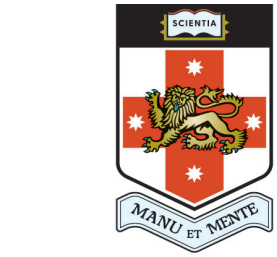

UNSW
SYDNEY

TITLE?

Justin Clarke

Supervisor: Associate Professor Yanan Fan

School of Mathematics and Statistics
UNSW Sydney

August 2023

Submitted in partial fulfillment of the requirements of the degree of
Bachelor of Science with Honours

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: Justin Clarke Date: 21/07/2023

Acknowledgements

Abstract

Graph sparification techniques for graph neural networks have traditionally been used to accelerate training and inference on real-world graphs which have billions of paramaters. There are also many different climate models which use complex mathematical models to model the interactions between energy and matter over the world. Many of these models share components and the structure of these relationships is not easily found due to the complexity of these climate models. The space of all possible graphs grows super-exponentially with the number of nodes and as such any correlation or causality is difficult to find. In this paper, I attempt to quantify these relationships with graph sparsification techniques. (Talk more about climate?)

Contents

Chapter 1

Introduction

1.1 Motivation

The simplest climate modeles have existed since the 1950's with the very first computers modelling small two-dimensional climates. Modern models have become increasingly more complex in part due to the increasing computational power available today and the large amount of data available worldwide to train these models on. Many of these models have become unexplainable due to the sheer complexity and number of their parts yet many share components and frameworks. One of the most important questions that climate science is attempting to answer today is what impact have humans had on the future of the climate. The prediction of climate change is important as it can guide us on the potential harms we may be causing to environment and life around us. As such, many models and 'scenario runs' have been developed which predict various outcomes in temperature, precipitation, air pressure and solar radiation given a certain level of societal development. On the lower end, SSP126 assumes an increasingly sustainable world where consumption is oriented towards minimising material resource and energy usage while SSP585 assumes a worst case scenario where fossil fuel usage and an energy-intensive lifestyle intensifies. (Talk more about the math behind these models? Stochastic Differential models or talk about a few of the main models in use today?)

In recent years, Graph neural networks have become the premier method of processing data with non-cartesian structure. Much of this data exists in the world in applications such as chemical analysis, social networks and link prediction (Insert references for each from reading). The main feature of GNNs is the message passing framework, where information from features on each node is passed to neighbouring nodes then aggregated and embedded. This is then propagated through a neural network structure to perform a range of tasks on the entire graph, individual nodes and edges.

Chapter 2

Background and Related Techniques

In this chapter we will give a brief overview on how the climate works and review current standards in climate modelling along with the basics behind neural networks and the extensions towards graph neural networks.

2.1 Climate Models

2.2 Neural Networks

Traditional machine learning techiques generally require meaningful data cleaning and feature creation which was costly to develop and often had many errors. The advent of deep learning allowed algorithms to progressively extract higher-level features from raw data. [1] The MLP

2.3 Graphs

2.4 Graph Neural Networks

2.5 Variational Autoencoder

Chapter 3

Framework

3.1 Dataset

The datasets used are from the CIMP6 scenario runs made available on the KNMI Climate Explorer website. The KNMI is part of the World Meteoriological Organization (WMO) []. The scenario runs include monthly predictions for temperature, min temperature, max temperature precipitation, radiation and pressure all over the globe in a 192x144 grid between 1850-2100 for 40 different models. For simplicity and brievity, this was filtered to just temperature during the 1960-1970 period in just Australia.

3.2 Problem Formulation

We define a graph as $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ represents a set of verticies which contains a list of nodes $\{v_1, \ldots, v_n\}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ the adjacentcy matrix which contains information on the graph topology. If an edge exists between two node v_i and v_j, then $\mathbf{A}_{ij} = 1$ else, $\mathbf{A}_{ij} = 0$. We also define the degree matrix as $\mathbf{D} = \sum_j A_{ij}$ where each entry on the diagonal is equal to the row sum of the adjacency matrix $\mathbf{A}$. Each node has a p-dimensional feature vector $x_i \in \mathbb{R}^p$ which describes some information about the node in the graph. By combining all n feature vectors from all nodes, we have a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. The graph also has a regression target $Y \in \mathbb{R}$ which refers to the historical temperature that each model from the graph is attempting to predict. As mentioned earlier, other variables such as precipitation, pressure and radiation are available but for simplicity, just temperature is currently being used. The two-layer GNN from [2] can be expressed as

$$f(\mathbf{A}, \mathbf{X}) = \sigma_2(\hat{\mathbf{A}}_2 \sigma_1(\hat{\mathbf{A}}_1 \mathbf{X} W^{(0)}) W^{(1)}) \tag{3.2.1}$$

where $\sigma_1(.)$ and $\sigma_2(.)$ are an activation function such as ReLU, and $\hat{A} = \tilde{D}^{-1/2}(A + I)\tilde{D}^{-1/2}$ is the symmetrically normalised adjacency matrix. The final regression problem can be formulated as

$$f : L \times X \to Y \tag{3.2.2}$$

where f denotes the learning function, L the graph, X denotes the time series input and Y the regression target.

Chapter 4

Conclusion

References

[1] Li Deng and Dong Yu. 2014.
[2] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.