Epidemics Graph Neural Network Node Classification and Link Prediction

Jaykumar Patel
patel.jay4802@utexas.edu

Afnan Mir
afnanmir@utexas.edu

Abstract

The COVID-19 pandemic showed that contact tracing helped mitigate the spread of the virus. However, manual contact tracing is slow and can be inaccurate. Thus, this project aims to automate contact tracing by utilizing Graph Neural Networks (GNNs). Our preliminary work on network analysis showed that the contact network is a mix of an exponential and scale-free network. Also, our simulation showed that during the first 12 hours, the infection does not spread much, but then it starts spreading steadily.

1. Introduction and Motivation

When COVID-19 first appeared, manual tracing was deployed to mitigate the initial outbreak. Contact tracing is the process of tracking how the virus spreads by identifying people who may have come in contact with an infected person, and then asking them to isolate and get tested.

However, the pandemic revealed that the COVID-19 virus spread faster than manual contact tracing [2]. Thus, this project's objective is to automate contact tracing by incorporating machine learning using GNNs to hopefully increase the mitigation of the spread of COVID-19 when compared to manual contact tracing.

2. Previous Work

Mathematical models, classical ML models, and graph-based ML models have been used to predict virus spread.

The SEIRD model is a mathematical model that predicts the change in Susceptible, Exposed, Infected, Recovered, and Deceased people over time by using differential equations. [3]. SIR is simpler version of the SIERD model [11].

The LSTM is an ML model that has been used to predict the number of cases over time [3]. A hybrid of SIRD and LSTM helps account for time dependent parameters of the SIRD model [1]. Also, GNNs, which are graph-based ML models, have been used on mobility data to predict virus spread and for link prediction for contact tracing [8][9].

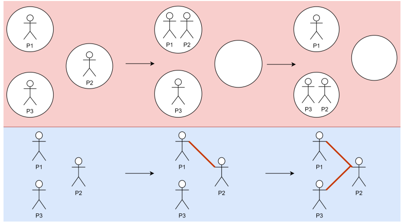

Figure 1: Example of contact network generation

3. Approach

We used the foursquare dataset to build a contact graph of Austin, TX [7]. Each entry contains a device ID, a location ID, UTC date and hour, and a dwell time, which tell us when and how long a person visited a location. Given this data, we generated a contact graph of Austin.

We used data only from July 1st, 2020 to generate a sample network. Our nodes were all the unique device IDs in the dataset, which correspond to people. For our edges, we used the following logic: we ignored entries with a dwell time less than 60 minutes, as we assumed this not enough time to make significant contact with others. Then, we used the UTC date and hour with the dwell time to determine the arrival and departure time interval for each entry. We then compared every entry with every other entry. If the entries' locations were the same and if intervals overlapped by at least 60 minutes, we considered this as a contact between the two people and added an edge between them. Figure 1 shows an example of a sample contact network generation.

Separately, we also created a sample Susceptible-Infected (SI) simulation using data from July 1st, 2020. We did not include a recovery/death or incubation period in this model mainly because we only performed the simulation over one day. For the simulation, we did the following: at hour 0, we randomly infected 20% of the nodes in the graph. Then we looked at all contacts between people for hour 1, and if an infected person came into contact with a susceptible person, we created an edge between them and infected the susceptible person with a probability of 1. We repeated the previous step for the remaining 22 hours of the day.

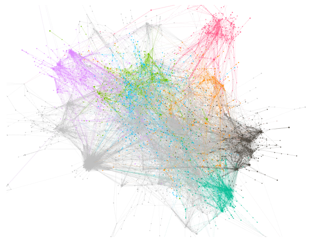

Figure 2: Contact network after one day

After generating and analyzing the contact network for one day and performing the SI simulation, we moved towards leveraging graph learning techniques to perfom link prediction, which is the fundamental task behind automating contact tracing. For this milestone, we focused on performing link prediction on a static graph. In order to have enough data to train and evaluate our model(s), we generated a contact network for the first five days of July. We used the same technique described in building our July 1st contact network to build the 5 day contact network; so the nodes are people who have a dwell time of at least 60 minutes at one location, and edges represent people who have come into contact with each other in the 5 day period.

Our first goal was to create a baseline link prediction model. We used the node2vec algorithm to generate node embeddings for each node in the graph [4]. We then had to generate our dataset of edges. In order to perform link prediction, we need the set of positive edges, which are all the edges present in the network, and we need aset of negative edges. This allows us to boil down the link prediction to a binary classification problem. Given our network, we created a set of negative edges that was equal in size to the set of positive edges to ensure balanced training. Using the node2vec embeddings and the set of positive and negative edges, we trained a GraphSAGE model to perform link prediction on the static 5 day graph [5].

After creating the baseline model, we searched for ways to improve the model's performance on the graph. This would include performing feature engineering techniques to add dimensions to our node embeddings and exploring the use of other GNN architectures such as GCN and/or GAT [6][10]. We hoped to be able to finetune the model and improve its performance to the point we could use it to perform link prediction on the dynamic contact network.

4. Experimental Setup and Results

In Figure 2, we can see the contact network after one day. Network properties for this network were also calculated. The average node degree is 10.41, the network diameter is 14, the average clustering coefficient is 0.687, and the average path length is 4.784. In addition to this, the degree distribution was mainly an exponential distribution with subtle hints at a power-law. This can also be seen from the network itself, as we can see the presence of a few hubs in the network. This makes sense, as we should expect a social network to be scale-free, but we do not have all the data points, so it is not fully scale-free on the sample network.

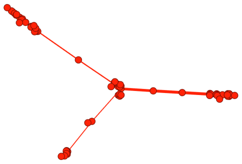

Figure 3: Infected graph after one day

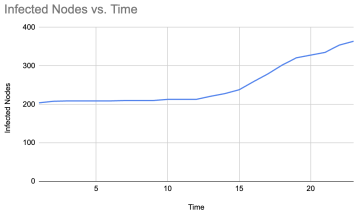

Figure 4: Number of infected nodes over 24 hours

In addition, we can see the resulting infected graph from our simulation in Figure 3. In this network, it is important to note that all of the visible nodes are infected, and the edges represent how the infection has been spread. We can see that not a lot of infected nodes have spread the virus, but the ones that have created one connected component that show the beginnings of a scale-free network.

In Figure 4, we can see the number of infected nodes over time. We can see that the infection doesn't spread much until the latter half of the day, after which it spreads steadily. We expect the rate of spread to be exponential when we simulate the spread over more days. This lines up with our intuition, as we would expect the infection to spread faster as more people get infected.

5. Conclusion and Short-Term Plans

Through the analysis of the network, we were able to determine the network of contacts is a combination of an exponential and scale-free network. Some people came into contact with many other people whereas others stayed within their cliques. The simulation showed that the number of infected people initially stayed relatively constant, but after 12 hours, infections began increasing steadily.

For M2, we plan to do a more complete analysis of the network by considering both July and August data. We also plan to run more realistic simulations by accounting for the incubation period and deceased/recovered people. We also plan to implement GNNs at a small scale to perform link prediction. Specifically, we will use GNNs to predict the people that come into contact with previously infected people, allowing us to predict the spread of the virus.

For this milestone, Afnan generated and analyzed the contact network for July 1st, 2020, and Jaykumar created and analyzed the simulation of the hourly spread of the virus.

References

[1] A. Bousquet, W. H. Conrad, S. O. Sadat, N. Vardanyan, and Y. Hong. Deep learning forecasting using time-varying parameters of the sird model for covid-19, February 22 2022.

[2] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, and et al. Estimating the number of infections and the impact of non-pharmaceutical interventions on covid-19 in european countries: technical description update, 2020.

[3] T. Geroski, A. Blagojevic, D. M. Cvetković, A. M. Cvetković, I. Lorencin, S. B. Šegota, D. Milovanovic, D. Baskic, Z. Car, and N. Filipovic. Epidemiological predictive modeling of covid-19 infection: Development, testing, and implementation on the population of the benelux union, October 28 2021.

[4] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks, 2016.

[5] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs, 2018.

[6] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2017.

[7] C. D. Lab. Foursquare Community Mobility Data with Basemap (US), 2020.

[8] K. Skianis, G. Nikolentzos, B. Gallix, R. Thiebaut, and G. Exarchakis. Predicting covid-19 positivity and hospitalization with multi-scale graph neural networks, March 31 2023.

[9] C. W. Tan, P.-D. Yu, S. Chen, and H. V. Poor. Deeptrace: Learning to optimize contact tracing in epidemic networks with graph neural networks, 2023.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks, 2018.

[11] R. S. Yadav. Mathematical modeling and simulation of sir model for covid-2019 epidemic outbreak: A case study of india, May 21 2020.