Network Representation Learning based on Community and Text Features

Yu Zhu[1], Zhonglin Ye[2], Haixing Zhao[1,2*], Ke Zhang[1]

[1]School of Computer, Qinghai Normal University, Xining 810008, China

[2]School of Computer Science, Shaanxi Normal University, Xi'an 710062, China

*Corresponding author: h.x.zhao@163.com

Abstract. Network representation learning (NRL) aims at building a low-dimensional vector for each vertex in a network, which is also increasingly recognized as an important aspect for network analysis. Some current NRL methods only focus on learning representations using the network structure. However, vertices in lots of networks may contain community information or text contents, which could be good for relevant evaluation tasks, such as vertex classification, link prediction and so on. Since it has been proved that DeepWalk is actually equivalent to matrix factorization, we propose community and text-enhanced DeepWalk (CTDW) based on the inductive matrix completion algorithm, which incorporates community features and text features of vertices into NRL under the framework of matrix factorization. In experiments, we evaluate the proposed CTDW compared with other state-of-the-art methods on vertex classification. The experimental results demonstrate that CTDW outperforms other baseline methods on three real-world datasets.

Keywords: Network Representation Learning, Community and Text Features, Inductive Matrix Completion

1 Introduction

With the constant development of the networks, modern society has entered an era of information explosion, and life is full of information. The relevance between the information forms all sorts of information networks, such as various social networks, citation networks between academic papers and so on. Recently, in order to extract useful information from massive network data, some researchers have already focused on network representation learning (NRL), which aims to build low-dimensional vectors for vertices and is applied to lots of machine learning tasks, such as vertex classification [1], recommendation system [2][3], and link prediction [4]. In particular, NRL can alleviate the sparse issue caused by the conventional representation method based on the graph spectrum.

Recently, there are mainly two types of NRL methods, one of which only takes network structures as input to learn vertex representations without considering other information, and the other simultaneously considers both network structure and auxiliary information including text information or community information of vertices. For example, based on a word representation model in NLP named as Skip-Gram [5], DeepWalk [6] learns vertex representations from random walk sequences in social networks. Shortly afterwards, various methods based on DeepWalk have been proposed for representation learning, such as Line [7], GraRep [8] and SDNE [9].The above-mentioned methods are only based on network structures. Nevertheless, vertices in real-world networks usually contain sufficient community or text information, which may also be important to NRL. For example, one paper is more easily referred by other papers similar or same research fields or communities with it. In addition, there is lots of text information in each paper regarded as a vertex of the network. Inspired by this, some researchers incorporate texts or communities into training models to learn better representations, such as CENR [10], CNRL [11], TriDNR [12]. CENR utilizes a neural network method to jointly learn the inter-node and node-sentence network relationships. CNRL decomposes all contents into sentences, and then it applies Wavg, RNN and BiRNN to verify the feasibility and reliability of this algorithm. TriDNR utilizes two neural networks to learn the representations based on inter-node, node-word, and label-word network relationships.

Besides the above-mentioned methods, Inductive Matrix Completion (IMC) [13] also takes advantage of additional information to complete gene-disease matrix. IMC is actually a matrix reduction algorithm and utilizes two feature matrices to factorize the objective matrix. The result matrix obtained from the objective matrix factorization contains influence factors of the feature matrices. Inspired by the idea of IMC, we propose a novel NRL method to take network structure, community features and text features together into consideration, named as community and text -enhanced DeepWalk (CTDW).

We test our method against several baselines on three real-world datasets. The vertex classification accuracy of our method outperforms the accuracies of other baselines when the ratio of training set ranges from 10% to 90%. Meanwhile, our method shows strong clustering abilities by node clustering visualizations on Citeseer, Cora and DBLP. In addition, we find that CTDW can learn better network representations than only network-based DeepWalk with the help of community and text information by case study.

2 Related Work

As a new technique, network representation learning (NRL) becomes more and more popular in network analysis. Some researchers' NRL methods mainly focus on representation learning based on network structures without taking other information into account. For example, Hofmann [14] first introduces the concept of network representation learning. Inspired by Skip-Gram, DeepWalk is proposed to learn the representations from network structures. Recently, there are some new methods to incorporate additional information into NRL. For example, NetPLSA [15] takes both network structures and text information into account for topic modeling. Max-margin DeepWalk (MMDW) [16] incorporates label information of vertices into NRL. Inspired by DeepWalk, node2vec [17] designs a biased random walk algorithm, which sufficiently utilizes different neighborhood information. The proposed models by Cao *et al* [8] and Wang *et al* [9] are applied to capture the vertices of the neighborhoods by incorporating global information into NRL. CENR, CNRL, TriDNR and M-NMF [18] incorporate text or community information into network representations.

The rest of this paper is organized as follows. Section 3 gives the formal definition of NRL and DeepWalk, and demonstrates that DeepWalk is equivalent to matrix factorization in fact. We put forward our method for NRL with community and text features in section 4.The datasets and experimental results are introduced in section 5. Section 6 concludes this paper.

3 DeepWalk Model

3.1 Formalization of NRL

Network representation learning is formalized as follows. Suppose that there is a network $G = (V, E)$, where V denotes the set of vertices and E denotes the set of edges. We want to build a low-dimensional representation vector $r_v \in R^k$ for each vertex v of G, where k is expected to be much smaller than $|V|$, which is the number of vertices of V.

3.2 DeepWalk

As a word representation method, Skip-Gram was introduced by DeepWalk into the study of social network to learn vertex representation from the network structure.

DeepWalk performs short random walks over the given network G to generate a sequence of vertices $S = \{v_1, v_2, \dots, v_{|S|}\}$. We regard the vertices $v \in \{v_{i-t}, \dots, v_{i+t}\} \backslash \{v_i\}$ as the context of the center vertex v_i, where t is the window size. The objective of DeepWalk is to maximize the average log likelihood of all vertex-context pairs in the random walk vertex sequence S

$$\frac{1}{|S|}\sum_{i=1}^{|S|}\sum_{-t\le j\le t, j\ne 0}\log\ p(v_{i+j} \mid v_i) \tag{1}$$

where $p(v_j|v_i)$ is defined by softmax function,

$$p(v_j \mid v_i) = \frac{\exp(c_{v_j}^T r_{v_i})}{\sum_{v\in V}\exp(c_v^T r_{v_i})}. \tag{2}$$

Here, r_{v_i} and c_{v_j} are the representation vectors of the center vertex v_i and its context vertex v_j, respectively. In other words, each vertex v has two representation vectors: r_v when v is a center vertex and c_v when v is a context vertex.

3.3 DeepWalk as Matrix Factorization

Fortunately, Yang *et al* [19] has proved that given a network $G = (V, E)$, DeepWalk actually factorizes a matrix $M \in R^{|V|\times|V|}$, where each entry M_{ij} is logarithm of the average likelihood to perform a random walk from the vertex v_i to the vertex v_j in fixed steps. As shown in Fig.1, the matrix M is factorized into the product of two low-dimensional matrices $W \in R^{k\times|V|}$ and $H \in R^{k\times|V|}$, where $k \ll |V|$. Note that we regard each column of the matrix W as a low-dimensional representation vector $r_v \in R^k$ for each vertex v.

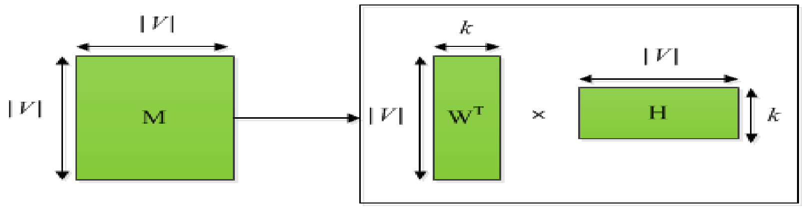


Fig.1. DeepWalk as Matrix Factorization

4 Our Method

In this section, we first introduce inductive matrix completion, and then give a detailed introduction to community and text matrix. At last, we formulate our NRL method to take network structure, community features and text features together into consideration, under the framework of matrix factorization.

4.1 Inductive Matrix Completion (IMC)

Matrix is a common way to represent relationships among the vertices of the networks, whose inherent structure information can be figured out by means of the matrix analysis. In order to make good use of abundant information of the networks, researchers resort to inductive matrix completion, which can make these features participate in representation learning by incorporating two feature matrices into the objective function, if there are additional features in the items of matrix $M \in R^{b \times d}$. Assume that two feature matrices are $X \in R^{f_x \times b}$ and $Y \in R^{f_y \times d}$.We want to solve matrices $W \in R^{k \times f_x}$ and $H \in R^{k \times f_y}$ to minimize square loss function, where $k \ll \{b, d\}$.

$$\min_{W,H} \sum_{(\mathrm{i},\mathrm{j}) \in \Omega} (M_{\mathrm{ij}} - (X^T W^T H Y)_{ij})^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2) \tag{3}$$

where Ω and λ are an observation set of matrix M and a harmonic factor to balance two components respectively. In addition, $\|\cdot\|_F$ is Frobenius norm of the matrix.

Note that some researchers originally put forward IMC to complete gene-disease matrix with gene and disease features. Although the goal of IMC differs from that of our method, we can utilize the idea of IMC, which takes the two aforementioned feature matrices as the auxiliary parameters to factorize the target matrix M. In this paper, we take community and text matrices as auxiliary matrices to learn better network representations.

4.2 Community and text matrix

The text in the dataset is converted to a matrix, which is factorized by Singular Value Decomposition (SVD) to obtain a text matrix $T \in R^{k \times |V|}$, where $\mathrm{k} \ll |V|$. The graph in the dataset is converted to adjacency matrix $A \in R^{|V| \times |V|}$, and then SVD is adopted to factorize the matrix $M = (A + A^2)/2 \in R^{|V| \times |V|}$ to obtain a matrix $B \in R^{n \times |V|}$, where $\mathrm{n} \ll |V|$. By means of the matrix B, we call K-means algorithm [20] to return a vector IDX containing community labels of each point in the matrix B. The community matrix $P_C \in R^{|V| \times |V|}$is obtained from the vector IDX, and then we use SVD algorithm to factorize the matrix P_C to obtain a community matrix $C \in R^{k \times |V|}$. In fact, the community and text matrix $C_T \in R^{2k \times |V|}$is a concatenation of the community matrix C and the text matrix T.

4.3 Community and text-Enhanced DeepWalk (CTDW)

Given a network $G = (V, E)$ and its corresponding community and text matrix $C\_T \in R^{2k \times |V|}$, we propose community and text-enhanced DeepWalk (CTDW) to learn representations of each vertex $v \in V$ from both network structure and community and text features. Since Yang *et al* [19] has found a tradeoff between speed and accuracy in their method by factorizing the matrix $M = (A + A^2)/2$ from the derivation of MF-style DeepWalk, we also factorize the same matrix in our method:

$$M = (A + A^2)/2 \tag{4}$$

As shown in Fig.2, the matrix M is factorized into the product of four matrices $E \in R^{|V|\times|V|}$, $W \in R^{2k\times|V|}$, $H \in R^{2k\times 2k}$, $C\_T \in R^{2k\times|V|}$ by using IMC algorithm, where E is an identity matrix, W and H are both target matrices and C_T is an additional community and text matrix.

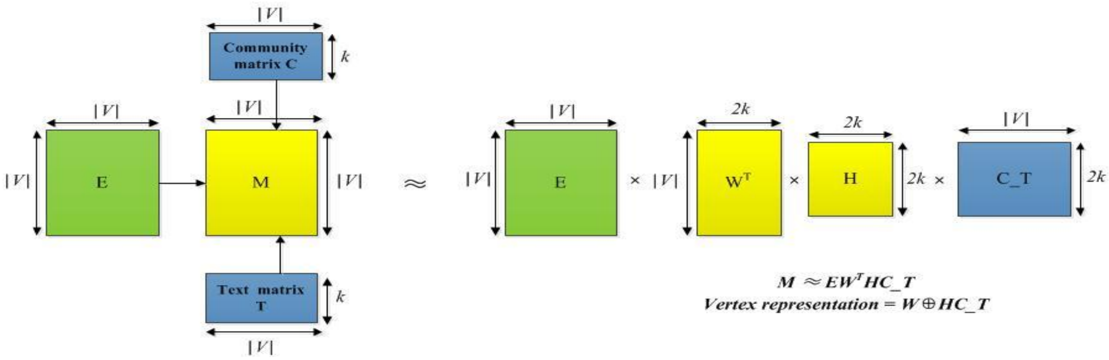


Fig.2. CTDW Framework

Here, we factorize M instead of $\log M$ for computational efficiency. The reason is that $\log M$ has much more non-zero entries than M, and the complexity of matrix factorization with square loss [3] is proportional to the number of non-zero elements of the matrix M. Our task is to solve matrices $W \in R^{2k\times|V|}$ and $H \in R^{2k\times 2k}$ to minimize

$$\min_{W,H} \left\| M - EW^T HC\_T \right\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2) \tag{5}$$

A direct method for representation learning is to independently train community or text features and network structure, and then we concatenate community or text representations along with network structure representations. However, this method leads to a loss of joint learning between network structure and community or text information. Therefore, similar with inductive matrix completion, CTDW incorporates community and text features to learn network representations. As shown in Fig.2, we factorize the matrix $M$with the help of the community matrix C and the text matrix T to obtain a target matrix W , which contains influence factors of the community and text matrix. The vertex representation is equivalent to a concatenation of W and HC_T, that is to say, vertex representation = $W \oplus HC\_T$. Since both $W$and HC_T obtained from CTDW can be regarded as low-dimensional representations of vertices, we build a unified $4k$-dimensional matrix for network representations.

5 Experiments

5.1 Datasets

Citeseer. Citeseer[1] contains 3312 publications from six classes and 4732 links, which are citation relationships between the documents. Each document is described by a binary vector of 3703 dimensions.

Cora. Cora[2] contains 2708 machine learning papers from seven classes and 5429 links, which are

1 http://citeseerx.ist.psu.edu/
2 https://people.cs.umass.edu/mccallum/data.html

citation relationships between the documents. Each document is described by a binary vector of 1433 dimensions indicating the presence of the corresponding word.

DBLP. DBLP[3] is a bibliographic network composed of authors and papers containing 3119 nodes from four classes and 39516 links.

5.2 Baseline methods

Structure-Based Method

node2vec. node2vec is an algorithmic framework for learning continuous feature representations for nodes in networks. In node2vec, we learn a mapping of nodes to a 200-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes.

LINE. LINE is proposed to learn network representations for large scale networks, which takes both 1-order and 2-order proximity into account, and the concatenation of these two representations is used as the final embedding. Same as DeepWalk, the dimension of representation vectors is 200.

DeepWalk. DeepWalk is a popular network structure-only representation learning method, which learns network representations by using the Skip-Gram model. We set parameters as follows, walk length $\gamma = 80$ and window size $t = 10$, representation dimension $k = 200$.

MFDW. MFDW is the abbreviated form of DeepWalk. MFDW factorizes the target matrix $M = (A + A^2)/2$, where A is the adjacency matrix of the network and then it uses the matrix $W \in R^{200\times|V|}$ to train classifiers.

Content-Based Method

Text. We take the text matrix $T \in R^{100\times|V|}$ as 100-dimensional representation. The method is content-only baseline.

Community. We take the community matrix $C \in R^{100\times|V|}$ as 100-dimensional representation. The method is also content-only baseline.

Community + Text. We can simply concatenate the vectors from both community features and text features into a 200-dimensional vector for network representations.

Combined Method

MV + Community + Text. We use SVD algorithm to factorize the matrix $M = (A + A^2)/2$ to obtain 100-dimensional vectors $MV$and then simply concatenate the vectors from the matrix, community features and text features into a 300-dimensional vector for network representations.

CT@E. CT@E is a variant of CTDW, based on IMC algorithm and its two auxiliary feature matrices are the matrix $P_C \in R^{|V|\times|V|}$ and the text matrix T to replace the identity matrix and the community and text matrix C_T in CTDW respectively. Its representation dimension is 200.

MMDW. As MFDW, MMDW also factorizes the matrix $M = (A + A^2)/2$ and uses the matrix $W \in R^{200\times|V|}$ to train classifiers. MMDW utilizes the max-margin approach to optimize matrix W, and thus the learnt representation possesses a discriminative ability.

TADW. TADW [19] incorporates text features of vertices into network representation learning under the framework of matrix factorization, and also factorizes the same matrix $M = (A + A^2)/2$ as MFDW and MMDW. It uses the concatenation matrix $W \oplus HT \in R^{200\times|V|}$ to train classifiers.

5.3 Classifiers and Experiment Setup

We conduct our experiments on three real-world network datasets. We adopt the classification tasks to

---

3 http://arnetminer.org/citation

verify the feasibility of our method. For all three datasets, we reduce the dimension of vectors to 100 via SVD decomposition of the related matrix, and obtain community and text matrix $C\_T \in R^{200\times|V|}$ by concatenating the community matrix C and the text matrix T. We also take the community and text matrix C_T as a content-only baseline. To evaluate our method, we randomly select a portion of documents as training set, and the rest are testing set. We take representation vectors of vertices as features to train classifiers, and calculate the accuracy of vertex classifications based on different training ratios, which range from 10% to 90%. Note that, the dimension of representation vectors from CTDW is $4k$. Our experiment is repeated for 10 times to record the average classification accuracy.

5.4 Experimental Results and Analysis

The classification experimental results for three datasets are shown in Table 1, Table 2 and Table 3. CTDW consistently outperforms the baseline methods on different datasets, which shows the feasibility of our method.

Table 1. Accuracy (%) of vertex classification on Citeseer

Training Ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%
node2vec	54.38	57.29	58.64	59.53	59.63	59.88	60.43	61.36	62.42
LINE	39.82	46.83	49.02	50.65	53.77	54.20	53.87	54.67	53.82
DeepWalk	49.09	55.96	60.65	63.97	65.42	67.49	66.80	66.82	63.91
MFDW	47.95	54.75	57.18	58.47	59.86	60.24	61.53	60.64	61.60
Text	57.76	66.02	69.70	70.12	70.54	70.58	70.76	70.34	70.41
Community	63.00	66.70	67.69	67.83	68.08	68.45	68.07	67.93	68.07
Community+Text	66.63	69.91	70.72	71.22	71.38	72.05	72.16	71.82	71.36
MV+Community+Text	68.29	70.54	71.34	71.90	71.47	72.56	72.52	72.39	72.51
CT@E	68.76	69.07	69.12	69.39	69.49	69.47	69.68	69.22	69.75
MMDW	55.49	60.70	63.66	65.27	66.02	69.14	69.34	69.47	69.72
TADW	70.20	71.23	73.17	73.45	74.02	74.06	75.48	76.74	75.12
CTDW	70.41	72.39	73.49	74.48	74.43	75.17	75.99	75.95	76.23

Table 2. Accuracy (%) of vertex classification on Cora

Training Ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%
node2vec	76.30	79.26	80.43	80.70	81.13	81.26	82.18	81.63	82.81
LINE	65.13	70.17	72.20	72.92	73.45	75.67	75.25	76.78	79.34
DeepWalk	68.51	73.73	76.87	78.64	81.35	82.47	84.31	85.58	85.61
MFDW	66.38	75.52	78.78	80.54	82.09	81.93	82.62	81.57	83.81
Text	57.70	67.26	70.10	71.05	71.47	71.74	72.04	73.17	74.02
Community	52.08	59.60	61.99	62.67	63.21	63.49	63.69	63.75	64.85
Community+Text	65.07	69.36	71.40	72.28	72.59	73.09	73.17	74.99	75.52
MV+Community+Text	75.18	79.69	81.46	81.90	82.48	83.67	83.70	84.16	84.33
CT@E	65.48	66.16	66.58	66.52	66.88	66.85	67.05	66.63	66.57
MMDW	73.61	79.99	80.43	81.92	83.76	84.97	86.39	86.70	87.45
TADW	80.09	80.70	83.47	84.94	85.37	85.87	85.94	85.26	86.01
CTDW	81.99	82.25	84.62	85.42	86.24	85.91	86.83	86.43	87.12

Table 1, Table 2 and Table 3 show classification accuracies on Citeseer, Cora and DBLP datasets. From the three tables, we have following observations:

(1) CTDW consistently outperforms all the other baselines on all three datasets. For example, CTDW outperforms the best baseline, i.e. TADW, by about 1% for the three datasets. Meanwhile, CTDW outperforms the remaining baselines more or less to some extent. These experiments demonstrate that CTDW is effective and robust.

(2) From the above tables, we find that content-based methods, i.e. Text, Community as well as Community and Text have good performances on the vertex classification. Meanwhile, a simple concatenation from Community and Text has better performance than separate content, i.e. Community or Text.

(3) Simple concatenation of representation vectors from the network structure, i.e. the matrix $M = (A + A^2)/2$ along with the community features and text features yields better improvements of classification accuracy than the content-based methods on all three datasets, showing the importance of both network structure and contents. But the performance of this kind of simple concatenation cannot do better than that of CTDW.

Table 3. Accuracy (%) of vertex classification on DBLP

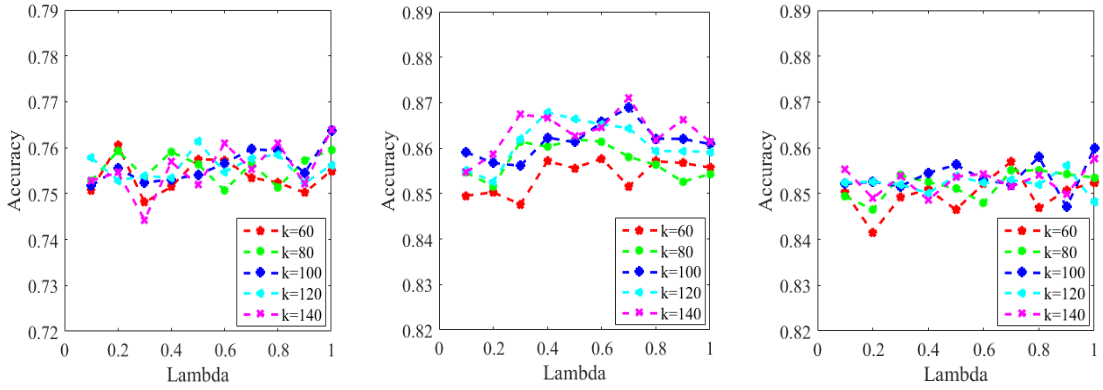
Training Ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%
node2vec	82.71	83.66	84.07	84.51	84.18	84.71	85.28	84.99	84.69
LINE	79.13	79.81	80.41	81.22	82.95	83.39	83.04	84.74	83.85
DeepWalk	81.53	81.61	83.07	83.78	83.76	84.32	84.42	85.13	84.17
MFDW	75.17	81.44	84.03	84.29	84.99	85.01	85.24	85.44	86.04
Text	60.69	68.15	70.62	72.76	73.79	73.46	74.37	74.16	74.85
Community	58.61	64.50	66.01	66.85	67.31	67.55	68.24	67.72	67.56
Community+Text	65.35	71.09	73.31	74.97	75.28	75.78	76.88	75.35	76.97
MV+Community+Text	76.22	80.10	81.78	82.75	82.94	83.76	84.53	84.93	85.16
CT@E	69.26	69.92	70.04	70.11	70.04	70.36	69.96	70.11	70.37
MMDW	79.70	82.05	84.23	84.84	83.45	85.42	84.96	85.78	84.49
TADW	79.09	81.43	82.42	82.94	83.50	84.40	84.91	85.26	85.72
CTDW	81.61	82.52	83.69	83.99	85.22	85.21	85.95	86.37	86.98

From these observations we find that CTDW generates high-quality representations, by incorporating community and text features into inductive matrix completion (IMC). Moreover, CTDW is not task-specific and the representations can be conveniently used for different tasks, such as link prediction, similarity computation and so on. The classification accuracy of CTDW is also competitive with several recent collective classification algorithms, although we don’t perform specific optimization for the tasks.

5.5 Parameter Sensitivity

CTDW has two hyper-parameters: dimension k and weight of regularization term λ. We fix training ratio to 60% and test classification accuracies with different k tand λ.

We let k vary from 60 to 140 and λ vary from 0. 1 to 1 for Citeseer, Cora and DBLP datasets. Fig.3 shows the variation of classification accuracies with different k and λ. The accuracies vary within 2%, 2. 4% and 1. 8% for fixed k on Citeseer, Cora and DBLP respectively. Therefore, CTDW can keep stable when k and λ vary within a reasonable range.

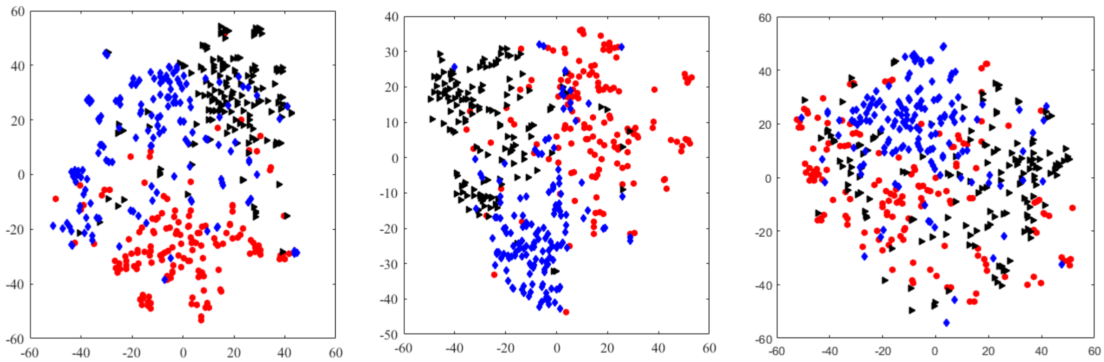


(a) Sensitivity on Citeseer (b) Sensitivity on Cora (c) Sensitivity on DBLP

Fig.3. Parameter sensitivity

5.6 Visualizations

In our research, we propose CTDW to learn network representations on Citeseer, Cora and DBLP datasets. To demonstrate whether the representations generated from CTDW show the discriminative classification ability or not, we randomly select three categories of networks, and each category contains 150 nodes. Fig.4 shows node clustering visualizations on Citeseer, Cora and DBLP.



(a) Visualization on Citeseer (b) Visualization on Cora (c) Visualization on DBLP

Fig.4. Clustering visualizations

As shown in Fig.4, we find that CTDW learns efficient representations with better clustering and representation ability. The representations on Citeseer and Cora datasets show strong clustering abilities, and the boundaries between the categories are clear and discriminative. However, the representation on DBLP dataset shows a relatively weaker clustering ability than Citeseer and Cora datasets. This reason is that there are more network links and more paths among communities in DBLP network, which leads to closer spatial distances among the vectors obtained from CTDW than Citeseer and Cora networks. In a word, the results of visualization demonstrate effectiveness of our method.

5.7 Case Study

Network representation learning aims to build a low-dimensional vector for each vertex in a network. To verify the performance of CTDW, we conduct an experiment on DBLP dataset. The document title is “distributional clustering of English words”, whose class label is “Artificial Intelligence”. As shown in Table 4, using representations generated by DeepWalk and CTDW, we find 5nearest documents of the above-mentioned selected document ranked by cosine similarity.

We find that all these documents are cited by the document “distributional clustering of English words” or some of these documents cite the document “distributional clustering of English words”. Since DeepWalk learns representations only based on network structures, 5 nearest documents by Deepwalk hardly contain relevant words with the selected document “distributional clustering of English words”, while 5 nearest documents by CTDW totally contain relevant words with the selected document “distributional clustering of English words”, such as “clustering”. This indicates that CTDW can learn better network representations with the help of community and text features than only network-based DeepWalk.

Table 4. Five nearest documents found by DeepWalk and CTDW

5 nearest documents by DeepWalk		
Title	Cosine Similarity	Label
a bootstrapping approach to unsupervised detection of cue phrase variants	0.7266	Artificial Intelligence
towards the automatic identification of adjectival scales clustering adjectives according to meaning	0.7137	Artificial Intelligence
similarity-based estimation of word cooccurrence probabilities	0.7002	Artificial Intelligence
statistical sense disambiguation with relatively small corpora using dictionary definitions	0.6999	Artificial Intelligence
the distributional inclusion hypotheses and lexical entailment	0.6908	Artificial Intelligence
5 nearest documents by CTDW		
Title	Cosine Similarity	Label
dimension induced clustering	0.8246	Artificial Intelligence
multiclass spectral clustering	0.8080	Artificial Intelligence
top-down induction of clustering trees	0.7978	Artificial Intelligence
refining initial points for k-means clustering	0.7829	Artificial Intelligence
generative model-based clustering of directional data	0.7634	Artificial Intelligence

6 Conclusion

In this paper, we propose community and text-enhanced DeepWalk (CTDW), which is a novel and discriminative network representation method to take network structure, community features and text features together into consideration based on inductive matrix completion. We conduct experiments with the tasks of vertex classification on three real-world datasets (Citeseer, Cora and DBLP). The experimental results show that CTDW is an effective and robust network representation method compared to other baseline methods. Meanwhile, the visualization results of learnt representations generated by CTDW demonstrate stronger discrimination ability. CTDW provides a normalized framework for joint learning with different types of resources via inductive matrix completion instead of simple concatenation towards these resources. For future work, we will extend our method to representation learning of large-scale networks. Meanwhile, we will explore some new technologies of matrix factorization, such as max-margin matrix factorization and matrix co-factorization.

Acknowledgements

The work is supported by the National Natural Science Foundation of China under grant 11661069, grant 61663041 and grant 61763041, the Program for Changjiang Scholars and Innovative Research Team in Universities under grant IRT_15R40, the Key Laboratory of Tibetan Intelligent Information Processing and Machine Translation in Qinghai Normal University, the Research Funds for Chunhui Program of Ministry of Education of China under grant Z2014022, the Natural Science Foundation of Qinghai Province under grant 2013-Z-Y17 and grant 2014-ZJ-721, the Fundamental Research Funds for the Central Universities under grant 2017TS045.

References

1. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. International Journal of Data Warehousing and Mining, 3(3), 1-13(2007).
2. Tu, C. C., Liu, Z. Y., Sun, M. S.: Inferring correspondences from multiple sources for microblog user tags. In: Chinese National Conference on Social Media Processing, pp. 1–12. Springer, Heidelberg (2014).
3. Yu, H. F., Jain, P., Kar, P., et al.: Large-scale multi-label learning with missing labels. In: Proceedings of ICML, pp. 593-601(2014).
4. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. Journal of the Association for Information Science and Technology, 58(7), 1019-1031(2007).
5. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, pp. 3111-3119(2013).
6. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701-710(2014).
7. Tang, J., Qu, M., Wang, M. Z., et al.: Line: large-scale information network embedding. In: Proceedings of WWW, pp. 1067-1077(2015).
8. Cao, S. S., Lu, W., Xu, Q. K.: GraRep: learning graph representations with global structural information. Conference on Information and Knowledge Management, pp. 891-900(2015).
9. Wang, D. X., Cui, P., Zhu, W. W.: Structural deep network embedding. The ACM SIGKDD International Conference, pp. 1225-1234(2016).
10. Sun, X. F., Guo, J., Ding, X., et al.: A general framework for content-enhanced network representation learning. arXiv:1610.02906, 2016.
11. Tu, C. C., Wang, H., Zeng, X. K., et al.: Community-enhanced network representation learning for network analysis. arXiv:1611.06645, 2016.
12. Pan, S. R., Wu, J., Zhu, X. Q., et al.: Tri-party deep network representation. In: Proceedings of IJCAI 2016, pp.1895-1901(2016).
13. Natarajan, N., Dhillon, I. S.: Inductive matrix completion for predicting gene-disease associations. Bioinformatics, 30(12), 60-68(2014).
14. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of ACM, pp. 50-57(2000).
15. Mei, Q. Z., Cai, D., Zhang, D., et al.: Topic modeling with network regularization. In: Proceedings of WWW, pp.101–110(2008).
16. Tu, C. C., Zhang W. C., Liu, Z. Y., et al.: Max-margin DeepWalk: discriminative learning of network representation. In: Proceedings of IJCAI, 2016.
17. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855-864(2016).

18. Wang, X., Cui, P., Wang, J., et al.: Community preserving network embedding. AAAI Conference on Artificial Intelligence, 2017.
19. Yang, C., Liu, Z. Y., Zhao, D. L., et al.: Network representation learning with rich text information. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
20. Krishna, K., Narasimha Murty, M.: Genetic K-means algorithm. IEEE Transactions on Systems, man, and cybernetics-part b: cybernetics, 29(3), 433-439(1999).