\hat{x}_i = \frac{x_i}{\sqrt{\sum_{i=0}^{n-1} (x_i)^2}}, (0 \leq i < |V|).$$

In this way, the x_i obtained from this iteration process will eventually converge.

Attributed network embedding. Given an attributed network $G = (V, E, \Delta, A)$, our goal is to extract the node information behavior features and learn an information behavior feature vector $\overrightarrow{z_v^{I}} \in \mathbb{R}^d$ ($d = K * \tau$, $d \ll |V|$) for each node v. The distance between $\overrightarrow{z_v^{I}}$ and $\overrightarrow{z_{v'}^{I}}$ is the information behavior similarity of two nodes v, v' ($v, v' \in V$). After that, node information behavior feature vector $\overrightarrow{z_v^{I}}$ is added to the node basic vector $\overrightarrow{z_v^{B}} \in \mathbb{R}^{d'}$ ($d' \ll |V|$) generated by one of existing embedding methods to get the ultimate joint embedding vector:

$$\overrightarrow{z_v} = ([\overrightarrow{z_v^{I}} \| \overrightarrow{z_v^{B}}]) \in \mathbb{R}^{d+d'} \tag{2}$$

where $[\cdot\|\cdot]$ denotes concatenating two vectors end to end. Nodes with similar network-structure features, node-attribute features, and information-behavior features are close to each other in the embedding space $\mathbb{R}^{d+d'}$.

4 Our Approach

In this section, we introduce our method of information behavior features extraction. We firstly propose our framework IBE (Section 4.1), which elaborates the components of node joint embedding vectors. Then, we present the model TNE (Section 4.2), which describes the process of generating information behavior feature vectors.

4.1 Information Behavior Extraction Framework (IBE)

As shown in Figure 2, data sources of the framework IBE consist of two parts. One is the network embedding Z^B generated by one of existing embedding methods, and the other is Z^I, where $Z^B \in \mathbb{R}^{|V|\times d'}$ and $Z^I \in \mathbb{R}^{|V|\times d}$ ($d', d \ll |V|$) are embedding matrix consisting of the embedding vectors $\overrightarrow{z_v^{B}}$ and $\overrightarrow{z_v^{I}}$ of nodes V, respectively. We linearly concatenate the embedding matrix Z^B and Z^I to generate a joint embedding matrix Z, which can be used for link prediction, recommendation, and other tasks in attributed networks.

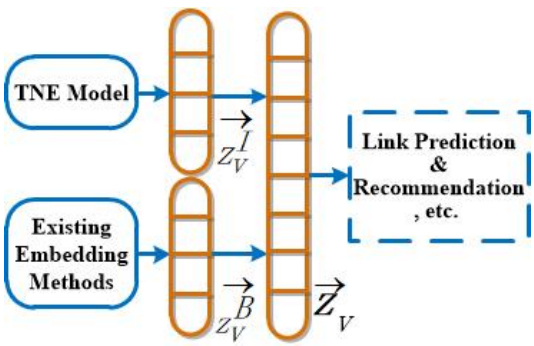


Fig. 2: Our information behavior extraction framework (IBE)

4.2 Topic-Sensitive Network Embedding Model (TNE)

In this section, we present the process of extracting node information behavior features. As shown in Figure 3, the TNE model consists of two parts: an ISR algorithm (Section 4.2.1) and a topic-sensitive embedding matrix (Γ) transposing (Section 4.2.2).

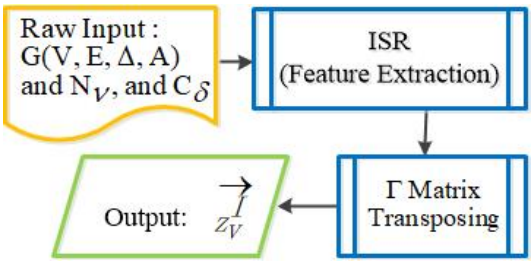


Fig. 3: Overview of TNE model.

4.2.1 Importance Score Rating Algorithm (ISR).

The ISR algorithm is used to get the importance scores of all nodes (illustrated by Equation 1 and Definition 4) and generates a topic-sensitive vector under a given topic. We firstly input raw data including a node set V, an adjacent-node set N_v, and a topic-based community C_δ to ISR (see Algorithm 1) and then simulate the iteration process of node information behavior under the given topic δ. When the importance scores of all nodes stabilize, the iteration is terminated. We propose $loss$ as a metric for iteration termination, which is calculated as follows.

$$loss = \sum_{i=0}^{|V|-1} (|x_i - x'_i|) \tag{3}$$

where x_i is the current iterative importance score for the node v_i, and x'_i is the importance score of the previous iteration for the node v_i.

After the iteration is done, we can obtain a $|V|$-dimensional topic-sensitive vector $\overrightarrow{\gamma^\delta} = (x_0, x_1, \ldots, x_{(|V|-1)})$ (illustrated by Definition 5), consisting of importance scores of all nodes under a given topic $\delta \in \Delta$.

Given a topic δ, the computing steps of topic-sensitive vector $\overrightarrow{\gamma^\delta}$ are given as follows (see Algorithm 1):

(1) For a node v_i, the line 6-8 is used to compute its adjacent scores m_i of all of neighbors N_i and the line 9-13 are used to compute its community score s_i. Using m_i and s_i, the importance scores x_i of node v_i is finally calculated out in the first statement of line 14.

(2) The second layer loop (line 5, line 14-15) is used to assemble the importance scores of all nodes to generate a list $\gamma^\delta = [x_0, x_1, ..., x_{(|V|-1)}]$ in program which is the topic-sensitive vector $\overrightarrow{\gamma^\delta} = (x_0, x_1, \ldots, x_{(|V|-1)})$ ($\overrightarrow{\gamma^\delta} \in \mathbb{R}^{|V|}$).

(3) The third statement of line 14 and line 16 are used to calculate the Euclidean norm $norm$. Line 17-19 is used to normalize the importance scores of all nodes by the Euclidean norm (the first statement of line 18) and calculate the sum of $loss$ for all nodes (the second statement of line 18). The third statement of line 18 is used to update the value $\tilde{\gamma}^\delta[v_i]$ of node v_i, which is the importance score and will be used in the next iteration.

(4) The first layer loop (line 3, line 20) is used to control the iterations.

Algorithm 1: ISR algorithm

Input : δ: a topic, $\delta \in \Delta$;
V: a node set of network G;
$|V|$: node number of network G;
N_i: a adjacent node set of node v_i;
v_i^{degree}: degree of node $v_i \in V$;
C_δ: a topic-based community, $C_\delta \in C^\Delta$;
$|C_\delta|$: node number of topic-based community δ;
β = 0.85: a hyper-parameter, imposing the ratio between m_i and s_i;
Output: γ^δ;

1 $\gamma^\delta = [x_0, x_1, ..., x_{(|V|-1)}] = [\frac{1}{|V|}, \frac{1}{|V|}, ..., \frac{1}{|V|}] = \tilde{\gamma}^\delta$, initializing every element of list γ^δ and $\tilde{\gamma}^\delta$ with $\frac{1}{|V|}$ in a given topic $\delta \in \Delta$, where $\tilde{\gamma}^\delta$ is used to temporarily store a topic sensitive vector;
2 loss = 30;
3 **while** $loss > \frac{1}{|V|}$ **do**
4 loss = 0; norm = 0;
5 **for** *each* $v_i \in V$ **do**
6 **for** *each* $v_k \in N_i$ **do**
7 $x_k = \gamma^\delta[v_k]$; $m_i = m_i + \frac{x_k}{(v_k^{degree})}$;
8 **end**
9 **if** $v_i \in C_\delta$ **then**
10 $s_i = \frac{1}{|C_\delta|}$;
11 **else**
12 $s_i = 0$;
13 **end**
14 $x_i = \beta * m_i + (1 - \beta) * s_i$; $\gamma^\delta[v_i] = x_i$; norm = norm + x_i^2;
15 **end**
16 norm = $\sqrt{norm}$;
17 **for** *each* $v_i \in V$ **do**
18 $\gamma^\delta[v_i] = \frac{\gamma^\delta[v_i]}{norm}$; loss = loss + ($|\gamma^\delta[v_i]$ - $\tilde{\gamma}^\delta[v_i]|$); $\tilde{\gamma}^\delta[v_i]) = \gamma^\delta[v_i]$;
19 **end**
20 **end**
21 return $\gamma^\delta = [x_0, x_1, ..., x_{(|V|-1)}]$;

4.2.2 Topic-Sensitive Embedding Matrix (Γ) Transposing.

For $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{(\tau-1)}\}$, the Γ matrix transposing calls the Algorithm 1 to get every topic-sensitive vectors $\overrightarrow{\gamma^{\delta}}$ in each topic $\delta \in \Delta$. After all τ topic-sensitive vectors are obtained, we combine the τ topic-sensitive vectors to form a topic-sensitive embedding matrix $\Gamma = (\overrightarrow{\gamma^{\delta_0}}, \overrightarrow{\gamma^{\delta_1}}, \ldots, \overrightarrow{\gamma^{\delta_{(\tau-1)}}})^{\mathrm{T}}$. Ultimately, Z^I is obtained by the Γ matrix transposing, as illustrated by Equation 4.

$$\begin{aligned} Z^I &= (\overrightarrow{z^I_{v_0}}, \overrightarrow{z^I_{v_1}}, \ldots, \overrightarrow{z^I_{v_{(|V|-1)}}})^{\mathrm{T}} \\ &= \Gamma^{\mathrm{T}} = (\overrightarrow{\gamma^{\delta_0}}, \overrightarrow{\gamma^{\delta_1}}, \ldots, \overrightarrow{\gamma^{\delta_{(\tau-1)}}}) \end{aligned} \tag{4}$$

Each row of Z^I is an information behavior feature vector $\overrightarrow{z^I_v}$ for node v and the dimension of $\overrightarrow{z^I_v}$ is $d = \tau$.

4.3 Generating Node Joint Embedding Vectors based on IBE

Z^B is a basic embedding matrix trained by one of the existing embedding methods. Each row of the Z^B is a basic vector $\overrightarrow{z^B_v}$ for a node v generated by one of the existing embedding methods.

Before getting Z by Equation 7 according to the framework IBE, we firstly enlarge Z^I or Z^B by λ (Equation 5) so that the element values of $\lambda * Z^I$ and Z^B or Z^I and $\frac{Z^B}{\lambda}$ are of the same order of magnitude, and the λ is calculated as follows.

$$\lambda = \frac{\overline{|b|}}{\overline{x}} = \left(\frac{\sum_{i=0}^{|V|-1}\sum_{j=0}^{d-1}|b_{ij}|}{|V| * d}\right) \div \left(\frac{\sum_{i=0}^{|V|-1}\sum_{j=0}^{\tau-1}x_{ij}}{|V| * \tau}\right) \tag{5}$$

where $\overline{|b|}$ is the average of all elements in Z^B, and $\overline{x}$ is the average of all elements in Z^I. And then we enlarge the element values of Z^I or Z^B who has the larger AUC [8] value by weight coefficient α (Equation 6) again.

$$\alpha = [auc(Z^I) \div auc(Z^B)]^{\psi} \tag{6}$$

where $auc()$ is a function used to calculate the value of AUC, ψ is an amplification factor of the ratio $\frac{auc(Z^I)}{auc(Z^B)}$.

Especially, we should not use the method of reducing the element values of Z^I or Z^B to make their element values of the same order of magnitude, because it may result in invalid results due to the element values are too small. So, according to the values of coefficients α and λ, we divide the methods of linearly concatenating Z^I and Z^B into four cases as follows.

$$\begin{aligned} Z &= (\overrightarrow{z_{v_0}}, \overrightarrow{z_{v_1}}, \ldots, \overrightarrow{z_{v_{(|V|-1)}}})^{\mathrm{T}} \\ &= \begin{cases} [(\alpha * \lambda * Z^I) \| Z^B], & \text{if } \lambda \geq 1 \; and \; \alpha \geq 1 \\ [(\lambda * Z^I) \| \frac{Z^B}{\alpha}], & \text{if } \lambda \geq 1 \; and \; \alpha < 1 \\ [(\alpha * Z^I) \| \frac{Z^B}{\lambda}], & \text{if } \lambda < 1 \; and \; \alpha \geq 1 \\ [(Z^I) \| \frac{Z^B}{\alpha * \lambda}], & \text{if } \lambda < 1 \; and \; \alpha < 1 \end{cases} \end{aligned} \tag{7}$$

where the operator $[\cdot\|\cdot]$ denotes concatenation, α is an enlarging coefficient to make the joint embedding matrix Z more similar to Z^I or Z^B who has the higher AUC value, and λ (Equation 5) denotes the enlargement factor who try to be adjusted to make the element values of $\lambda * Z^I$ and Z^B or Z^I and $\frac{Z^B}{\lambda}$ in the same order of magnitude.

For the case of $[(\alpha * \lambda * Z^I) \| Z^B]$ ($\lambda \geq 1$ and $\alpha \geq 1$) in Equation 7, Z is displayed in matrix form as follows:

$$Z = \begin{bmatrix} \alpha * \lambda * x_{00} & \cdots & \alpha * \lambda * x_{0(\tau-1)} & b_{00} & \cdots & b_{0(d-1)} \\ \alpha * \lambda * x_{10} & \cdots & \alpha * \lambda * x_{1(\tau-1)} & b_{10} & \cdots & b_{1(d-1)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha * \lambda * x_{(|V|-1)0} & \cdots & \alpha * \lambda * x_{(|V|-1)(\tau-1)} & b_{(|V|-1)0} & \cdots & b_{(|V|-1)(d-1)} \end{bmatrix},$$

where each row of Z is the final joint embedding vector $\overrightarrow{z_v}$ for node v based on the framework of IBE and x_{ij} ($0 \leq i < |V|, 0 \leq j < \tau$) is the element of $\overrightarrow{z^I_{v_i}}$, b_{ij} ($0 \leq j < d$) is the element of $\overrightarrow{z^B_{v_i}}$. The other three cases of Equation 7 have similar matrix representations.

5 Experiments

In this section, we describe our datasets, baseline models and present the experimental results to demonstrate the performance of the IBE framework in link prediction tasks. The source code and datasets can be obtained from https://github.com/swurise/IBE.

5.1 Datasets

In Table 1, we consider the following real-world network datasets. Zhihu is the largest online Q&A website in China. Users follow each other and answer questions on this site. We randomly crawl 10,000 active users from Zhihu and take the descriptions of their concerned topics as text information [23]. Cora is a research paper classification citation network constructed by McCallum et al [12]. After filtering out papers without text information, 2,277 machine learning papers are divided into seven categories and 36 subcategories

Table 1: Statistics of the real-world information networks.

	Social Network	Language Network		Citation Network	
Dataset / Name	BlogCatalog	Zhihu	Wiki	Cora	Citeseer
Nodes	10,312	10,000	2,408	2,277	3,312
Edges	667,966	43,894	17,981	5,214	4,732
Attributes	-	10,000	-	2,277	-
Number of topics (τ)	39	-	17	7	6

in this network. Citeseer is divided into six communities: Agents, AI, DB, IR, ML, and HCI and 4,732 edges between them. Similar to Cora, it records the citing and cited information between papers. WiKi contains 2,408 documents from 17 classes and 17,981 edges between them. BlogCatalog[1] is asocial blog directory. The dataset contains 39 topic labels, 10312 users, and 667,966 links.

5.2 Baselines

To validate the performance of our approach, we employ several state-of-the-art network embedding methods as baselines to compare with our IBE framework. A number of existing embedding methods are introduced as follows.

- CANE [23] learns context-aware embeddings with mutual attention mechanism for nodes, and the semantic relationship features are extracted between nodes. It jointly leverages network structure and textural information by regarding text content as a special kind of node.
- DeepWalk [17] transforms a graph structure into a sample set of linear sequences consisting of nodes using uniform sampling. These linear sequences are transported to the Skip-Gram model to learn the distributed node embeddings.
- HOPE [14] is a graph embedding algorithm, which is scalable to preserve high-order proximities of large-scale graphs and capable of capturing the asymmetric transitivity.
- LAP [2] is a geometrically motivated algorithm for constructing a representation for data sampled from a low dimensional manifold embedded in a higher-dimensional space.
- LINE [22] learns node embeddings in large-scale networks using first-order and second-order proximity between the nodes.
- Node2vec [7] has the same idea as DeepWalk using random walk sampling to get the combinational sequences of node context, and then the network embeddings of nodes are obtained by using the method of word2vec.

[1] http://networkrepository.com/soc-BlogCatalog.php

5.3 Evaluation Metrics and Parameter Settings

We randomly divide all edges into two sets, a training set, and a testing set, and take a standard evaluation metric AUC scores (area under the ROC curve) [8] as evaluation metrics to measure the link prediction performance. AUC represents the probability that nodes in a random unobserved link are more similar than those in a random nonexistent link. Because the number of topics is different in each dataset, we use τ to denote the maximum number of topics for each dataset. In concatenating weight coefficient $\alpha = [auc(Z^I) \div auc(Z^B)]^{\psi}$, we set the factor ψ equal to 4 except that ψ has a specified value.

5.4 Experimental Results

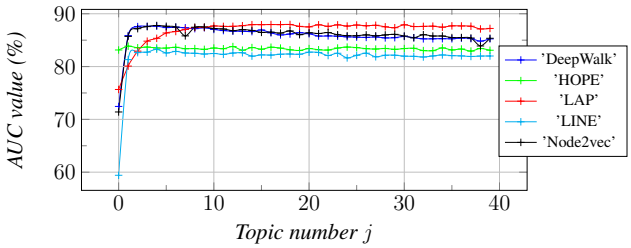
For experiments, we evaluate the effectiveness and efficiency of our IBE on five networks for the link prediction task. For each dataset, we compare the AUC data of basic embedding matrix Z^B generating by one of the existing embedding methods, the information behavior feature vectors Z^I, and their joint embedding vectors Z generating by the framework IBE. We employ six state-of-the-art embedding methods as baselines, including Node2vec, DeepWalk, LINE, LAP, CANE, HOPE, for comparisons with their extending frameworks IBE in the following experiments. Table 2 compares AUCs over five datasets. By concatenating Z^B and Z^I linearly, the joint embedding vectors Z achieves the best performance. Especially on the BlogCatalog and Zhihu datasets, the AUC values of the joint embedding vectors Z are higher, compared with their baselines, than 14.6% and 27.6% respectively on average. One of the reasons may be that the average AUC of information behavior feature vectors Z^I are 81.7 and 82.4, respectively, which are more than 11% higher compared with the other three datasets on average. The other reason is that the maximum topic number τ of BlogCatalog and Zhihu are larger than those of Cora and Citeseer. In Tabl 2, we can also see that most of the AUC values of the joint embedding vectors Z for datasets Wiki, Cora exceed 90%. The reason is that their AUC values of baselines are relatively high, and most of them are more than 85%. On the Citeseer dataset of Table 2, we also can see that the improvement of the AUC values of the joint embedding vectors Z is less, compared with their baselines. The result can

Table 2: The AUC values of Z^I, $Z^B(baseline)$ and $Z(baseline)$ in different datasets where '(baseline)' in $Z^B(baseline)$, $Z(baseline)$ is to distinguish all kind of network embeddings Z^B and their extension embeddings Z, and the ψ equals 4 except that ψ has a specified value.

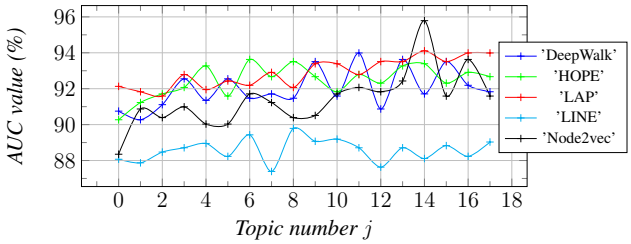
	Social Network	Language Network		Citation Network	
Dataset / Embedding	BlogCatalog ($\tau = 39$)	Zhihu ($\tau = 13$)	Wiki ($\tau = 17$)	Cora ($\tau = 7$)	Citeseer ($\tau = 6$)
Z^I	81.5	81.5	69.1	69.9	54.5
$Z^B(CANE)$	-	71.2	-	94.5	-
$Z(CANE)$	-	**85.6**	-	**94.7**	-
$Z^B(DeepWalk)$	72.1	58.2	90.6	89.5	84.5
$Z(DeepWalk)$	**85.0**	**84.9**	**93.5**	**89.5** (ψ=1)	**85.7**
$Z^B(HOPE)$	83.3	64.3	92.6	84.6	70.2
$Z(HOPE)$	**83.0**	**82.0** (ψ=6)	**92.7**	**86.7**	**71.2**
$Z^B(LAP)$	75.2	74.2	92.2	87.9	80.3
$Z(LAP)$	**87.3**	**86.0**	**93.2**	**89.1**	**81.5**
$Z^B(LINE)$	59.1	51.6	87.8	78.3	69.8
$Z(LINE)$	**82.1**	**82.8**	**88.1** (ψ=7)	**80.7**	**73.8**
$Z^B(Node2vec)$	71.0	56.1	89.0	86.7	83.2
$Z(Node2vec)$	**85.2**	**84.5** (ψ=6)	**93.1**	**88.1**	**84.5** (ψ=2)

be explained by the fact that the AUC values of information behavior feature vectors Z^I are all low, about 55% and the direct reason for the low AUC values is that the number of topics is too small. Due to the number of topics is small, the topic subdivision degree of nodes is low, and the classification of node labels will not be too detailed.

In general, in Table 2, it can be seen that the AUC values of concatenating embedding vectors Z are higher than that of Z^B and Z^I, which indicates that the concatenating method can properly integrate the features of all parties.



(a) τ = 39, BlogCatalog dataset.



(b) τ = 17, Wiki dataset.

Fig. 4: ψ= 5. With the increase of τ, the AUC values are calculuated for different datasets. j is a topic number, if j=0 then Z=Z^B else Z=[$Z^B \| Z^I$].

5.5 Parameter sensitivity analysis

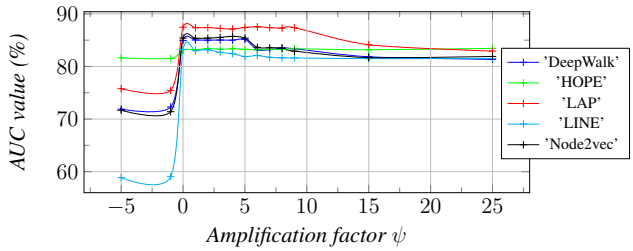
We further performed parameter sensitivity analysis in this section, and the results are summarized in Figure 4 and Figure 5. Due to space limitations, we only take the dataset of Wiki, Zhihu as examples to estimate the topic number j ($0 \leq j \leq \tau$) and the amplification factor ψ of vector concatenation can affect the link prediction results.

The topic number j ($0 \leq j \leq \tau$)**:** In Figure 4, we illustrate the relationship between the number of topics and link prediction. When j = 0, Z^I does not exist, Z degenerates to Z^B. As shown in Figure 4, we can see that as j increases from 1 to τ, Z^B linearly combines with more topic-based feature dimensionality from Z^I, and the AUC values keep changing. When the AUC values of Z^B are below 82% in Figure 4(a), AUC values increase sharply with an increase of j. When the AUC values of Z^B are higher than a certain critical value, the AUC values increase more slowly or even stop growing with an increase of j.

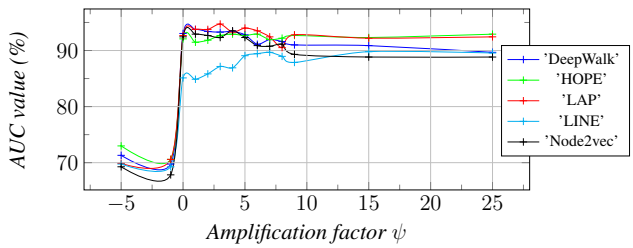
So, we can see that when the number of topics is large, each node can be classified in detail by the topic classification labels, which helps to improve the AUC values using a small number of topics. These also show that the AUC values of the concatenating embedding vectors Zs will be higher than that of all parties for concatenating, that is Z^Is and Z^Bs, but it will not increase indefinitely.

The amplification factor ψ of vector concatenation: ψ is an amplification factor for α (Equation 6) which is a weight coefficient for enlarging the element values of Z^I or Z^B who has the larger AUC (Equation 7). From Figure 5,

we can see that the AUC value, when ψ is less than 0, is less than that when ψ is greater than 0. The reason is that the weight coefficient α, when ψ is less than 0, enlarges the Z^I or the Z^B who has the smaller AUC value. As a result, the joint embedding Z is more similar to that with a lower AUC value. When ψ is between 1 and 5, the prediction result is the best. However, when the value of ψ increases gradually, the AUC values decrease slightly and tend to the $Z^I s$ or the $Z^B s$ who have the larger AUC values.



(a) BlogCatalog dataset.



(b) Wiki dataset.

Fig. 5: The topic number is τ, and ψ is an amplification factor. With the increase of ψ, the AUC values are calculated for different datasets.

6 Conclusion and Future Work

This paper has presented an effective network embedding framework IBE, which can easily incorporate topology features, attributed features, and features of topic-based information behavior into network embedding. In IBE, we linearly combinate Z^I and Z^B to generate node joint embedding matrix Z. To get the Z^I, we have proposed the TNE model to extract the node information behavior features. The model contains an ISR algorithm to generate the topic-sensitive embedding matrix (Γ) and a Γ matrix transposing algorithm. The Γ matrix transposing algorithm transposes Γ matrix into the information behavior feature matrix Z^I for nodes eventually. Experimental results in various real-world networks have shown the efficiency and effectiveness of joint embedding vectors in link prediction. In the future, we plan to investigate other methods of extracting features that may better integrate with the TNE model. Moreover, we will further investigate how the TNE model works in heterogeneous information networks.

Authors contribution All authors (Ganglin Hu, Jun Pang and Xian Mo) contributed to the conception and design and to the acquisition, analysis and interpretation of the data; drafted and revised the article and approved the version to be published; agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Acknowledgements This paper is funded in part by the National Natural Science Foundation of China (62032019, 61732019, 61672435), and the Capacity Development Grant of Southwest University (SWU116007).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not containany studies with human participants or animalsperformed by any of the authors.

References

1. Alam, M.H., Ryu, W.J., Lee, S.: Joint multi-grain topic sentiment: modeling semantic aspects for online reviews. Information Sciences **339**, 206–223 (2016)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems **14**(6), 585–591 (2001)
3. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. CIKM '15: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management p. 891–900 (2015)
4. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. IEEE Trans. Knowl. Data Eng. **31**(5), 833–852 (2019). DOI 10.1109/TKDE.2018.2849727. URL https://doi.org/10.1109/TKDE.2018.2849727
5. Daokun, Z., Jie, Y., Xingquan, Z., Chengqi, Z.: Network representation learning: A survey. IEEE Transactions on Big Data **PP**(99), 1–1 (2017)
6. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowledge Based Systems **151**(JUL.1), 78–94 (2017)
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks **2016**, 855–864 (2016)
8. Hanley, J.A., Mcneil, B.J.: The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology **143**(1), 29–36 (1982)

9. Hou, C., He, S., Tang, K.: Attributed network embedding for incomplete attributed networks (2018)
10. Li, S., Chua, T.S., Zhu, J., Miao, C.: Generative topic embedding: a continuous representation of documents. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 666–675 (2016)
11. Liao, L., He, X., Zhang, H., Chua, T.: Attributed social network embedding. IEEE Transactions on Knowledge and Data Engineering **30**(12), 2257–2270 (2018)
12. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval Journal **3**, 127–163 (2000). Www.research.whizbang.com/data
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
14. Ou, M., Peng, C., Jian, P., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. Proceedings of the 22nd ACM SIGKDD International Conference, 2016 p. 1105–1114 (2016)
15. Pang, J., Zhang, Y.: DeepCity: A feature learning framework for mining location check-ins. In: Proceedings of 11th International AAAI Conference on Web and Social Media, pp. 652–655 (2017)
16. Peng, C., Xiao, W., Jian, P., Wenwu, Z.: A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering **PP**(99), 1–1 (2017)
17. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710. ACM (2014)
18. Perrault-Joncas, D.C., Meilă, M.: Directed graph embedding: an algorithm based on continuous limits of laplacian-type operators. Advances in Neural Information Processing Systems pp. 990–998 (2011)
19. Pettigrew, K.E., Fidel, R., Bruce, H.: Conceptual frameworks in information behavior. Annual Review of Information ence & Technology **35** (2001)
20. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
21. Sun, X., Guo, J., Ding, X., Liu, T.: A general framework for content-enhanced network representation learning (2016)
22. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
23. Tu, C., Liu, H., Liu, Z., Sun, M.: Cane: Context-aware network embedding for relation modeling. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1722–1731 (2017)
24. Velikovi, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2017)
25. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1225–1234. ACM (2016)
26. Wang, J.: Laplacian eigenmaps. Springer Berlin Heidelberg (2012)
27. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: IJCAI, vol. 2015, pp. 2111–2117 (2015)
28. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Network representation learning: A survey. IEEE Trans. Big Data **6**(1), 3–28 (2020). DOI 10.1109/TBDATA.2018.2850013. URL https://doi.org/10.1109/TBDATA.2018.2850013

Figures

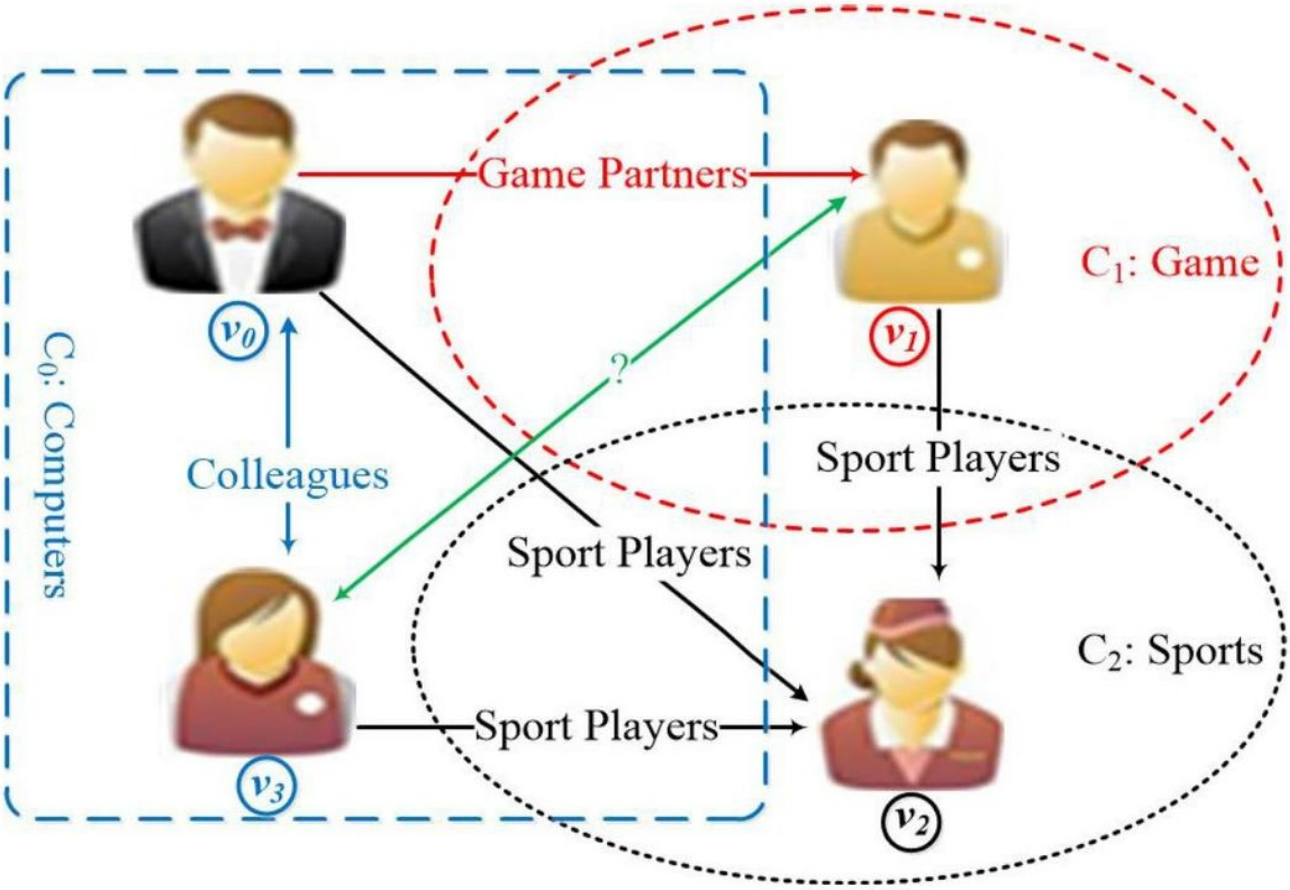


Figure 1

Please see the Manuscript PDF file for the complete figure caption

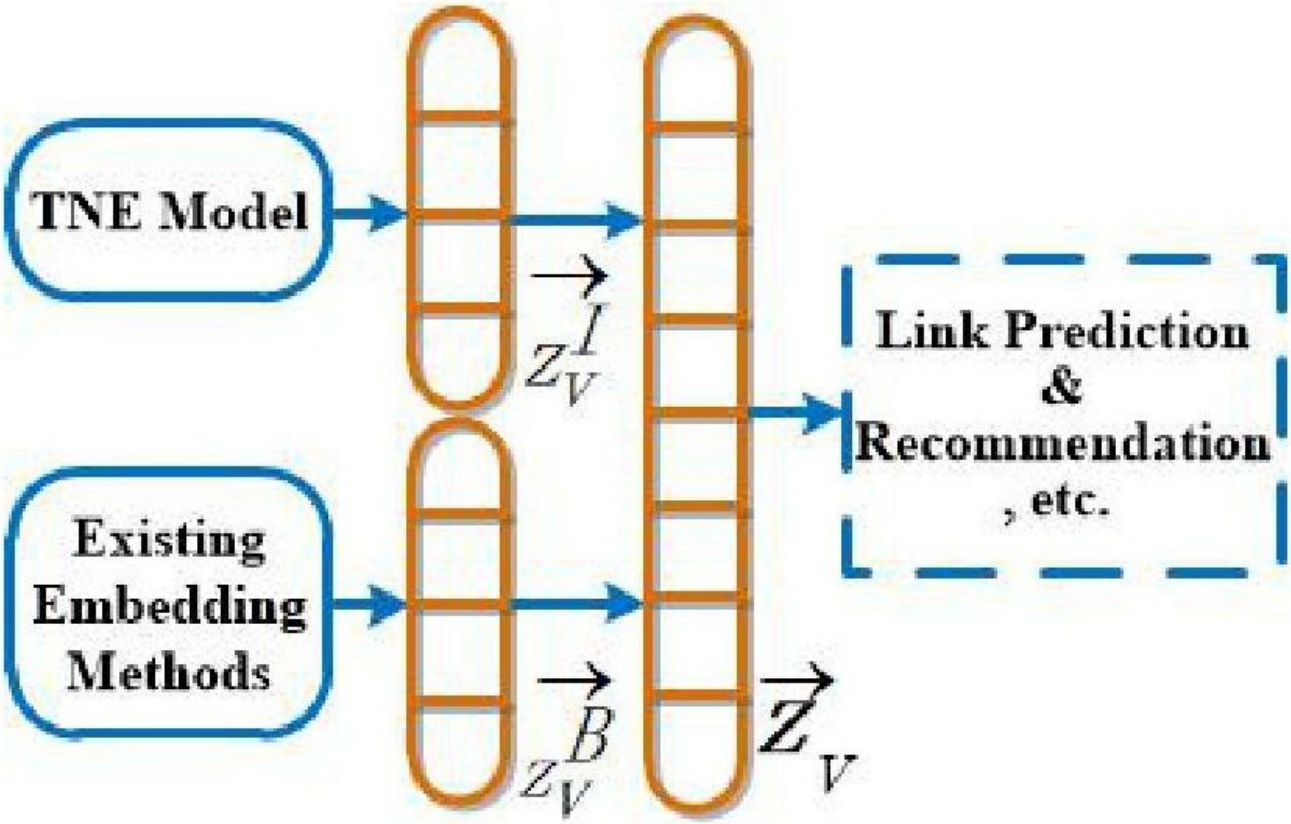


Figure 2

Please see the Manuscript PDF file for the complete figure caption

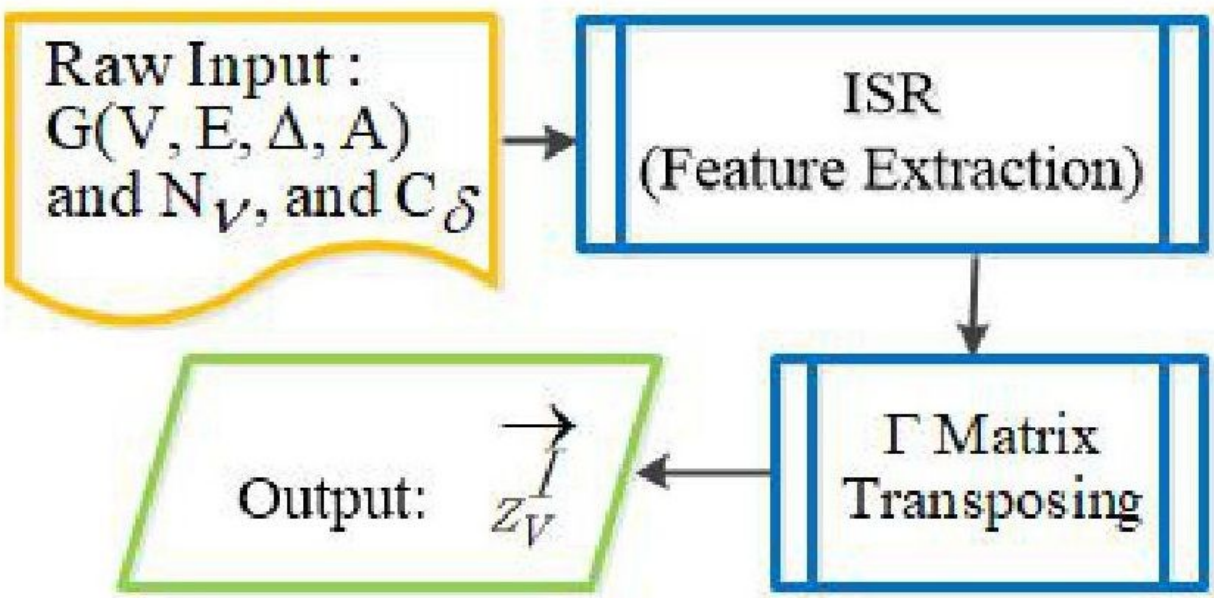


Figure 3

Please see the Manuscript PDF file for the complete figure caption

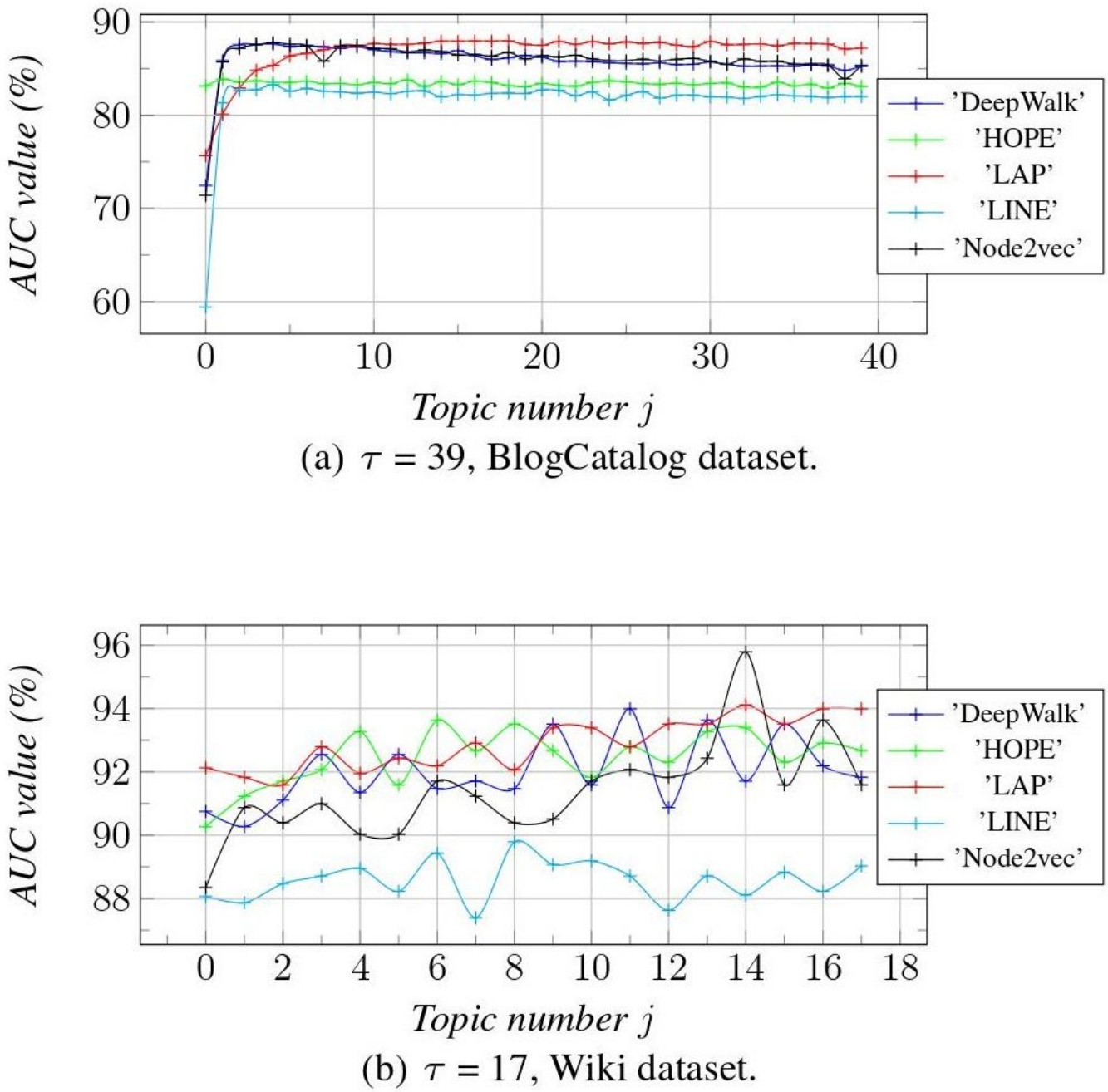
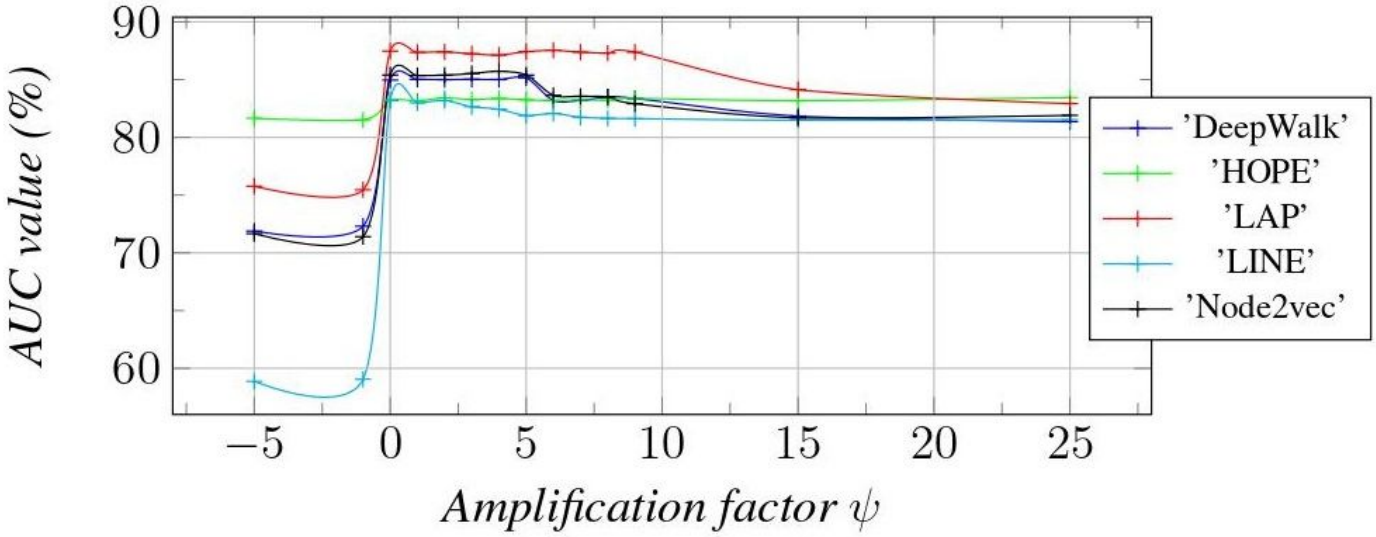
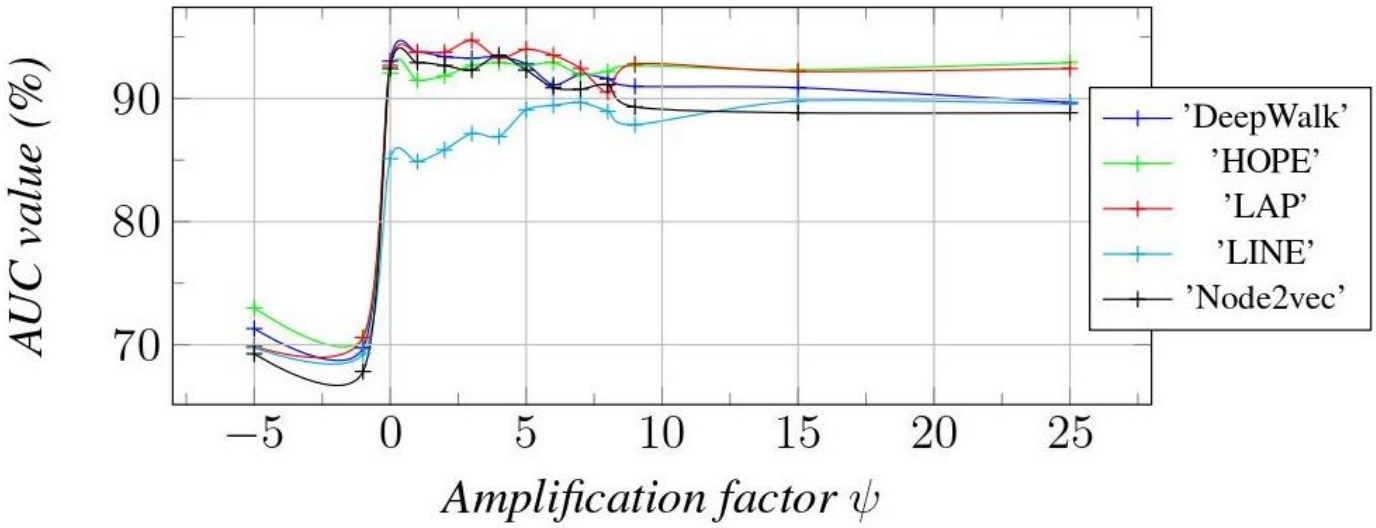


Figure 4

Please see the Manuscript PDF file for the complete figure caption



(a) BlogCatalog dataset.



(b) Wiki dataset.

Figure 5

Please see the Manuscript PDF file for the complete figure caption