

Understanding Drug Abuse Social Network Using Weighted Graph Neural Networks Explainer

Zuanjie Ma(✉), Hongming Gu, and Zhenhua Liu

Emerging Technology Research Division,
China Telecom Research Institute, Beijing, China
{guhm,liuzhh11}@chinatelecom.cn

Abstract. Relational data are universal in today's society, such as social media network, citation network, and molecules. They can be transformed into graph data consisting of nodes and edges. To classify the class of nodes in graph, GNNs is a state-of-the-art model in this domain. For example, anti-drug police rely on social data to identify hidden drug abusers. However, in the deployment of GNNs, it still require interpretability techniques to support. Because GNNs embed the node information and pass them through the adjacency network, resulting in the lack of transparency in its model. However, the common explanation technique used in other models are not directly applicable on GNNs. And those explanation methods in CNNs are not applicable either considering the topological variety of graph. In order to explain our GNN model on drug abuse social network, we proposed a framework to preprocess graph data, train GNN model, and apply our Maximal Mutual Information (MMI) method to help explain the prediction made by the GNN model. We evaluate the performance of the explanation model with evaluation metrics *consistency* and *contrastivity*. The explanation model performs impressively in both real world data and two synthetic data compared to other baselines. The results also show that our model can mimic GNN explanation made by human in node classification.

Keywords: Interpretability · Graph Neural Networks · Max mutual information

1 Introduction

Many real-world applications, including social networks, rely heavily on structured data as graphs. Graph is a powerful data structure to represent relational data. It maps the features of entities as node features and the relational information between them as edges. In order to take advantage of such data structure, Graph Neural Networks (GNNs) have emerged as state-of-the-art learning approach on these data to perform node classification task, such as drug abuser identification. The principle of GNNs is to recursively incorporate information

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12951, pp. 52–61, 2021.
https://doi.org/10.1007/978-3-030-86970-0_5

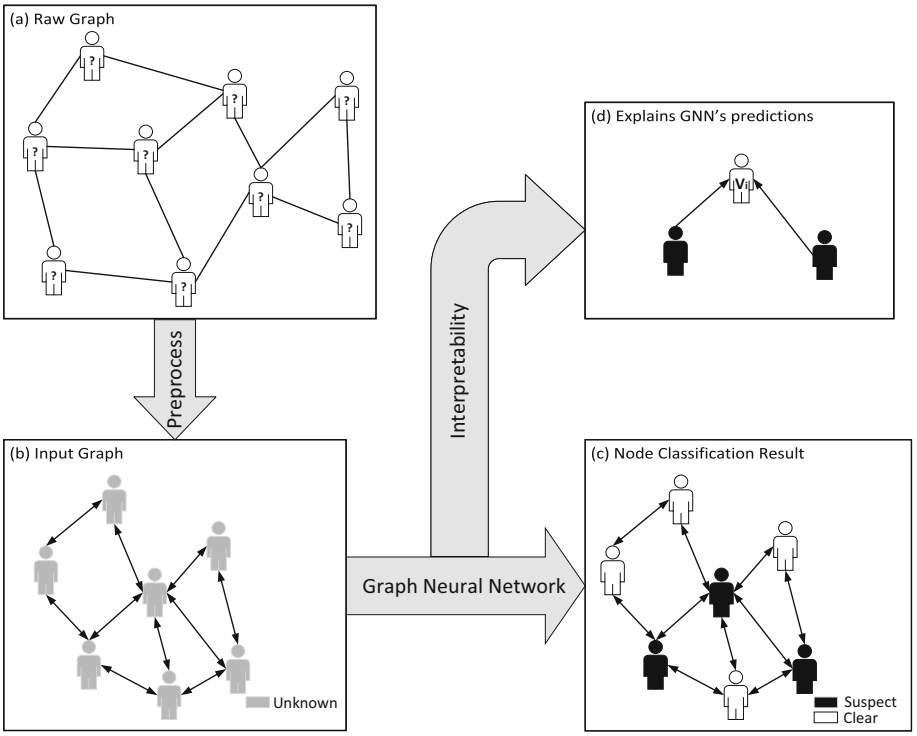

Fig. 1. The pipeline of the explanation framework. (a) represents the raw drug abuse graph to be preprocessed. (b) and (c) refer to the training and inference process of node classification task by GNN. (d) explains the prediction result of the trained model on drug abuse social data.

from neighboring nodes in graph, thus capturing both graph structure information and node or edge features. Despite their powerful representation ability, GNNs lack transparency for a human-friendly explanation for their predictions. Besides, the ability to explain the prediction of GNNs is important and useful because it can increase trust in GNNs model and transparency in decision-critical applications so that practitioners can improve its performance before deploying them in the real world. However, due to the complex data representation and non-linear transformations performed on the data, explaining decisions made by GNNs is a challenging problem. Therefore we formulate our explanation to identify the critical pattern in the input data used by the model to perform prediction. Although there are many explanation methods developed in deep learning models such as convolution neural network (CNN) [2,7,9,12], those methods are not directly applicable to explain GNNs. Although some work have been done in explaining GNN [5,8,11,13], but they fails in keeping the relational information in non-linear transformation or taking the edge weight information into account. Here we design a framework as denoted in Fig. 1 to preprocess, train, and explain

drug abuse graph data based on GNNs model. We first extract feature from the raw graph to build the input graph for GNNs. Given the node classification results from GNNs, we apply a post-hoc graph feature explanation method to formulate an explanation. Our experiments on drug abuse data and synthetic data demonstrate that our the applied methods and pipeline can generate similar explanations and evidence as human interpretation.

2 Data Preprocessing

We are given a spatial-temporal heterogeneous drug abuse social data to perform node classification task. This data consist of social connections between nodes of suspect and clear collected in 17 consecutive months. There are four kinds of edges responding to different ways of contact between nodes. In order to extract information from this dynamic dataset, we apply sliding window based data preprocessing method on this dataset. The size of window here we applied is one month. Specifically, y_u represents the ground truth label of each node v_u, it is defined by the drug abuse record in the past 17 months, 0 for clear who never exist in abuse record, 1 for suspect who have been appeared in the drug abuse record. For each node v_u, the associated features, $X_u \in \mathbb{R}^d$ is the boolean indicator of the drug abuse record in the past 10 months. And the edge weight e_{uv} is the concatenation of original features consisting of the frequency of statistic values such as number of calls and call duration. There are 17 graphs in the final processed data, representing 17 months data. Each graph contains around 2000 nodes.

3 Graph Neural Networks

3.1 Data Representation

In this section, we introduce the notation and definitions of weighted and directed graphs. We denote the graph by $G = (V, \mathcal{E})$ where V is the set of nodes, $\mathcal{E}$ the set of edges linking the nodes and X the set of nodes' features. For connected nodes, $u, v \in V$, we denote by $e_{vu} \in \mathbb{R}$ the weight of the directed edge $\langle v, u \rangle \in \mathcal{E}$ linking them. We denote $E[v, u] = e_{vu}$, where $E \in \mathbb{R}^{|\mathcal{E}|}$. For each node, v_u, we associate a d-dimensional vector of features, $X_u \in \mathbb{R}^d$ and denote the set of all features as $X = \{X_u : u \in V\} \in \left(\mathbb{R}^d\right)^{|V|}$.

Edge features contains important features about graphs. For instances, the graph G may represent a drug abuse social network, where the nodes V represent suspects or clears and the edges E are the different kinds of social connections between them.

We consider a node classification task, where each node u is assigned a label $y_u \in I_C = \{0, \ldots, C-1\}$. The two perspectives of the explanation correspond to the informative subset on E and X of the weighted graph.

3.2 Node Classification

GNN features its advantage by keeping the flow of information across neighborhoods and graphs. Besides, it is invariant to permutations on ordering. Different from the traditional GNN architectures, we focus on GNNs exploiting edge information on weighted and directed graphs. The per-layer update of our GNN model are as follows. At layer l, the update of GNN model Φ involves three key computation steps. (1) First, the model computes neural messages between every pair of nodes. The message for node pair $\langle v_u, v_v \rangle$ is a function MSG of v_u's and v_v's representations h_u^{l-1} and h_v^{l-1} in the previous layer and of the weight of edge e_{uv} pointing from u to v: $m_{uv}^l = \text{MSG}\left(\mathbf{h}_u^{l-1}, \mathbf{h}_v^{l-1}, e_{uv}\right)$. (2) Second, for each node v_u, GNN aggregates messages from v_u's neighborhood $\mathcal{N}_{v_u}$ and aggregated message M_u via an aggregation method AGG: $M_u^l = \text{AGG}\left(\left\{\mathbf{m}_{uv}^l, e_{uv}\right\} \mid v_v \in \mathcal{N}_{v_u}\right)$, where $\mathcal{N}_{v_u}$ is neighborhood of node v_u whose definition depends on a particular GNN variant. (3) Finally, GNN takes the aggregated message M_u^l along with v_i's representation $\mathbf{h}_u^{l-l}$ from the previous layer, and it non-linearly transforms them to obtain v_u's representation $\mathbf{h}^l$ at layer l: $\mathbf{h}_u^l = \text{UPDATE}\left(M_u^l, \mathbf{h}_u^{l-1}\right)$. The final embedding for node v_u after L layers of computation is $\mathbf{z}_i = \mathbf{h}_i^L$. In summary, our explainer provides explanation for any GNN that follows the computations in terms of MSG, AGG, and UPDATE.

4 Graph Feature Explanation

Next we describe our explanation approach in forms of a two-stage pipeline.

First, given the input graph $G = \{V, \mathcal{E}\}$, node feature X and true nodes labels Y, we will get a trained GNN model Φ: $G \mapsto (u \mapsto y_u)$, where prediction $y_u \in I_C$.

Second, the explanation part will provide a subgraph and a subset of features retrieved from the k-hop neighborhoods of each node u based on the classification model Φ and the node u's true label where $k \in \mathbb{N}$ and $u \in V$. The subgraph and the subset of node features are theoretically the minimal set of information the trained model Φ needed to perform the node classification task.

Here, we define the subgraph of G as $G_S = (V_S, \mathcal{E}_S)$, where $G_S \in G$, $V_S \in V$, and $\mathcal{E} \in \mathcal{E}$.

4.1 Informative Components Detection

Following the Maximal Mutual Information (MMI) Mask method in [6]. For node u, mutual information (MI) quantifies the change in the probability of prediction when u's computational graph is limited to subgraph G_S and its node features are limited to X_S. The explainable subgraph is chosen to maximize the mutual information:

$$\max_{G_S} I\left(Y, (G_S, X_S)\right) = H(Y \mid G, X) - H\left(Y \mid G_S, X_S\right) \tag{1}$$

As the trained classifier Φ is fixed, the $H(Y)$ term of Eq. 1 is constant. Therefore, it is equivalent to minimize the conditional entropy $H(Y \mid G_S, X_S)$:

$$H(Y \mid G_S, X_S) = -\mathbb{E}_{Y|G_S,X_S}\left[\log P_\Phi(Y \mid G_S, X_S)\right] \tag{2}$$

In this way, the graph component explanation w.r.t node u's prediction $\hat{y}_u$ is the subgraph G_S that minimize. Some edges in node u's computational graph may not be useful in node prediction, while others are most informative edges to aggregate information from neighborhoods. We thus aim to find the most important subgraph component in computational graph to explain the useful edges in node prediction task. In order to optimize Eq. 2, we also extends the edge mask $\mathcal{M}_{vw}$ by considering edge weight e_{vw} and adding extra regularization to make sure a connected component to be formed in the subgraph. We add constraints to the value of mask before every iteration of optimization:

$$\begin{cases} \sum_w \mathcal{M}_{vw} e_{vw} = 1 \\ \mathcal{M}_{vw} \geq 0, \end{cases} \quad \text{for } \langle v, w\rangle \in \mathcal{E}_c(u) \tag{3}$$

In this way, we optimize a relaxed adjacency matrix in our explanation rather than optimizing a relaxed adjacency matrix in graph. Thus we optimize a mask $\mathcal{M} \in [0,1]^Q$ on weighted edges, where there are Q edges in $G_c(u)$. Then the masked edge $E_c^{\mathcal{M}} = E_c \odot \mathcal{M}$, where $\odot$ is element-wise multiplication, is subject to the constraint that $E_c^{\mathcal{M}}[v,w] \leq E_c[v,w]$, $\forall (v,w) \in \mathcal{E}_c(u)$. So the objective function can be written as:

$$\min_M -\sum_{c=1}^{C} \mathbb{I}[y=c] \log P_\Phi\left(Y \mid G_c = (V_c, E_c \odot \mathcal{M}), X_c\right) \tag{4}$$

4.2 Node Feature Importance

Node features also contribute to information aggregation in propagation of message layer. Thus we also take into account the node feature in our explanation. In addition to learn a mask on weighted graph, we follow [6] to learn a mask on node attribute to filter node features in subgraph G_S. The filtered node feature $X_S^T = X_S \odot M_T$, where M_T is the feature mask to be learned, is optimized by

$$\min_{M_T} -\sum_{c=1}^{C} \mathbb{I}[y=c] \log P_\Phi\left(Y \mid G_S, X_S \odot \mathcal{M}_T\right) \tag{5}$$

The reparameterization on node attribute X is also applied in the optimization of M_T as in [10].

5 Evaluation Metrics

We can use metrics to evaluate accuracy of classification with our drug abuse data. But for these real world data, we can hardly know the ground truth of the explanation. However, we propose the evaluation metrics to quantify the explanation results.

In this paper, we define *consistency* and *contrastivity* to quantify the explanation result. We first introduce graph edit distance (GED) [1], which is a measure of similarity between two graphs. It is defined as the minimum number of graph edit operations to transform one graph to another. Thus, we follow the sense of classification that informative components detected for the node in the same class should be consist, we design *consistency* as the GED of informative components between nodes in the same class as whether the informative components detected for the node in the same class are consist. Besides, we design *contrastivity* as the GED of informative components of nodes across classes, as whether the informative components detected for the node in the different class are contrastive, to measure the contrastivity of informative components of nodes in different class.

Specifically, *consistency* is defined as below:

$$\frac{\sum_{a=1}^{N_i}\sum_{b=1}^{N_i} GED\left(G_a, G_b\right)}{N_i \times (N_i - 1)} \qquad a \neq b \tag{6}$$

where G_a and G_b are the informative components from node a and b in same class i, N_i is the number of nodes in class i. And *contrastivity* is defined as:

$$\frac{\sum_{a=1}^{N_i}\sum_{b=1}^{N_j} GED\left(G_a, G_b\right)}{N_i \times N_j} \tag{7}$$

where G_a is the informative component from node in class i, G_b the informative component from node in class j, N_i and N_j the number of nodes in class i and j.

6 Experiments

6.1 Alternative Baseline Approaches

We here consider the following alternative approaches that can provide explanation of prediction made by GNNs: (1) Grad [3] is a gradient-based method. We compute gradient of the GNN's loss function w.r.t the model input. The gradient-based score can be used to indicate the relative importance of the input feature since it represents the change in input space which corresponds to the maximizing positive rate of change in the model output. (2) GNNExplainer is also a MMI-based method to explain the prediction made by GNNs. However, it does not take the weight of edges into consideration. The experiments in this section of both real world and synthetic dataset reveals the outstanding performance of our proposed method by keeping relational information in non-linear transformation and taking edge weight information of graph into account.

6.2 Drug Abuse Social Network

In order to evaluate the performance of our explanation method, we perform node classification and different explanation methods. The node classification task on this dataset is to classify each node in graph as suspect or clear in drug abuse. Thus, we choose every pair of adjacent months to be train graph and test graph. In this way, we have 17 pair of train graph and test graph. Both node and edge in graph are associated with categorical features, and the edge features are normalized to [0, 1] as 1-D feature. We perform inductive learning on the train graph and generalize the trained GNN on test graph. The learning rate of GNN is 0.001 and we trained the model for 100 epochs until convergence. Based on the trained model, we performed the explanation methods to compare their performance in Table 1. We measured consistency and contrastivity of informative components within and between classes by selecting top 6 edges. The performance of explanation methods are averaged on all train and test graph pairs and repeated by 100 times. The lower consistency value shows that nodes in the same class have similar topology structure. The higher contrastivity value means GNN relies more on topological difference to classify nodes into different class. From the results in Table 1, GRAD performs worst since it fails to keep the structural information in propagation. While GNNExplainer performs worse than our MMI method, because it ignores the edge features information.

Table 1. Model performance

	Consistency	Contrastivity
MMI	0.42	1.24
GNNExplainer	0.47	1.18
GRAD	0.63	0.87

6.3 Synthetic Data

In order to evaluate the applicability of our model, following [10], We construct a synthetic dataset called $BA - SHAPES$ with a base Barabási–Albert (BA) [4] graph with 100 nodes and a set of 60 five-node house structured network motifs randomly attached to the nodes of the base graph. Besides, we add the extra $0.1N$ random edges to perturb the graph, where N is the number of nodes poorly connected. We assigned different weights for edges in graph as denoted in Fig. 2. We haved tried different combinations of edges weights, and as the weight of different roles of edges are sparse enough, the explanation result remains similar. The four classes of nodes are assigned according to their structure role: the top, middle, and bottom of the house and the nodes that don't belong to a house. Several natural and human-made systems, including the Internet, the world wide web, citation networks, and some social networks are thought to be approximately scale-free and certainly contain few nodes (called hubs) with

unusually high degree as compared to the other nodes of the network. All the node feature X_i was designed the same constant to constrain the node label to be determined by motif only. We randomly split 60% of the nodes for training and the rest for testing. We performed informative component detection and compared the result with ground truth in Table 2. The GRAD performed worst in this case because it fails to keep the relational structure in graph data. The GNNExplainer performed worse in this case because it ignore the attribute of edges in graph so that the nodes of base graph may usually be included into the informative subgraphs.

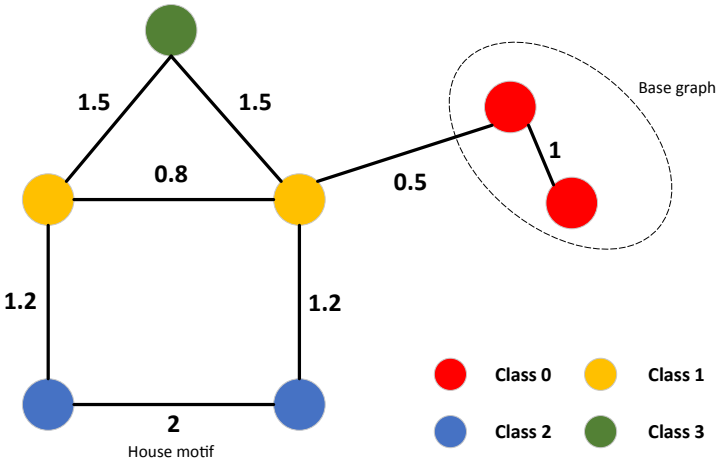

Fig. 2. The edge weights of corresponding edges in graph.

Table 2. Average AUC score of nodes in house shape

Method	MMI mask	GRAD	GNNExplainer
AUC	0.945	0.893	0.831

We also constructed a graph called $BA-FEAT$ with BA graph and designed the 2-D node attributes for each node in graph to constrain the node labels to be determined by node features only. We assigned the normal distribution $S_u \sim N(0, 0, 1)$ as the first entry of node u's feature, and $s_u + (y_u + 1) * 0.2$ as the second entry, where y_u is the ground truth label of node u. We thus constructed a graph with 100 nodes with the node features above. And we followed the same training model and methods in BA-SHAPES. To measure the performance of the explanation for node importance, we calculated the accuracy of classifying the second entry of node features as the important features in Table 3.

Table 3. Node feature importance performance (averaged on 10 times)

Method	MMI mask	GNNExplainer	GRAD
ACC	1	1	1
MSE	0.23	0.27	0.33

7 Conclusion

In this paper, we introduce the framework to preprocess and train on graph data, formulate the MMI based explanation method to explain its predictions. In order to evaluate its explanation performance, we applied it on both real world social data and synthetic data with evaluation metrics and baselines. The experiment results show that our explanation method provides comprehensive and impressive explanation of data used by GNN, while revealing the important information in topological structure. This explanation methods can benefit debugging in GNNs model, deployment in real word applications and those decision-critical area.

References

1. Abu-Aisheh, Z., Raveaux, R., Ramel, J.Y., Martineau, P.: An exact graph edit distance algorithm for solving pattern recognition problems. In: ICPRAM 2015–4th International Conference on Pattern Recognition Applications and Methods, Proceedings, vol. 1, pp. 271–278. SciTePress (2015). https://doi.org/10.5220/0005209202710278. http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005209202710278
2. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Mueller, K.R.: How to explain individual classification decisions. J. Mach. Learn. Res. **11**, 1803–1831 (2009). http://arxiv.org/abs/0912.1128
3. Baldassarre, F., Azizpour, H.: Explainability techniques for graph convolutional networks. Technical report (2019)
4. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999). https://doi.org/10.1126/science.286.5439.509. https://science.sciencemag.org/content/286/5439/509science.sciencemag.org/content/286/5439/509.abstract
5. Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., Chang, Y.: GraphLIME: local interpretable model explanations for graph neural networks (2020). http://arxiv.org/abs/2001.06216
6. Li, X., Saude, J., Reddy, P., Veloso, M., Morgan, J.P., Research, A.I.: Classifying and Understanding Financial Data Using Graph Neural Network. Technical report. www.aaai.org
7. Liu, X., Wang, X., Matwin, S.: Interpretable deep convolutional neural networks via meta-learning. In: Proceedings of the International Joint Conference on Neural Networks 2018-July, February 2018. http://arxiv.org/abs/1802.00560
8. Neil, D., Briody, J., Lacoste, A., Sim, A., Creed, P., Saffari, A.: Interpretable graph convolutional neural networks for inference on noisy knowledge graphs, December 2018. http://arxiv.org/abs/1812.00279

9. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: 2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings, December 2013. http://arxiv.org/abs/1312.6034
10. Ying, R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: generating explanations for graph neural networks, March 2019. http://arxiv.org/abs/1903.03894
11. Yuan, H., Tang, J., Hu, X., Ji, S.: XGNN: towards model-level explanations of graph neural networks. virtual event, pp. 430–438, June 2020. https://doi.org/10.1145/3394486.3403085. http://arxiv.org/abs/2006.02587
12. Zhang, J., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. Int. J. Comput. Vis. **126**(10), 1084–1102 (2016). http://arxiv.org/abs/1608.00507
13. Zhang, Y., Defazio, D., Ramesh, A.: RelEx: a model-agnostic relational model explainer, May 2020. http://arxiv.org/abs/2006.00305