CSGNN: Improving Graph Neural Networks with Contrastive Semi-supervised Learning

Yumeng Song(✉), Yu Gu, Xiaohua Li, Chuanwen Li, and Ge Yu

School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning, China
ymsong94@163.com, {guyu,lixiaohua,lichuanwen,yuge}@mail.neu.edu.cn

Abstract. The Graph Neural Network (GNN) is a rising graph analysis model family that encodes node features into low-dimensional representation vectors by aggregating local neighbor information. Nevertheless, the performance of GNNs is limited since GNNs are trained only over predictions of the labeled data. Hence, effectively incorporating a great number of unlabeled nodes into GNNs will upgrade the performance of GNNs. To address this issue, we propose a Contrastive Semi-supervised learning based GNN (CSGNN) that improves the GNN from extra supervision predicted by contrastive learning. Firstly, CSGNN utilizes multi-loss contrast to learn node representations via maximizing the agreement between nodes, edges and labels of different views. Then, a semi-supervised fine-tuner learns from few labeled examples while making the best use of unlabeled nodes. Finally, we introduce the knowledge distillation based on label reliability, which further distills the node labels predicted by contrastive learning into the GNN. Experimentally, CSGNN effectively improves the classification performance of GNNs and outperforms other state-of-the-art methods in accuracy over a variety of real-world datasets.

Keywords: Contrastive learning · Semi-supervised learning · Graph Neural Network

1 Introduction

Graph Neural Networks (GNNs) have aroused more and more attention on account of the ability to handle the graph-structured data defined on irregular or non-Euclidean domains. GNNs compute graph node representations through a propagation process which iteratively aggregates local structural information. GNNs are clearly superior to traditional graph-based algorithms in quite a few tasks [6]. Unfortunately, GNNs, as data-driven inference models, are also not free of the bottleneck when training data is inadequate. The reason is that GNNs are trained only over predictions of labeled nodes by minimizing the supervised loss, and predictions of unlabeled nodes do not contribute to the training. In order to tackle the intrinsic hardness, various researches have emphasized incorporating

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13245, pp. 731–738, 2022.
https://doi.org/10.1007/978-3-031-00123-9_58

unlabeled data into GNNs via combining them with self-supervised learning [8] or augmenting topology and attributes of graphs in different ways [12].

Recently, Contrastive Learning (CL) achieves great success in graph representation learning [15,16]. As a popular form of self-supervised learning, CL seeks to maximize the mutual information between the input and its representations by contrasting positive pairs with negative-sampled counterparts. However, CL learns embeddings in a task-agnostic way without using labeled data. This leads us to explore a fusion mechanism of CL and GNNs for graph-based semi-supervised learning. In the latest studies on graph-based CL, [11] proposes a contrastive semi-supervised model CG, which minimizes the contrastive loss, the graph generative loss and the classification loss between graph views together. But CG does not bring out the full power of CL. One of the state-of-the-art researches in computer vision also proposes a contrastive semi-supervised model SIMCLRv2 [1]. SIMCLRv2 distills the generated embeddings of unlabeled data into the downstream student model. Nevertheless, SIMCLRv2 cannot be directly applied to graph-based data.

Based on the above discussion, we propose a Contrastive Semi-supervised learning based GNN (CSGNN) which utilizes knowledge distillation to combine CL with GNNs, with a CL model as the teacher model and a GNN as the student model. To the best of our knowledge, CSGNN is the first research on combining contrastive learning with GNNs through knowledge distillation. Our contributions are summarized as follows:

- This paper provides a contrastive semi-supervised based GNN which could comprehensively leverage the abundant structural and semantic information of unlabeled nodes.
- In the teacher model, a multi-loss contrastive learning method is introduced to learn representations by contrasting positive and negative examples between nodes, edges and labels.
- We design a reliable knowledge distillation method via computing the label reliability based on the Shannon entropy of teacher and student's predictions.
- In experiments, we demonstrate that CSGNN can greatly improve the performance of GNNs in node classification task compared with the state-of-the-art methods on real-world datasets.

2 Related Works

For graph data, graph contrastive learning applies the idea of CL on GNNs. These methods can be categorized based on how the positive and negative samples are constructed. One is to measure the loss of different parts of a graph in latent space by contrasting nodes and the whole graph, nodes and nodes or nodes and subgraphs [5]. The other one uses different data augmentation methods to generate contrastive pairs. GraphCL [13] develops contrastive learning with node dropping, edge perturbation, subgraph sampling and feature masking. MVGRL [3] constructs multiple graph views by sampling subgraphs based

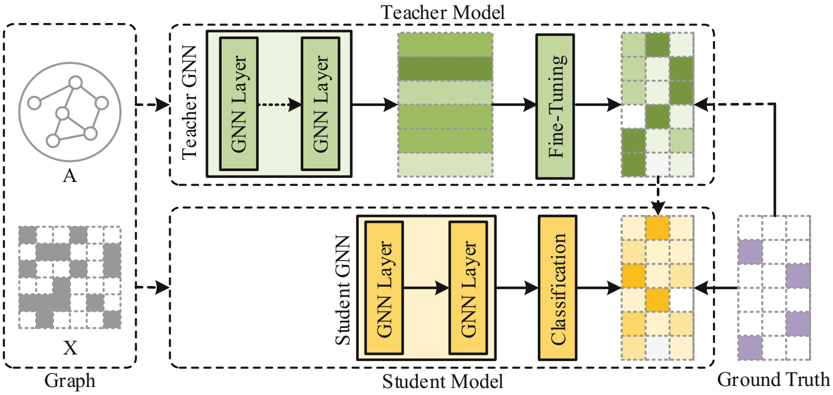

Fig. 1. The overview of CSGNN

on random walks. GCA [16] leverages the network centrality to augment the graph adaptively on both topology and attribute levels. CG [11] augments the graph via localized graph convolution and hierarchical graph convolution and designs a new semi-supervised contrastive loss. Most CL models cannot achieve the best performance for specific tasks through unsupervised learning.

3 Overview

The overall framework of CSGNN is shown in Fig. 1. Given a graph $G = (V, A, X)$ with a node set $V = \{v_1, v_2, ..., v_N\}$, a graph adjacent matrix $A \in R^{N\times N}$ and a node attribute matrix $X \in R^{N\times F}$ where F is the dimension of node attributes, we firstly input A and X of into a trained teacher GNN to generate the node general representation. The teacher GNN is trained via multi-loss contrastive learning, which can measure mutual information in multiple dimensions and obtain the main features of nodes without focusing on the details. The node embedding after fine tuning represents the category probability of each node. For further improving predictive performance and obtaining a compact model, we train the student GNN on the labeled data with ground truths and the unlabeled data with predicted labels from the fine-tuned teacher GNN. Finally, we can generate predictions directly from the student GNN, regardless of the teacher model.

4 Teacher Model with Contrastive Learning

The training process of the teacher model involves the following stages: (1) the adaptive graph augmentation stage, which transforms the original graph into different views; (2) the encoding stage, which generates the node representations via the teacher GNN; (3) the contrasting stage, which contrasts the latent vectors between nodes, edges and labels.

Graph Augmentation. We augment the graph by perturbing possibly unimportant links and features following the schemes proposed in GCA [16] which tends to keep important structures and attributes unchanged. Augmentation is divided into topology-level and attribute-level. On topology-level, we define edge centrality s^e_{uv} for edge e_{uv} to measure its influence based on PageRank centrality of two connected nodes. On attribute-level, we add noises to node attributes via randomly masking a fraction of dimensions with zeros in node attributes. We sample a random mask m_i $Bernoulli(P^{a_i}_{perturbing})$ for each attribute i. To evaluate the importance of attribute i, we assume that attributes frequently appearing in influential nodes should be important.

Augmented Graphs Encoding. At each iteration, we generate two graph views based on the augmentation scheme above, denoted as G_1 and G_2. Then G_1 and G_2 are input into the teacher GNN with shared parameters. The node embeddings are denoted as $U = f(X_1, A_1)$ and $V = f(X_2, A_2)$.

Contrastive Learning. After that, we employ the contrastive loss to train the teacher GNN. We conduct joint contrastive losses between nodes, edges and labels to make the embeddings more conducive to classification. For any node i, its embedding is u_i in view G_1 and v_i in view G_2. The node contrastive loss between a pair of positive examples u_i and v_i is given as follows:

$$\ell_{u_i,v_i} = -\log \frac{\exp(sim(g(u_i), g(v_i))/\tau)}{\sum_{k\neq i}\exp(sim(g(u_i), g(v_k))/\tau) + \sum_{k\neq i}\exp(sim(g(u_i), g(u_k))/\tau)} \tag{1}$$

where $sim(\cdot,\cdot)$ is the cosine similarity, $g(\cdot)$ is a non-linear transformation network, $\sum_{k\neq i}\exp(sim(g(u_i), g(v_k))/\tau)$ is the loss between inter-view negative pairs and $\sum_{k\neq i}\exp(sim(g(u_i), g(u_k))/\tau)$ is the loss between intra-view negative pairs. Since the symmetric among the views, our unsupervised node contrastive loss $\mathcal{L}_{nodes}$ can be presented as:

$$\mathcal{L}_{nodes} = \frac{1}{2N}\sum_{i=1}^{N}(\ell_{u_i,v_i} + \ell_{v_i,u_i}) \tag{2}$$

The goal of the edge contrastive loss is to distinguish between existing edges and non-existing edges within and between views. We reconstruct the adjacency matrix A^*_1 and A^*_2 based on the node embedding of each view. We also reconstruct the adjacency matrix $A^*_{1,2}$ between two views. We calculate the inner product of node embeddings as the possibility that two nodes have edges for reconstructing the adjacency matrix. Given edge $e_{i,j}$ in graph G, the corresponding edge in A^*_1, A^*_2 and $A^*_{1,2}$ are positive examples, and non-existing edges are negative examples. Here, the unsupervised edge contrastive loss can be computed as:

$$\mathcal{L}_{edges} = \frac{1}{3|E|}(\ell^{edges}_{G_1} + \ell^{edges}_{G_2} + \ell^{edges}_{G_1,G_2}) \tag{3}$$

$$\ell^{edges}_{G_1} = -\log \frac{\sum_{e\in E}\exp(A^*_{1e}/\tau)}{\sum_{\widehat{e}\notin E}\exp(A^*_{1\widehat{e}}/\tau)} \tag{4}$$

where E is the edge set of graph G and A^*_{1e} is the value of edge e in A^*_1. $\ell^{edges}_{G_2}$ and $\ell^{edges}_{G_1,G_2}$ are similar to Eq. 4.

Our supervised contrastive learning loss will distinguish nodes of the same category and nodes of different categories within and between views, which is defined as:

$$\mathcal{L}_{labels} = \frac{1}{2\,L|} \sum_{l \in L} (\ell_l^{labels,G_1} + \ell_l^{labels,G_2}) \tag{5}$$

$$\ell_l^{labels,G_1} = -\log \frac{\sum_{k \in S(l)} \exp(sim(g(l), g(k))/\tau)}{\sum_{\widehat{k} \in Diff(l)} \exp(sim(g(l), g(\widehat{k}))/\tau)} \tag{6}$$

where L is the set of labeled nodes, $S(\cdot)$ is the set of nodes with the same label, $Diff(\cdot)$ is the set of nodes with different labels. ℓ_l^{labels,G_2} is similar to Eq. 6.

By combining node, edge and label contrastive losses, we arrive at the following multi-loss contrastive learning:

$$\mathcal{L} = \mathcal{L}_{nodes} + \lambda_1 \mathcal{L}_{edges} + \lambda_2 \mathcal{L}_{labels} \tag{7}$$

where λ_1 and λ_2 are hyperparameters that control the proportion of the corresponding loss. After training, we input the node embeddings into an L2-regularized logistic regression classifier to generate fine-tuned prediction results.

5 Student Model with Reliable Distillation

5.1 Label Reliability Based on Shannon Entropy

Since the correctness of unlabeled nodes' label predictions is difficult to evaluate, Shannon entropy is used to evaluate the probability of reliable label predictions. However, there are also correct predictions for nodes with high entropy. Therefore, we can compare the prediction results of the student model and the teacher model to enhance the evaluation of label reliability. Formally, we define the label reliability of a node i prediction as follows:

$$R_i = \begin{cases} 1, & \text{if } i \in L \\ 1, & \text{if } t(i) = s(i) \text{ and } H(T_i) < H_{max} \\ \exp(-(H(T_i) + H(S_i))) & \text{if } t(i) = s(i) \text{ and } H(T_i) > H_{max} \\ 0, & \text{if } t(i) \neq s(i) \end{cases} \tag{8}$$

where $H(\cdot)$ computes the Shannon entropy of the vector, $t(\cdot)$ is the label of the teacher's prediction, $s(\cdot)$ is the label of the student's prediction, T is the node prediction vector of the teacher model, S is the prediction vector of student model and H_{max} is the max reliable threshold of entropy.

5.2 Model Training

We train the model based on labeled nodes with ground truths and unlabeled nodes with reliable labels. For each iteration, we update the reliability of the unlabeled nodes. With the improvement of the accuracy of the student model, more and more reliability nodes can be chosen to teach the student GNN. The training loss of the student GNN is defined as:

$$\mathcal{L}_{student} = \frac{1}{|L|}\sum_{l \in L} CE(y_l, \widehat{y}_l) + \frac{1}{|U|}\sum_{u \in U} R(u) CE(\widetilde{y}_u, \widehat{y}_u) \tag{9}$$

where L is the set of labeled nodes, U is the set of unlabeled nodes, y_l is the label of labeled node l, $\widetilde{y}_u$ is the label of unlabeled node u which is learned from the teacher model, $\widehat{y}$ is the prediction of the student GNN and $CE(\cdot)$ is the cross-entropy loss function.

6 Experiments

6.1 Experiment Setting

Five real-world graph datasets are used for the experiments including Cora, Citeseer, Pubmed, Amazon Computers and Amazon Photo [11]. As for baselines, we opt a series of methods including the Label Propagation(LP) [14], Chebyshev [2], GCN [4], GAT [9], DGI [10], GMI [7], MVGRL [3], GCA [16] and CG [11]. For Cora, Citeseer and Pubmed datasets, we use 20 nodes per class as the training set and 30 nodes per class as the validation set. For Amazon Computers and Amazon Photo datasets, we use 30 labeled nodes per class as the training set, 30 nodes per class as the validation set. We report the mean accuracy and the stand derivations of 20 runs. For the hyperparameters of different GNNs, we set them as suggested by their authors. For CSGNN, we set a 2-layer GCN as the teacher GNN and a 2-layer GAT as the student GNN. The hyperparameters of CSGNN are the optimal parameters selected based on experimental results.

6.2 Semi-supervised Classification

The semi-supervised node classification results are reported in Table 1. The results for five datasets exhibit similar trends: CSGNN yields predictions comparable or superior to those of the other contestants. For example, compared to GCN, CSGNN reaches nearly 3.3%, 3.5%, 1.1%, 7.1%, 3.5% gain on five datasets respectively. We also have the following observations: (1) Some unsupervised contrastive learning methods present better performance than baseline semi-supervised learning methods; (2) Two contrastive learning methods GCA and CG are strong competitors for the best performance. They perform well on some datasets, but also fail in some datasets, while CSGNN consistently performs well on all datasets. Hence, we believe that CSGNN can steadily improve GNNs' performance, even better than the state-of-the-art methods.

Table 1. Results of semi-supervised node classification (%)

	Cora	Citeseer	Pubmed	Computers	Photo
LP	68.0	45.3	63.0	70.8 ± 0.0	67.8 ± 0.0
Chebyshev	79.3 ± 1.3	67.4 ± 1.5	75.3 ± 0.5	62.6 ± 0.8	74. 3 ± 0.5
GCN	81.5 ± 0.6	70.7 ± 0.4	79.3 ± 0.2	76.3 ± 0.5	87.3 ± 1.0
GAT	83.1 ± 0.5	72.5 ± 0.7	79.5 ± 0.5	79.3 ± 1.1	86.2 ± 1.5
DGI	81.7 ± 0.6	71.5 ± 0.7	77.3 ± 0.6	75.9 ± 0.6	83.1 ± 0.5
GMI	82.7 ± 0.2	73.0 ± 0.3	80.1 ± 0.2	76.8 ± 0.1	85.1 ± 0.1
MVGRL	82.9 ± 0.7	72.6 ± 0.7	79.4 ± 0.3	79.0 ± 0.6	87.3 ± 0.3
GCA	80.9 ± 0.6	68.1 ± 2.0	80.3 ± 0.9	82.3 ± 0.3	90.4 ± 0.2
CG	83.4 ± 0.7	73.6 ± 0.8	80.2 ± 0.8	79.9 ± 0.6	89.4 ± 0.5
CSGNN	**84.8 ± 1.0**	**74.2 ± 1.2**	**80.8 ± 0.4**	**83.4 ± 1.4**	**90.8 ± 0.1**

6.3 Ablation Study

This section provides an ablation analysis to validate the contributions of different components of CSGNN on three citation datasets. For the variants, we use "T" as the teacher model, "S/R" as the student model without evaluating label reliability, "S" as the student model, "w/o KD" as the GAT without knowledge distillation, "CL-N" as CL with the node loss, "CL-N-E" as CL with node and edge losses, and "CL-ALL" as CL with all losses. The results are summarized in Table 2. It exhibits three interesting patterns: (1) The node, edge and label losses benefit the contrastive learning; (2) Without label reliability, distillation will reduce the performance of the student model, and the performance of "S/R" variants are even lower than the model without distillation; (3) Among different techniques, distillation improves performance more than contrastive learning.

Table 2. Ablation results of semi-supervised node classification(%)

	Cora			CiteSeer			PubMed		
	T	S/R	S	T	S/R	S	T	S/R	S
w/o KD	–	–	83.1	–	–	72.5	–	–	79.5
CL-N	80.9	81.3	84.0	68.1	69.6	72.8	80.3	78.2	80.4
CL-N-E	81.1	83.4	84.2	69.3	71.0	74.2	80.4	77.3	80.7
CL-ALL	81.3	83.6	84.4	69.4	70.8	74.6	80.5	79.5	80.9

7 Conclusion

In this paper, we explore contrastive learning methods for graph-based data and propose a contrastive semi-supervised learning based GNN by knowledge distillation, named CSGNN. CSGNN is able to learn from reliable unlabeled nodes when we distill the predictions of contrastive learning with multi-loss into the downstreaming student model. Extensive experiments demonstrate that CSGNN can consistently outperform the state-of-the-art models in node classification accuracy on real-world datasets.

Acknowledgements. This work is supported by the National Natural Science Fondation of China (62072083 and 61872071).

References

1. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020)
2. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. NIPS **29**, 3844–3852 (2016)
3. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: ICML, pp. 4116–4126 (2020)
4. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
5. Liu, Y., Pan, S., Jin, M., Zhou, C., Xia, F., Yu, P.S.: Graph self-supervised learning: A survey. arXiv preprint arXiv:2103.00111 (2021)
6. Peng, Y., Choi, B., Xu, J.: Graph learning for combinatorial optimization: a survey of state-of-the-art. Data Sci. Eng. **6**(2), 119–141 (2021)
7. Peng, Z., et al.: Graph representation learning via graphical mutual information maximization. In: WWW, pp. 259–270 (2020)
8. Sun, K., Lin, Z., Zhu, Z.: Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In: AAAI, vol. 34, pp. 5892–5899 (2020)
9. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
10. Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. ICLR **2**(3), 4 (2019)
11. Wan, S., Pan, S., Yang, J., Gong, C.: Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In: AAAI, vol. 35, pp. 10049–10057 (2021)
12. Wang, Y., Wang, W., Liang, Y., Cai, Y., Liu, J., Hooi, B.: Nodeaug: semi-supervised node classification with data augmentation. In: KDD, pp. 207–217 (2020)
13. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. NIPS **33**, 5812–5823 (2020)
14. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)
15. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131 (2020)
16. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: WWW, pp. 2069–2080 (2021)