Integrity and Robust Network Embedding of Information Network with AAE

Bin LIU, Yun-fang CHEN and Wei ZHANG*

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

*Corresponding author

Keywords: Network embedding, Joint information, Adversarial learning.

Abstract. Most existing network representation methods only consider network structure information or node content information independently, not making full use of them. Furthermore, noise that exists in real networks also causes the deterioration of the efficiency of traditional methods. In this paper, we propose IANE (Integrity Adversarial Network Embedding) framework, adversarial learning to regularize the representation learning in Adversarial AutoEncoders (AAE). IANE includes an information joint module and an adversarial learning module. The former aims to obtain complete input information, while the latter captures the highly non-linear structure of the network in the process of dimension reduction. Besides IANE has better robustness against noise by matching the posterior distribution of the embedding vectors to the given prior distribution. Our experiment results show that our method performs higher Macro-F1 than several baselines proposed in recent years.

Introduction

Network is a ubiquitous structure and can be found in various fields, such as social network, protein-protein interaction networks. Real-world information networks consist of massive nodes, and the rich information in the networks is important for us to understand the networks and discover the hidden networks laws. The content information of the node includes the title or abstract of the article, which exists in the form of word text. The structure information includes the reference relationship between the articles, which represented by an adjacency matrix. Effective representation of two kinds of information is crucial.

Our works are mainly concerned with how to use the two types of information together effectively. Although there has been a lot of work on this subject [1-3], learning network representation still faces many challenges as follows: (1) High non-linearity: As [4], because a node may have multiple direct predecessors and successors, the underlying structure of the network is highly non-linear. (2) Full use of reconstructed information: We found that the nodes not only have structural information, but also rich semantic (text) information. (3) Insufficient robustness: The existing representation methods lack the consideration of robustness. In real noisy network, the embedded expression technologies often fail to maintain stable performance.

The main contributions of our method are as follows:

(1) Learn feature representations from the content information and the network structure independently, and then merge two separate representation vectors to make full use of heterogeneous information.

(2) With the encoding part of the AAE [5], map data into a highly nonlinear space by multi-layer non-linear function and treat them as a reduced dimensional block to capture highly nonlinear structures.

(3) Realize the function of inverse mapping of traditional GAN [6] in the adversarial learning process for robust representation.
Integral Adversarial Network Embedding

Problem Definition and Notations

For an information network (i.e., a paper citation network), we can obtain information about links between nodes and node content (i.e., abstracts, titles, and author names). Our goal is to learn an effective feature representation vector that preserves complete structural information and node content information so that it can be applied to many tasks (i.e., text classification).

Let \( G = (V, E, C) \) denote a given network, where \( V = \{v_i\}_{i=1,2,\ldots,N} \) 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 \( v_j \), \( e_{ij} = 1 \), otherwise, \( e_{ij} = 0 \) when network is unweighted. \( C = \{c_i\} \) is the set of content information. Let \( A \) denote the adjacency matrix for a network, and let \( a_i = \{e_{i,1}, \ldots, e_{i,n}\} \) be an adjacency vector. Combine node content information \( c_i \) and node structure information \( a_i \) into \( x_i \in X \) as input. Our goal is to map \( x_i \in X \) to the vector \( z_i \in \mathbb{R}^d \) in low-dimensional space for each node \( v_i \) of a given network. The format is as follows: \( f: V \rightarrow Z \), where \( z_i \) is the \( i \)th row of \( Z \) (\( Z \in \mathbb{R}^{N \times d} \), \( N = |V| \)), \( d \) is the dimension of representations, \( Z \) is the representation matrix.

Union-training Module

In Figure 1, we clearly show the modular structure of our model, and we will explain the structure of each module and design details.

![Figure 1. An Overview of the IANE Framework.](image)

The part of Information extraction in Figure 1 contains content vector and structure vector. Content vector \( c_i \) can be obtained from nodes text information coding. The original model that encodes text information is the Bag-of-Words(BOW). However, the BOW has significant drawbacks: First, it loses the sequence relationship of words, Second, it ignores the semantic relationship between words. Moreover, high-dimensional vectors will lead to data sparsity problem. We exploit an advanced method called doc2vec [7] for paragraph vectors. Doc2vec can learn the fixed-length representation vectors from the texts of indefinite length, and obtain the similarly semantic contents that are close in low-latitude space. Finally, we can obtain \( c_i \) by doc2vec.

Structure vector \( a_i \) comes from adjacency matrix which describes the mutual reference relationship of nodes. We merge the two types of vectors into \( x_i \) as follow:

\[
    x_i = \text{Concatenate}[c_i, a_i] \quad (1)
\]

The part of Adversarial learning in Figure 1 is the core of IANE. We exploit AAE for dimensionality reduction and regularization. The key to AAE is the adversarial network that consists of the generator \( G \) acted as encoder and the discriminator \( D \), here \( G \) and \( D \) are
associated with $Z^{(k)}$. We treat $Z^{(k)}$ as fake sample and Gaussian distribution as real sample, and the discriminator $D$ will distinguish whether the input is from fake sample or real sample after training.

The advantages of using AAE as a joint module in the embedding space are as follows:

1. The encoder will be an efficient generator if adversarial training happens in embedding space, while general generators are usually weak.
2. AutoEncoder can extract feature while GAN play a role in noise reduction.

During the encoding phase, we adapt several fully connected layers composed of multiple non-linear mapping functions to map the input data to a highly nonlinear latent space.

Therefore, given the input $x_i$, the output $z_i^{(k)}$ for the $k^{th}$ layer is shown as follows:

$$z_i^{(1)} = \sigma(W^{(1)}x_i + b^{(1)})$$

$$z_i^{(k)} = \sigma(W^{(k)}z_i^{(k-1)} + b^{(k)}), k=2, \ldots, K$$

where $\sigma$ is the non-linear activation function of each layer, the encoder is a three-layer neural networks with tanh activations. The value of $K$ varies with the data. After obtaining $z_i^{(k)}$ at the last layer of the encoder, we can get the output $\hat{x}_i$ by inverting the encoding calculation process, so the goal of the Autoencoder is to minimize the reconstruction error. The loss function is defined as follow:

$$L_R(x_i, \hat{x}_i) = \sum_{i=1}^{m} ||\hat{x}_i - x_i||^2$$

(3)

Since the encoder layer of Autoencoder also acts as the generator of Adversarial network simultaneously, so $z_i^{(k)}$ from (2) formula still plays the role of fake samples during the adversarial learning. Then, a prior distribution $p(z)$ (Gaussian distribution) is selected as the data distribution for generating real samples. In the training process, the discriminator is trained to tell apart the prior samples from the embedding vectors, while the generator is aimed to fit embedding vectors to the prior distribution. In the experiments, the discriminator loss function for each mini-batch is as follows:

$$L_D = -\frac{1}{m} \sum_{i=1}^{m} \log(D(z_i^{(k)})) + \log(1 - D(z_i^{(k)}))$$

(4)

We disguise its output as prior data to better confuse the discriminator, the generator is trained to improve the following payoff:

$$L_G = \frac{1}{m} \sum_{i=1}^{m} \log\left(\frac{1}{m} \sum_{i=1}^{m} \log\left(D\left(z_i^{(k)}\right)\right)\right)$$

(5)

The decoding stage is the reconstruction of the encoding stage and the output should be as close as possible to the input, loss function consists of three parts: (3), (4), and (5).

**Experiment**

The label information can represent the attributes, interests, or other characteristics of the node which can play a catalytic role in many applications, such as friend recommendation or accurate advertising. However, only some nodes in a real social network are labeled. Therefore, node classification helps to mine information of unlabeled nodes. The comparison of the classification results is intuitively shown in Table 1 and Table 2. A significant hyperparameter is the embedding dimension $d$. Here we use the results when $d$ is 200. $p$ denotes the proportion of the training set used when training the SVM classifier. Tables 1 and 2 compare the effect of our model and One-Hot, DeepWalk[1], Doc2vec[7], and CANE[8] on CiteSeerM10 and DBLP when $p$ is 10%, 30%, 50%, and 70% respectively.
Table 1. Macro-F1 score on CiteSeerM10 Network.

| Method   | 10  | 30  | 40  | 50  |
|----------|-----|-----|-----|-----|
| onehot   | 0.244 | 0.301 | 0.352 | 0.359 |
| DeepWalk | 0.297 | 0.335 | 0.346 | 0.361 |
| doc2vec  | 0.503 | 0.542 | 0.547 | 0.573 |
| CANE     | 0.514 | 0.572 | 0.608 | 0.613 |
| IANE     | 0.626 | 0.658 | 0.675 | 0.695 |

Table 2. Macro-F1 score on DBLP Network.

| Method   | 10  | 30  | 40  | 50  |
|----------|-----|-----|-----|-----|
| onehot   | 0.328 | 0.362 | 0.371 | 0.372 |
| DeepWalk | 0.379 | 0.454 | 0.459 | 0.461 |
| doc2vec  | 0.574 | 0.598 | 0.604 | 0.605 |
| CANE     | 0.583 | 0.602 | 0.634 | 0.653 |
| IANE     | 0.731 | 0.755 | 0.768 | 0.793 |

The performance of different methods with varying dimensions has been evaluated. In Figure 2, the performance of different models with dimensions from 100 to 500 when $p$ is 30% and 70% is compared on two datasets. We find that with the increase of the dimensions, the effect of classification has slightly increased. We think it may be that higher dimensions can capture more information.

Figure 2. Performance of each strategy on different training proportion $p$.

**Conclusion**

In this paper, we propose an effective network representation learning method, which focuses on the use of rich heterogeneous information in the network and then obtains a certain level of robustness against noise by adversarial training. Furthermore, we demonstrate the generalization of IANE with the two different prior distributions.
Reference

[1] Perozzi, B., Al-Rfou, R., & Skiena, S.: Deepwalk: Online learning of social representations. In: 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701-710. ACM, New York (2014).

[2] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q.: Line: Large-scale information network embedding. In: 24th International Conference on World Wide Web, pp. 1067-1077. International World Wide Web Conferences Steering Committee (2015).

[3] Grover, A., & Leskovec, J.: node2vec: Scalable feature learning for networks. In: 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855-864. ACM, California (2016).

[4] Luo, D., Nie, F., Huang, H., and Ding, C. H.: Cauchy graph embedding. In: 28th International Conference on Machine Learning, pp 553–560. ICML, Washington (2011).

[5] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644, (2015).

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672-2680. MIT press, (2014).

[7] Le, Q., & Mikolov, T.: Distributed representations of sentences and documents. In: In International Conference on Machine Learning, pp. 1188-1196. BeiJing China (2014).

[8] Tu, C., Liu, H., Liu, Z., & Sun, M.: Cane: Context-aware network embedding for relation modeling. In: 55th Annual Meeting of the Association for Computational Linguistics, pp. 1722-1731. Canada (2017).