Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-019-00659-7

Topical network embedding

Min Shi[1] · Yufei Tang[1] · Xingquan Zhu[1] · Jianxun Liu[2] · Haibo He[3]

Received: 26 November 2018 / Accepted: 18 October 2019

Abstract
Networked data involve complex information from multifaceted channels, including topology structures, node content, and/or node labels etc., where structure and content are often correlated but are not always consistent. A typical scenario is the citation relationships in scholarly publications where a paper is cited by others not because they have the same content, but because they share one or multiple subject matters. To date, while many network embedding methods exist to take the node content into consideration, they all consider node content as simple flat word/attribute set and nodes sharing connections are assumed to have dependency with respect to all words or attributes. In this paper, we argue that considering topic-level semantic interactions between nodes is crucial to learn discriminative node embedding vectors. In order to model pairwise topic relevance between linked text nodes, we propose topical network embedding, where interactions between nodes are built on the shared latent topics. Accordingly, we propose a unified optimization framework to simultaneously learn topic and node representations from the network text contents and structures, respectively. Meanwhile, the structure modeling takes the learned topic representations as conditional context under the principle that two nodes can infer each other contingent on the shared latent topics. Experiments on three real-world datasets demonstrate that our approach can learn significantly better network representations, i.e., 4.1% improvement over the state-of-the-art methods in terms of Micro-F1 on Cora dataset. (The source code of the proposed method is available through the github link: https://github.com/codeshareabc/TopicalNE.)

Responsible editor: Po-ling Loh, Evimaria Terzi, Antti Ukkonen, Karsten Borgwardt.

✉ Yufei Tang
tangy@fau.edu

1 Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, USA

2 School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China

3 Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, USA

Published online: 24 October 2019

Keywords Network embedding · Network representation · Topic model · Semantic mining

1 Introduction

Networked data commonly exist in all aspects of social, economic, industrial, and personal life, such as citation networks (Le and Lauw 2014), social networks (Oro et al. 2018), and invocation networks of web application programming interfaces (Dojchinovski and Vitvar 2018). Analyzing these networked data can help gain insights from many perspectives (Verma and Bharadwaj 2017; Wang et al. 2018), such as social behaviors, community structures, and information diffusion patterns. Recently, network embedding, also known as network representation learning (Zhang et al. 2018), has been proposed to represent each node as a low-dimensional vector by preserving rich network structures and side information (Huang et al. 2017), allowing network analytic tasks such as node classification and link prediction benefit from the learned continuous feature vectors.

To date, intensive studies have been carried out on learning meaningful and distinguishing network representations. Early works mainly focus on structure-based methods, where nodes with similar topological structures are mapped to be close in the latent space. Representative methods include DeepWalk (Perozzi et al. 2014), LINE (Tang et al. 2015) and Node2vec (Grover and Leskovec 2016) that consider either the direct neighborhood relationships or the high-order node proximity in a network. In real world, networks associated with substantial text content are ubiquitous. For examples, Wikipedia articles connect each other to form a hyperlink network and research papers quote each other to form a citation network. Numerous research studies have shown that preserving content information can significantly improve the embedding performance (Pan et al. 2016; Yang et al. 2015) because nodes sharing similar content often have a higher chance of sharing a linkage or a connection. However, when leveraging node content for embedding learning, existing methods typically take node content as a flat word set or an attribute set and directly incorporate node content set for representation learning. Such designs have been severely challenged by the following realities.

- **Rich node content** Many networks have rich node content, such as user profiles or research papers, where each node will have a high dimensional content information. This results in significant difficulty in measuring content similarity between nodes.
- **Sparse connections** While each node often has rich content, its connections to others are often sparse. On one hand, most nodes in the network only have a handful of connections. On the other hand, for any two connected nodes, their connection is usually summarized as a single edge (possibly with some weight values). As a result, it is difficult to attribute which aspect of the node content triggers the node interactions.
- **Context representation** In a network setting, each node and its connections form a small neighborhood which is triggered by certain context. To capture and model

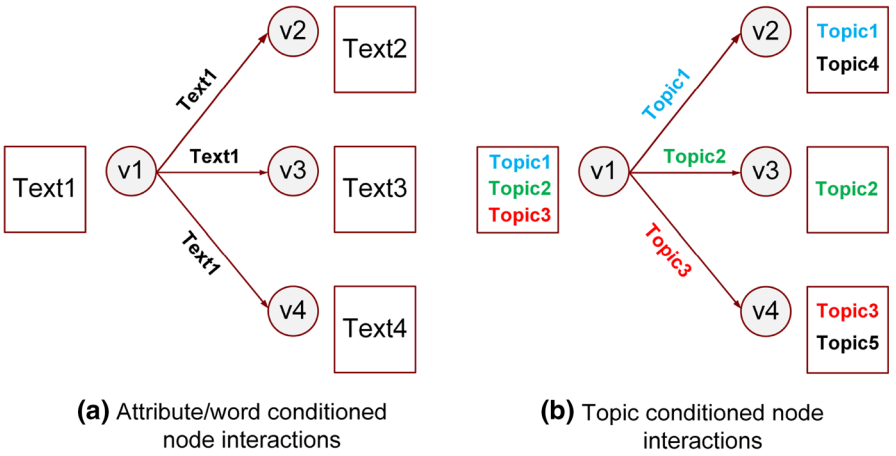

Fig. 1 **a** Traditional content-assisted network embedding, where the text in each node are treated as a flat word set and nodes with a link are assumed to have dependency w.r.t. all their node contents. Such a paradigm results in ambiguity or distortion of the node relationships, because a node may exhibit different aspects when interacting with different neighborhood nodes. **b** Presents our proposed approach, where each node contains multiple topics derived from the textual content, and links between nodes are attributed to the corresponding shared semantic topics

context information, existing methods often rely on random walks to explore node and its neighbors as a whole. This does not allow explicit representation of the context of each node, and therefore cannot explain why node pair (A,B) is connected and what are the contextual difference between two connections (A,B) versus (A,C).

In summary, network node contents are often rich and informative, but knowledge-fragmented. Simple text information preserving, like most existing methods do, would cause the node representation to be semantically ambiguous and less distinguishing. Indeed, studies (Le and Lauw 2014; Shi et al. 2018b) have shown that text or document can be represented by a collection of latent topics, representing different knowledge aspects of the content and further explaining how documents are related/similar for one or several shared latent topics (Le and Lauw 2014). Analogously, interactions between nodes in networks might be resulted by different topics included in the node contents. In other words, interactions between neighborhood nodes can be attributed to their shared latent topics.

The above observations motivated our research to model semantic relevance between linked nodes for topical network embedding (TNE). The main difference between TNE and traditional content-assisted embedding is summarized in Fig. 1, where each node is described by a collection of latent topics instead of treating text in the node as a flat word set. Therefore, link relationships between nodes are built on shared topics that are fine-grained in interpreting and measuring node affinities.

The main contribution of the paper, compared to existing work in the filed, is threefold:

1. We propose to use topic model to allow node content being summarized as multi-aspect semantics, and further model node connections though their semantic interactions. As a result, our method advances the existing research which aligns node connections at word/attribute level to the alignment at the node semantic level.
2. We propose a deep learning model to simultaneously learn node and topic representations, where the mutually related structure and content can enhance each other in a unified optimization framework.
3. Our model allows explicit context characterization for individual node. As a result, it can not only explain why several nodes are connected to form a neighborhood, but also explain the difference between two connections in a network.

Our experiments and validations show that the topic-aware node relation modeling can achieve significant performance gain compared to state-of-the-art baselines. In addition, our method delivers a transparent and interpretable way to explain interactions between nodes in a network.

2 Problem definition and preliminary

This section first formulates the problem of topical network embedding (TNE), followed by a brief description of the LDA and DeepWalk models, which are preliminary materials for the studied topic-aware network representation learning problem. For easy retrieval, the commonly used symbols are summarized in Table 1.

2.1 Formulation of TNE

Let an information network be represented as $G = (V, E, \mathbb{C})$, where $V = \{v_i\}_{i=1,\ldots,|V|}$ is a set of unique nodes ($|\cdot|$ denotes the cardinality, number of elements, in a set); and $E = \left\{e_{i,j}\right\}_{i,j=1,\ldots,|V|}$ and $i \neq j$ represents the set of edges in G. For each node v_i, we use cnt_i to denote its node content, which consists of a sequence of attributes or words $cnt_i = \left\{w_j\right\}_{j=1,\ldots,|cnt_i|}$. For all nodes in G, their content forms the content corpus $\mathbb{C} = \{cnt_i\}_{i=1,\ldots,|V|}$. To capture semantics of node content, we can learn a set of topics $\mathbb{T}$ from the content corpus $\mathbb{C}$, and for each node v_i, we consider its content cnt_i consisting of a set of k topics (multi-faceted semantics) denoted by $\mathbb{T}_i = \{t_n\}_{n=1,\ldots,k}$. The proposed TNE **aims** to represent each node v_i with a continuous low-dimensional vector $\mathbf{h}_{v_i} \in \mathbb{R}^{1\times d_v}$, i.e., learning a mapping $f : G \rightarrow \{\mathbf{h}_{v_i}\}_{i=1,\ldots,|V|}$ so that network structure, content, and topic can be fully preserved, where d_v is the dimension of the learned node vectors. Specifically, in a topic-aware information network, the relationship between two nodes is built on their shared topics. Therefore, an optimal TNE model could capture such fine-grained and topic-oriented node relations.

2.2 LDA: latent Dirichlet allocation

LDA is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over a set of latent topics ($\mathbb{T}$), where each topic

Table 1 Summary of key symbols and notations used in the proposed models

Symbols	Description
$v_i \in V$	The ith node (v_i) in the set of network nodes (V)
$e_{i,j} \in E$	An edge connecting v_i and v_j, also a member of the network edge set (E)
cnt_i	The content associated with node $v_i \in V$
$w_j \in cnt_i$	the jth word/attribute of the node content cnt_i
$\mathbb{C}$	The content corpus of all nodes. $\mathbb{C} = \{cnt_i\}_{i=1,...,\|V\|}$
$\mathbb{T}$	The list of all semantic topics learned by LDA
θ_i	The real-value topic distribution for node content cnt_i
$\mathbb{T}_i \subset \mathbb{T}$	The list of latent topics of node content cnt_i
$z_{j,q}$	The shared topics between nodes v_j and v_{j+q}, $q \neq 0$
k	The number of most relevant topics selected for each node content
$t_n \in \mathbb{T}_i$	The nth latent topic of cnt_i
r	The length of the truncated random walk
$\mathbb{S}$	A collection of node sequences generated by the random walk
$s = \{v_1, \rightsquigarrow, v_r\}$	A random walk node sequence, $s \in \mathbb{S}$
d_v	The node embedding size (the dimension of the node embedding vector)
d_t	The topic embedding size (the dimension of the topic embedding vector)
$\mathbf{h}_{v_j} \in \mathbb{R}^{1\times d_v}$	Vector representation of node v_j
$\mathbf{h}_{t_n} \in \mathbb{R}^{1\times d_t}$	Vector representation of topic t_n
$\mathbf{h}_{w_i}$	Vector representation of word w_i
d	The node sliding window size
h	The word sliding window size
α	A parameter balancing topology structure and node content

$t_i \in \mathbb{T}$ is characterized by a distribution over words (Blei et al. 2003). LDA assumes the following generative process for a document doc_i:

1. Choose $\theta_i \sim Dirichlet(\sigma)$.
2. For each word w_i in a given document doc_i:
   (a) Choose a topic $t_i \sim Multinomial(\theta_i)$.
   (b) Choose a word w_i from $p(w_i|t_i, \beta)$, multinomial probability conditioned on the topic t_i.

where σ and β are hyper-parameters need to be set before the model training. θ_i is the topic distribution of document doc_i and can be learned by the Gibbs sampling process (Griffiths 2002). After the training of LDA model, each document in the corpus corresponds to a unique topic distribution over all shared topics $\mathbb{T} = \{t_n\}_{n=1,...,|\mathbb{T}|}$.

2.3 DeepWalk

DeepWalk (Perozzi et al. 2014) is proposed to represent network with low-dimensional vectors by preserving only the neighborhood structures of nodes. It first conducts a

truncated random walk over the whole network to generate a set of node sequences that are analogical to natural language sentences. Then, similar to the way of learning word representations in a sentence based on the word occurrences (Pennington et al. 2014), DeepWalk learns node representations based on the SkipGram model (Mikolov et al. 2013) that keeps the affinity of linked nodes by representing nodes that are useful to predict their surrounding nodes. More formally, given a node sequence $s = \{v_1, \rightsquigarrow, v_r\}$, the goal of DeekWalk is to maximize the following log probability:

$$\mathscr{L} = \sum_{j=1}^{r} \log p\left(v_{j-d} : v_{j+d} | v_j\right) = \sum_{j=1}^{r} \left[\sum_{-d \le q \le d,\ q \ne 0} \log p\left(v_{j+q} | v_j\right) \right], \quad (1)$$

where d is the node sliding window size that controls the number of surrounding nodes need to be predicted by v_j. The conditional probability $p(v_{j+q}|v_j)$ is given by:

$$p\left(v_{j+q} | v_j\right) = \frac{\exp\left(\mathbf{h}_{v_j}^T \mathbf{h}_{v_{j+q}}\right)}{\sum_{i=1}^{|V|} \exp\left(\mathbf{h}_{v_j}^T \mathbf{h}_{v_i}\right)}, \quad (2)$$

in which $\mathbf{h}_{v_j}^T$ and $\mathbf{h}_{v_{j+q}}$ are input and output vectors of node v_j, respectively. DeepWalk considers the network structure information only, ignoring the rich text content information that usually reveal and explain the most direct reasons (e.g., shared semantics) of nodes linking each other from human's understanding.

3 TNE: topic network embedding model

3.1 Topic conditioned network embedding

As discussed in the Introduction section, since the node content may simultaneously exhibits multi-aspect semantics, conventional way of considering the content as a whole without reaching the topic level would confuse the relationships between nodes. For instance, a review article may discuss a wide range of latent topics and it connects to many cited articles of differing topics. In this paper, we abstract different aspects of each node content as a collection of topics trained based on the LDA model, i.e., the topic distribution of node v_i can be represented as a length $|\mathbb{T}|$ real-value vector $\theta_i = \left\{\theta_1^i, \theta_2^i, \ldots, \theta_{|\mathbb{T}|}^i\right\}$, with each value $\theta_{n,n=1,\ldots,|\mathbb{T}|}^i$ indicating the probability that v_i belongs to the corresponding topic $t_{n,n=1,\ldots,|\mathbb{T}|}$. However, each node could highlight only several important topics (e.g., with top probability values) aligned with different aspects of its content. In the setting of this paper, although the global topic list $\mathbb{T} = \{t_n\}_{n=1,\ldots,|\mathbb{T}|}$ is shared across all node contents during training of the LDA model, each node content is finally associated with its top-k ($k \le |\mathbb{T}|$) most relevant topics, thus resulting in different nodes may have different topic lists.

Our idea is that the relationship between each pair of nodes is built on their shared topics to increase the interpretability of node interactions. For example, for two linked nodes v_1 and v_2, assume that their topic lists (e.g., $k = 3$) are represented

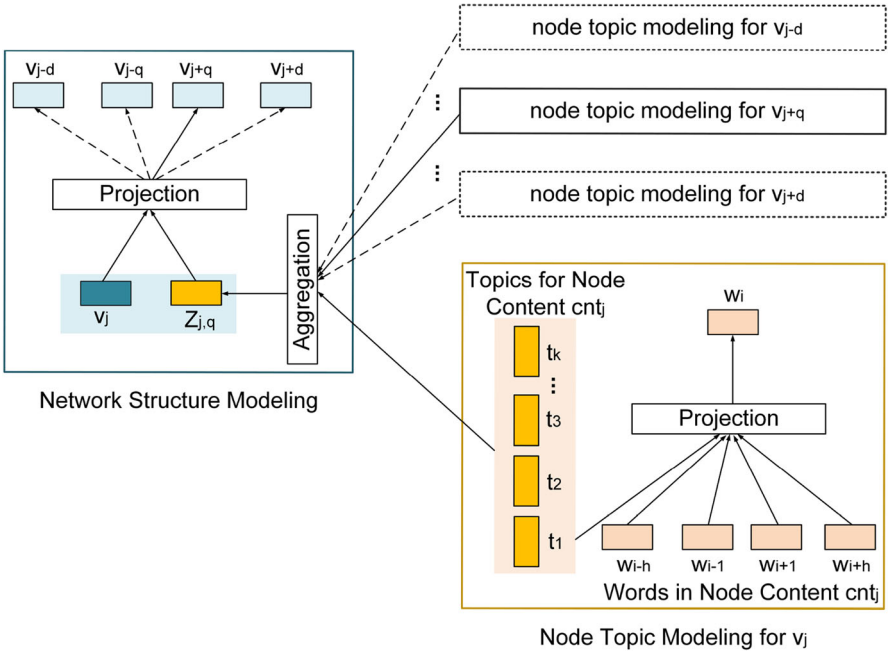

Fig. 2 The proposed TNE framework for node representation learning. For each node v_j, the node topic modeling (right panel) first learns semantic vectors of the top-k topics $t_1, \ldots, t_k$ based on the node content cnt_j (this learning process also applies to every other nodes $v_{j-d}, \ldots, v_{j+d}$, *etc.*). After that, the network structure modeling (left panel) is used to learn representation of $v_{j,j=1,\ldots,|V|}$ from the network structure, where the relationship modeling between each pair of nodes (e.g., v_j and v_{j+q}) is conditioned on the dynamically aggregated shared topics, i.e., $Z_{j,q}$. The node topic modeling and network structure modeling are optimized in a reciprocally enhanced fashion. (Refer to text for details)

as $\mathbb{T}_1 = \{t_1, t_3, t_4\}$ and $\mathbb{T}_2 = \{t_1, t_4, t_7\}$, respectively. Then, the relationship modeling between v_1 and v_2 will be conditioned on their shared topics $\{t_1, t_4\}$. Accordingly, we propose a unified optimization framework for this purpose shown in Fig. 2. It involves two integrate and reciprocally enhanced learning components to learn topic vector representations and node vector representations, respectively. First, the model (right panel in Fig. 2) learns the semantic representation for each of the k most relevant topics associated with node v_j from its node content cnt_j (e.g., represented by a collection of attributes or words). Meanwhile, the derived topic vectors are dynamically aggregated as context (e.g., $z_{j,q}$) for node relation modeling and representation (left panel in Fig. 2) learning from the network topology structure, where only the shared topics are collected for each pair of nodes. These two related learning processes are described in detail as follows.

3.1.1 Node topic modeling

Each node content cnt_j corresponds to a k-length topic list that is selected according to its topic distribution $\theta_j = \{\theta_1^j, \theta_2^j, \ldots, \theta_{|\mathbb{T}|}^j\}$ learned by LDA, i.e., ranking θ_j and selecting the corresponding top-k relevant topics to form the final topic list $\mathbb{T}_j =$

$\{t_n\}_{n=1,\ldots,k}$. We can conceive that each topic t_n is a semantic abstraction of all words from the text contents, where different topics highlight different words. In the right panel of Fig. 2, we adopt a similar way as the paragraph vector model (Le and Mikolov 2014) to learn the semantic representation of each topic t_n from the content of node v_j. During training, each topic is considered and mapped to a paragraph-level vector and together with other words in a window to predict the target word. The global learning objective of the node topic modeling is to maximize the following probability over all node contents and topics:

$$\mathscr{L} = \sum_{n=1}^{k} \sum_{j=1}^{|V|} \sum_{i=1}^{|cnt_j|} \log p\left(w_i | w_{i-h}, \ldots, w_{i+h}, t_n\right), \tag{3}$$

where $|V|$ is the total number of unique nodes in the network. $|cnt_j|$ is the total number of words in the content of node v_j, and h is the size of word sliding window set in the paragraph vector model. It is worth noting that each topic $t_{n,n=1,\ldots,|\mathbb{T}|}$ may be shared by many nodes (e.g., t_n is within their top-k-topic lists) at the same time, thus its vector representation can be collectively learned and trained across all corresponding node contents. The prediction task in Eq. (3) is typically done via a multi-class classifier defined by the softmax:

$$\log p\left(w_i | w_{i-h}, \ldots, w_{i+h}, t_n\right) = \frac{\exp\left(\bar{\mathbf{h}}_{w_i}^T \mathbf{h}_{w_i}\right)}{\sum_{j=1}^{|\mathbb{C}|} \exp(\bar{\mathbf{h}}_{w_i}^T \mathbf{h}_{w_j})}, \tag{4}$$

where $|\mathbb{C}|$ is the total number of words in the whole corpus. $\bar{\mathbf{h}}_{w_i}^T$ and $\mathbf{h}_{w_i}$ are respectively the input and output vectors w.r.t. word w_i. $\bar{\mathbf{h}}_{w_i}$ is computed by concatenating vectors of topic $t_{n,n=1,\ldots,k}$ and the averaged vectors of all words in the corresponding window by:

$$\bar{\mathbf{h}}_{w_i} = \mathbf{h}_{t_n} \oplus \left(\frac{1}{2h} \sum_{-h \le u \le h, u \ne 0} \mathbf{h}_{w_{i+u}} \right), \tag{5}$$

where $\mathbf{h}_{t_n}$ is the vector representation of topic t_n and $\oplus$ is the concatenation operation. After above learning and optimization processes, we can obtain the semantic representations of all associated relevant topics $t_{n,n=1,\ldots,k}$ for each node $v_{i,i=1,\ldots,|V|}$. In the next, the modeling of node relationships will be conditioned on the shared topics in the form of semantic vectors.

3.1.2 Network structure modeling

Similar to DeepWalk in modeling the network structure, we perform a truncated random walk over the whole network to capture the structural relations between nodes, with each walk rooting at a starting node and each step randomly jumping to one neighborhood node without bias. This process will generate a collection of fixed-length

node sequences $\mathbb{S}$ with the node neighborhood relationships well captured. However, different from conventional methods such as DeepWalk (Perozzi et al. 2014) and TriDNR (Pan et al. 2016) that either directly encode the structure-based node relationships (e.g., nodes with neighborhood relations have similar embeddings) or coarsely consider content-enhanced node relation modeling (e.g., nodes with neighborhood relations and shared contents have similar embeddings), we propose a more interpretable way by introducing topics to explain the relationships between nodes within each random-walk node sequence. In other words, two neighborhood nodes are mapped to similar representations conditioned on their shared topics. With shared topics as conditional context, the network structure modeling aims to maximize the following log-likelihood:

$$\mathscr{L} = \sum_{j=1}^{r} \sum_{-d \le q \le d, q \ne 0} \log p\left(v_{j+q} | v_j\right) + \log p\left(v_{j+q} | z_{j,q}\right), \tag{6}$$

where $z_{j,q}$ is the aggregation of all shared topics between nodes v_j and v_{j+q} represented by:

$$z_{j,q} = \{t_n |\ t_n \in \mathbb{T}_j,\ t_n \in \mathbb{T}_{j+q}\}, \tag{7}$$

where $\mathbb{T}_j$ and $\mathbb{T}_{j+q}$ are topic lists associated with node v_j and v_{j+q}, respectively. We can observe from Eq. (6), compared to Eq. (1) used in DeepWalk, the relationship modeling between nodes v_j and v_{j+q} is conditioned on their shared topics $z_{j,q}$. If no shared topics (e.g., $z_{j,q}$ is empty) exist between v_j and v_{j+q}, our model can still work. In such case, the left panel in Fig. 2 will degrade to the basic DeepWalk model and accordingly the optimization target changes from Eqs. (1) to (6).

The above two aspects of node topic modeling and network structure modeling are not learned independently but in a collective and reciprocally enhanced manner. To combine Eqs. (3) and (6), the collective training procedure of the TNE model is summarized in Algorithm 1, where our model finally seeks to optimize the following integrate probability:

$$\begin{aligned}\mathscr{L} = & \sum_{n=1}^{k} \sum_{j=1}^{|V|} \sum_{i=1}^{|cnt_j|} \alpha \log p\left(w_i | w_{i-h}, \ldots, w_{i+h}, t_n\right) \\ & + \sum_{j=1}^{|V|} \sum_{s \in \mathbb{S}} \sum_{-d \le q \le d, q \ne 0} (1-\alpha) \log p\left(v_{j+q} | v_j\right) + \alpha \log p\left(v_{j+q} | z_{j,q}\right), \end{aligned} \tag{8}$$

where α is the weight parameter to balance the learning of network structure, text content and topic information, i.e., with lower value of α, the optimization will emphasize more on the network structure information than the content/attribute information in the node representation learning process. In Eq. (8), $\log p\left(v_{j+q} | z_{j,q}\right)$ is computed by:

$$\log p\left(v_{j+q} | z_{j,q}\right) = \frac{\exp\left(\bar{\mathbf{h}}_{z_{j,q}}^T \mathbf{h}_{v_{j+q}}\right)}{\sum_{i=1}^{M} \exp\left(\bar{\mathbf{h}}_{z_{j,q}}^T \mathbf{h}_{v_i}\right)}, \tag{9}$$

in which $\bar{\mathbf{h}}_{z_{j,q}}$ is the averaged vectors of all shared topics in $z_{j,q}$:

$$\bar{\mathbf{h}}_{z_{j,q}} = \frac{1}{|z_{j,q}|} \sum_{t_n}^{z_{j,q}} \mathbf{h}_{t_n}. \tag{10}$$

3.1.3 Network representation

After above unified learning process, we finally represent each node in the network by two ways: (1) using the output vector of node v_j directly, i.e.,, the $\mathbf{h}_{v_j}$ optimized by Eq. (8); or (2) concatenating the vector of node v_j with the averaged vector of all its latent topics in $\mathbb{T}_j$ by:

$$\mathbf{h}_{v_j} = \mathbf{h}_{v_j} \oplus \left(\frac{1}{\mathbb{T}_j} \sum_{t_n}^{\mathbb{T}_j} \mathbf{h}_{t_n} \right). \tag{11}$$

Compared with the first way of representing nodes, the second representation way of concatenating node and topic vectors emphasizes more on the combined similarities of network structures and topic-represented contents between nodes.

Algorithm 1 Training TNE

Input: The given information network $G = (V, E, \mathbb{C})$
Input: The total number of topics $|\mathbb{T}|$ trained for corpus $\mathbb{C}$ and the number of relevant topics k selected for each node
Output: The node representations $\{\mathbf{h}_{v_j}\}_{j=1,\ldots,|V|}$

procedure RepresentationLearning
  Train topic distributions $\{\theta_i\}_{i=1,\ldots,|V|}$ over all shared topics $\mathbb{T} = \{t_n\}_{n=1,\ldots,|\mathbb{T}|}$ based on LDA from node contents $\mathbb{C}$
  for each node $v_j \in V$ **do**
    Rank its topic distribution $\theta_j = \left\{\theta_1^j, \theta_2^j, \ldots, \theta_{|\mathbb{T}|}^j\right\}$ by descent order
    $\mathbb{T}_j \leftarrow$ select top k relevant topics from $\mathbb{T}$ according to θ_j
    for each topic $t_n \in \mathbb{T}_j$ **do**
      Learn its topic embedding $\mathbf{h}_{t_n}$ from node content cnt_j by Eq. (3)
    end for
  end for
  for each node $v_j \in V$ **do**
    Learn its node embedding $\mathbf{h}_{v_j}$ by Eq. (6)
  end for
  Optimize Eqs. (3) and (6) by Eq. (8) with mini-batch training
end procedure

3.2 Optimization and parameter estimation

The optimization aims to maximize the collective probability in Eq. (8) over all observed node contents C and node sequences S. In this paper, we adopt the noise

contrastive estimation (NCE) (Gutmann and Hyvärinen 2010) to optimize and estimate the model parameters. NCE transforms the language model estimation problem to the problem of estimating the parameters of a probabilistic binary classifier that uses the same parameters to distinguish samples of the empirical distribution from samples generated by the noise distribution.

To optimize $\log p\left(v_{j+q}|v_j\right)$ in Eq. (8), for each node $v_j \in V$ together with the targeting node v_{j+q} (e.g., seen as a positive sample) within a node sequence $s \in \mathbb{S}$, NCE first generates f noisy/negative sample nodes $\left\{\tilde{v}_{ji}\right\}_{i=1,\ldots,f}$ from a noise distribution Q, with the class label $y = 1$ for positive samples and $y = 0$ for negative samples. Then, the goal is to optimize the sample distribution with a model parameterized by θ_1, where the conditional class distributions are defined as $P(v_{j+q}|y = 1, v_j) = P_{\theta_1}(v_{j+q}|v_j)$ and $P(v_{j+q}|y = 0, v_j) = P_n(v_{j+q})$, respectively. Accordingly, the posterior probabilities of class y associated to the positive and negative samples can be estimated by:

$$P(y = 1|v_{j+q}, v_j) = \frac{P_{\theta_1}(v_{j+1}|v_j)}{P_{\theta_1}(v_{j+q}|v_j) + fP_n(v_{j+q})}, \tag{12}$$

$$P(y = 0|v_{j+q}, v_j) = \frac{fP_n(v_{j+q})}{P_{\theta_1}(v_{j+q}|v_j) + fP_n(v_{j+q})}. \tag{13}$$

Finally, the unified classification objective of node structure modeling over all nodes in V is defined by maximizing the log-likelihood of positive examples belonging to class $y = 1$ and noisy samples belonging to $y = 0$:

$$J_{\theta_1} = \sum_{j=1}^{|V|}\left[\log\frac{P_{\theta_1}(v_{j+q}|v_j)}{P_{\theta_1}(v_{j+q}|v_j) + fP_n(v_{j+q})} + \sum_{i=1}^{f}\log\frac{fP_n(\tilde{v}_i)}{P_{\theta_1}(\tilde{v}_i|v_j) + fP_n(\tilde{v}_i)}\right]. \tag{14}$$

Analogically, the optimization of $\log p\left(w_i|w_{i-h}, \ldots, w_{i+h}, t_n\right)$ and $\log p\left(v_{j+q}|z_{j,q}\right)$ in Eq. (8) takes the similar forms as Eq. (14) and are parameterized by θ_2 and θ_3, respectively. Therefore, the federal NCE loss function for optimizing Eq. (8) is represented by:

$$J = (1 - \alpha)J_{\theta_1} + \alpha J_{\theta_2} + \alpha J_{\theta_3}, \tag{15}$$

where all parameters can be jointly learned based on the stochastic gradient descent algorithm (Bottou 2010). For example, the gradient of Eq. (14) w.r.t. θ_1 on the positive sample node v_{j+q} together with the generated f negative sample nodes is calculated by:

$$\begin{aligned}\frac{\partial}{\partial\theta_1}J_{\theta_1,v_{j+q}} = &\frac{fP_n(v_{j+q})}{P_{\theta_1}(v_{j+q}|v_j) + fP_n(v_{j+q})}\frac{\partial}{\partial\theta_1}P_{\theta_1}(v_{j+q}|v_j)\\ &-\sum_{i=1}^{f}[\log\frac{P_{\theta_1}(\tilde{v}_i|v_j)}{P_{\theta_1}(\tilde{v}_i|v_j) + kP_n(\tilde{v}_i)}\frac{\partial}{\partial\theta_1}P_{\theta_1}(\tilde{v}_j|v_j)].\end{aligned} \tag{16}$$

Note that Eq. (16) involves a sum over f noise samples instead of a sum over the entire $|V|$ nodes, thus making the model training time linear in the number of noise samples and independent of the network size.

3.3 Methodology discussion

Some methods exist to simultaneously embed network structures and textual contents, including probabilistic topic model-based techniques such as relational topic model (RTM) (Chang and Blei 2009) and its extension PLANE (Le and Lauw 2014), and neural network-based techniques such as TriDNR (Pan et al. 2016). In this section, we briefly compare and summarize the differences between the proposed TNE approach and the existing methods.

3.3.1 TNE versus RTM

RTM is a generative probabilistic topic model based on LDA and mainly used for embedding documents and their links between them. In the RTM, each document is first generated from topics as in LDA. The links between documents are then modeled as binary variables, one for each pair of documents. These are distributed according to a distribution that depends on the topics used to generate each of the constituent documents. Therefore, the content of the documents are statistically connected to the link structure between them. However, RTM only models the immediate links between nodes, which overlooks the valuable information of high-order node relations (e.g., nodes can reach each other through their neighbors) (Tang et al. 2015). In addition, RTM treats each document as a set of semantically equivalent features which are then used to measure the affinities with different linked documents. In comparison, TNE is able to capture the high-order node relations based on the random walk process and models each node content as a set of semantically different topics, where a node interacts with different neighbors for different shared topics.

3.3.2 TNE versus PLANE

PLANE is a probabilistic topic model built on RTM. PLANE not only learns a topic distribution for each document but also learns a low-rank representation expressed as coordinates on a D-dimensional space. However, similar to RTM, PLANE only preserves the direct document links and assumes that each link is built on the integrate similarity of document contents.

3.3.3 TNE versus TriDNR

Similar to TNE, TriDNR first adopts random walk to capture the node relationships over the whole network. Then, it learns node representations from both network structures and node contents based on a coupled neural network model. It enforces nodes with neighborhood relations and similar contents to also have close representations. However, the relationship between each pair of nodes is assumed to have dependency

Table 2 Dataset characteristics

Items	Cora	Wiki	Citeseer
# Nodes	2708	2405	3312
# Edges	5214	17,981	4732
# Unique words	14,694	4973	3703
# Average words per node	90	647	32
# Categories	7	17	6

with the whole content. Such a paradigm fails to differentiate the discrepancy when a node interacts with different neighbors. While TNE introduces topics to represent various aspects demonstrated by each node content, where nodes connect each others for different factors (e.g., the shared topics).

4 Experiments

This section compares the proposed approach for node representation learning against various strong baselines over three real-world datasets. First, similar to the literature (Pan et al. 2016; Perozzi et al. 2014), we evaluate the performance of all methods by conducting supervised node classification task based on the learned node representations. We then visualize the node representations in a 2-dimensional space to gain a straightforward performance comparison of the proposed approach with other state-of-the-art methods.

4.1 Datasets

We evaluate our approach by performing multi-class classification task on data collected from three real-word networks described by either rich content (e.g., Wiki and Cora) or sparse content (e.g., Citeseer). Their characteristics are summarized in Table 2.

Cora is a citation network dataset that contains 2708 machine learning papers from 7 research categories. Each paper corresponds to a category label. There are 5214 citation relations between these papers. Each paper is described by its abstract. There are 14,694 unique words in the vocabulary, and the average number of words for each node is 90.

Wiki dataset contains 2405 Web pages from 17 categories. Each web page belongs to a category. There are 17,981 hyperlinks between these Web pages. Each paper is described by a long text with an average number of words of 647. There are 4973 unique words in the vocabulary.

Citeseer dataset contains 3312 literature from 6 categories, and 4732 links between them. Each publication is described by a text with an average number of words of 32. There are 3703 unique words in the vocabulary.

4.2 Baselines

We choose baseline methods from three categories: (1) methods only preserving the network structure; (2) methods preserving both the structure and static text content; and (3) methods simultaneously considering the network structure, content, and latent topics.

Structure only

- **DeepWalk** (Perozzi et al. 2014) preserves only the neighborhood relations between nodes by the truncated random walk, and uses SkipGram model to learn the node embeddings.
- **LINE** (Tang et al. 2015) is a structure preserving embedding method, which can preserve both first-order and second-order node proximity of a large-scale network.
- **Node2vec** (Grover and Leskovec 2016) adopts a more flexible neighborhood sampling process than DeepWalk, i.e., biased random walk, to better capture the local structure (the second-order node proximity) and the global structure (the high-order node proximity).

Both structure and content:

- **TriDNR** (Pan et al. 2016) is a state-of-the-art method that exploits network structure, node content and label information for node representation learning.

Structure, content, and latent topics:

- **RTM** (Chang and Blei 2009) is the relational topic model that captures both text and network structure to learn the topic distribution representation of each document.
- **PLANE** (Le and Lauw 2014) extends the relational topic model and performs the topic-based embedding of document networks by incorporating text, links and latent topics in a unified model.
- **TNE** is our approach that introduces topics to supervise the interactions between nodes. It finally represents nodes without concatenating the topic vectors.
- **TNE**$_c$ is a variant of TNE. The only difference is that we represent nodes by concatenating the node and topic vectors, which is defined in Eq. (11).

With above comparative baselines, we mainly explore answers to the following two questions as well as their possible reasons. First, can textual contents help to enhance the relationship modeling between nodes, especially when the network node connectivity is sparse. Second, for methods (e.g., TriDNR) that simply model and preserve contents as a set of plain words or attributes, whether introducing latent topics to uncover multi-aspect semantics of each node content can further increase the interpretability of node interactions and improve the network representation learning performance.

4.3 Experiment settings

We perform the node classification task to evaluate the performance (Jian et al. 2018). For all baselines, we build a SVM classifier with linear kernel (Abraham et al. 2014) based on the Scikit-learn tool (sklearn.svm.SVC) on the training data and then predict

labels for all nodes from the test data, where the ratio ($p\%$) of training data ranges from 10 to 70%. Similar to the literature (Pan et al. 2016; Perozzi et al. 2014), we adopt Macro-F1 and Micro-F1 as metrics, which are defined as follows:

$$\text{Micro-F1} = \frac{\sum_{i=1}^{\mathbb{L}} 2TP^i}{\sum_{i=1}^{\mathbb{L}} \left(2TP^i + FP^i + FN^i\right)}, \tag{17}$$

$$\text{Macro-F1} = \frac{1}{|\mathbb{L}|} \sum_{i=1}^{\mathbb{L}} \frac{2TP^i}{\left(2TP^i + FP^i + FN^i\right)}, \tag{18}$$

where $\mathbb{L}$ is the set of label categories (e.g., Cora dataset has 7 label categories, so $|\mathbb{L}| = 7$). TP^i, FN^i and FP^i denote the number of true positives, false negatives and false positives w.r.t. the ith label category, respectively.

For all baselines w.r.t. two metrics, each experiment is repeated 20 times with randomly sampled training data, where the average performance and standard deviation are finally reported. In the experiment, 70% of all network nodes are labeled for training use (e.g., 80% for training the classifier and 20% for parameter selection) and the remaining are for evaluation (e.g., predict a label for each test node based on the trained classifier and then compare the prediction with its actual label). There are many hyperparameters involved in the proposed model. For the LDA model, following previous experience (Shi et al. 2018a), we set the prior hyper parameters σ, β and the iteration time based on Gibbs sampling as 0.1, 0.05 and 2000, respectively. To balance the model performance and the training efficiency, we set the total number of topics $|\mathbb{T}|$ trained by LDA as 20, the number of most relevant topics k for every node content as 2, the balance parameter α as 0.2, the node dimension size d_v as 100, the topic dimension size d_t as 100, and the training data ratio p as 70%. In addition, similar to the comparative method (Pan et al. 2016), settings of the rest parameters such as node window size d, word window size h, learning rate are kept the same for all the baselines and they are set as 2, 2 and 0.05, respectively.

4.4 Node classification performance

The node classification results of different algorithms on Cora, Wiki, Citeseer datasets are presented in Tables 3, 4 and 5, respectively. From these tables, we have the following three significant observations:

– The results on both Cora and Wiki datasets show that methods only preserving the network structure (e.g., DeepWalk and LINE) are generally inferior to content-assisted methods (TriDNR, TNE and TNE$_c$). Moreover, it is interesting to note that the content-preserving methods would improve more especially when the network structure is relatively sparse, i.e., the node connectivity of Cora dataset (e.g., average 1.93 connections per node) is sparser than that of Wiki dataset (e.g., average 7.48 connections per node), but the average Macro-F1 of TriDNR improved 28.5% over the combined average Macro-F1 of DeepWalk, LINE and Node2vec on Cora dataset, compared with that improved 12.7% on Wiki dataset. There are two major reasons behind. The first is that the networked data are generally sparse,

Table 3 Classification results on Cora dataset

%p	Macro-F1				Micro-F1			
	10	30	50	70	10	30	50	70
DeepWalk	0.395 ± 0.020	0.487 ± 0.009	0.520 ± 0.013	0.537 ± 0.012	0.429 ± 0.016	0.508 ± 0.009	0.544 ± 0.013	0.558 ± 0.013
LINE	0.453 ± 0.016	0.562 ± 0.014	0.589 ± 0.016	0.609 ± 0.023	0.474 ± 0.015	0.573 ± 0.012	0.599 ± 0.016	0.619 ± 0.021
Node2vec	0.679 ± 0.018	0.754 ± 0.013	*0.786 ± 0.011*	0.797 ± 0.016	0.687 ± 0.019	0.762 ± 0.012	*0.791 ± 0.010*	0.798 ± 0.016
RTM	0.372 ± 0.007	0.437 ± 0.014	*0.786 ± 0.011*	0.445 ± 0.016	0.404 ± 0.005	0.459 ± 0.016	0.466 ± 0.021	0.469 ± 0.015
PLANE	0.305 ± 0.018	0.374 ± 0.011	0.401 ± 0.011	0.420 ± 0.015	0.342 ± 0.017	0.403 ± 0.011	0.431 ± 0.008	0.450 ± 0.012
TriDNR	0.716 ± 0.012	0.771 ± 0.010	0.779 ± 0.009	0.789 ± 0.014	0.737 ± 0.008	0.781 ± 0.009	0.790 ± 0.009	*0.799 ± 0.014*
TNE	*0.708 ± 0.011*	*0.768 ± 0.011*	0.788 ± 0.011	*0.796 ± 0.014*	*0.718 ± 0.011*	*0.774 ± 0.011*	0.795 ± 0.011	0.803 ± 0.014
TNE_c	$\mathbf{0.746 \pm 0.011}$	$\mathbf{0.800 \pm 0.011}$	$\mathbf{0.816 \pm 0.011}$	$\mathbf{0.828 \pm 0.015}$	$\mathbf{0.759 \pm 0.009}$	$\mathbf{0.809 \pm 0.010}$	$\mathbf{0.827 \pm 0.009}$	$\mathbf{0.839 \pm 0.013}$

The 1st, 2nd, and 3rd best results are bold-faced, underscored and italic-formatted, respectively

Table 4 Classification results on Wiki dataset

%p	Macro-F1				Micro-F1			
	10	30	50	70	10	30	50	70
DeepWalk	0.230 ± 0.011	0.276 ± 0.007	0.291 ± 0.011	0.300 ± 0.013	0.329 ± 0.013	0.394 ± 0.009	0.424 ± 0.010	0.442 ± 0.017
LINE	0.339 ± 0.019	0.435 ± 0.017	0.472 ± 0.023	0.487 ± 0.018	0.434 ± 0.015	0.532 ± 0.011	0.567 ± 0.011	0.587 ± 0.017
Node2vec	0.379 ± 0.020	0.455 ± 0.019	*0.501 ± 0.020*	*0.524 ± 0.025*	0.488 ± 0.014	0.564 ± 0.013	0.603 ± 0.009	0.625 ± 0.011
RTM	0.443 ± 0.032	*0.473 ± 0.036*	0.390 ± 0.034	0.395 ± 0.060	*0.564 ± 0.028*	*0.587 ± 0.046*	0.449 ± 0.052	0.453 ± 0.075
PLANE	0.163 ± 0.013	0.214 ± 0.006	0.236 ± 0.008	0.246 ± 0.019	0.243 ± 0.014	0.307 ± 0.010	0.342 ± 0.011	0.360 ± 0.020
TriDNR	0.385 ± 0.017	0.434 ± 0.015	0.444 ± 0.008	0.490 ± 0.044	0.567 ± 0.011	0.616 ± 0.005	*0.627 ± 0.009*	*0.635 ± 0.024*
TNE	*0.408 ± 0.022*	0.479 ± 0.015	0.508 ± 0.028	0.530 ± 0.031	0.559 ± 0.011	0.564 ± 0.016	0.632 ± 0.011	0.636 ± 0.019
TNE_c	$\mathbf{0.491 \pm 0.019}$	$\mathbf{0.612 \pm 0.013}$	$\mathbf{0.594 \pm 0.016}$	$\mathbf{0.626 \pm 0.026}$	$\mathbf{0.660 \pm 0.012}$	$\mathbf{0.706 \pm 0.010}$	$\mathbf{0.725 \pm 0.010}$	$\mathbf{0.733 \pm 0.013}$

Table 5 Classification results on Citeseer dataset

%p	Macro-F1				Micro-F1			
	10	30	50	70	10	30	50	70
DeepWalk	0.247 ± 0.011	0.284 ± 0.008	0.292 ± 0.009	0.304 ± 0.014	0.281 ± 0.009	0.331 ± 0.008	0.348 ± 0.010	0.362 ± 0.016
LINE	0.311 ± 0.012	0.346 ± 0.010	0.353 ± 0.011	0.360 ± 0.013	0.341 ± 0.012	0.386 ± 0.009	0.403 ± 0.010	0.414 ± 0.013
Node2vec	*0.444 ± 0.015*	0.501 ± 0.010	0.515 ± 0.011	0.514 ± 0.013	*0.476 ± 0.016*	*0.549 ± 0.009*	*0.569 ± 0.011*	0.574 ± 0.015
RTM	0.262 ± 0.012	0.292 ± 0.013	0.296 ± 0.015	0.300 ± 0.018	0.280 ± 0.014	0.324 ± 0.013	0.327 ± 0.016	0.328 ± 0.019
PLANE	0.210 ± 0.008	0.234 ± 0.008	0.246 ± 0.008	0.246 ± 0.012	0.236 ± 0.007	0.274 ± 0.008	0.294 ± 0.009	0.299 ± 0.015
TriDNR	0.451 ± 0.009	*0.490 ± 0.009*	*0.502 ± 0.009*	*0.505 ± 0.014*	0.511 ± 0.009	0.555 ± 0.006	0.570 ± 0.009	0.574 ± 0.015
TNE	0.432 ± 0.012	0.468 ± 0.051	0.478 ± 0.011	0.489 ± 0.012	0.474 ± 0.014	0.516 ± 0.012	0.532 ± 0.009	*0.547 ± 0.011*
TNE_c	$\mathbf{0.530 \pm 0.012}$	$\mathbf{0.575 \pm 0.011}$	$\mathbf{0.585 \pm 0.012}$	$\mathbf{0.591 \pm 0.014}$	$\mathbf{0.578 \pm 0.013}$	$\mathbf{0.626 \pm 0.009}$	$\mathbf{0.643 \pm 0.010}$	$\mathbf{0.655 \pm 0.012}$

methods mining only the network structures cannot uncover the holistic relationships between nodes, i.e., many nodes actually have the same labeling information although they do not connect each other in the network. In addition, the rich contents leveraged by other methods are helpful to reveal and predict the absent or implicit relations between nodes, i.e., shared content features would strengthen the closeness between two nodes that is originally not well captured by the static network structure. Nevertheless, we can observe that RTM and PLANE with content information preserved take no advantages compared with the structure-based methods such as Node2vec. This is because RTM and PLANE consider only the immediate neighborhood relationships (e.g., one-hop relations), failing to model relationships between nodes whom can reach each other through their neighbors over the whole the network. Above phenomenons demonstrate that both network structure and text content are of important in learning quality network representations.

– In this paper, both TriDNR and the proposed TNE models use random walk to preserve network structures and meanwhile leverage contents to enhance the node relationships modeling. However, when compared with TriDNR, our proposed TNE model is competitive on Cora and Wiki datasets, i.e., after p is set larger than 50%, TNE performs generally better than TriDNR. The reason is that TriDNR incorporates the node content as a set of flat words for network structure relationships inference, which assumes each network connection has dependency with the whole contents of the two corresponding end nodes. Such a content modeling fashion fails to differentiate various purposes of a node while interacting with different neighborhood nodes. In comparison, TNE assumes that node relationships are built on the shared topics, which is more interpretable and also respects the fact that a node could show different aspects of the content when interacting with other nodes. Despite of above observations, results in Table 5 on Citeseer dataset show that TriDNR performs better than TNE in most cases. The reason is probably that the average content for each Citeseer network node (e.g., average 32 words per node) is much shorter than those of Cora and Wiki networks (e.g., average 90 and 647 words per node, respectively). This phenomenon demonstrates that long texts generally manifest mixed semantics and topics can be introduced to characterize them for more accurate node relations modeling. While short texts tend to reveal simple semantics, where the TriDNR model could be efficient for such type of environment setting. However, by concatenating the node and topic vectors as the final network representation, TNE_c has achieved the best performance over all three tested datasets, with the average Micro-F1 performance on Cora dataset 58.7% over DeepWalk, 42.8% over LINE, 64.7% over Node2vec, 98.9% over PLANE, 79.9% over RTM, and 4.1% over TriDNR.
– In the category of topic-based baselines, our models outperform both RTM and PLANE. The reason lies in the different mechanisms of leveraging the latent topics. RTM and PLANE model the text content as a set of topics and force the structurally linked nodes (e.g., documents) to follow similar topic distributions as a whole (e.g., all nodes have the same list of topics). Such a paradigm actually boils down to highlight the overall content similarity as do in TriDNR. In comparison, our methods consider topics as hidden information to explain the node relations in a

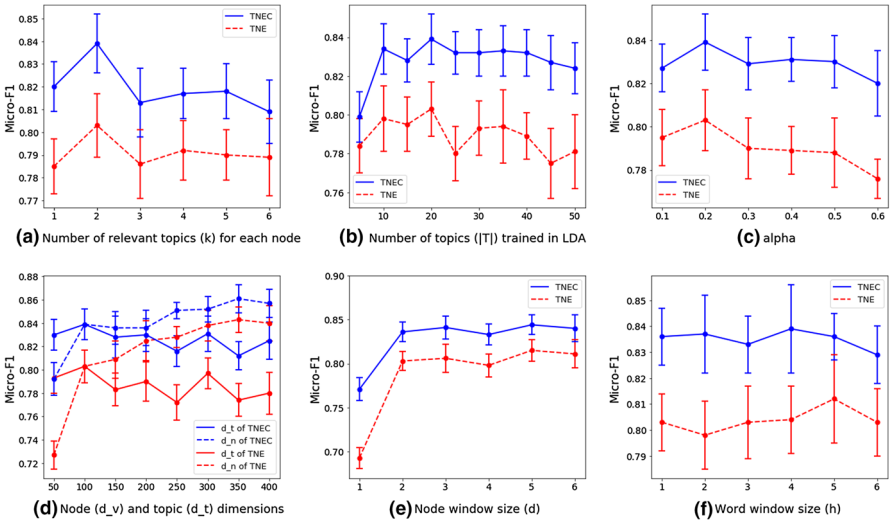

Fig. 3 Parameter influence study results

more fine-grained manner, i.e., each node has its own significant unique list of topics that correspond to multi-aspect content semantics. The experimental results have verified the reasonability of considering the topic-level similarity between nodes, especially for long text-described networks.

4.5 Parameter sensitivity study

Parameters that may significantly influence the classification performance are studied, including the number of relevant topics (k) demonstrated by each node content, the total number of topics ($|\mathbb{T}|$) learned from the whole content corpus based on the LDA, the sliding window sizes d and h, the balance parameter α, and the embedding sizes d_v and d_t for nodes and topics, respectively.

Figure 3a shows the influence of parameter k. A larger k value means that the node content exhibits more different semantics and each shared topic tends to reveal a specialized semantic relevance between two neighborhood nodes. We can observe that the Cora dataset obtains the best performance when each node has two latent topics. The number of topics ($|\mathbb{T}|$) trained in LDA reveals how many aspects of different semantics could be involved in the whole content corpus. The larger value of $|\mathbb{T}|$ means the semantic revealed by each single topic is more sparse and specialized, and meanwhile each node content is allowed to demonstrate more aspects of semantics. Figure 3b shows the impact of $|\mathbb{T}|$ and the best setting is 20 on the Cora dataset. The balance parameter α is used to control the weights of content, topic and structure information, where Fig. 3c shows that the performance fluctuates slightly with α, and the best setting for α is 0.2. We vary the number of node dimensions (d_v) and topic dimensions (d_t), and their effects are shown in Fig. 3d. Changes of these two parameters both have influence on the proposed approaches TNE and TNE$_c$, whereas

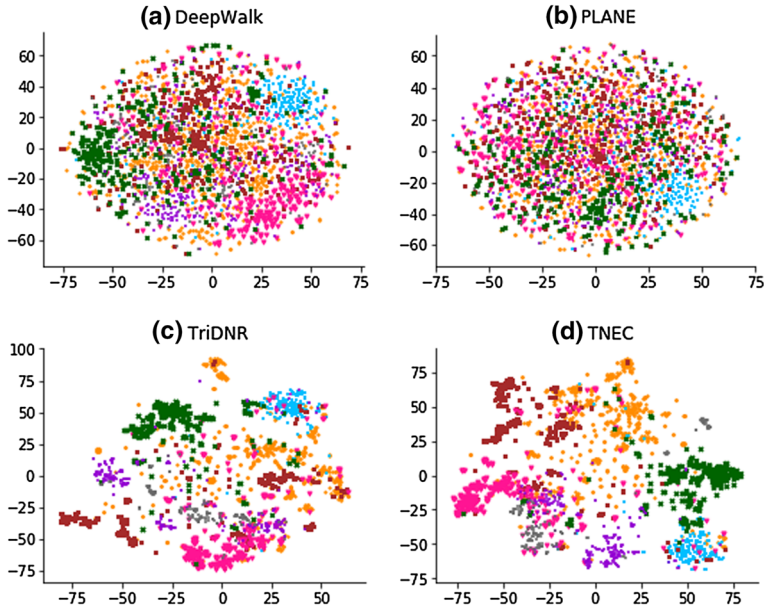

Fig. 4 Visualization of low-dimensional vectors on the Cora dataset

the dimension of node vector has a larger impact on the results, showing an increasing trend for lager value of d_v. The influences of node window size d used while modeling node relations are shown in Fig. 3e. It experiences a significant improvement (e.g., more than 6% for TNE$_c$) with d changing from 1 to 2, and then maintains at a high level with slight fluctuations. The best setting for d is 5 among all tested values. Similarly, Fig. 3f shows that the results are moderately affected with the change of word window size h, where the best settings for TNE and TNE$_c$ are 5 and 4, respectively.

4.6 Network visualization

We map node representations learned by different algorithms on Cora dataset onto a 2-dimensional space based on the t-SNE tool. There are 7 categories in total, and the more clear and far away they can be separated from each other generally means the better quality of the learned node representations (Maaten and Hinton 2008).

We can observe from Fig. 4a, b that DeepWalk and PLANE generate relatively poor visualization results (e.g., nodes from different categories overlap largely with each other), which means that the low-dimensional feature vectors learned by these two models are least distinguishing compared with other methods. From Fig. 4c, d we can see that TNE$_c$ performs significantly better than TriDNR, i.e., the resulting clusters of TNE$_c$ are more compact and clear in most cases. The results demonstrate that topic-level semantic relevance is more likely to be the reason of nodes clustering together (e.g., belonging to the same category) compared with the overall similarity of the node content.

4.7 Examples of topical network embedding

To demonstrate the impact of topics in node relations modeling, we showcase the similarities of two pairs of neighborhood nodes $\langle v_1, v_2\rangle$ and $\langle v_2, v_3\rangle$ by their Euclidean distances in the learned embedding space, i.e., the smaller distance generally means the better affinity characterization between a pair of nodes linking each other. In Table 6, each node is associated with three topics ($k = 3$) and for each topic two relevant word features are chosen from the corresponding node content (e.g., in the content of node v_1, words *evolve* and *population* are assigned with topic t_0). We can observe from Table 7 that: (1) Content-preserved models (e.g., TriDNR and TNE) can achieve more enhanced neighborhood affinities than the structure-based DeepWalk model; (2) TNE and TNE$_c$ have better constrains from the content perspective on node structures modeling than the TriDNR model.

DeepWalk only considers the structure-based node relations, which is hard to differentiate various neighborhood node relationships subtly, i.e., the affinity between v_1 and v_2 is determined by an edge that has no difference with others in the structure. While TriDNR considers the text content-enhanced node relations modeling, it assumes that the similarity of $\langle v_1, v_2\rangle$ (or $\langle v_2, v_3\rangle$) is built on the similarity of their whole content features (e.g., cnt_1 and cnt_2 are similar as a whole). However, node contents may contain many irrelevant or noisy features (from Table 6 we observe the neighborhood nodes v_2 and v_3 have many semantic-irrelevant words), which may confuse or even distort the original node relations reflected by the edges. In comparison, in our proposed models the relations of $\langle v_1, v_2\rangle$ and $\langle v_2, v_3\rangle$ are built on their respective shared topics $\{t_0, t_2\}$ and $\{t_{13}\}$, where each topic can be seen as an abstraction of similar words to identify the similar parts between different node contents. Therefore, it is more accurate and reasonable to incorporate topics to characterize the shared content features between nodes than to simply model the content as a set of flat word features and measure node affinities as a whole.

5 Related work

Graph mining is a popular topic (Kimura et al. 2010), where Network embedding is an emerging research area. The learned node representations can significantly facilitate

Table 6 Three example nodes with their topics, relevant words, and neighbors from the Cora dataset, where two relevant words are demonstrated for each respective topic

Nodes	Top-3 topics	Top-6 relevant word features	Neighbors
v_1	t_0; t_2; t_{15}	Evolve, population; classification, feature; disease, diagnostic	v_2
v_2	t_0; t_2; t_{13}	Selection adaption; tree, feature; network, topology	v_1, v_3
v_3	t_8; t_4; t_{13}	Reinforcement, learning; search, heuristic; neural, network	v_2

Table 7 Calculated Euclidean distances between nodes by different models

Distances	DeepWalk	TriDNR	Our models
$\langle v_1, v_2 \rangle$	1.04	0.58	0.55 (TNE); **0.45** (TNE_c)
$\langle v_2, v_3 \rangle$	1.15	0.65	0.56 (TNE); **0.52** (TNE_c)

The smaller the distance value, the better the affinity characterization between a pair of nodes linking each other

various downstream tasks, such as item recommendation (Cai et al. 2018a), link prediction (Cai et al. 2018b) and node classification (Jian et al. 2018). A large spectrum of research and their applications have been studied so far. We summarize existing works mainly in two branches: structure-preserving and attribute-preserving methods.

5.1 Structure-preserving methods

Mine only the network structure such as neighborhood relation and community information. It is based on the intuitive assumption that nodes appear with similar network structures also have similar semantic representations. DeepWalk (Perozzi et al. 2014) first performs a truncated random walk process over the whole network to capture the node structures. Then, it adopts the SkipGram model (Mikolov et al. 2013) for node relations modeling. However, Node2vec (Grover and Leskovec 2016) argues that DeepWalk is not flexible enough to capture the diversity of connectivity patterns in a network. To address this issue, Node2vec designs a second order random walk strategy to sample the neighborhood nodes, which can embed nodes with the same network community closely. Similarly, LINE (Tang et al. 2015) is proposed to simultaneously preserve the first (e.g., direct neighbors) and second order (e.g., relations by shared neighbors) proximity. The high-level community similarity is also widely considered in the past. For instance, MNMF (Wang et al. 2017) is a non-negative matrix factorization-based model to preserve both the microscopic neighborhood structure and the macroscopic community structure.

5.2 Attribute-preserving methods

Preserve both the network structure and the auxiliary attribute information such as labels and textual contents. It based on the assumption that the content information help to interpret the affinities between nodes aligned with that revealed by the network structures. MMDW (Tu et al. 2016) and GENE (Chen et al. 2016) are proposed to encode the labeling information of nodes, which assume that label and cross-label information is important for high-quality network embedding. LANE (Huang et al. 2017) is further proposed to model the affinities and correlations between node attributes, network structure and labels. Recent studies show that rich texts of nodes are very helpful to enhance and predict the node relationships, especially when the networked data are sparse. For example, the relational topic model (RTM) (Le and Lauw 2014) is utilized to model both the network content and link relationships. TADW (Yang et al. 2015)

absorbs from the rich texts for enhancing the structure-based representation learning based on an equivalent matrix factorization method as the DeepWalk. TriDNR (Pan et al. 2016) is proposed to integrate the node structure, content and labels. It enforces the node representations to be learned from simultaneously the network structure and text content under the shared model parameters.

Although different methods are developed to integrate node content, nearly all of them take the text content as static auxiliary information, and simply model them as a simple flat word/attribute set. This paradigm is not reasonable when the content of a node actually reveal multiple aspects of semantics, with each aspect illustrated by a part of word features. To address this problem, we introduce the concept of topics to represent node content in our work, which allows the text content of a specific node to describe rich semantics.

6 Conclusion and future work

In this paper, we studied topical network embedding for content-rich networks. Different from existing content-assisted embedding methods that incorporate content as simple flat attribute/word set, we introduced topics to manifest different aspects of node content, allowing each node to exhibit different semantics when interacting with other nodes. The proposed method, topical network embedding (TNE), incorporates content to perform topic-conditioned node structure/relationship modeling. A unified optimization framework is proposed to learn network node and topic representation, by simultaneously leveraging node structure, content and latent topics. Experiments and validations showed that by enabling topic-aware node relations modeling, TNE achieves significant performance gain compared to state-of-the-art baselines, especially for long text-described networks. On the other hand, as we have discussed in the experimental results, the short text associated with some networks tend to demonstrate simplified semantics and the various topics intentionally introduced for complicated semantic characterization could produce undesirable embedding results.

Future work can emphasize on the following two aspects: (1) for networks with sparse or noisy content, LDA might be ineffective to capture topics. Alternatively, we recommend to use advanced topic models such as RTM (Le and Lauw 2014) to fully utilize the document link relationships for more accurate topics eliciting; and (2) for networks with multi-labels (e.g. an image with multiple labels), we recommend to assign weights to different topics and perform weighted topic-aware network embedding.

Acknowledgements This work is supported in part by the US National Science Foundation (NSF) through Grants Nos. IIS-1763452 and CNS-1828181.

References

Abraham A, Pedregosa F, Eickenberg M, Gervais P, Mueller A, Kossaifi J, Gramfort A, Thirion B, Varoquaux G (2014) Machine learning for neuroimaging with scikit-learn. Front Neuroinform 8(2):14

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3(1):993–1022

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of the 19th international symposium on computational statistics, pp 177–186
Cai X, Han J, Pan S, Yang L (2018a) Heterogeneous information network embedding based personalized query-focused astronomy reference paper recommendation. Int J Comput Intell Syst 11(1):591–599
Cai X, Han J, Yang L (2018b) Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation. In: Proceedings of the 32nd AAAI conference on artificial intelligence, pp 5747–5754
Chang J, Blei D (2009) Relational topic models for document networks. In: Proceedings of the 12th international conference on artificial intelligence and statistics, pp 81–88
Chen J, Zhang Q, Huang X (2016) Incorporate group information to enhance network embedding. In: Proceedings of the 25th ACM international conference on information and knowledge management, pp 1901–1904
Dojchinovski M, Vitvar T (2018) Linked web apis dataset. Semant Web 9(4):1–11
Griffiths T (2002) Gibbs sampling in the generative model of Latent Dirichlet Allocation. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.138.3760
Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 855–864
Gutmann M, Hyvärinen A (2010) Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: Proceedings of the 13th international conference on artificial intelligent and statistics, pp 297–304
Huang X, Li J, Hu X (2017) Label informed attributed network embedding. In: Proceedings of the 10th ACM international conference on web search and data mining, pp 731–739
Jian L, Li J, Liu H (2018) Toward online node classification on streaming networks. Data Min Knowl Discov 32(1):231–257
Kimura M, Saito K, Nakano R, Motoda H (2010) Extracting influential nodes on a social network for information diffusion. Data Min Knowl Discov 20(1):70
Le TM, Lauw HW (2014) Probabilistic latent document network embedding. In: Proceedings of the 14th international conference on data mining, pp 270–279
Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31st international conference on machine learning, pp 1188–1196
Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(11):2579–2605
Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:1301.3781
Oro E, Pizzuti C, Procopio N, Ruffolo M (2018) Detecting topic authoritative social media users: a multilayer network approach. IEEE Trans Multimed 20(5):1195–1208
Pan S, Wu J, Zhu X, Zhang C, Wang Y (2016) Tri-party deep network representation. In: Proceedings of the 25th international joint conference on artificial intelligence, pp 1895–1901
Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 1532–1543
Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 701–710
Shi M, Liu J, Zhou D, Tang Y (2018a) A topic-sensitive method for mashup tag recommendation utilizing multi-relational service data. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2018.2805826
Shi T, Kang K, Choo J, Reddy CK (2018b) Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: Proceedings of the 27th international conference on world wide web, pp 1105–1114
Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, pp 1067–1077
Tu C, Zhang W, Liu Z, Sun M et al (2016) Max-margin DeepWalk: discriminative learning of network representation. In: Proceedings of the 25th international joint conference on artificial intelligence, pp 3889–3895
Verma A, Bharadwaj KK (2017) Identifying community structure in a multi-relational network employing non-negative tensor factorization and GA k-means clustering. Wiley Interdiscip Rev Data Min Knowl Discov 7(1):e1196
Wang X, Cui P, Wang J, Pei J, Zhu W, Yang S (2017) Community preserving network embedding. In: Proceedings of the 31st AAAI conference on artificial intelligence, pp 203–209

Wang C, Song Y, Li H, Zhang M, Han J (2018) Unsupervised meta-path selection for text similarity measure based on heterogeneous information networks. Data Min Knowl Discov 32(6):1735–1767
Yang C, Liu Z, Zhao D, Sun M, Chang EY (2015) Network representation learning with rich text information. In: Proceedings of the 24th international joint conference on artificial intelligence, pp 2111–2117
Zhang D, Yin J, Zhu X, Zhang C (2018) Network representation learning: a survey. IEEE Trans Big Data. https://doi.org/10.1109/TBDATA.2018.2850013