Effective Representing of Information Network by Variational Autoencoder

Hang Li and **Haozheng Wang**
College of Computer and Control Engineering, Nankai University, Tianjin, China
{hangl,hzwang}@mail.nankai.edu.cn

Zhenglu Yang*
College of Computer and Control Engineering, Nankai University, Tianjin, China
yangzl@nankai.edu.cn

Haochen Liu
Institute of Statistics, Nankai University, Tianjin, China
lhaochen@mail.nankai.edu.cn

Abstract

Network representation is the basis of many applications and of extensive interest in various fields, such as information retrieval, social network analysis, and recommendation systems. Most previous methods for network representation only consider the incomplete aspects of a problem, including link structure, node information, and partial integration. The present study proposes a deep network representation model that seamlessly integrates the text information and structure of a network. Our model captures highly non-linear relationships between nodes and complex features of a network by exploiting the variational autoencoder (VAE), which is a deep unsupervised generation algorithm. We also merge the representation learned with a paragraph vector model and that learned with the VAE to obtain the network representation that preserves both structure and text information. We conduct comprehensive empirical experiments on benchmark datasets and find our model performs better than state-of-the-art techniques by a large margin.

1 Introduction

Information network representation is an important research issue because it is the basis of many applications, such as document classification in citation networks, functional label prediction in protein-protein interaction networks, and potential friend recommendations in social networks. Although there are not a few recent work proposed to study the issue [Belkin and Niyogi, 2003; Tenenbaum *et al.*, 2001; Cao *et al.*, 2015; Tian *et al.*, 2014; Cao, 2016], it is still far from satisfactory because of the intrinsic difficulty. In essence, the rich and complex information (i.e., link structure and node contents) embedded in information networks poses a significant challenge in the effective representation of networks.

For an example on the scenario of paper classification, a reference relationship exists between papers, and each paper has its own title. Papers with similar titles are likely to be similar in terms of contents and topic, and papers tend to be close if they have a citation link. Through representation learning, the network can be mapped to a low-dimensional space for classification.

Network-distributed representation learning can be viewed as a problem using low-dimensional vectors to represent nodes in a network. Most network representation methods are based on a network structure. The traditional representation is based on matrix decomposition and uses eigenvectors as representation [Belkin and Niyogi, 2003; Roweis and Saul, 2000b; Tenenbaum *et al.*, 2001]. Furthermore, they extend to high-order information [Cao *et al.*, 2015]. However, these methods are not applicable to large-scale networks, and although many approximate approaches have been developed to solve this problem, they are not effective enough. Some methods are based on optimization objective functions [Tang *et al.*, 2015; Pan *et al.*, 2016; Yang *et al.*, 2015]. Although they are suitable for large-scale network data, they adopt shallow models that are limited in terms of performance and are difficult to use to obtain highly non-linear relationships that are vital to the preservation of network structure. Inspired by deep learning techniques in natural language processing, [Perozzi *et al.*, 2014; Grover and Leskovec, 2016] adopted several stunted random walks in networks to generate node sequences serving as sentence corpus and then applied the skip-gram model to these sequences to learn node representation. However, they cannot easily handle additional information during random walks in a network.

To capture highly non-linear structures for large-scale networks, [Tian *et al.*, 2014; Cao, 2016] introduced an autoencoder to model training instead of using a sampling based method to generate linear sequences. Motivated by this model, we develop the variational autoencoder (VAE) [Kingma and Welling, 2014], which is a deep generation model, instead of a basic autoencoder. Most previous studies utilized only one type of information in networks. The work in [Le and Mikolov, 2014] focused on node content, and others [Grover and Leskovec, 2016; Perozzi *et al.*, 2014] explored link structure. Although a few previous models [Pan *et al.*, 2016; Yang *et al.*, 2015] combined both text information and network structure, they did not preserve the complete network structure and only partially utilized node content. A straightforward method is to learn a node vector by using DeepWalk for network structure and learn a text vector via Paragraph Vectors model [Le and Mikolov, 2014] (the state-

*Corresponding author.

of-the art model to embed text into a vector space), and then concatenate these two vectors into a unified representation. However this simple combination is suboptimal.

To address the above issues, we propose a deep generative model to learn network representation by modeling both node content information and network structure comprehensively. First, we obtain representation based on node content through the paragraph vector model. Then, we feed the network adjacency matrix and representation obtained into a deep generative model, the building block of which is the VAE. After stacking several layers of the VAE, we choose the result of the first layer before decoding as the final representation. Intuitively, we can obtain the representation containing both content information and structure in a d-dimensional feature space. The experimental evaluation demonstrates the superior performance of our model on the benchmark datasets.

Overall, our paper makes the following contributions:

- We propose a model to learn social information network representation. The model can obtain highly non-linear relationships in network structures. To the best of our knowledge, this work is among the first to use generative models to learn network representations.
- We explore the application of the VAE as an unsupervised architecture, which simultaneously integrates node content and network structure.
- We extensively evaluate the proposed method and demonstrate that our model is significantly better than state-of-the-art techniques.

2 Related Work

Distributed representation learning is the basis of many applications, and it has been studied extensively in recent years. The aim of representation learning is to seek a low-dimensional vector to describe an objective. Our goal is to extract enough features in a social information network.

Traditional dimensionality reduction techniques [Belkin and Niyogi, 2001; Roweis and Saul, 2000a] typically construct the affinity network using the feature vectors of the vertexes and then compute the eigenvectors of the affinity network. Network representation was first proposed by [Hoff *et al.*, 2002], and it was later followed by numerous methods [Belkin and Niyogi, 2003; Roweis and Saul, 2000b; Tenenbaum *et al.*, 2001] based on matrix factorization, which treats eigenvectors as representations. Furthermore, they extend to high-order information [Cao *et al.*, 2015], while they adopt shallow models and the complex structure can not be capture. [Tian *et al.*, 2014; Cao, 2016] introduced basic autoencoder to extract complex features and model non-linear structure of the network. SDNE [Wang *et al.*, 2016] is a semi-supervised deep model that captures the non-linear structural information over the network. The source code of SDNE is not available, so this method cannot be reproduced and compared to ours. Recently, DeepWalk [Perozzi *et al.*, 2014] exploit the classical model with natural language processing to learn network representation, that can effectively solve the sparse problem of data. DeepWalk adopt random walk to generate the standard input sequence at first, and then used Skip-Gram [Mikolov *et al.*, 2013] to model the network. Inspired by DeepWalk, Walklets [Perozzi *et al.*, 2016] focus on multi-scale vertex relationships, Node2vec [Grover and Leskovec, 2016] design a biased random walk procedure which efficiently explores diverse neighborhoods, LINE [Tang *et al.*, 2015] optimizes a designed objective function that preserves first-order and second-order proximity structures. However, all the above method learning network representation are structure-based, they ignore the node content information, which is a significant feature of the network.

Early text representation models, such as bag-of-words approaches (e.g., TF-IDF) are designed for discrete forms of words, and they do not consider the context of words. Word2vec [Mikolov *et al.*, 2013] is an effective method that projects a word into a continuous vector space based on a statistical language model and it has become the basis for most natural language processing tasks. Mikolov explored similar methods to learn document representation(e.g. Doc2vec [Le and Mikolov, 2014]), many of which are carried out on Word2vec. [Socher *et al.*, 2013; Socher, 2013] applied word vectors to recursive neural networks to describe sentences. Convolutional neural networks(CNN) [Kim, 2014] and LSTM-based [Kiros *et al.*, 2015] methods were developed to capture text information. However, in terms of social information network, the text among networks cannot represent all information, and the structure is significant.

As the first to consider both node structure and text information, TADW [Yang *et al.*, 2015] find out that DeepWalk is equivalent to matrix factorization so they incorporated the text features of vertices into network representation learning under the framework of matrix factorization. TriDNR [Pan *et al.*, 2016] used information from node structures, node text, and node labels to jointly learn network representation; their model combines DeepWalk and Doc2vec. However, these models are both shallow and can not capture highly non-linear structures. Moreover, the content information they use is not comprehensive (e.g., the abstract of a paper in a citation network).

3 Preliminary

Notation: Let $G = (V, E, C, L)$ denote a given network, where $V = \{v_i\}_{i=1...N}$ is the node set and $E = \{e_{ij}\}$ is the edge set that indicates the relation of nodes. If a direct link exists between v_i and v_j then $e_{ij} = 1$; otherwise, $e_{ij} = 0$ when network is unweighted. $C = \{c_i\}$ is the set of content information; c_i can be regarded as a word. $L = \{l_i\}$ is the set of class labels. let A denote the adjacency matrix for a network, and let $x = \{e_{i,k}, ..., e_{n,k}\}$ be an adjacency vector. Our goal is to seek a low-dimensional vector $\vec{u}_j$ for each node v_i of a given network.

Autoencoder: We first provide a brief description of a basic autoencoder and the VAE. The basic autoencoder first compresses the input into a small form and then transforms it back into an approximation of the input. The encoding part aims to find the compression representation z of a given data x, and the decoding part is a reflection of the encoder used to reconstruct the original input x. The VAE [Kingma and Welling, 2014] imposes a prior distribution on the hidden layer vector

of the autoencoder and re-parameterizes the network according to the parameters of the prior distribution. Through the parameterization process, the means and variance values of the input data can be learned. We extended VAE to generate two means and variances of input data, which can be considered correspond to the content and structure respectively.

4 Model Description

The architecture of the proposed model is shown in Fig. 1. The whole architecture consists of two main modules, namely, the content2vec module and the union training module. For an information network, such as a paper citation network, we can obtain the node link and content information (e.g., paper abstract). We learn an effective feature representation vector that preserves both structure information and node content information and can thus be applied to many tasks (e.g., paper classification).

4.1 Content2vec Module

We employ the state-of-the-art approach called doc2vec [Le and Mikolov, 2014], which utilizes text information to learn vector representations for each documents, as our content2vec module. Specifically, if one node contains other information (e.g., author name), we treat it as a word and merge it into the comprehensive text information (e.g., the abstract of the paper in the citation network) as the content of the node. A representation u_i that includes the node content information is obtained from this module. Therefore, we can maximize the following objective:

$$\mathcal{O} = \sum_{i=1}^{N} \log P(w_{-b} : w_b | v_i) \tag{1}$$

where w is a word in the text information of node v_i, b is the window size of word sequence. After optimizing this objective, we can obtain the representation u_i for v_i.

4.2 Union-training Module

The union training module is the core part of our model, in which content information and structure information are integrated. The details are shown in Fig 1. The VAE is adopted as the main block. Given a network, the adjacency matrix A can be obtained. A is able to describe the relationship among the nodes and reflect the overall structure of the network. We extract each adjacency vector a_i and concatenate it with the corresponding u_i as the input x_i of our model. Therefore, the content and structure information is able to be learned simultaneously.

During the encoding phase, we adapt several fully connected layers composed of multiple nonlinear mapping functions to map the input data to a highly nonlinear latent space. Therefore, given the input x_i, the output h^k for the k^{th} layer is shown as follow:

$$\begin{aligned} h^1 &= \pi(W^1 x_i + b^1) \\ h^k &= \pi(W^k h^{k-1} + b^k), k = 1, 2...K \end{aligned} \tag{2}$$

where π is the nonlinear activation function of each layer. The value of K varies with the data.

In the last layer of encoder, we obtain four output: μ_{i1}, σ_{i1} μ_{i2} and σ_{i2}. They can be treated as the means and variances of the distribution of content information and structure information respectively. Furthermore, we sample two values ε_{i1} and ε_{i2} from two previous distributions (e.g., Gaussian distribution). Then we can obtain the re-parameterized z_i1 and z_21. Through concatenate z_i1 and z_21, content and structure information can be integrated together, y_i is the representation of the network. Nonlinear operations are not performed in this phase. Thus, the gradient descent method can be safely applied in optimization. The operations can be expressed as follows:

$$\begin{aligned} z_{ik} &= f(\mu_{ik}, \sigma_{ik}, \varepsilon_i), k = 1, 2 \\ y_i &= Merge[z_{i1}, z_{i2}] \end{aligned} \tag{3}$$

where f is a linear function that can re-parameterize y_i, $Merge$ concatenate the two vectors together directly.

The joint-input x_i of our model includes the content information u_i and structure information a_i. We consider that separating the two kinds of latent representations is necessary because each of them has its own meaning and individual effect.

Algorithm 1 Training Algorithm for Our Model

Input: Adjacency matrix A ,content information C
Output: Network representations Y and parameters W, B

```
Train a paragraph vector model based on C, obtain the representations U
X = Merge[A, U]
repeated:
Feed x_i into encoder, obtain σ_i1, σ_i2, μ_i1, μ_i2
Sample ε_i1 and ε_i2 from two Gaussian distributions
z_i1 = f(σ_i1, μ_i1, ε_i1), z_i2 = f(σ_i2, μ_i2, ε_i2)
y_i = Merge[z_i1, z_i2]
Decode y_i to obtain x̂_i
Based on Eq. 4 update W, B
until convergence
Obtain the network representations Y = {y_i}
```

The decoding phase is a reflection of the encoder; its output $\hat{x_i}$ should be close to the input x_i. The loss function of this module that should be minimized is as follows:

$$\mathcal{L}(x_i) = -\sum_{k=1}^{2} \mathcal{KL}(q(z_{ik}|x_i)||p(z_{ik})) + \mathcal{H}(x_i, \hat{x_i}) \tag{4}$$

where KL is the KL divergence which is always used as a measure of the distinction between two distributions, $\mathcal{H}$ is a cross-entropy function that is applied to measure the difference between x_i and $\hat{x_i}$. Finally, We choose the output of the layer y_i as the final representation of each node.

The full algorithm is presented in Algorithm 1.

4.3 Analysis and Discussion

In terms of the content2vec module, one intuitive approach to optimize Eq.(1) is to use the stochastic gradient ascent, although computing the gradient is expensive. To solve this problem, we apply the hierarchical soft-max

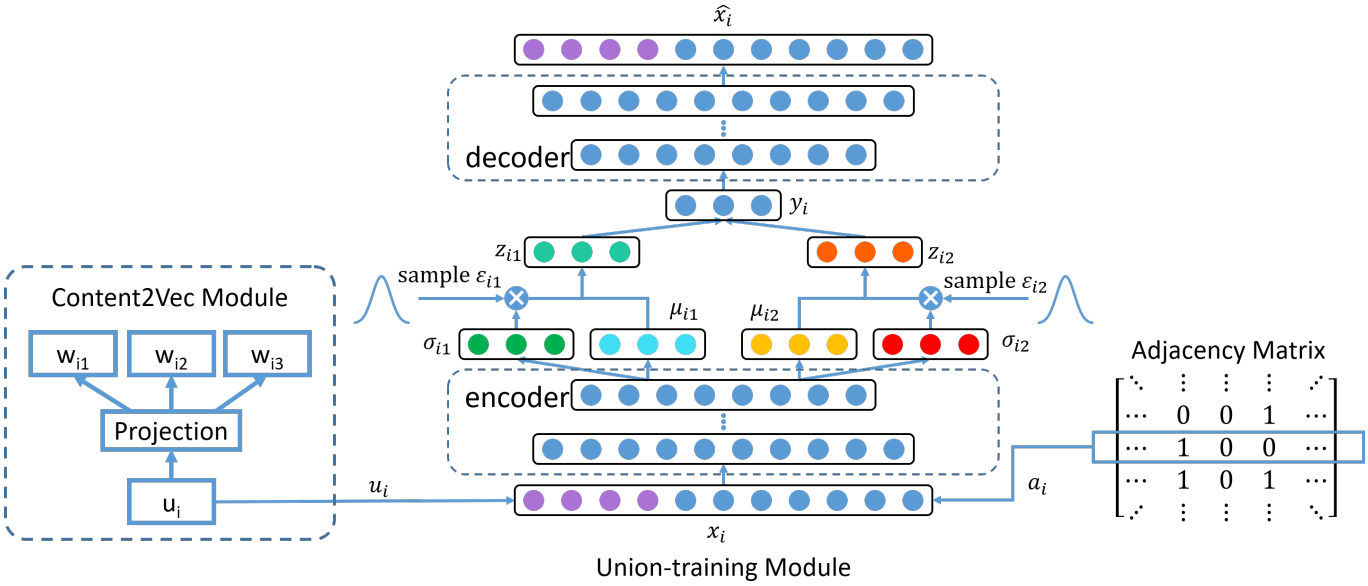

Figure 1: Architecture of our model. $\mathbf{w}_i$ can be viewed as a word of the content information, $\mathbf{v}_i$ is a node in the network, u_i is a representation vector learned by the Content2Vec Module, x_i is a vector of the adjacency matrix. The input of the union-training module is combination of x_i and u_i, the encoder and decoder are stack full-connected layer, $\sigma_{i1}, \sigma_{i2}, \mu_{i1}, \mu_{i2}$ can be seen the mean and variance of the distribution of the content and structure data, respectively. ε_{i1} and ε_{i2} are the sample data from two Gaussian distributions.

method [Mikolov *et al.*, 2013], which reduces the time complexity to $O(R\log(W) + N\log(N))$, where R is the total number of words in the content. For the union training module, we can easily infer that the VAE used in our model involves a low time complexity following previous analysis [Tian *et al.*, 2014]. Time complexity is related to the number of nodes n, degree network layer k, and number of iterations I; thus, it is approximate to $O(nkI)$.

One of the advantages of our model is that when handling new nodes, we do not need to re-train the union training module. When we obtain the adjacency vector x, we can feed it into our model and obtain the representation at a complexity of O(1). If no link exists between the new node and the network, we can exploit its content information.

5 Experiments

5.1 Datasets

Paper citation networks is a classical social information network. To evaluate the quality of the proposed model, we conduct three important tasks on two benchmark citation network datasets:(1) CiteseerM10[1]. It contains 10 distinct categories with 10,310 papers and 77,218 citations. Titles are treated as the text information because no more text information is available. (2) DBLP dataset[2]. We treat abstracts as text information and choose 4 research areas with the same setting as that of [Pan *et al.*, 2016], which are database (SIGMOD, ICDE, VLDB, EDBT, PODS, ICDT, DASFAA, SSDBM, CIKM), data mining (KDD, ICDM, SDM, PKDD, PAKDD), artificial intelligent (IJCAI, AAAI, NIPS, ICML, ECML, ACML, IJCNN, UAI, ECAI, COLT, ACL, KR), computer vision (CVPR, ICCV, ECCV, ACCV, MM, ICPR, ICIP, ICME). Therefore we get a network contains 30,422 nodes and 41,206 edges.

5.2 Performance on Node Classification

In terms of node classification task, we compare our approach [3] with the following methods:

- **One-Hot** uses adjacency matrix, which carries the structure information as the high-dimension representation, and directly feed into the classifier.
- **DeepWalk** [Perozzi *et al.*, 2014] is exploited by statistical models, which employs truncated random walks to learns nodes embedding by treating walk as the equivalent of sentences.
- **Node2vec** [Grover and Leskovec, 2016] learns the network representation by designing a biased random walk procedure which efficiently explores diverse neighborhoods.
- **Doc2vec** [Le and Mikolov, 2014] is the Paragraph Vector model that learns document representation by predicting the words appeared.
- **DW+D2V** is simply to concatenate the representation result learned by DeepWalk and Doc2vec.
- **TADW** [Yang *et al.*, 2015] is text-based DeepWalk, which incorporates text information into network structure by matrix factorization.

[1] http://citeseerx.ist.psu.edu/

[2] http://arnetminer.org/citation (V4 version is used)

[3] https://github.com/Algorithm216/RIN/

Table 1: Macro-F1 score on Citeseer-M10 Network

%p	One-Hot	Deepwalk	Node2vec	Doc2vec	DW+D2V	TADW	TriDNR	Ours
10%	0.254	0.297	0.314	0.503	0.526	0.475	0.683	**0.889**
30%	0.321	0.334	0.331	0.536	0.615	0.488	0.744	**0.913**
50%	0.352	0.346	0.346	0.547	0.633	0.495	0.760	**0.924**
70%	0.363	0.344	0.339	0.534	0.630	0.495	0.773	**0.940**

Table 2: Macro-F1 score on DBLP Network

%p	One-Hot	Deepwalk	Node2vec	Doc2vec	DW+D2V	TADW	TriDNR	Ours
10%	0.328	0.379	0.448	0.574	0.495	0.660	0.724	**0.751**
30%	0.362	0.454	0.473	0.598	0.586	0.687	0.742	**0.753**
50%	0.371	0.459	0.475	0.604	0.614	0.697	0.747	**0.762**
70%	0.372	0.461	0.476	0.605	0.628	0.699	0.748	**0.763**

- **TriDNR** [Pan *et al.*, 2016] uses node text, label, and structure to jointly learn node representation.

We conduct the paper classification task on two benchmark citation networks to evaluate the performance of our method. To reduce the influence of the differences between classifiers, a common linear SVM is employed by all the methods. The results are shown in Table 1 and Table 2, respectively. The reported parameters for our model are set: dimension d=100 on CiteseerM10 and d=300 on DBLP. The dimension for other algorithms is the same as ours, and the other parameters are set as their papers report, i.e., window size b=10 in DeepWalk and Node2vec, in-out parameter q=2 in Node2vec, text weight ∂=0.8 in TADW and TriDNR.

For the classification task, we use Macro-F1 which is the same as that adopted by other algorithms to measure the classification performance.

The experiments are independently conducted 10 times for each setting, and the average values are reported. The proportion of training data with labels is range from 10% to 70%.

Our model is evaluated by comparing it with seven approaches. One-Hot uses the original structure data, and its performance is poor because it is discrete and the context relation of nodes can not be captured. DeepWalk and Node2vec are structure-based methods that exhibit inferior performance mainly because they only use the shallow structure information and the network is rather sparse, while the information of the complex non-linear structure cannot be employed. The performance of Doc2vec is not as good as ours which demonstrates the effectiveness of our proposed model. TADW and TriDNR are inferior to our approach, although these two methods also consider the text and structure. Nevertheless, they cannot capture the complex non-linear structure.

In Table 1(Citeseer-M10), the improvement margin of our method over the baselines is more obvious when the training percentage range from 10% to 70%. It demonstrates that our method can achieve a more significant improvement than baselines when the labelled data is limited. Such an advantage is especially important for real-world applications because the labelled data is usually scarce.

Our model exhibits consistent superior performance, and is up to 16% better than the state-of-the-art methods (i.e., the Macro-F1 score of our model is 94% when the proportion of training data with labels is 70% conducted on the Citeseer-M10 Network dataset).

5.3 Parameter Setting

A significant hyperparameter in our model is the dimension d. The performance of different methods with varying dimensions has been evaluated. The result is illustrated in Fig.2. We obtain very good performance on the Citeseer-M10 dataset, i.e., the Macro-F1 score is 94% and the performance tends to be stable as b becomes larger. It validates the effectiveness of our algorithm and the reason is due to the ability of our model that can capture the complex network structure and the text information. From Fig.2, we can see that the performance gets better when d increases from 100 to 600. Intuitively, the main reason is that more information can be preserved in higher dimensional space of the datasets.

5.4 Case Study

To demonstrate the advantage of our method intuitively, we present an example in Citeseer-M10 dataset. With regard to a given query paper, we use the vector representation generated by different algorithms to determine the five most similar papers. The value of the cosine similarity between vectors is adopted as the metric.

Table 3 illustrates the concrete example. A query is conducted on three representative methods, DeepWalk, Doc2Vec and TriDNR, which represents the structure-based methods, the content-based methods and the combined methods, respectively.

The title of the target paper is "Redistribution of Slurry Components as Influenced by Injection Method, Soil, and Slurry Properties". Its class label is "agriculture". The paper focuses on the impact of livestocks slurry on soil. It is published on Journal of Environmental Quality. As shown in Table 3, the class value 0 indicates "agriculture" and 1 indicates "archeology".

We observe that the five papers obtained by our method exactly have the same class label as the target paper. However, for the other three methods, they all have 2 out of the 5 papers that do not match with the query paper. Those papers focus on archeology instead of agriculture.

Doc2vec merely use the text information to learn the node representation, and it tends to simply consider that papers

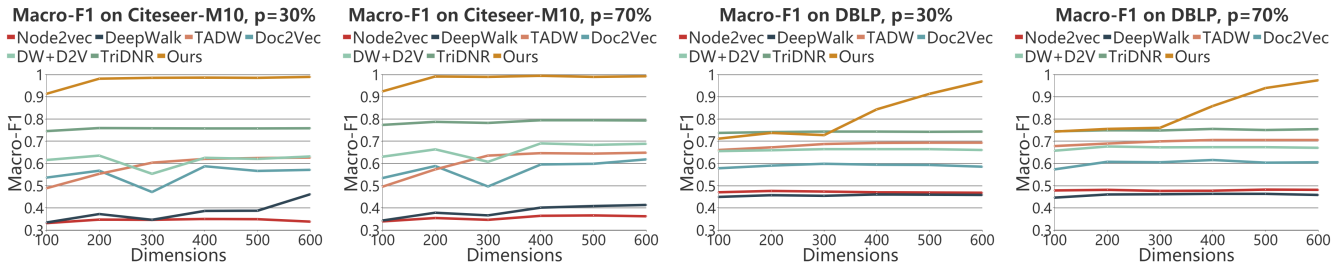

Figure 2: Performance of each strategy on different training proportion p

Table 3: Top-5 Similar Node Search, class 0 and 1 represent the category agriculture and archaeology respectively

Paper	Class
Query: Redistribution of Slurry Components as Influenced by Injection Method, Soil, and Slurry Properties	0
Our Mode:	
1.KLUM: A Simple Model of Global Agricultural Land Use as A Coupling tool of Economy and Vegetation	0
2.Harvests and Business Cycles in Nineteenth-Century America Joseph H. Davis, Vanguard Group	
3.Faculty of Civil Eng. and Geodetic Sci.	0
4.Acidic Deposition, Ecosystem Processes, And Nitrogen Saturation In A High Elevation Southern Appalachian Watershed	0
5.Trends and Contributing Factors Enhanced Output Traits and Market Coordination Issue	0
Deepwalk:	
1.Symposium no.16 Paper no.2231 Presentation: poster 2231-1	0
2.Mesolithic Hunter-Gatherers in the Northwestern Part of the Great Hungarian Plain	1
3.An FAO Type Crop Factor Modification to SWB for Inclusion of Crops with Limited Data: Examples for Vegetable Crops	0
4.Discrepancies in the Radiocarbon Dating Area of the Turin Shroud	1
5.Use Caution When Harvesting and Feeding Ditch Hay	0
Doc2vec:	
1.Laser Scanner Survey of An Archaeological Site-Scala Di Furno (Lecce, Italy)	1
2.Faculty of Civil Eng. and Geodetic Sci.	0
3.Description of Map Units	1
4.Heavy Metals Transport in the Soil Profiles under the Application of Sludge and Wastewater	0
5.Vegetated Treatment of Vehicle Wash Sediments: Mathematical Modeling of Groundwater and Solute Transport	0
TriDNR:	
1.Printed in U.S.A Rapid Water Flow and Transport of Inorganic and Organic Nitrogen in A Highly Aggregated Tropical Soil	0
2.Water Quality in Drainage Ditches Influenced by Agricultural Subsurface Drainage	0
3.Description of Map Units	1
4.By	1
5.New Constraints on Northern Hemisphere Growing Season Net Flux	0

which have similar titles are more proximate. The paper "Laser Scanner Survey of An Archaeological Site - Scala Di Furno (Lecce, Italy)" found by Doc2vec is obviously an archaeological survey, which is not related to the query paper. Doc2vec may extract the incorrect papers because of the correlation between the titles of the papers.

Deepwalk measures the similarity between papers mainly depending on their citation relationship. The paper "Discrepancies in the Radiocarbon Dating Area of the Turin Shroud" found by Deepwalk is also an archaeological one. Since the query paper is widely cited among several research areas such as chemistry and biology, it may have some citation relationship with this wrong paper. Deepwalk deems it as a similar paper to the query paper by mistake.

For TriDNR, it performs well for most of the tasks because it integrates both the text and the structure information. However, it still perform poor in some specific cases. For instance, the inappropriate paper "By" found by it has a partially missing title. Since it has a weakness that it only utilizes the shallow structure information of nodes and it relies heavily on the text information. On the contrary, our method extracts the highly non-linear relationship between the nodes by exploiting Variational Autoencoder so that we achieve good results.

6 Conclusions

In this paper, we have introduced an effective network representation model, which comprehensively integrates the text information and the network structure. We introduced Paragraph Model as a preliminary module. Furthermore, we have exploited Variational Autoencoder as the main block of our model, that could capture highly non-linear structure of the network. The comprehensive experimental evaluation on two benchmark datasets has demonstrated the effectiveness of our model.

Acknowledgements

This research is supported by the National Natural Science Foundation of China: U1636116, 11431006, Research Fund for International Young Scientists: 61650110510, Ministry of Education of Humanities and Social Science: 16YJC790123.

References

[Belkin and Niyogi, 2001] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for

embedding and clustering. In *International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 585–591, 2001.

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep:learning graph representations with global structural information. pages 891–900, 2015.

[Cao, 2016] Shaosheng Cao. deep neural network for learning graph representations. In *AAAI*, 2016.

[Grover and Leskovec, 2016] A Grover and J Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, 2016.

[Hoff *et al.*, 2002] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *Eprint Arxiv*, 2014.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2014.

[Kiros *et al.*, 2015] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *Computer Science*, 2015.

[Le and Mikolov, 2014] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, 2013.

[Pan *et al.*, 2016] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. In *IJCAI*, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, 2014.

[Perozzi *et al.*, 2016] Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. Walklets: Multiscale graph embeddings for interpretable network classification. 2016.

[Roweis and Saul, 2000a] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, 2000.

[Roweis and Saul, 2000b] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Socher *et al.*, 2013] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.

[Socher, 2013] Richard Socher. Grounded compositional semantics for finding and describing images with sentences. *Nlp.stanford.edu*, 2013.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, 2015.

[Tenenbaum *et al.*, 2001] Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2001.

[Tian *et al.*, 2014] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Y. Liu. Learning deep representations for graph clustering. *Inproceedings*, 2014.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *SIGKDD*, pages 1225–1234, 2016.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, 2015.