Estimating epidemic infection spread using Graph Neural Networks

CS 768 End-Term Project

Arif Ahmad Eeshaan Jain Tushar Nandy

Nov 28, 2021

Indian Institute of Technology Bombay

Outline

Project Description

Project Description

Given the spread of an epidemic in a population, modelled as nodes in a network, we try to predict each person's health status by monitoring a smaller subset of nodes over a fixed period. Our model estimates the states of all neighbours of the known node via message-passing.

Challenges Faced

List of challenges faced till now

1. Epidemic state prediction has been modeled using classical approaches such as node centrality measures, and deep learning approaches using MLP and CNNs, but there has been little work using GNNs.
2. Finding the viable dataset for the task took a lot of time, with no result in the end. Hence, synthetic data was generated.

Work Done

Summary of what is done till now

1. Literature review of epidemic models, random graphs, common graph neural networks, and previous work in this domain
2. Learning to use PyTorch Geometric
3. Generation of synthetic dataset
4. Wrapper code for running models
5. Experimentation on smaller dataset
6. Report has been started

Dataset

Synthetic Data

As mentioned earlier, finding datasets had been an issue we faced from the start. To tackle the problem, we generated synthetic data as follows:

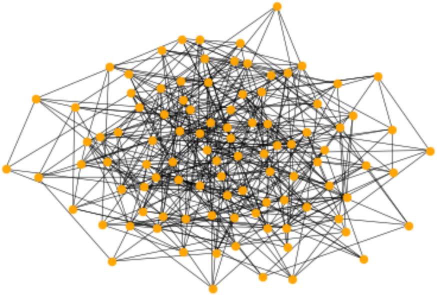
1. Random graphs were generated using the Erdős–Rényi model, Watts–Strogatz model and the Barabási–Albert model. In each graph, we kept the number of nodes as 100, and considering realistic situations, the average degree of each graph was kept near 10.
2. On each graph, we started with some infected nodes (between 1 to 5), and the progress of the epidemic was done in correspondence to the Kermack–McKendrick theory (i.e the Susceptible-Infected-Recovered model).
3. For each model, 80 random graphs were generated, and 90 time-steps were considered to bring out the temporal information using the sum aggregator.

Synthetic Data

4 In a larger experiment, 500 nodes were taken, with average degree kept near 30, which represents a realistic amount of people one can interact with.

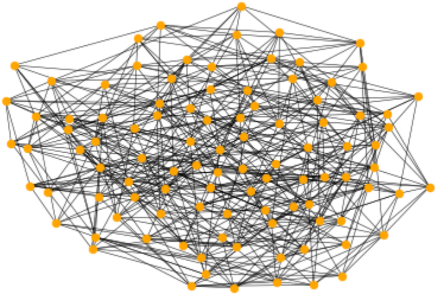
5 In the above case, 256 graphs were generated for 90 time-steps each, and as before, the sum aggregator was used.

Sample ER graph



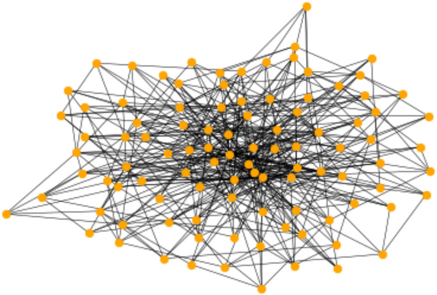
ER graph with $p = \frac{10}{99}$

Sample WS graph



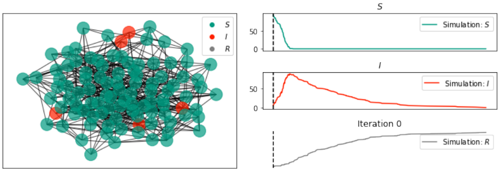
WS graph with $k = 10$ and $p = 0.6$

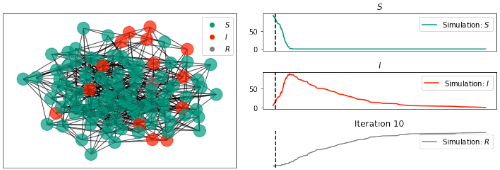
Sample BA graph



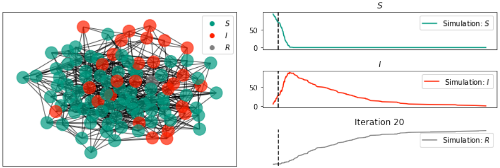
BA graph with $m = 5$

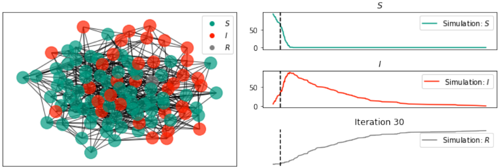
SIR Simulation



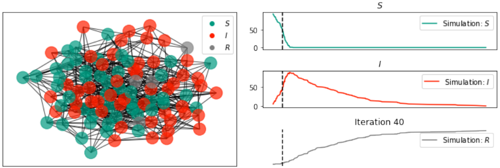


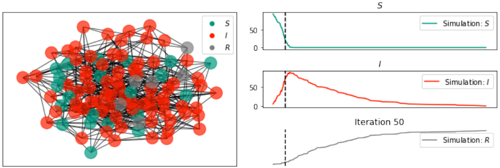
SIR Simulation



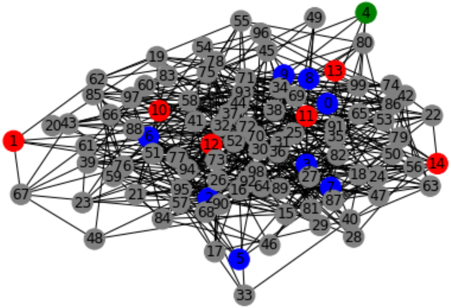


SIR Simulation





Fraction of dataset



Visualizing the amount of data we are seeing (15% here).
(*green: susceptible, red: infected, blue: recovered*))

Model

Description of our model

The network comprises of two GCNConv layers which map the initial 3-dimension feature space to a 64-dimension embedding space ($3 \rightarrow 32 \rightarrow 64$). Each of these layers is followed by ReLU activation function and 30% dropout. This is followed by a two-layer linear classifier followed by log-softmax which predicts the class of each node from the embeddings. The loss criterion is negative log-likelihood and we train the model using an Adam optimizer with learning rate 0.0002.

Initial results

100 node networks

For 80 graphs generated using the ER model, with $\beta = 0.01$ and $\gamma = 0.005$, with 100 nodes,

1. We achieved a test accuracy of 79.7% knowing just 20% of the graph.
2. We achieved a test accuracy of 81.5% knowing just 30% of the graph.

Future Plan

Plan in the coming 2 days

1. Finish all the experimenting, consolidate all results and finish the report work by evening.
2. The experimentation will have the following components:
   - The training would occur over all three random graph models
   - Comparison will be made between the amount of graph known and accuracy
   - Comparison would be made between different layers such as GCNConv etc.
3. If time permits, add results of a GVAE to benchmarking.

Thank You

- Arif Ahmad
- Eeshaan Jain
- Tushar Nandy