GNN Interpretability Using Bayesian Inference

Shalin Patel[1,2]

[1] *Division of Applied Mathematics, Brown University*
[2] *Department of Computer Science, Brown University*

Abstract

1 Introduction

Graphs serve as a natural repository for information in many real-world applications ranging from social, informational, chemical, and biological domains [1]. Especially as data becomes more and more unstructured, graphs represent a flexible manner for storing and relating different nodes and their related features [2]. Indeed, graphs represent one of the most general mathematical structures for relating data and are seeing increasing use in modeling phenomenon such as social networks and gene regulatory networks [2, 3]. For the purposes of this work, given a set of vertices V, node features $\mathcal{X} : V \to \mathbb{R}^d$, a set of edges $E \subseteq V \times V$, and weights on the edges $W : E \to \mathbb{R}$, we consider the graph $G = \{V, \mathcal{X}, E, W\}$. Additionally, we let the space of all graphs for a given set of vertices V be $\mathcal{G}$.

1.1 Graph Neural Networks

To deal with the proliferation of graphs in computing and the need to construct models that consider graphs as a first-class member of the modeling process, a class of models known as Graph Neural Networks have emerged (GNN) with state of the art performance on a variety of classification and regression tasks [4]. Specifically, GNNs and their early iterations in GCNs took inspiration from CNNs that represented applying successive convolution operations on regular grids of information, such as images, to compose local features in the grid in to higher-level predictions [5]. At a high level, most GNN frameworks can be split into three steps `MSG`, `AGG`, and `UPD` representing a messaging step, aggregation step, and update step, respectively.

At a layer l in a GNN model ϕ, the update of the hidden state of the model occurs first by sending messages for all $(v_i, v_j) \in E$ as a function of the hidden state $\mathcal{H}_i^{l-1}$ and $\mathcal{H}_j^{l-1}$ as well as the weight $W_{ij} := W(v_i, v_j)$. Specifically, we have

$$m_{ij}^l := \texttt{MSG}(\mathcal{H}_i^{l-1}, \mathcal{H}_j^{l-1}, W_{ij})$$

Then, a GNN performs an aggregation step wherein it calculates an aggregate message for every vertex $v \in V$. Let $\mathcal{N}_k : V \times E \to \mathcal{P}(E)$ be a function that returns the edges in the k-hop neighborhood of a node. Then, we can formally write the aggregation step as

$$M_i^l := \texttt{AGG}(\{m_{ij}^l \mid v_j \in \mathcal{N}_k(v_i)\})$$

Then, finally, at each node, the GNN takes a nonlinear function (often a neural network of some sort) and applies it to this aggregated message M_i^l along with the hidden state $\mathcal{H}_i^{l-1}$ to get the new hidden state.

$$\mathcal{H}_i^l := \texttt{UPD}(\mathcal{H}_i^{l-1}, M_i^l)$$

When composed in layers, this forms a full Graph Neural Network. Note that in this framework, $\mathcal{H}_i^0 := \mathcal{X}_i$. Based on the task type, either node or graph classification in this work, further layers may be added on top of the final output. For example, in graph classification, it is often the case that the final node embeddings are concatenated and then run through an MLP to get a final classification for the whole graph [4]

1.2 Interpretation on GNNs

Given a GNN, it is natural in many fields such as computational biology to perform interpretation on the model in order to gain further insights. For example, in the case of RNA-seq data, a natural question is to determine important regulatory pathways between genes that could be related to eventual up or down regulation of a target gene [3]. One natural way to perform this task is to train a GNN on the RNA-seq data for a node-classification task and feed it a large graph G with many redundant edges. Given a GNN model ϕ with l layers and a target gene $v_i \in V$, we would like to determine $\mathcal{E}_i \subseteq \mathcal{N}_l(V_i, E)$ as well as $\mathcal{W}_i : \mathcal{E}_i \to \mathbb{R}$ such that $\mathcal{E}_i, \mathcal{W}_i$ represent the most important subgraph for the model ϕ to perform its predictions for the input vertex V_i. While there are a few criteria for determining importance, we consider importance to be the maximal mutual information between the original model with its inputs and the same model with the estimated subgraph $\mathcal{E}_i, \mathcal{W}_i$. Written more formally, we wish to discover

$$\arg\max_{\mathcal{E}_i, \mathcal{W}_i} \texttt{MI}(\phi(V_i, \mathcal{X}, E, W), \phi(V_i, \mathcal{X}, \mathcal{E}_i, \mathcal{W}_i))$$

Clearly, this is a hard problem as a brute force search would take $O(2^{|E|})$ time even when discounting the weight array $\mathcal{W}_i$.

1.3 Bayesian Inference

2 Related Work

3 Methods

4 Experimental Setup

5 Results

6 Discussion

References

[1] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1082–1090. Association for Computing Machinery.

[2] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. 5(1):59–68.

[3] Francesca Petralia, Won-Min Song, Zhidong Tu, and Pei Wang. New method for joint

network analysis reveals common and different coexpression patterns among genes and proteins in breast cancer. 15(3):743–754.

[4] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering.