Epidemics Graph Neural Network 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 prone to inaccuracies. Thus, this project aims to automate contact tracing by utilizing Graph Neural Networks (GNNs) for link prediction. Our network analysis shows that the contact network is mostly exponential with hints of scale-free properties. Additionally, our analysis identifies a discernible pattern in mobility, characterized by an increase in travel during the weekends. Furthermore, we were able to achieve an AUC of 0.91 for static link prediction by using the GCN architecture and performing feature engineering. Future plans include performing temporal link prediction to automate contact tracing.

1. Introduction and Motivation

When COVID-19 first appeared, manual contact 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. Firstly, we will create and analyze contact networks. Then we will use GNNs to perform static link prediction. Finally, for M3, we will perform temporal link prediction on the contact networks.

2. Previous Work

Mathematical models, classical machine learning models, and graph-based machine learning 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]. The Susceptible-Infected-Recovered (SIR) model is a simpler version of the SEIRD model [12].

The Long Short Term Memory model is a machine learning 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 [9][10].

We hope to build off of these previous works by utilizing GNNs and other deep learning techniques to be able to predict contacts between individuals. This is a non-trivial and novel task, as it requires the model to potentially learn a graph's structure based on its structure at previous time steps as well as node features.

3. Approach

3.1 Network Generation and Simulation

We used the Foursquare dataset from the months of July and August 2020 to build contact networks of Austin, TX [8]. Each entry contains a device ID, a location ID, a 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.

Firstly, we used data from July 1st, 2020 to July 5th, 2020 to create a sample contact 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 of less than 60 minutes, as we assumed this was 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. We will call this the 5-Day Contact Network. This network captures the meaningful

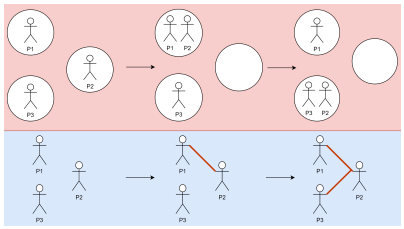

Figure 1: Example of contact network generation

contacts that occurred between people from July 1st, 2020 to July 5th, 2020. Figure 1 shows an example of a sample contact network generation.

Then we also created a set of 62 contact networks, one for each day from July 1st, 2020 to August 31st, 2020. We used the same logic as before to create these individual contact networks. If two people were at the same location at the same time for at least 60 minutes, we created an edge between them. We will call this set of networks the Temporal Contact Networks. Each day's network captures the meaningful contacts that occurred between people for that day. Furthermore, we analyzed how the clustering coefficient and average node degree changed over time by calculating these metrics for each network in the Temporal Contact Networks.

Furthermore, we created an SIR simulation using the Temporal Contact Networks, which we ran from July 1st, 2020 to August 31st, 2020. Here are the parameters and assumptions that were made for the simulation:

- Contact between people that is less than 60 minutes is not considered significant enough to spread the virus.
- If a susceptible person comes into contact with an infected person for at least one hour, then they get infected with a probability of 0.30. This is called the infection rate (IR) and it is constant.
- An infected person will recover after seven days. This is called the recovery period (RP) and it is constant.
- A person can only be infected if they were previously susceptible, and a person can only be recovered if they were previously infected.
- Initially, 20% of the people, chosen at random, are infected. The rest are susceptible.
- Only infected people can infect others.

The simulation algorithm is shown in Algorithm 1. The simulation was run on each network in the Temporal Contact Networks. The simulation results were analyzed to see how the virus spread over time. This approach is scaleable, as the simulation can be run on any number of Temporal Contact Networks. Thus, we can use this approach to simulate the spread of the virus over a longer period of time.

Algorithm 1 SIR Simulation

Input: Temporal Contact Networks (TCN: Array of Contact Networks), IR, RP
Initialize: Set 20% nodes with $state = I$, Set 80% nodes with $state = S$, Set all nodes with $time\_of\_recovery = \infty$
for i = 1 to len(TCN) **do**
  for each node n where $n.state = I$ **do**
    for each neighbor m of n in TCN[i] where $m.state = S$ **do**
      if $rand(0, 1) <=$ IR **then**
        $m.state = I$
        $m.time\_of\_recovery = i + RP$
      end if
    end for
    if $i = n.time\_of\_recovery$ **then**
      $n.state = R$
    end if
  end for
end for

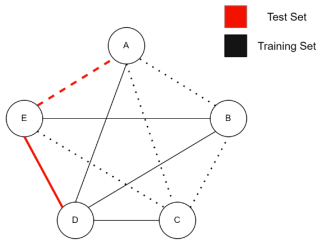

Figure 2: Example of positive and negative edges for train and test data

3.2 Machine Learning

After generating and analyzing the 5-Day Contact Network, and Temporal Contact Networks, and performing the SIR simulation, we moved towards leveraging graph learning techniques to perform 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 models, we used the 5-Day Contact Network.

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 generated the dataset of edges. To perform link prediction, we need a set of positive edges, which are the edges present in the network, and we need a set of negative edges, which are the edges not present in the network. 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 Contact Network [5].

A sample example of training and testing data is shown in Figure 2. The black lines represent training data and the red lines represent testing data. The solid lines represent positive edges and dotted lines represent negative edges.

The model is trained on the black lines, and attempts to predict the red lines. This allows us to boil down the link prediction to a binary classification problem.

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 the Graph Convolutional Network (GCN) and/or the Graph Attention Network (GAT) [7] [11]. 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 Temporal Contact Networks. This approach is scalable because generating the training and testing edge sample, node embeddings, and node features can be done on a network of any size; however, generating the node2vec embeddings for a large network can be computationally expensive.

4. Experimental Setup and Results

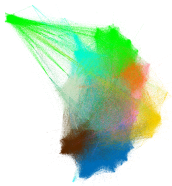

Figure 3: 5-Day Contact Network from July 1st, 2020 to July 5th, 2020

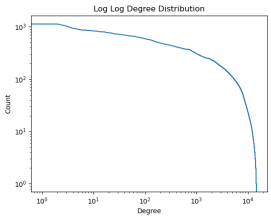

Figure 4: 5-Day Contact Network Degree Distribution

In Figure 3, we can see the 5-Day Contact Network. Network properties for this network were calculated. The average node degree is 102.599, the network diameter is 7, the average clustering coefficient is 0.627, and the average path length is 2.849. In addition to this, the degree distribution was mainly an exponential distribution with subtle hints of power-law as shown in Figure 4. This can be seen from the network itself, as we can see the presence of a few hubs in the network.

In addition, the simulation results are shown in Figure 5. The simulation results show that the number of infected people initially increases, but then decreases significantly and approaches zero. However, the number of recovered

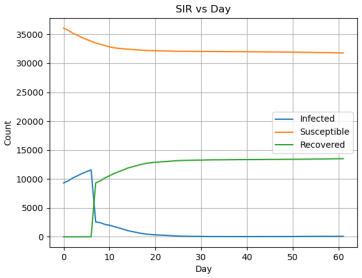

Figure 5: SIR Simulation Results

people exhibits the opposite behavior. This makes sense since the people who started with the infection at the beginning of the simulation recover after seven days; thus, they can no longer infect others nor become susceptible again. Also, the number of susceptible people decreases over time, which makes sense since they are getting infected and recovering. The maximum number of infected people on any given day is about 11,500 and the total number of people infected over the simulation is about 14,300.

It is important to note that this simulation was run on limited data. The total number of nodes in the simulation, which is the total number of people, is around 45,000, whereas the actual population in the Austin metropolitan area in 2020 was around 2 million. Furthermore, this model assumes a closed population, which is not the case in Austin, TX. People are constantly moving in and out of the city. Thus, the simulation results are not representative of the actual spread of the virus in Austin, TX. Having access to more data would allow the simulation to provide realistic results. However, this simulation does show that the virus can spread quickly, as seen in the first week of the simulation. Thus, it is important to have an efficient and accurate contact tracing system to mitigate the spread of the virus.

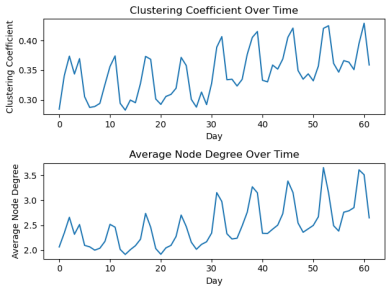

Figure 6: Clustering Coefficient and Average Node Degree for Temporal Contact Networks

We also analyzed the Temporal Contact Networks. The clustering coefficient and the average node degree for these networks are shown in Figure 6. The results reveal a seven-

day periodic pattern with peaks occurring on the weekends. These results make sense since people are more likely to go out and socialize on the weekends. This increase in socialization on the weekends leads to more contacts, which leads to a higher clustering coefficient and average node degree.

After performing the analysis on the 5-Day Contact Network and Temporal Contact Networks, we moved towards performing link prediction on the 5-Day Contact Network. For our baseline model, as stated in the approach, we used a GraphSAGE model with node2vec embeddings. We trained the model for 1000 epochs on a 90/10 train/test split with the Adam optimizer [6]. The model achieved an AUC score of 0.49 on the test data, which is worse than random guessing. Part of this could be because no internal node features were used in the node embeddings.

To improve upon this, we performed feature engineering to generate additional features for each node. The features we added included average number of locations traveled to per day, average distance traveled per day, age, gender, and State-Age-Gender (SAG) score. By adding these features to the node2vec embeddings, we were able to improve the AUC score to 0.63, which is a significant improvement over the baseline model. We can see that the additional features helped the model learn the graph structure better.

We noticed that the features were of different scales, so we hypothesized that normalizing the features would improve the model's accuracy. This adjustment improved the performance of the model, leading to an AUC of 0.75.

Seeking further enhancement, we explored a different model architecture, specifically the GCN model. Utilizing the same features and data, the GCN model achieved an AUC score of 0.91, demonstrating a significant improvement over the initial model.

While these performance metrics establish the feasibility of link prediction on contact networks, it's crucial to note a limitation in the current approach—namely, its inability to capture temporal data. Recognizing this limitation, our next objective for M3 is to implement temporal link prediction using the GCN model. This aligns with our goal of automating contact tracing.

5. Conclusion and Short-Term Plans

Through the analysis of the 5-Day Contact Network, we were able to determine that the network is mostly exponential in degree distribution, with hints of scale-free properties. 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 increases, but after a few days, decreases significantly.

The analysis of the Temporal Contact Networks highlights a pattern in mobility. Particularly, people tend to socialize more on the weekends, which leads to a higher clustering coefficient and average node degree.

Finally, we were able to perform static link prediction on the 5-Day Contact Network. We were able to improve the model's performance by performing feature engineering and using the GCN model.

For M3, we plan to use the Temporal Contact Networks for temporal link prediction. Essentially, given contact networks for July 1st, 2020 to August 15th, 2020, the model will attempt to predict the contacts that will occur from August 16th, 2020 to August 31st, 2020. We will also attempt to improve the model's performance by performing feature engineering and using other GNN architectures.

For this milestone, Afnan generated and analyzed the 5-Day Contact Network and performed static link prediction on it. Jaykumar created and analyzed the Temporal Contact Networks and ran the SIR simulation on them.

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] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

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

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

[9] 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.

[10] 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.

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

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