The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

A Semi-Supervised Network Embedding Model for Protein Complexes Detection*

Wei Zhao,[1] Jia Zhu,[2] Min Yang,[1] Danyang Xiao,[2] Gabriel Pui Cheong Fung,[3] Xiaojun Chen[4]
[1]SIAT, Chinese Academy of Sciences
[2]South China Normal University
[3]The Chinese University of Hong Kong
[4]Shenzhen University

Abstract

Protein complex is a group of associated polypeptide chains which plays essential roles in biological process. Given a graph representing protein-protein interactions (PPI) network, it is critical but non-trivial to detect protein complexes. In this paper, we propose a semi-supervised network embedding model by adopting graph convolutional networks to effectively detect densely connected subgraphs. We conduct extensive experiment on two popular PPI networks with various data sizes and densities. The experimental results show our approach achieves state-of-the-art performance.

Introduction

Protein complex is a complex graph structure that is linked by protein-protein interactions (PPI), which plays an essential role in biological process. In general, the PPI network can be represented as an undirected and unweighted graph where proteins are represented as vertices and their interactions as edges. Each protein complex consists of two or more proteins that are shown as densely connected subgraphs, which indicates graph based clustering methods should be utilized to discover them.

Recently, network embedding has gained significant attention on improving the performance of many graph clustering methods (Wang, Cui, and Zhu 2016). However, most network embedding methods heavily rely on the attributes of each vertex in the network, which is not suitable for PPI network since there is no any metadata associated with each node except protein name.

In this paper, we propose a semi-supervised network embedding model by adopting graph convolutional network (GCN) that is capable of capturing both local and global structure of PPI network even there is no any information associated with each vertex in PPI network. We conduct extensive experiments on two widely used PPI datasets. The experimental results demonstrate that our method consistently outperforms the previous methods.

Model Description

PPI data come in the form of connections between proteins, which is easily described as a graph model. Proteins are represented as vertices and their interactions are represented as edges in the graph. Assume we have a graph $G = (V, E)$, where V represents a set of vertices in the graph, $V = v_1, ..., v_n$. E represent a set of edges in the graph, $E = e_1, ..., e_n$. Each edge is associated with two vertices. Let $H = h_1, \ldots, h_n$ be the set of neighbor vertices of v, and n is the number of the neighbor vertices of v. For PPI network, there is no weight for edges. In this section, we elaborate our semi-supervised network embedding model.

Component for the First-Order proximity. As described earlier, there is no attributes attached to each vertex in PPI network, we propose a method to select the important neighbor vertices of each vertex by using the vectors generated by DeepWalk as its attributes. We first apply DeepWalk to the graph to get a 64-dimensions vector for each vertex. And then the Euclidean metric is employed to compute the tightness score $Score_{v,h_i}$ between v and each h_i. Finally, we keep these neighbor vertices that have tightness score higher than the average score as attributes of v.

Once we have the attributes for each vertex, we then can use the attributes as supervised information to exploit the first-order proximity and refine the representations in the latent space to constrain the similarity of a pair of vertexes.

Component for the Second-Order proximity. The second-order proximity describes the pairwise similarity between vertices' neighborhood structure.

We design a GCN (Kipf and Welling 2016) based auto-encoder (Yang et al. 2015) to preserve the second-order proximity of the graph. Here, we use the attributes from each vertex as input channels of the GCN, and then after encoding of l convolutional layers, we can get a representation that is learned from the original graph. For the decoding part, we simply use an inner product decoder.

In our proposed model, we can naturally incorporate vertices' attributes to simultaneously optimize the first-order and second-order proximity referring to the following definition:

Definition 1 *Given an undirected, unweighted graph $G = (V, E)$ with $N = |V|$ vertices. We have an adjacency matrix A of G and an $N \times D$ matrix X as input, D is the number of attributes per vertex. With a stochastic latent variables z_i, we can summarize an $N \times F$ output matrix Z. where F is the number of output attributes.*

*Jia Zhu is the corresponding author (jzhu@m.scnu.edu.cn).

Data set	Vertices	Edges	Ave. degree	Density
Krogan	5364	61289	22.85	0.0043
Dip	4972	17836	7.17	0.0014

Table 1: Features of PPI datasets

Every network layer can then be written as a non-linear function:

$$H^{(l+1)} = f(H^{(l)}, A) = ReLU(AH^{(l)}W^{(l)})$$

where $H^{(0)} = X$ and $H^{(L)} = Z$, L is the number of layers, $W^{(l)}$ is a weight matrix for the l-th network layer and ReLU is the activation function.

Model Optimization

We use a common graph Laplacian regularization term loss function to optimize:

$$L = L_{first} + \lambda L_{second} = \sum_{i,j=1}^{n} ||y_i - y_j||^2 + \lambda \sum_{0}^{L} ||H^{(l+1)} - H^l||^2$$

where L_{first} denotes the supervised loss of the first-order proximity (the labeled part of the graph). L_{second} denotes the unsupervised loss of the second-order proximity, and λ is a trade-off factor, y_i and y_j are two matrices constructed based on selected neighbor vertices and each vertex is 64-dimensions vector generated by DeepWalk.

Experiments

Experimental setup

Datasets We conduct experiments on two widely used datasets: Krogan[1] and Dip[2]. The detailed statistics are presented in Table 1.

Baseline Methods We choose three different types of clustering methods: K-means, DBSCAN (Ester et al. 1996) and COACH (Min et al. 2009). We also compare our model with two network embedding models: DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and SDNE-SN (Wang, Cui, and Zhu 2016).

Implementation Details For COACH, we set DENSITY, AFFINITY, and CLOSENESS, to 0.7, 0.2 and 0.5, respectively. For K-means and DBSCAN, we use the default settings. Also, we set dropout rate to 0.2 for all convolutional layers. We train the model for a maximum of 200 epochs using Adam optimizer with a learning rate of 0.01 and early stopping with a window size of 10.

Experimental Results

In our experiments, the results are evaluated with F-measure. As shown in Figure 1, compared to previous methods, our approach achieves better results on both two datasets. On Dip data, our model achieves the 0.528 F-Measure, which is around 20% higher than using COACH only. COACH+Our Model (denoted as Our Model) is also 9.5% higher than the COACH+SDNE-SN (denoted as SDEN-SN) method that is the second best method, and 17% higher than the COACH+DeepWalk (denoted as DeepWalk) method. Also, we compared the clustering quality of each method summarized in Table 2. As expected, our model can detect more protein complexes than other methods on two datasets.

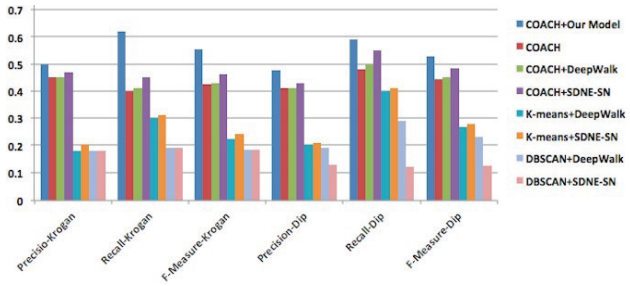

Figure 1: Comparison results on Krogan and Dip dataset

Data set	Our Model	COACH	DeepWalk	SDNE-SN
Krogan	610	570	570	580
Dip	808	748	750	840

Table 2: Number of Protein Complexes Detection

Conclusions and Future Work

In this paper, we proposed a network embedding model to capture the local and global structure of PPI networks effectively. Extensive experiments show that our model achieves state-of-art performance.

Acknowledgments This work was supported by the National Science Foundation of China (No. 61772211, 61750110516), the S&T Projects of Guangdong Province (No.2016A030303055, 2016B030305004, 2016B010109008) and the Guangzhou Science and Technology Project (No.201604046017).

References

Ester, M.; Kriegel, H. P.; Sander, J.; and Xu, X. W. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *In SIGKDD,* 226–231.

Kipf, T. N., and Welling, M. 2016. Variational Graph Auto-Encoders. *ArXiv e-prints:1611.07308*.

Min, W.; Li, X. L.; Kwoh, C. K.; and Ng, S. K. 2009. A core-attachment based method to detect protein complexes in ppi networks. *BMC Bioinformatics* 10(1):169.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. *In SIGKDD*.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. *In SIGKDD,* 1225–1234.

Yang, M.; Tu, W.; Lu, Z.; Yin, W.; and Chow, K.-P. 2015. Lcct: a semisupervised model for sentiment classification. In *NAACL*.

[1]http://interactome-cmp.ucsf.edu/

[2]http://dip.doe-mbi.ucla.edu/