PROJECT COL865: FRCGNN

Harsh Pandey
Department of Computer Science
Indian Institute of Technology, Delhi
New Delhi, India
harsh.pandey@cse.iitd.ac.in

Srikanta Bedathur
Department of Computer Science
Indian Institute of Technology, Delhi
New Delhi, India
srikanta@cse.iitd.ac.in

ABSTRACT

This is the final project of course COL865, under the guidance of Prof. Sirkanta Bedathur. The main aim of this paper is to combine the use of FROCC Bhattacharya et al. (2020) and GNNs and compare the results by testing it against the paper Ivanov & Prokhorenkova (2021).

1 INTRODUCTION

Graph neural networks (GNNs) are well known algorithms for graph structures. They are widely used in areas like recommender systems, combinatorial optimization and many more.

Although they perform well with sparse data either homogeneous data or bag-of-words representation. But this is not always the case in real life. There are many instances where the data is more detailed and has rich semantics among nodes, and at these places GNNs take a hit.

As we already saw in the paper Ivanov & Prokhorenkova (2021) it is possible to boost the GNNs performance over these dense tabular data. They did this using Gradient Boosted Decision Trees (GBDT). GBDTs are very successful on tabular dataset but they don't perform well on graphical data, where there is some relation between the connected nodes.

FROCC Bhattacharya et al. (2020) is another once class classifier algorithm which takes the projection of data along different directions and then use distance to categorize whether the given point belongs to the class or is an outlier.

Our main motive is to try to use this projection technique as an advantage to get a better representation of tabular data, and then use GNNs to finally train the model on this new representation. We show an simple architecture FrcGNN, which somewhat similar to ResGNN and not BGNN of Ivanov & Prokhorenkova (2021) because we don't train FROCC and GNN combined, as FROCC is not a gradient based algorithm.

2 BACKGROUND

Let $G = (V, E)$ be a graph with nodes having features and target labels. In node prediction tasks (classification or regression), some target labels are known, and the goal is to predict the remaining ones. Throughout the text, by lowercase variables $x_v (v \in V)$ or x we denote features of individual nodes, and X represents the matrix of all features for $v \in V$. Individual target labels are denoted by y_v, while Y is the vector of known labels.
Graph Neural Networks (GNNs) use both the network's connectivity and the node features to learn latent representations for all nodes $v \in V$. Many popular GNNs use a neighborhood aggregation approach, also called the message-passing mechanism, where the representation of a node v is updated by applying a non-linear aggregation function of v's neighbors representation (Fey & Lenssen, 2019). Formally, GNN is a differentiable, permutation-invariant function $g_\theta : (G, X) \to \hat{Y}$, where $\hat{Y}$ is the vector of predicted labels. Similar to traditional neural networks, GNNs are composed of multiple layers, each representing a non-linear message-passing function:

$$x_v^t = \text{COMBINE}^t(X_v^{t-1}, \text{AGGREGATE}^t(\{(X_w^{t-1}, x_v^{t-1}) : (w, v) \in E\})), \tag{1}$$

where x_v^t is the representation of node v at layer t, and COMBINE^t and AGGREGATE^t are (parametric) functions that aggregate representations from the local neighborhood of a node. Then, the GNN mapping g_θ includes multiple layers of aggregation 1. Parameters of GNN model θ are optimized with gradient descent by minimizing an empirical loss function $L_{GNN}(Y, g_\theta(G, X))$.

Fast Random projection based Once Class Classifier (FROCC) Bhattacharya et al. (2020) is a one-class classifier using random projections, defined as follows:

Definition 1 (FROCC (S, N, ε, K))**.** *Assume that we are given a set of training points $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subseteq \mathbb{R}^d$. Then, given an integer parameter $N > 0$, a real parameter $\varepsilon \in (0, 1]$ and a kernel function $K(\cdot, \cdot)$, the ε-*separated Fast Random-projection based One-Class Classifier *FROCC (S, N, ε, K), comprises, for each i, $1 \leq i \leq N$,*

1. *A* classifying direction $\boldsymbol{w}_i$ *that is a unit vector chosen uniformly at random from $\mathbf{1}_d$, the set of all unit vectors of $\mathbb{R}^d$, independent of all other classifying directions, and*
2. *a set of intervals R_i defined as follows: Let $S_i' = \{K(\boldsymbol{w}_i, \boldsymbol{x}_j) : 1 \leq j \leq m\}$ and $S_i = \{y_1, \ldots, y_m\}$ is a shifted and scaled version of S_i' such that $y_i \in (0, 1), 1 \leq i \leq m$. Assume $y_1 \leq \cdots \leq y_m$. Then each interval of R_i has the property that it is of the form $[y_j, y_{j+k}]$ for some $j \geq 1, k \geq 0$ such that*
   (a) *for all t such that $0 \leq t < k$, $y_{j+t+1} - y_{j+t} < \varepsilon$,*
   (b) *$y_j - y_{j-1} > \varepsilon$ whenever $j - 1 > 0$, and*
   (c) *$y_{j+k+1} - y_{j+k} > \varepsilon$ whenever $j + k + 1 \leq m$.*

Given a query point $\boldsymbol{y} \in \mathbb{R}^d$, FROCC (S, N, ε, K) calculates the loss as

$$\ell(FROCC, \boldsymbol{y}) = 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{K(\boldsymbol{w}_i, \boldsymbol{y}) \text{ lies within some interval of } R_i}.$$

FROCC returns YES if the loss is less than a threshold, otherwise it returns NO.

Note this is a once class classifier, for using it as a multi-class classifier, we can just train one classifier for each class and use any aggregation method to combine the results of each classifier to get the prediction.

3 FROCC WITH GNN

In this section we will introduce how we combine these two methods.

We first have different FROCC classifier for each unique label (L) of the dataset. The FROCC is trained as a one class classifier for every label of the dataset. This is also visible in the figure where we show multiple FROCC blocks in green. The output of the FROCC will not be the prediction of label for every class rather the score which is

$$\text{Score}(i) = 1 - \ell(FROCC, \boldsymbol{y})$$

We train FROCC for every label we then combine the outputs using some aggregation function. The functions used are as follows

1. A direct approach is to use the max value over all the scores generated.
$$\max_{i \in L}(\text{Score}(i))$$
2. The option we tried was to concatenate all the predicted scores for every class and then pass it on to the GNN.
3. Here we used a single layered neural network to which takes a vector from $\mathbb{R}^{|L|}$ and converts it into just a scalar value ($\mathbb{R}$). This model is trained independently, where the input is the prediction ofFROCC and the output is the expected prediction. Note that we train this like a regression model and not a classification model. Just using classification model will decrease the information content as at last we will have to do some thresholding.

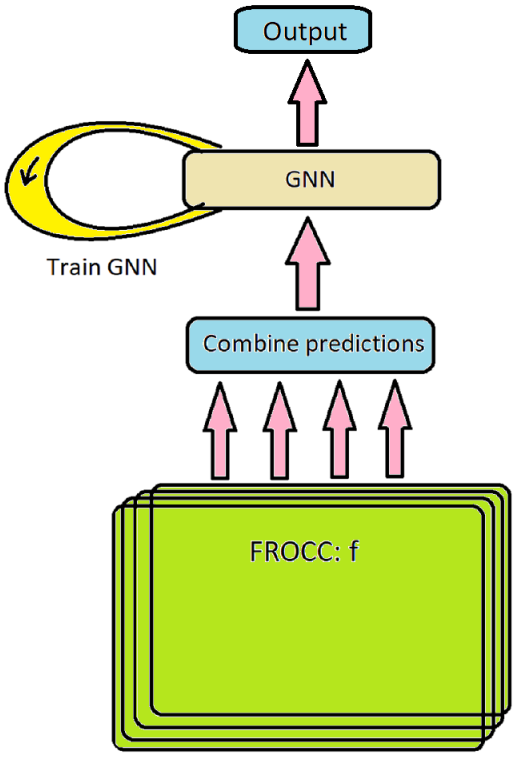

Figure 1: This is a picture showing the architecture of FrcGNN. Note multiple green boxes represent FROCC for every label of dataset.

4 EXPERIMENTS

We have performed multiple experiments using this model. We also show the comparison of these models with ResGNN and BGNN from paper Ivanov & Prokhorenkova (2021). The results are put in the table1.

We noted that when training just on FROCC and combining them using the arg max approach the accuracies are very low (as visible in table 1 Row 1, so we dove deep into FROCC. It was interesting to note that the label wise FROCC scores are very good but when combining them they take a hit. **This was the main inspiration to use the individual scores of classes as encoding and passing them to GNNs**. We show one more table having class wise comparison of the results for every dataset 2

Accuracy on Different Datasets			
Models	house	dblp	slap
LightGBM	0.55	0.913	**0.963**
GNN	0.625	0.83	0.895
ResGNN	**0.625**	**0.892**	0.905
Frocc	0.314	0.165	0.147
FrcGNN, 1d	0.600	0.804	0.913
FrcGNN, 2d	0.595	0.830	**0.916**
FrcGNN, 1d MLP	0.612	0.807	0.909
FrcGNN, enc_1	0.602	**0.847**	**0.916**

Table 1: This is the table showing comparison with different datasets and models, top results are highlighted in **bold** excluding LightGBM model

Frocc	house	dblp	slap
Label's best acc.	0.582	0.873	1.0
Label's worst acc.	0.525	0.603	0.858
Average acc.	0.557	0.755	0.968

Table 2: This table shows the label wise accuracy of FROCC, which is way better then the combined we got using arg max over all

REFERENCES

Arindam Bhattacharya, Sumanth Varambally, Amitabha Bagchi, and Srikanta J. Bedathur. Frocc: Fast random projection-based one-class classification. *ArXiv*, abs/2011.14317, 2020.

Sergei Ivanov and Liudmila Prokhorenkova. Boost then convolve: Gradient boosting meets graph neural networks. *ArXiv*, abs/2101.08543, 2021.