GNN Interpretation Thesis Project

Shalin Patel

GNNs and Computational Biology

- GNNs useful for creating multi-modal models of complex biological systems (e.g. Gene Expression)
  - Based off HiC contact data
  - Based off Gene Regulatory Networks (GRNs)
- These models often demonstrate state of the art performance
- Many biological datasets demonstrate graphical structure and are often coerced into unnatural forms
  - GNNs remove a lot of these issues

Interpretation of GNNs

- While GNNs give better performance than previous models, interpretating them is harder than before
- Many traditional models have well established interpretations
  - SVMs, Random Forests, Logistic Regression all have natural interpretations
  - CNNs also have many interpretation methods including saliency maps, LIME, SHAP, ...
- To derive understanding, and new biological value from the improved performance, interpretations for GNNs need to be generated
- Specifically, we want to understand what parts of the underlying graph structure are driving newfound performance

Problem Formulation

Let $\mathcal{G}$ denote a graph on edges E and nodes V such that each node is associated with $\mathcal{X} = \{x_1, \ldots, x_n\}$.

Let $f : V \mapsto \{1, \ldots, C\}$ represent a classification task on nodes in V.

Specifically, for a node v, a GNN ϕ that approximates f learns the conditional distribution $P_\phi(Y \mid G_c(v), X_c(v))$ where G_c and X_c represent the computaitonal graph of v (typical the n-hop neighborhood). Predictions are given as $\hat{y} = \phi(G_c(v), X_c(v))$.

The goal of GNN interpretation is to find $\psi : V \mapsto \mathcal{G}$ where $\psi(v) = G_s(v)$ such that $G_s(v) \subseteq G_c(v)$ and

$$MI(\hat{y}, \phi(G_s(v), X_c(v)))$$

is maximized.

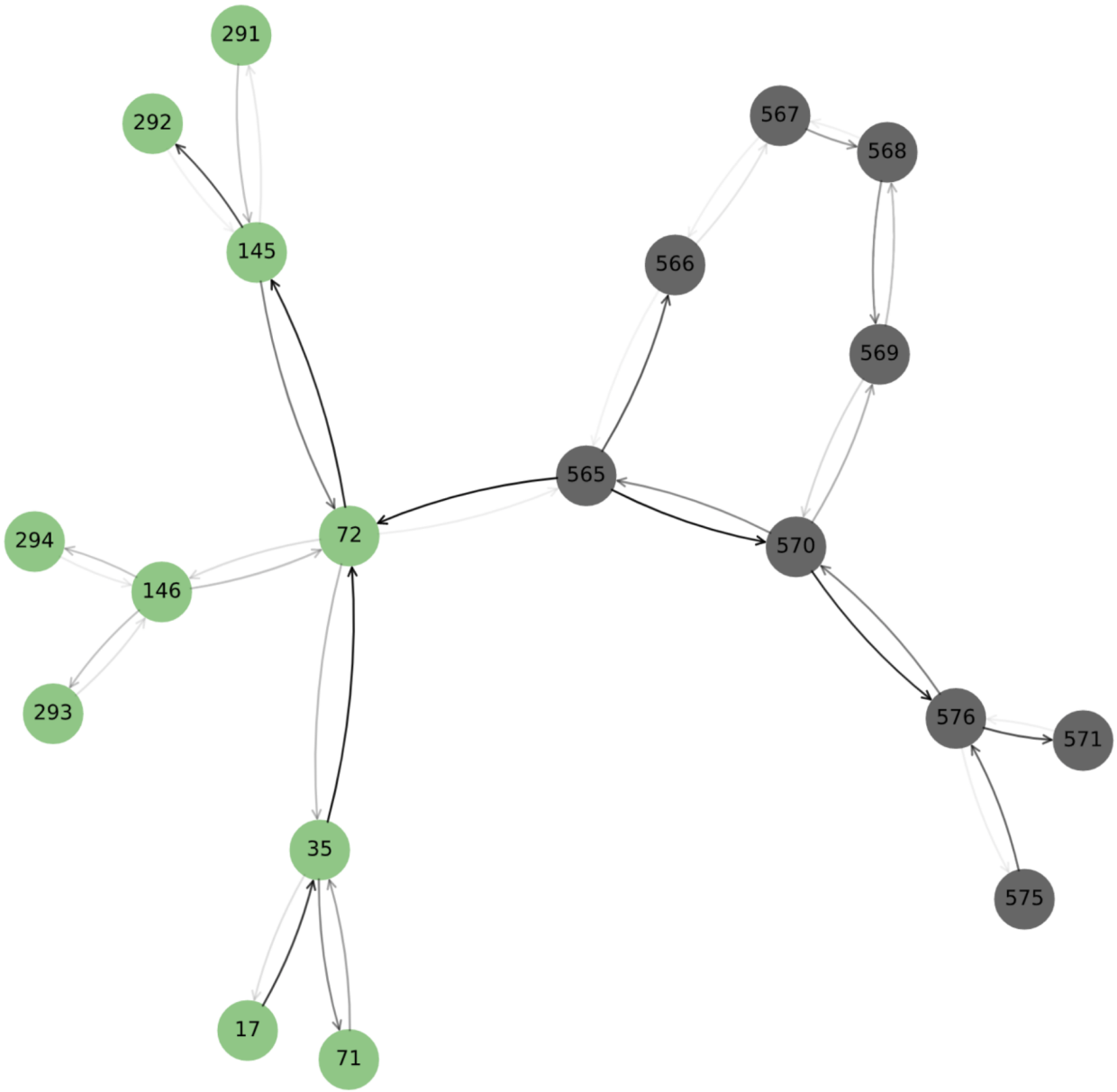
GNNExplainer

- GNNExplainer works by learning a continuous mask $\sigma(M)$ such that $A_c \odot \sigma(M)$ represents the interpretation
- GNNExplainer assumes $P_{\mathcal{G}}(G_S) = \prod_{(j,k) \in G_c} A_s[j,k]$ meaning that each edge is treated as an independent Bernoulli RV
- This mask is learnt via backpropogation on the mutual information with respect ot these $A_s[j,k]$.
- In practice, this ignores all sorts of conditionality between the edges and is not rich enough to capture complex interactions between edges
  - Does not perform well in replicaiton studies
  - Has extremely high variance and does not converge consistently

PGExplainer

- The current SOTA (based on some flawed benchmarks)
- Has about 0.78 AUC in replication studies
- Requires a split between graph embedding and inference model to work

Benchmark Dataset

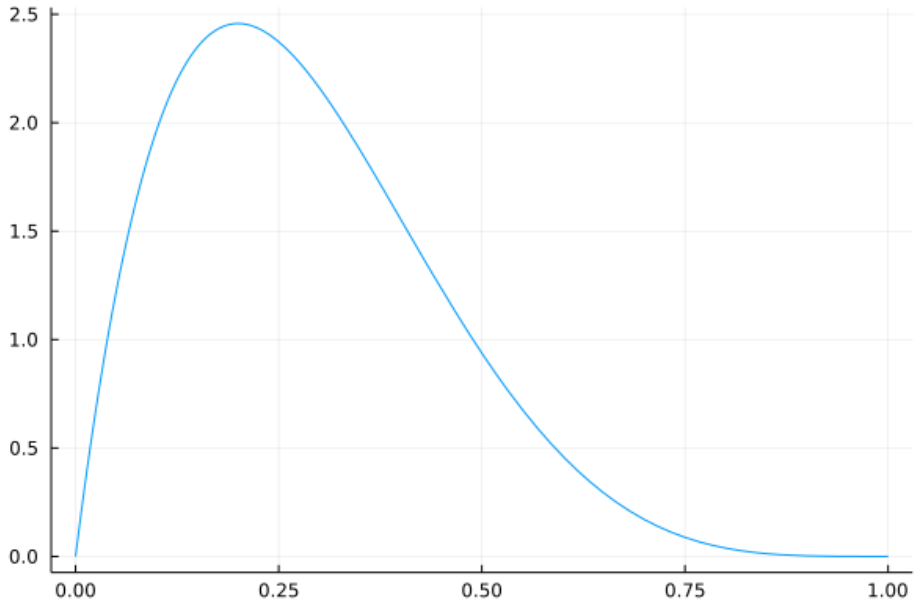


Bayesian Formulation

- Specify a model $\bar{\psi}$ for sampling $G_s(v)$
- Given a mask M sampled from this distribution, condition $\phi(M, X_c(v))$ on $\phi(G_c(v), X_c(v))$ and update the parameters of $\bar{\psi}$ using SVI
- There are many different models we can try
- We want a good tradeoff between structure and flexibility to capture the complex conditional interactions that are happening in the underlying dataset

Beta-Bernoulli Model

- Sampled each edge from a Beta-Bernoulli variable with the following prior on a given edge



- Does not perform super well $\sim$ on par with GNNExplainer (makes sense these are essentially the same model)

Random Walk Model

- Worked by randomly choosing an node adjacent to v
- Sampled edges in the set adjacent to all added nodes and continued until no more nodes to add with each edge being drawn from an independent Bernoulli distribution
- Performed a bit better reaching ~ 0.55 AUC

Normalizing Flow Model

- If there are e edges in $G_c(v)$, we have an e-dimensional base mv-normal distribution
- Uses and invertible spline transform-based normalizing flows to transform base distribution to a target distribution over all edges
- Sampled from this target distribution to build an edge mask
- Gives the most flexibility
- Performs around ~ 0.60 AUC

Next Steps

- Defintely have some bugs that are leading to suboptimal training
- Reevaluate how to evaluate these models
- Evaluate PGExplainer against actual ground truth examples
- Explore different transforms in the normalizing flow model
- Combine random walk with normalizing flow