Qualifying Exam

Art Tay

Introduction

- Overview of the problem area.
  - Graphs are an important data structure.
    - GRAPHS provide an incredibly flexible structure for modeling complex data. Data can naturally appear as graphs, like molecules. We can reduce data to a graph, such as the key points of a image. We can even use graphs to add structure, such as grammatical relationships.
  - GNN models are good at prediction and inference on graph data.
    - Graph Neural Networks (GNNs) have become a popular choice for prediction and inference on graph data. At their core, GNNs work by iteratively updating node embeddings based on information from neighboring nodes. The idea is to use the graph's structure to engineer better features. This message passing scheme allows GNNs to capture complex dependencies and patterns present within the graph structure. GNN architectures typically consist of multiple layers, each performing message passing and aggregation operations to refine the embeddings. These layers are often followed by pooling and dense prediction layers to produce the final output.
  - There are many important applications for graph classification models.
    - Some important applications of graph classification include predicting chemical toxicity (Bai et al. 2019), classifying proteins (Gallicchio and Micheli 2019), and even detecting cancer from pathology slides (Xiao et al. 2023).
  - **Problem:** While GNNs achieve remarkable predictive power, their complexity prevents the exaction of the scientific rationale.
- Why is the problem important?
  - Explaining or interpreting GNN predictions would
    - help with the adoption of such models for critical applications,
    - prevent adversarial attacks,
    - detect potential implicit discrimination,
    - guide scientific as well as machine learning research.
- How does the problem relate to the fundamentals areas of Statistics?
  - Explain-ability vs Interpretability
    - Yuan et al. (2022)
    - A model is interpretable if the models decision process can be readily understood by humans. For example, a linear regression model is interpretable because the coefficient clearly define how any prediction get made.

        - A model is explainable if the models prediction can be reasoned post-hoc. Permuting each variable and measuring the variation in the predictions can be used to estimate each variables marginal effect [cite].
    - One goal would be to create a GNN type model whose decision process is human interpretable. A straight translation from statistics would be a circuit type analysis [cite]. For graphs, this would mean some form of coefficients on subgraphs producing the prediction.
    - Another goal might be to develope a method that determines if a feature is statistical significant to the GNN model. The challenge is that the graph features that matter to researchers aren't necessarily tabular.
- What is the impact of solving this problem?
    - In the application where GNNs have shown strong predictive power, we can exact a testable scientific hypothesis for the nature of the classification.
    - In the application where GNNs have weak predictive power, highlight the potential misunderstandings the model is having.

Notation

- Let G denote a graph.
- Any graph G can be describe by X, A, E. The node feature matrix, edge feature matrix, and adjacency matrix respectively
- Let $X = [X_c,\ X_d]$, where X_c is the subset of continuous node features and X_d is the subset of one-hot discrete node features.
- Let $E = [E_c,\ E_d]$, denoted in the same manner.
- Let n represent the number of nodes in the graph and v represent the number of edges.
- Let $\text{feat}_{(.)}$ denote the number of features or columns in the the corresponding feature matrix.
- A is a binary $n \times n$ matrix where $A[i,\ j] = 1$ indicates that an edge exists between nodes labeled i and j.
- Let $\text{explainee}(G;\ \Omega) = h_G^{(1)}, \dots h_G^{(L)}, \rho_G$ be an L layer GNN model with parameters Ω that we would like to explain.

Analysis of Core Papers

GNNInterpreter

(Wang and Shen 2024)

- Overview
    - Instance v. Model Level
        - In general, explanation methods serve to elucidate which features within the data influence disparate predictions. These methods typically fall into two categories: instance-level and model-level. Instance-level explanations aim to unveil the model's

rationale behind a particular prediction. In domains such as image and text analysis, a prevalent approach involves masking or perturbing the instance and assessing the impact on the model's prediction. On the other hand, model-level explanations seek to understand how a model generally distinguishes between classes. In image and text analysis, for instance, one common technique involves treating the input as a trainable parameter and optimizing the model's prediction towards a specific class. Consequently, the resulting optimized input comprises a set of features strongly associated with the targeted class.

  - GNNInterpreter provides model level explanations for GNN in this manner.
  - Formally, GNNInterpreter tries to learn the graph generating distribution for each class.
  - GNNInterpreter works by optimizing the parameters of a generic graph generating distribution to produce samples that closely match the explainee's understanding of the targeted class.

- Explanation of the graph generating distribution.
  - Graph generating distributions are hard to specify because there can be discrete and continuous elements of X, E and A. Furthermore, the interactions between these matrices can be complex.
  - The authors tackle these issues by making two simplify assumptions.
    1. Assume that G is a *Gilbert* random graph, every possible edge as an independent fixed probability of occurring.

       $$\forall(i,\ j) \neq (k, l)\ Pr(A[i,\ j] = 1) \perp Pr(A[k,\ l] = 1)$$

    2. The features of every node and edge are independently distributed.
  - The author justify these assumptions by:
    1. The other graph distributions aren't suitable.
       a. Erdo-Renyi graphs have a fixed number of edges and nodes.

       b. Rado graphs are infinite in size.
       c. The random dot-product graph model is just a generalization of Gilbert random graphs.
    2. Because the parameters of the independent distributions will be updated jointly using the *explainee* model, the *explainee's* understanding of the latent correlation structure should be contained in the final estimates.
  - X_c and E_c can be sampled from any continuous distribution that can be expressed as a location-scale family. Separating the stochastic and systematic components is necessary for gradient based optimization. It is commonly known as the "re-parametrization trick".
  - X_d, E_d as well as A need to be sampled from a continuous distribution for gradient based optimization, but the distribution has to have sampling properties close to a discrete distribution.

- The author assume that the true underlying distribution for every discrete node and edge feature is *categorical.* The categorical distribution is also know as the multi-bernoulli, where every sample has a fixed probability of being in one of the discrete categories.

- Suppose there are D categories with probabilities $\pi_\omega = \dfrac{\theta_\omega}{\sum_{i \in D} \theta_i}$. Then

$$I = \underset{i \in D}{\operatorname{argmax}} \; \log \theta_i + G^{(i)} \sim \operatorname{Cat}(\pi)$$

  where $G^{(i)} \overset{i.i.d.}{\sim} \operatorname{Gumbel}(0, 1)$.

- The intuition is that the Gumbel or extreme value distribution is the density of the maximum order statistic of i.i.d. standard normals which makes it a good candidate for model the winning or maximum probability category. Adding Gumbel noise to the logits should maintain the true relative proportions, but enough skewness such that every category has some probability of having the maximum noised logit.

- **Proof 1:** In order for I to be a true categorical distribution, $Pr[I = \omega] = \pi_\omega$. $I = \omega$ if and only if $\log \theta_\omega + G^{(\omega)} > \log \theta_i + G^i \; \forall i \in D \setminus \omega$. Let M_i denote a random variable that follows a $\operatorname{Gumbel}(\log \theta_i, 1)$ distribution.

$$\begin{aligned}
Pr[I = \omega] &= \mathbb{E}_{M_\omega} \prod_{i \in D \setminus \omega} Pr(M_i < m_\omega) \text{ i.i.d location shifted Gumbel distributions.} \\
&= \mathbb{E}_{M_\omega} \prod_{i \in D \setminus \omega} \exp\left(-e^{\log \theta_i - m_\omega}\right) \text{ Gumbel CDF.} \\
&= \mathbb{E}_{M_\omega} \exp\left(- \sum_{i \in D \setminus \omega} e^{\log \theta_i - m_\omega}\right) \\
&= \int_{-\infty}^{\infty} \exp\left(\log \theta_\omega - m_\omega\right) \exp\left(-e^{\log \theta_\omega - m_\omega}\right) \cdot \exp\left(- \sum_{i \in D \setminus \omega} e^{\log \theta_i - m_\omega}\right) \; dm \\
&\text{Gumbel PDF.} \\
&= \int_{-\infty}^{\infty} \exp\left(\log \theta_\omega - m_\omega\right) \exp\left(- \sum_{i \in D} e^{\log \theta_i - m_\omega}\right) \; dm \\
&= \int_{-\infty}^{\infty} \theta_\omega \exp\left(-m_\omega\right) \exp\left(-e^{-m_\omega} \sum_{i \in D} \theta_i\right) \; dm \\
&= \pi_\omega \sum_{i \in D} \theta_i \int_{-\infty}^{\infty} \exp\left(-m_\omega\right) \exp\left(-e^{-m_\omega} \sum_{i \in D} \theta_i\right) \; dm \text{ From the above definition of } \pi_\omega \\
&= \pi_\omega \sum_{i \in D} \theta_i \frac{\exp\left(-e^{-m_\omega} \sum_{i \in D} \theta_i\right)}{\sum_{i \in D} \theta_i} \Bigg|_{-\infty}^{\infty} \\
&= \pi_\omega \sum_{i \in D} \theta_i \frac{1}{\sum_{i \in D} \theta_i} = \pi_\omega
\end{aligned}$$

  Reference: Huijben et al. (2022)

- Using inverse CDF sampling and and relaxing the argmax to a Softmax, we can sample one-hot categorical vectors based on two parameters θ_{Cat}, a trainable parameter vector of length equal to the number of categories, and τ, a hyperparameter that controls the degree of relaxation (smaller value approximate the discrete sampling better, but can result in numerical issues).

$$\text{Softmax}\left(\frac{\theta_{\text{Cat}} - log(-log\ \epsilon)}{\tau}\right), \quad \epsilon \sim U[0,1]$$

This method, known as the concrete distribution (Maddison, Mnih, and Teh 2017), yields a reasonable smooth gradient w.r.t. to the probability parameters.

- The adjacency matrix can be sampled in a similar manner since the Bernoulli is just a special case of the categorical.

$$\text{sigmoid}\left(\frac{\theta_A + \log\ \epsilon - \log(1 - \log\ \epsilon)}{\tau}\right)$$

This is known as the binary concrete distribution (Maddison, Mnih, and Teh 2017).

- Notate the combined graph generating distribution as:

$$G_{\text{gen}} \sim \text{gen}(\Theta)$$

where Θ is the set of all parameters from the independently sampled distributions.

- Prediction objective.
  - An obvious objective is to maximize the likelihood that the *explainee* model predicts a sampled graph to be a member of the target class.
  - Let $\tilde{\rho}$ denote the desired predicted probability vector. Then the above objective can be expressed as:

$$\underset{\Theta}{\text{argmin}}\ \mathcal{L}_{\text{pred}}(\Theta) = \mathbb{E}_{G_{\text{gen}}}\ \text{CrtEnt}(\text{explainee}(G_{\text{gen}}), \tilde{\rho})$$

- Embedding objective.
  - The authors note that …
- Regularization terms.
- Summary of Results + Figures
  - Metrics

D4Explainer

(Chen et al. 2023)

- Note on counter-factual explanations.
- Graph diffusion.

ProtGNN

(Zhang et al. 2021)

Synthesis of Core Papers

- Comparison of generation methods.
  - GNNInterpreter uses continuously relaxed discrete distributions.
  - D4Explainer uses diffusion.
  - Diffusion is slower, but can be more realistic. Probably because diffusion is less subject to the **out-of-distribution (OOD) problem**.
  - Prototype projection are like generative methods. Restricted to in distribution, but realism is all but guaranteed.

Technical Details

- Minimal reproduction of each method on MUTAG.

Future Directions

References

Bai, Yunsheng, Hao Ding, Yang Qiao, Agustin Marinovic, Ken Gu, Ting Chen, Yizhou Sun, and Wei Wang. 2019. "Unsupervised Inductive Graph-Level Representation Learning via Graph-Graph Proximity." https://arxiv.org/abs/1904.01098.

Chen, Jialin, Shirley Wu, Abhijit Gupta, and Rex Ying. 2023. "D4Explainer: In-Distribution GNN Explanations via Discrete Denoising Diffusion," no. arXiv:2310.19321 (October). https://doi.org/10.48550/arXiv.2310.19321.

Gallicchio, Claudio, and Alessio Micheli. 2019. "Fast and Deep Graph Neural Networks." https://arxiv.org/abs/1911.08941.

Huijben, Iris A. M., Wouter Kool, Max B. Paulus, and Ruud J. G. van Sloun. 2022. "A Review of the Gumbel-Max Trick and Its Extensions for Discrete Stochasticity in Machine Learning," no. arXiv:2110.01515 (March). https://doi.org/10.48550/arXiv.2110.01515.

Maddison, Chris J., Andriy Mnih, and Yee Whye Teh. 2017. "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables," no. arXiv:1611.00712 (March). https://doi.org/10.48550/arXiv.1611.00712.

Wang, Xiaoqi, and Han-Wei Shen. 2024. "GNNInterpreter: A Probabilistic Generative Model-Level Explanation for Graph Neural Networks," no. arXiv:2209.07924 (February). https://doi.org/10.48550/arXiv.2209.07924.

Xiao, Guanghua, Shidan Wang, Ruichen Rong, Donghan Yang, Xinyi Zhang, Xiaowei Zhan, Justin Bishop, et al. 2023. *Deep Learning of Cell Spatial Organizations Identifies Clinically Relevant Insights in Tissue Images*. Preprint. In Review. https://doi.org/10.21203/rs.3.rs-2928838/v1.

Yuan, Hao, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2022. "Explainability in Graph Neural Networks: A Taxonomic Survey," no. arXiv:2012.15445 (July). https://doi.org/10.48550/arXiv.2012.15445.

Zhang, Zaixi, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2021. "ProtGNN: Towards Self-Explaining Graph Neural Networks," no. arXiv:2112.00911 (December). https://doi.org/10.48550/arXiv.2112.00911.