Multi-scale Subgraph Contrastive Learning for Link Prediction

Shilin Sun, Zehua Zhang (✉), Runze Wang, and Hua Tian

College of Information and Computer,Taiyuan University of Technology,Jinzhong 030600,China
zehua_zhang@163.com

Abstract. Existing works about link prediction rely mainly on pooling operations which cause loss of edge information or similarity assumptions, so that they are limited in specific networks, and mainly supervised learning methods. We propose a Multi-scale Subgraph Contrastive Learning(MSCL) method. To adapt to networks of different sizes and make direct use of edge information, MSCL converts a sampled subgraph centered on the target link into a line graph as a node-scale to represent links, and mines deep representations by combining two scales information, subgraph-scale and line graph node-scale. After learning the information of the two subgraphs separately by encoders, we use contrastive learning to balance the information of two scales to alleviate the over-reliance of the model on labels and enhance the model's robustness. MSCL outperforms a set of state-of-the-art graph representation learning solutions on link prediction task in a variety of graphs including biology networks and social networks.

Keywords: Multi-scale · Contrastive Learning · Subgraph · Link Prediction.

1 Introduction

Link prediction methods combined with specific applications can be considered to predict the presence of interactions between pairs of nodes or the type of interactions. And link prediction methods have many specific applications in the real world, such as drug interaction prediction, community discovery, etc.

Traditional methods for link prediction score links by calculating nodes similarity. Based on the assumption that similar nodes are more inclined to connect, these heuristic methods use known node information in the network to score nodes. Some methods consider local information about the network, such as Common Neighbors (CN). Other methods take a global perspective, for example, Katz [5] and Pagerank [1] directly aggregate over all the paths between node pairs to score links. Although such methods improve performance, they lack universal applicability to various networks, for example, similar proteins do not tend to interact [7].

To design a universal model, based on the fact that subgraphs already contain enough information and are suitable for networks of different sizes, Weisfeiler-Lehman Neural Machine(WLNM) [9] extracts local subgraphs around the links, and learns the subgraphs corresponding to link existence through the fully connected layer to achieve better performance. To enhance the graph feature learning capability and incorporate latent features, SEAL [10] uses a graph neural network to replace the fully connected neural network in WLNM. However, the information loss of pooling operations with subgraphs is not negligible. LGLP [2] converts subgraphs into line graphs to obtain a unique node to represent every link directly for more edge information. Although it works well, only a single scale of information is considered in LGLP and it is based on a supervised learning approach. So the performance improvement is limited by the noise from the growing edges of the line graph and over-reliance on labels.

In this paper, we propose a **M**ulti-scale **S**ubgraph **C**ontrastive **L**earning (MSCL) method for the link prediction task. Firstly, the subgraphs are extracted centered on the links, and then the subgraphs are converted into line graph subgraphs so that each link has its corresponding node representation. Thus, a new view is obtained. Secondly, the original subgraph and the line graph subgraph are learned by different encoders to obtain link representation, and finally, multi-scale information is balanced using contrastive learning. Our work is summarized as follows:

- We propose a multi-scale subgraph contrastive learning framework, which converts subgraphs into line graphs to improve the efficiency of aggregating information and reduce the information loss of subgraph pooling.
- We adopt a contrastive learning component to achieve multi-scale learning and balance information of two views. So the robustness of the model is enhanced and the dependence of models on labels is decreased.
- The multi-scale information is learned using different encoders, and the model achieves state-of-the-art results on two public available datasets.

2 Proposed method

2.1 Problem formulation

Through multi-scale contrastive learning, the model integrates line graph and subgraph information. The line graph node transformed from the subgraph of the target link is the positive sample g^+, and the node of the line graph corresponding to the other link is negative sample g^-, and the anchor g is the subgraph of target link. The model then trains the mapping function using contrastive learning to enhance model performance. Our contrastive learning component is shown in Equation (1). $score(\cdot)$ is a function to calculate the similarity of two representations for a target link. And $f(\cdot)$ denotes graph encoder.

$$\text{score}(f(g), f(g^+)) >> \text{score}(f(g), f(g^-)) \tag{1}$$

2.2 Overview

Figure 1 illustrates the overall framework of MSCL. The line graph transformation component samples the original graph for subgraphs and transforms them to line graphs. Next, the encoder encodes the subgraph and line graph respectively. Finally, the contrastive learning component balances multi-scale information.

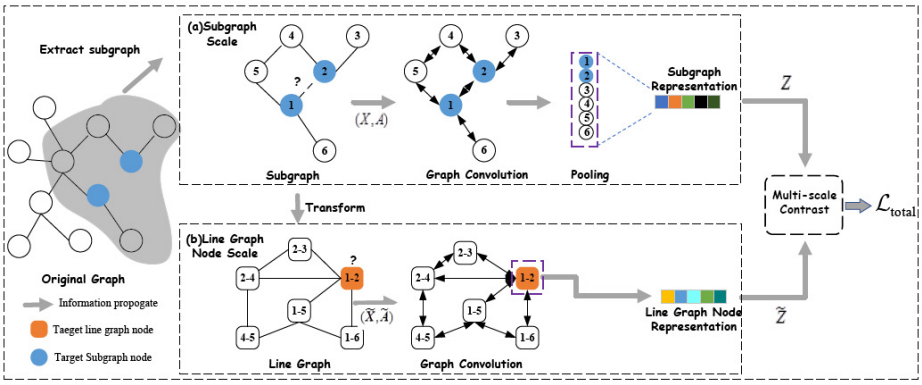


Fig. 1: Overview of the MSCL framework

2.3 Line Graph Transformation

MSCL mainly uses the different scale information of subgraph $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{X}, \boldsymbol{A})$ for link prediction, and its core point is to balance the scale information of line graph nodes by contrastive learning.

Line Graph Transformation: To transform a subgraph into a line graph, we transform the subgraph's edges to the line graph's set of nodes, as illustrated in Equation (2). If any two nodes of a line graph correspond to two edges of a subgraph that share a common node, then these two nodes form an edge [2]. $\widetilde{\mathbf{V}}$ is the line graph node identity matrix, and $\widetilde{\mathbf{A}}$ is the set of edges of the line graph, which is identical to the representation of the adjacency matrix. Equation (2) is used to generate the line graph representation $\widetilde{\mathbf{G}}(\widetilde{\mathbf{V}}, \widetilde{\mathbf{E}}, \widetilde{\mathbf{X}}, \widetilde{\mathbf{A}})$.

$$\widetilde{\mathbf{V}} = \{e\}, \forall e \in \mathbf{E}$$

$$\widetilde{\mathbf{X}} = \{\text{concate}(x_i, x_j) | \forall e_{(v_i, v_j)} \in \widetilde{\mathbf{V}}\}$$

$$\widetilde{\mathbf{E}} = \{l_{(e_i, e_j)} | e_i \cap e_j \neq \emptyset\}, \forall e_i, e_j \in \widetilde{\mathbf{V}} \tag{2}$$

$$\text{Num}(\widetilde{\mathbf{E}}) = \frac{1}{2} \sum_{i=1}^{m=\text{Num}(\boldsymbol{V})} \text{Deg}(v_i)^2 - \text{Num}(\boldsymbol{E}), \forall v \in \boldsymbol{V} \tag{3}$$

As shown in Equation (3), the number of edges in the line graph increases exponentially compared to the number of edges in the original graph. $Deg(\cdot)$ is a function to calculate the degree of nodes.

2.4 Graph Encoder

The graph encoder $f(\cdot)$ is separated into two parts: a subgraph encoder $f_S(\cdot)$, a line graph encoder $f_L(\cdot)$. And the graph encoder handles the feature and adjacency matrices of the two views. The representation of subgraphs and line graphs is conducted independently as $\boldsymbol{Z}, \widetilde{\mathbf{Z}}$.

The graph encoder extracts substructure features by stacking multiple graph convolution layers as $\boldsymbol{Z}^{1:h} := [\boldsymbol{Z}^1 \dots \boldsymbol{Z}^h]$, and h represents the number of layer. Finally, the obtained graph is pooled through the SortPooling layer [11] to obtain a graph-scale representation, thereby obtaining a cross entropy loss $\mathcal{L}$.

The design of the line graph encoder convolution layer mainly follows GCN [6]. Finally, the pooling layer selects nodes to represent target links, and the supervised loss $\widetilde{\mathcal{L}}$ of the line graph can be obtained.

2.5 Contrastive Learning

To combine multi-scale information which is line graph node information and subgraph information, we mainly follow the contrastive model in GraphCL [8].

Take $z^{(n)}, \widetilde{z}^{(n)}$ in $\boldsymbol{Z}, \widetilde{\mathbf{Z}}$ respectively, to denote the two views of the nth graph in the small batch. Negative samples are generated from the other $n-1$ line graphs. $sim(\cdot)$ is cosine similarity function, and contrastive loss $\mathcal{L}_{CON}$ is defined as:

$$\mathcal{L}_{\mathrm{CON}} = \frac{1}{|T|} \sum\nolimits_{n=1}^{|T|} -\log \frac{\exp(\mathrm{sim}(z^{(n)}, \widetilde{z}^{(n)})/\tau)}{\sum_{n'=1, n' \neq n}^{N} \exp(\mathrm{sim}(z^{(n)}, \widetilde{z}^{(n')})/\tau)} \tag{4}$$

$|T|$ denotes the number of links in the training set and τ is a hyperparameter.

Finally, the total loss function $\mathcal{L}_{total}$ is obtained by combining the self-supervised task loss with the supervised loss of both views.

$$\mathcal{L}_{\mathrm{total}} = \alpha\mathcal{L} + \beta\widetilde{\mathcal{L}} + \lambda\mathcal{L}_{\mathrm{CON}} \tag{5}$$

α, β, λ are the hyperparameters used to balance the different losses.

3 Experiment

During the training process of the model, we use the Adam optimizer with a learning rate of 0.05.To evaluate the effectiveness of link prediction, we use Area Under the Curve(AUC) and Average Precision(AP) as evaluation metrics.

3.1 Datasets and Baseline Models

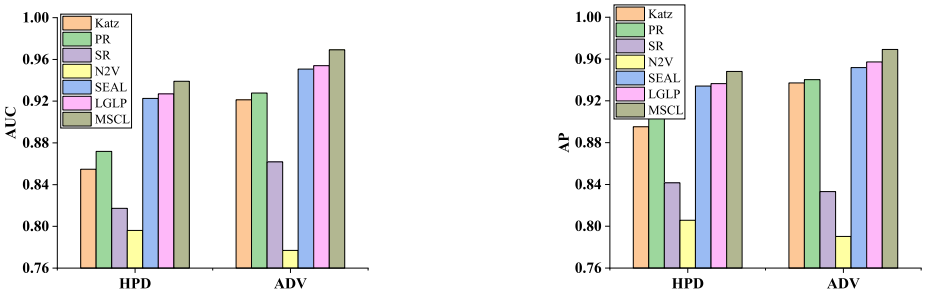
We conduct experiments on two datasets in different areas, HPD, ADV [2] to verify MSCL's effectiveness. In this work, we compare three network similarity methods, including Katz [5], PageRank (PR) [1] and SimRank (SR) [4]. Also, the network embedding methods Node2vec(N2V) [3] and graph representation learning methods SEAL [10] and LGLP [2].

Table 1: Summary of datasets used in our experiments.

Datasets	Nodes	Links	Degree	Area
HPD	8756	32331	7.38	Biology
ADV	5155	39285	15.24	Social Network

3.2 Comparison With Baselines

Fig. 2 shows that the performance of MSCL method has been improved compared with the other three types of methods on 80% training percentage of links. The graph representation learning method can learn deeper feature information and topological information than other methods, so the performance is significantly improved. MSCL combines multi-scale information and therefore has the best performance among all methods.

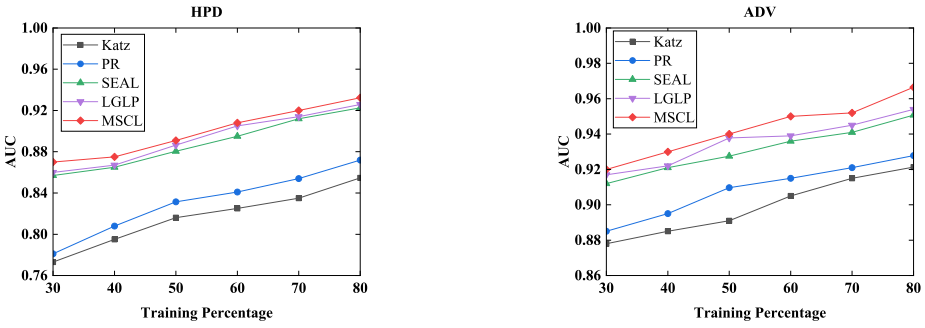


(a) AUC comparison with baselines

(b) AP comparison with baselines

Fig. 2: Performances of MSCL and baselines

3.3 Model Robustness Analysis



(a) AUC comparison on HPD

(b) AUC comparison on ADV

Fig. 3: Robustness analysis on HPD and ADV with different training percentage

To explore the robustness of the model to network sparsity, experiments are conducted on edge datasets of different sizes from 30% to 80%. Fig.3 shows the

robustness of MSCL to network sparsity. MSCL outperforms at all assignments with various levels of network sparsity. The performance of Katz and PageRank, which are based on network similarity, is poor compared to other methods due to their assumptions, and the gap becomes larger as the density increases. MSCL alleviates the model's over-reliance on labels and the line graph noise problem.

4 Conclusions

In this paper, we propose a method for balancing line graph node scale and subgraph scale information by contrastive learning. Final experiments show that MSCL performs well. Future work will continue to explore the information yield of graphs at different scales for diverse tasks.

Acknowledgements This work was supported by the National Natural Science Foundation of China (61702356), Industry-University Cooperation Education Program of the Ministry of Education, and Shanxi Scholarship Council of China.

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30**(1), 107–117 (1998), proceedings of the Seventh International World Wide Web Conference
2. Cai, L., Li, J., Wang, J., Ji, S.: Line graph neural networks for link prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
3. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
4. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 538–543 (2002)
5. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39 – 43 (1953)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
7. Kovács, I.A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.K., Kishore, N., Hao, T., et al.: Network-based prediction of protein interactions. Nature communications **10**(1), 1–8 (2019)
8. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems **33**, 5812–5823 (2020)
9. Zhang, M., Chen, Y.: Weisfeiler-lehman neural machine for link prediction. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 575–583 (2017)
10. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. Advances in neural information processing systems **31** (2018)
11. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)