Variable Proximity-Based Graph Structure for Emission Modeling and Prediction

Mark Lisi
Department of Computer Science
Yale University
New Haven, CT, 06511
`mark.lisi@yale.edu`

1 Introduction

As the world continues to industrialize, it becomes more and more important to understand the effects of human-generated emissions on our planet. Accurate and timely climate modeling and prediction are essential for addressing pressing environmental and societal challenges, such as extreme weather events, sea-level rise, and food security.

Traditional climate models often rely on grid-based representations of Earth's surface and atmosphere, which can lead to challenges in capturing fine-grained spatial dependencies and intricate climate phenomena. In contrast, graph neural networks lend themselves well to tasks involving structured data, such as geographic emission records. This project introduces a novel approach to representing geographic data in a graph format based on physical proximity - we are then able to optimize accuracy in node prediction tasks by tuning the distance threshold required to denote an edge.

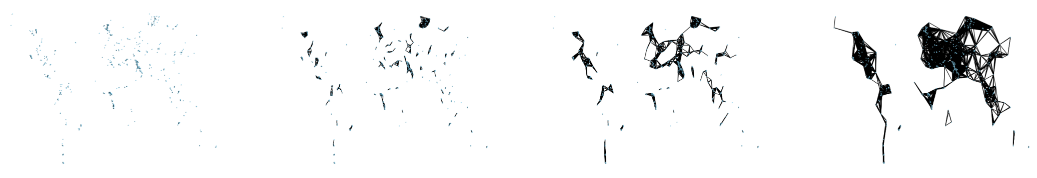
Through a series of experiments, this project highlights a clear improvement in the accuracy of a graph convolutional network over the performance of a multi-layer perceptron. Both regression and classification tasks saw improvement by converting standard tabular data into a graph-structured format based on physical proximity. Varying the distance threshold required to draw an edge seems to have limited effect on the ultimate result of the model, but affects the rate of convergence. Overall, the merit of converting standard tabular data to an artificially structured dataset is made clear by significant improvements in performance on classification and regression tasks.

2 Related Work

Similar structure-manipulating approaches have been studied to capture local detail. The 2021 paper "EdgeNets: Edge Varying Graph Neural Networks" introduces a general framework called EdgeNet which allows nodes to use different parameters to weigh information from neighbors, learning edge-neighbor dependent weights. The key characteristic of EdgeNet as described in the paper is the ability to assign trainable parameters to every edge within each shift. Although this approach enhances the GNN's ability to represent information, it can hinder its capacity to adapt to new, unseen graphs - also known as inductive capability. Still, the performance of EdgeNet demonstrated some interesting phenomena in this research: specifically, a tight connection between graph convolution and graph attention mechanisms. EdgeNet was tested on synthetic and real-world graph signal classification problems, including some standard datasets such as the Facebook dataset from SNAP Stanford[1]. They used a variety of sophisticated architectures including a graph attention network (GAT) and a spectral edge varying GNN.

There has also been some work done using GNNs in the environmental science sphere. Researchers from DAMO Academy and Zhejiang University have created WeatherGNN, a graph neural network

[1]https://snap.stanford.edu/data/egonets-Facebook.html



((a)) α = 100 ((b)) α = 500 ((c)) α = 1000 ((d)) α = 2000

Figure 1: Visualization of how connectivity threshold α affects graph structure

for numerical weather prediction. The model, which acts as a bias-correction method on traditional numerical weather predictions (NWP), interprets spatial weather data in a grid format, then adaptively captures meteorological interactions within each grid. Their model was able to significantly improve on the results generated by NWP, demonstrating an average improvement of 40.5% RMSE when tested on two real-world datasets covering different terrain types in China. Their approach includes two separate GNNs for different purposes: one is a factor-wise GNN that captures meteorological interactions within each grid adaptively, and the other is a computationally efficient hierarchical GNN that captures dependencies between grids.

3 Method

For this project we worked with the Climate TRACE 2021 dataset. The dataset is a publicly available record of emissions data partitioned into sectors such as agriculture, transportation, fossil fuel operations, and manufacturing (full list available on the Climate TRACE website). Each sector contains emission measurements of four common pollutants (CO_2E, CO_2, CH_4, N_2O) at discrete geographic locations, and each measurement site is binned into a subcategory. The manufacturing sector, for instance, has measurements for aluminum, cement, pulp/paper, and steel plants.

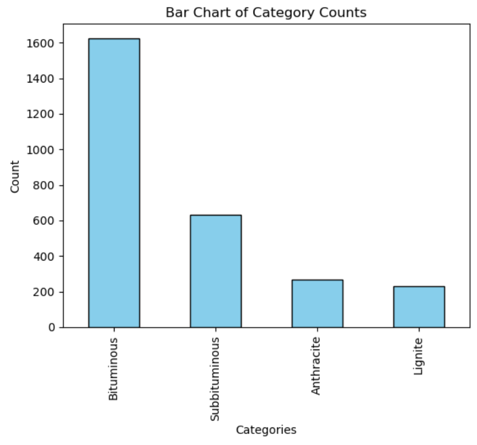
Notably, this data is not inherently graph structured - thus, the first step of our methodology is to generate edge representations for the given tabular data based on physical proximity. Each measurement site included in the dataset comes with a latitude and longitude indicating position - from this information we can generate edge representations where all nodes within a certain distance threshold are connected. We will then vary this threshold as a hyperparameter to find an optimal connectivity for node prediction tasks. As a distance metric we can use the Haversine formula[2], which gives physical distance in kilometers directly from latitude and longitude. We draw edge representations for proximity radii of 1, 2, 3, 4, 5, 25, 50, 100, 200, 500, 1000, 2000, 5000, and 10000 kilometers (see fig. 1).

After creating the edge indices in a $2 \times N$ tensor to adhere to PyG's edge index format, we can use statistics from each measurement site as node features. We include categorical features for the country each site is located in as well as the type of site. Our numerical feature is simply the amount of emissions generated at that site (for each polluting gas, if there are multiple).

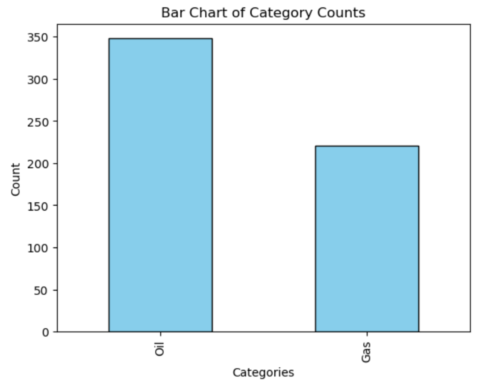
Once the data is processed into a graph format, we will use a relatively standard GCN - PyG's builtin 'GCN' class adapts the graph convolutional operator from Semi-Supervised Classification with Graph Convolutional Networks (Kipf & Welling 2017). Our GCN for classification uses two convolutional layers, equipped with the ReLU activation function as a nonlinearity. For regression we use the same model, with only slight alterations to account for shifts in output dimensions. We will compare the testing results from this GCN to a baseline by passing the original, unstructured data through a standard multi-layer perceptron (MLP) with two linear layers to mimic the architecture of the GCN. For classification, we will aim to predict the subclass of an emission site based on its position and amount of emissions. For regression we invert this task - given the type of an emission site, we estimate the emissions generated in a year.

This projects attempts to combine some traits of the previous works mentioned and expand upon them - namely, we extrapolate the usefulness of GNNs in the field of environmental science to data that is not inherently structural and generate our own artificial structure. Due to the limitation of non-structural data, we cannot give each edge a learnable parameter as in the implementation of EdgeNet -

[2]https://en.wikipedia.org/wiki/Haversine_formula



((a)) α = 100



((b)) Caption for image 4

Figure 2: Distribution of classes in coal and oil/gas datasets

the edges do not exist! Rather, these experiments treat edge connectivity as a hyperparameter that can be manually adjusted dependent on the task to create edges.

4 Experiments

4.1 Datasets

We prepare two main subsets of the Climate TRACE 2021 dataset for experimentation: emissions records for the coal mining industry and for oil/gas fields. Finding practically usable data in this dataset was a challenge - in many sectors there are well-established majorities, making class imbalance a major hurdle particularly in the coal dataset. In an effort to resolve the coal dataset's class imbalance we attempt to use synthetic data upsampling (SMOTE-ENN) with limited success, as discussed in the results section.

We report the number of edges at each chosen connectivity increment below:

α (km)	Coal edges	Oil/Gas edges
1	452	10
2	968	12
3	1,788	12
4	2,772	18
5	3,930	20
25	33,032	228
50	74,210	596
100	160,670	1,446
200	348,012	3,028
500	942,714	7,192
1000	1,348,632	12,312
2000	1,796,560	26,152
5000	3,927,088	90,260
10000	5,349,564	209,010

Table 1: Number of edges per α on each dataset

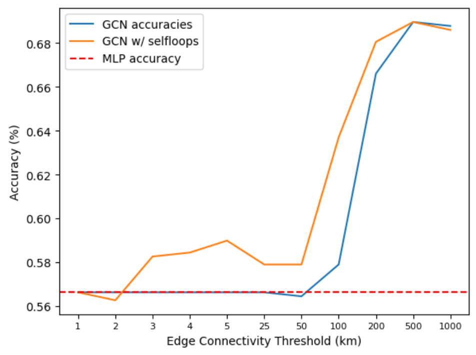
We can observe that the coal dataset is much more densely connected than the oil/gas dataset. For computational reasons, the highest connectivity thresholds ($\alpha = 5000, 10000$) will be omitted from some tests.

4.2 Training Details

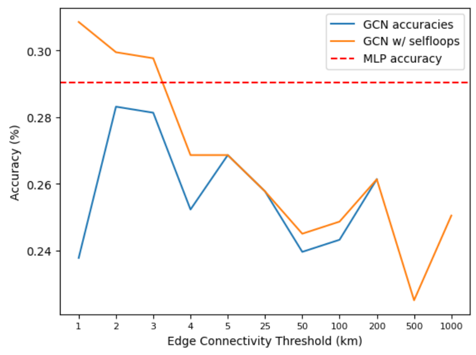
To preprocess the data we followed the protocol from section 3, converting regular tabular data into a graph-structured dataset. We also normalized the features using PyG's builtin 'transforms' library. Additionally, we prepare secondary versions of each dataset containing self loops in efforts to improve convergence - we will compare these results with the performance on unaltered graphs. We run the model five times and average the results - the plots below reflect mean errors. We choose the Nesterov-accelerated Adaptive Moment Estimation (NAdam) optimizer with a learning rate of 0.001 and weight decay 0.0005. All models were trained for 500 epochs with the exception of the MLP used for regression tasks, which was trained for 2000 epochs.

4.3 Results: Classification

With extremely low edge connectivity thresholds the model seems to function more or less identically to a standard MLP.



((a)) Accuracy on unaltered coal dataset



((b)) Accuracy on SMOTE-ENN enhanced data

Figure 3: Classification accuracy on coal datasets with varied α

When testing on the coal dataset models with low edge connectivity predicted the majority class almost every time - at higher connectivity levels the model learned to be more discerning, improving to maximum accuracies of 70 percent. The addition of self loops seemed to improve accuracy at middling thresholds with limited effectiveness at extremely high or low values.

As previously mentioned, the coal dataset has a relatively major class imbalance, with the majority class comprising 1625/2756 total nodes. To address this we attempted to use Synthetic Minority Over-sampling Technique with Edited Nearest Neighbors (SMOTE-ENN), an upsampling technique that generates synthetic examples of the minority class(es). In our case this seemed to hamper accuracy significantly (Fig. 3(b)). The dataset containing self loops had the best accuracy at low edge connectivities, but all models including the MLP performed significantly worse than on the non-upsampled dataset. This could be due to the already irregular nature of real world data being particularly difficult to synthesize reliably with upsampling techniques.

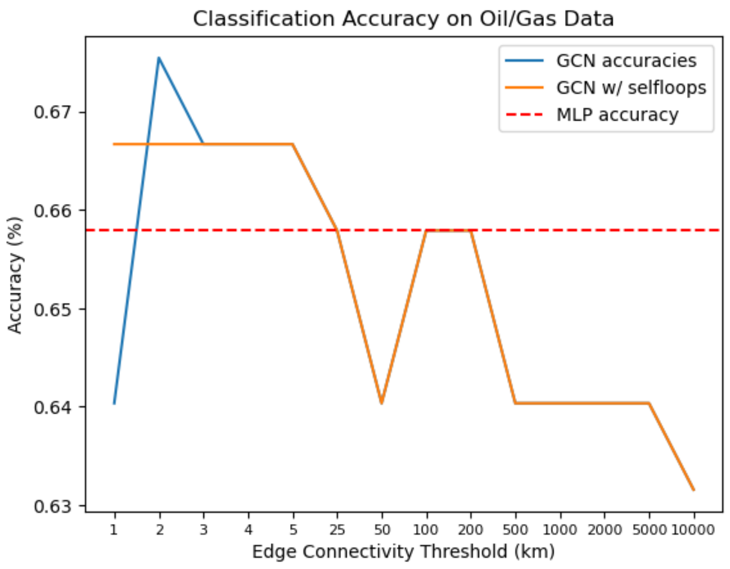


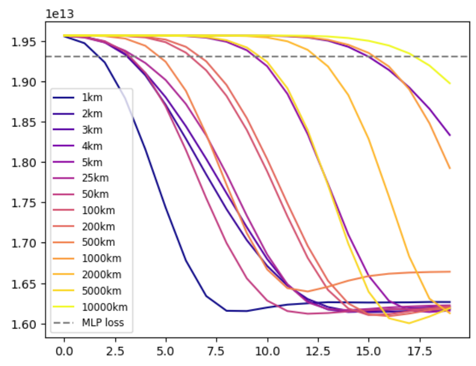
Figure 4: Various connectivities vs. MLP for classification on oil/gas data

Next we analyze the performance of the GCN on the less densely connected oil and gas dataset. Self-loops were markedly less effective on this dataset - our GCN produced identical accuracy with or without self-loops, with the exception of very small values of α. Smaller connectivity thresholds gave the best results on this dataset, implying that local information is more important than global information in this particular prediction task.

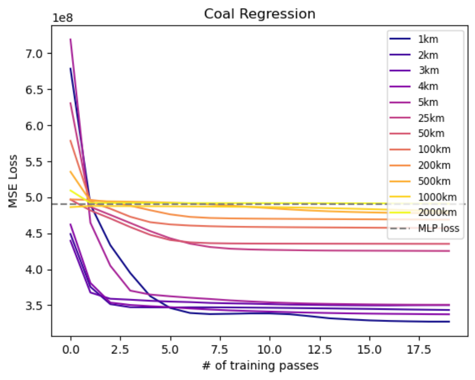
4.4 Results: Regression

On regression tasks, the GCN model once again yielded superior performance to the MLP - in runs on the oil dataset tuning the connectivity hyperparameter seemed to have limited effect on final loss, but clearly affected rate of convergence. The much more densely connected coal dataset showed much more noticeable variance in final loss with adjustment of the connectivity parameter. Additionally, the presence of self-loops in either dataset did not have a significant affect on performance.

On the coal dataset there were two noticeable clusters in the results: sparsely connected configurations and densely connected configurations. The most densely connected coal dataset tested $(\alpha = 2000)$ performed very similarly to the MLP, but all others ultimately exceeded its performance. In general, it appears that sparsely connected configurations are better suited to the tasks laid out in this project - the losses exhibited by the least connected configurations are the lowest of those tested.



((a)) Oil/Gas regression results



((b)) Coal regression results

Figure 5: Regression losses across both datasets

5 Conclusion

The results of these experiments demonstrate clearly that there is observable computational value in artifically integrating structural components to standard datasets. In this project, the protocol of where to draw edges in the graph's structure was determined by the connectivity hyperparameter α. Like most hyperparameters in the field of deep learning, varying α affected the results in non-uniform and often unexpected ways.

There are ample opportunities for future work building upon the concepts touched on in this project. Graph structure can be determined by other factors than physical proximity in other fields: time-based edges for temporal data, for instance. Any heuristic/rule relevant to a specific area of study could be used as an edge-drawing criterion. Additionally, a much more granular fine-tuning/optimization of α on specific tasks could yield even better results.

6 Reproducibility

The code used to generate the results found in this paper, as well as information about dependencies and usage, can be found at this github repository!