Graph of Graphs: A New Knowledge Representation Mechanism for Graph Learning (Student Abstract)

Zhiwei Zhen[1], **Yuzhou Chen**[2], **Murat Kantarcioglu**[1], **Yulia R. Gel**[1]

[1] University of Texas at Dallas, 800 W Campbell Rd, Richardson, TX 75080
[2] Temple University, 1801 N Broad St, Philadelphia, PA 19122
zhiwei.zhen@utdallas.edu, yuzhou.chen@temple.edu, ygl@utdallas.edu, muratk@utdallas.edu

Abstract

Supervised graph classification is one of the most actively developing areas in machine learning (ML), with a broad range of domain applications, from social media to bioinformatics. Given a collection of graphs with categorical labels, the goal is to predict correct classes for unlabelled graphs. However, currently available ML tools view each such graph as a standalone entity and, as such, do not account for complex interdependencies among graphs. We propose a novel knowledge representation for graph learning called a *Graph of Graphs* (GoG). The key idea is to construct a new abstraction where each graph in the collection is represented by a node, while an edge then reflects similarity among the graphs. Such similarity can be assessed via a suitable graph distance. As a result, the graph classification problem can be then reformulated as a node classification problem. We show that the proposed new knowledge representation approach not only improves classification performance but substantially enhances robustness against label perturbation attacks.

Introduction

Graph classification tasks by inferring the geometric structures has attracted increasing attention in recent years due to its diverse applications in cybersecurity, social network analysis, and financial risk analysis among others. More recently, Geometric Deep Learning (GDL) has led to significant gains in graph classification performance by incorporating the information of structure space into the learning process and extending the power of Machine Learning (ML) to non-Euclidean domains. However, the relationships among graphs are barely integrated into neural networks, even though such relationships may play a significant role for a graph learning task. Indeed, graphs with similar structures often tend to be in the same group. Here, we advance GDL for graph classification tasks, by constructing a new network representation of the task-related graph relationships with different graph comparison metrics.

Related Work

Graph Convolution Networks (GCNs) for Graph Classification GCNs have been shown to be very successful in graph classification tasks (Wu et al. 2020). While powerful, GCNs do not explicitly address the similarities among graphs, show limited capability to filter permutation, and are sensitive to attacks. To address these limitations, we advance the currently existing GCN framework by integrating our new *Graph of Graphs* (GoG) solution.

Problem Statement and Methods

Formally, the graph classification problem can be formulated as follows. Let $\tilde{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N\}$ be a set of graphs, where N is the number of graphs in the graph set. Each graph can be represented as $\mathcal{G}_i = \{V_i, E_i\}$ where $V_i = \{v_1^i, v_2^i, \ldots, v_p^i\}$ is a set of p nodes and E_i is the set of edges. Furthermore, each node in the graph is associated with node features, $X \in \mathbb{R}^{N \times F}$, where F denotes the dimension of features. Here we focus on two problems: (i) graph classification with original training labels and (ii) graph classification with *attacked* training labels.

`Problem 1`: Learn a mapping function $\mathcal{H}(\{\mathcal{G}^{(n)}\}_{n=1}^N)$ which maps the records to a binary anomaly output $l^{(n)}$.
`Problem 2`: With attacked training labels $l_*^{(n)}, n = 1, 2, \ldots, N_{train}$. Learn a mapping function $\mathcal{H}(\{\mathcal{G}^{(k)}\}_{n=1}^N)$ which maps the records to a binary anomaly output $l^{(n)}$. We assume that the training labels are corrupted by noises.

Metrics for Graph Comparison The distance between graphs provide a way to determine the relationships between graphs, which is critical for the classification task. To compare the graph structures, in our study, we consider 3 graph comparison metrics: vertex-edge overlap distance, lambda distance and Frobenius norm distance (Wills and Meyer 2020). Specially, for any two graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ (where $i, j \in \{1, 2, \ldots, N\}$), we calculate the above 3 metrics to determine the similarities between graphs. In general, smaller distance indicates higher similarity among two graphs.

Methodology

Graph of Graphs We propose *Graph of Graphs* (*GoG*) as a general solution for graph classification tasks. First, we compute distances between any two graphs by using distance measure $\mathcal{M}_h(i, j)$ (here h represents the metric for graph comparison, i.e., distance between graphs) and

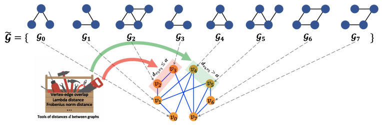
$\widetilde{\mathcal{G}} = \{ \mathcal{G}_0 \quad \mathcal{G}_1 \quad \mathcal{G}_2 \quad \mathcal{G}_3 \quad \mathcal{G}_4 \quad \mathcal{G}_5 \quad \mathcal{G}_6 \quad \mathcal{G}_7 \}$
$d_{v_2,v_3} \leq \alpha$
$d_{v_4,v_5} > \alpha$
v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7
Vertex-edge overlap
Lambda distance
Frobenius norm distance
...
Tools of distances d between graphs

Figure 1: GoG representation learning framework.

$i, j \in \{1, 2, \ldots, N\}$). We then generate a distance matrix for N graphs, denoted as $\mathcal{D}_h^{(i,j)} \in \mathbb{R}^{N \times N}$. Finally, we use the threshold τ to filter noisy connections between graphs, that is, $\hat{A}_{ij} = 1$ if $\mathcal{D}_h^{(i,j)} \leq \tau$ and 0 otherwise. (Here $\hat{A}$ is the binary correlation matrix.) As a result, we build a new graph $\hat{\mathcal{G}}$ based on the N graphs $\tilde{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N\}$, which we call *Graph of Graphs (GoG)*. Specifically, $\hat{\mathcal{G}} = (\hat{V}, \hat{E}, \hat{A})$ with node set $\hat{V}$ with $|\hat{V}| = N$ (i.e., each graph $\mathcal{G}_i \in \tilde{g}$ denotes a node in GOG $\hat{G}$) and edge set $\hat{E}$. The $N \times N$-matrix $\hat{A}$ represents the adjacency matrix of GoG $\hat{G}$ with entries $\{a_{ij}\}_{1 \leq i,j \leq N}$. Furthermore, for each node $i \in \hat{V}$, we also obtain its node features F_i through calculating the network statistics of the graph $\mathcal{G}_i$. We consider 7 network statistics: average degree, betweenness, closeness, eigenvector, current flow betweenness, subgraph, and current flow closeness centralities. The description of GoG is illustrated in Figure 1.

Graph Convolutional Networks To classify each node, we adopt the Graph Convolutional Network (GCN) proposed by (Kipf and Welling 2017). The GCN conducts graph filtering operation in each convolutional layer with the filter g and the signal matrix $Z^{(\ell-1)}$, where $g = I - L$ and the initial input signal matrix $Z^{(0)} = X$. Hence, we can obtain the following expression $g_{\theta'} \star x_{\hat{V}} \approx \theta'_0 x_{\hat{V}} - \theta'_1 \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} x_{\hat{V}}$. When constraining a single parameter in this convolutional filter, the filter operation becomes $g_\theta \star x_{\hat{V}} \approx \theta(I + \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2}) x_{\hat{V}}$. To avoid the exploding and vanishing gradient, we transform $I + \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2}$ to $\tilde{\hat{D}}^{-1/2} \tilde{\hat{A}} \tilde{\hat{D}}^{-1/2}$ using the re-normalization trick, where $\tilde{\hat{A}} = I + \hat{A}$ and $\tilde{\hat{D}}_{ii} = \sum_j \tilde{\hat{A}}_{ij}$ is the degree matrix. Finally, the graph convolution operation on node-level representation learning can be formulated as $Z^{(\ell)} = \sigma(\tilde{\hat{D}}^{-1/2} \tilde{\hat{A}} \tilde{\hat{D}}^{-1/2} Z^{(\ell-1)} \Theta^{(\ell-1)})$, where $\sigma(\cdot)$ is the activation function and $\Theta^{(\ell-1)}$ is the trainable weight matrix.

Experiments

We conduct experiments on the biological benchmarks MUTAG dataset with 188 graphs of 2 classes (Yanardag and Vishwanathan 2015), and BZR dataset with 405 graphs of 2 classes (Sutherland, O'Brien, and Weaver 2003). We use 90/10% random training/test split and provide the average accuracy in 10 runs for model evaluation. All experiments are conducted on a single GeForce RTX 3090 with CUDA 11.7. We evaluate (i) Graph Convolution Network(GCN) on graphs (i.e., graph classification task) and (ii) GCN-VERTEX, GCN-LAMBDA and GCN-DELTANCE0 (i.e., based on 3 graph comparison metrics) on GoG. To evaluate the robustness of our proposed GoG framework, we also apply label noise on two graph benchmarks. In our experiment, we set the noise label rate at 20% for training labels.

	MUTAG		BZR	
Model	Before	After	Before	After
GCN	75.00 (6.00)	55.00 (10.94)	81.75 (5.28)	48.33 (13.37)
GCN-VERTEX	**88.95 (5.80)**	83.16 (7.21)	82.44 (3.91)	80.73 (5.80)
GCN-LAMBDA	87.37 (7.51)	82.10 (6.66)	82.46 (6.82)	**81.46 (4.77)**
GCN-DELTANCE0	**88.95 (5.62)**	**84.21 (9.35)**	**84.44 (6.00)**	81.22 (5.98)

Table 1: Performance of 4 models; the accuracies under "Before" and "After" are from experiments without labels attack and with attacked labels respectively.

Table 1 reports the mean accuracy and standard deviation across all models on MUTAG and BZR. From the table, our proposed method beat the vanilla GCN for both MUTAG and BZR datasets with or without labels attacked. In particular, our graph to graph method is much more robust with perturbations when vanilla GCN has a significant drop in accuracy. In terms of relative gain, the best results from our method have 18.6% and 3.29% higher accuracy without attacked labels and 53.11% and 68.55% with attacked labels for MUTAG and BZR. Hence, our Graph to Graph method approach maybe viewed as one of the most reliable methods for graph classification tasks.

Conclusion

We proposed a novel knowledge representation approach for graph learning, Graph of Graphs (GoG). GoG allows us to reformulate the problem of multiple graph classification as a node classification task. Results on bioinformatics graphs have indicated that GoG does not only improve classification accuracy, but substantially enhances robustness with respect to label perturbation attacks. In the future, we will expand GoG to prediction tasks for time-evolving graphs.

Acknowledgments

This work was supported in part by the NSF grants # ECCS 2039701 and its INTERN supplement, # OAC 2115094 and the ONR grant # N00014-21-1-2530. The views expressed in the article do not necessarily represent the views of NSF and ONR.

References

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Sutherland, J. J.; O'Brien, L. A.; and Weaver, D. F. 2003. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *J. Chem. Inf. Comput. Sci.*, 43(6): 1906–1915.

Wills, P.; and Meyer, F. G. 2020. Metrics for graph comparison: A practitioner's guide. *PLOS ONE*, 15(2): e0228728.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst*.

Yanardag, P.; and Vishwanathan, S. 2015. Deep graph kernels. In *SIGKDD*.