Updated Graph Neural Network to Predict Patient Zero(s)

Aditya Sasanur
asasanur@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Ninaad Lakshman
ninaadlakshman@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

ACM Reference Format:
Aditya Sasanur and Ninaad Lakshman. 2023. Updated Graph Neural Network to Predict Patient Zero(s). In *Proceedings of CSE 8803 (CSE 8803 EPI).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM

In the midst of an outbreak, the imperative task of identifying the outbreak's root cause becomes increasingly pronounced. This endeavor carries significant implications, as pinpointing the origin of the outbreak empowers researchers and scientists to delve into the underlying factors responsible for the initial emergence of the disease. Such insights are invaluable for not only mitigating the ongoing outbreak but also for implementing strategies to contain and prevent future outbreaks from occurring. However, the quest to locate a patient zero, or potentially multiple patient zeros, becomes considerably more complex and intricate as the outbreak unfolds. The complexity arises from the diverse and intricate network of connections through which the disease can propagate.

From a computational standpoint, this challenge poses considerable difficulties. As the outbreak progresses, tracing the precise source or sources of the infection becomes computationally expensive and often impractical. This is primarily due to the stochastic and unpredictable nature of disease spread within a dynamic population and network. Consequently, the formidable challenge at hand revolves around not only determining the number of initial patient zeros but also accurately identifying their identities amidst the intricate web of transmission pathways. This multifaceted challenge underscores the need for innovative approaches, such as the one we are pursuing, which harness advanced technologies like graph neural networks to navigate the complexity of disease dynamics and patient zero identification.

2 LITERATURE REVIEW

2.1 Patient 0 localization and foundations

Shelke and Attar's [5] paper on source detection in social networks was a foundational way for our team to understand what factors should be considered when finding a patient zero. They mention several factors from network structure such as its topology to diffusion models used (such as epidemiological models) to centrality measures. For locating patient zeros, these factors are the foundation for future source localization methods. This paper is helpful in listing out what avenues one could pursue when generating a source localization estimator (such as whether to identify a single source or multiple sources). Shelke and Attar do not propose a novel method but rather test out different approaches.

There have been multiple novel methods used for identifying and predicting a patient zero. One of the foundational papers by Shah and Zaman [4] looks into finding the source of a "rumor" in a network (analogous to the source of a disease). To minimize estimation error in their rumor spreading model, they leverage the maximum likelihood estimator. In this case, they use the underlying assumption that the best source is one that maximizes data likelihood in an SI model. This is a strength of the paper, as the algorithm itself is quite explainable in comparison to other neural network counterparts. This method works efficiently in k-regular trees, but one of the underlying weaknesses is that it is computationally inefficient in general graphs (without converting the graph to a tree). This is because computing the summation of the likelihoods of all possible permutations is expensive (which for a regular tree is not the case). However, in realistic scenarios, it is often the case where we are dealing with networks that follow a general graph pattern. Approaches to change general graphs into trees (minimum spanning tree and BFS) would work in formatting this network in a tree, making the task more efficient.

2.2 Dynamic message-passing algorithm

Another approach involves a modified dynamic message-passing algorithm by Lokhov [2] and other researchers. They derive this dynamic message-passing (DMP) algorithm from a pre-existing belief propagation (BP) method, which estimates marginal probability distribution in a variety of problems. Using this DMP method, they were able to compute a patient zero with high probability (in comparison to other methods such as distance and Jordan centralities). There are some noted weaknesses such as if the graph has self-loops and that there could be better ways to approximate their likelihood outside of mean-field-like approximation. Outside of these minor weaknesses, this is a valid and accurate way to compute patient zeros outside of neural networks as well. We would ideally like to find an even more accurate way to compute patient zero in our graph neural network approach.

2.3 Graph neural networks

Graph neural networks (GNNs) do have broad reaching applications within epidemiology. Papers like Song and others' [6] publication on COVID-19 infections with graph neural networks to predict influence of current infected patients on future infections. This helped establish the general structure of graph neural networks and step-by-step implementation to handle graph-structure data

CSE 8803 EPI, September 2023, Atlanta, GA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

from a COVID-19 epidemiological standpoint. Their GNN-specific approach, including both graph convolutional networks and graph attention networks, did have success with capturing relational information between nodes which helped its accuracy. Their research provides a clear baseline and functionality of epidemiological graph neural networks in the case of forecasting. Their paper focused on whether each infected node had an effect on each subsequent round of infection. However, it did not delve into the opposite case: localizing the infection source by finding a patient zero. Shah and other researchers [3] have a paper that learn a single patient zero through a graph neural network. They seek to model relationships within the graph training on labeled data which normally use a graph as input. In this case, the paper uses a graph neural network to identify the source of the contagion dynamics of this graph, modeling a contagion using Susceptible-Infected-Recovered (SIR) and Susceptible-Exposed-Infected-Recovered (SEIR). Compared to dynamic message-passing, they state the algorithmic time complexity of a trained GNN in the inference stage has complexity $O(N^2L)$ where N is the number of nodes (with max complexity of N^2 per layer) and L layers. This is polynomial in time, and they state that this time complexity results in a computational speed that's 50-100 times faster than dynamic message-passing. Utilizing the past GNN literature and the works of [8], we plan on improving upon the GNN architecture to find patient 0s.

Their GNN architecture follows a modern GNN design with linear, convolutional, batch normalization, and leaky-ReLu activation layers. They found a high accuracy in the first two weeks of spread with accuracy degrading by 50% after the two week period, and their efficiency exceeded classical methods by over 100 times. Additionally, they specify that they approach the theoretical upper bound in patient zero detection accuracy, but this accuracy only applies to identifying single patient zeros. However, due to the use of a graph neural network, the patterns and computations the network generates are not very explainable compared to dynamic message-passing and other algorithm-based approaches. Furthermore, this paper assumes a single source, whereas additional changes could be made to the neural network to identify multiple patient zeros. In fact, this was an extension they mention in their conclusion.

2.4 Multiple source detection model

Wang and other researchers [7] have created a propagation model, named label propagation-based source identification (LPSI), for the specific case of identifying multiple sources without knowing the underlying propagation model. This was done algorithmically in both an iterative and convergence fashion, which both yielded high accuracy. This paper explained that the convergence method for LPSI did suffer from a large time complexity but the iterative method was faster than other counterparts.

2.5 Variational autoencoder

Ling and others found variational autoencoders [1] to have a great use in graph diffusion problems such as finding the source. After training their source localization variational autoencoder (SL-VAE), they plan on predicting an optimal diffusion source given the observation. This SL-VAE is an ensemble of multiple information propagation models, including a graph neural network. Essentially, their SL-VAE leverages a GNN among others coupled with their decoder and encoder, which are hallmarks of a variational autoencoder. They tested their SL-VAE performance using Susceptible-Infected (SI) and SIR models. For SI and SIR, even though the immunity of individual nodes brought more randomness to the localization task, they noticed that SL-VAE has the best performance in comparison to other models. This SL-VAE also has the ability to predict multiple sources as their predicted optimal diffusion source variable can hold more than one source. This SL-VAE approach has a clear advantage of accuracy but does suffer from an additional layer of complexity on top of graph neural networks and therefore less explainability.

3 ALGORITHMS

To predict the identities of patient zero or zeros we plan on utilizing a Graph Neural Network to make node level predictions. Multiple avenues are currently under consideration and the final algorithm that we will implement will be based on the experiments we run and which one produces the highest accuracy.

The first experiment will involve training our GNN on tree-like networks with a passed in time-step T and each network will have one labeled patient zero. This should provide the most accurate results since outputting the identity of exactly one patient 0 at a specific timestep leaves less variability for the GNN to learn.

To build on this, we will then aim to train the GNN to not predict the identity of a patient 0 but instead the quantity of patient 0s in a tree-like network at time-step T. This will be done by training the GNN with passed in time-step and networks labeled with the number of patient 0s.

Subsequently, if the GNN performance is not up to par, we aim to supplement the GNN with multiple snapshots of the network outbreak at different timesteps. This would provide the GNN with more data to draw patterns from and perhaps produce better results.

Lastly, though variational autoencoders do add a fair bit of complexity to the model, we plan on experimenting with adding them to our architecture as Ling and other researchers found their VAE to outperform existing methods. Our VAE will be different as we plan on providing additional features such as snapshots and time-step as an attempt to increase accuracy and add realistic parameters to our model.

After performing all the aforementioned experiments, we intend to compare and contrast the different models and extract the strengths of each one in order to create a final model capable of pinpointing patient 0s with very high accuracy. We will furthermore extend our work to general graphs and evaluate the performance.

4 DATA

To acquire the data required for both training and testing, we will adopt a methodology akin to the data collection process described in Shah's 2020 study [3]. Our approach involves a series of steps aimed at generating a diverse and comprehensive dataset.

Firstly, we will define a range of node quantities, including 100, 500, 1,000, 2,000, 5,000, and 10,000 nodes, and specify various edge structures, such as dense, sparse, and tree-like configurations. This combination of node counts and edge structures will form the basis for our experimentation.

To create the actual network graphs, we will employ the Python package called NetworkX. This versatile library enables us to generate random networks, compute various network properties, and visualize the resulting networks. Leveraging NetworkX will allow us to ensure the diversity and complexity of the generated graphs.

Once the networks are constructed, our next step is to conduct multiple simulations on each of them. During these simulations, we will systematically vary the values of key infection parameters, including β, γ, λ, and κ. This variation will provide us with a comprehensive set of infection dynamics, covering a wide range of scenarios and scenarios.

After configuring the graph structures and setting the parameters, we will designate K nodes as patient zero(s) to initiate the disease outbreak. The propagation of the outbreak will be tracked over a variable time span, T, ranging from 1 to 100 time steps. This flexibility in time step selection ensures that we capture a broad spectrum of outbreak durations and behaviors.

Once the outbreak propagation simulations are completed, we will have generated all the requisite data for training and testing our model. The input data for the model will include the network structure represented as an adjacency matrix, the infection parameters (β, γ, λ, and κ), the chosen time step (T), and the number of initial infected nodes (K). This comprehensive dataset of about 20,000 graphs (based on estimates for necessary datapoints needed for training but subject to change) will empower our model to learn and generalize from a wide array of infection scenarios, making it robust and effective in predicting disease dynamics.

5 EVALUATION

To assess the effectiveness of our approach, we will subject our novel model to rigorous testing to evaluate its ability to accurately identify the initial infection sources (patient zeros). This evaluation will be carried out using a dedicated testing dataset comprising a substantial 10,000 generated graphs, all constructed using the versatile NetworkX Python library. These graphs will exhibit diverse characteristics, encompassing a wide spectrum of structural complexity, density levels, and node/edge quantities. The deliberate diversity within our testing dataset ensures a comprehensive assessment of our model's performance under varying scenarios.

To quantify the model's performance, we will employ the Kullback-Leibler Divergence (KL-Divergence). KL-Divergence is a valuable metric that measures the dissimilarity between probability distributions, in this case, the distributions of predicted patient zeros. By utilizing KL-Divergence, we can precisely gauge how closely the model's predictions align with the actual patient 0s across the diverse set of test graphs.

Additionally, to benchmark our model against existing methodologies and to assess its competitive edge, we will employ two more evaluation metrics: top-k accuracy and normalized rank. Top-k accuracy measures the model's ability to correctly rank the patient zeros among a list of candidates, providing insights into its predictive accuracy. Normalized rank, on the other hand, offers a comprehensive assessment of the model's performance by considering the relative ranking of the true patient zeros within the model's predictions.

By employing these rigorous evaluation techniques, we aim to not only validate the efficacy of our novel model but also demonstrate its potential for significant improvement over current state-of-the-art techniques in the domain of Graph Neural Networks (GNNs) and predicting patient zeros in disease outbreaks. This multifaceted evaluation approach will provide a holistic view of our model's capabilities and its contributions to advancing the field of disease dynamics prediction.

6 EXPECTED OUTCOME

Our overarching goal for this semester is to develop a pioneering graph neural network architecture that possesses the unique capability to take into account two critical factors: the specific time step at which the disease outbreak has progressed and the number of patient zeros involved. This innovative approach represents a significant advancement over previous research efforts, which predominantly focused on predicting a single patient zero while neglecting temporal information. By considering both time and the number of initial infections, our model aims to revolutionize our ability to pinpoint the identities of the patient zeros accurately.

The incorporation of time-step information is particularly groundbreaking as it allows us to track the evolution of outbreaks over time. This temporal awareness is crucial for modeling and understanding the dynamics of multi-source pandemics comprehensively. By capturing the temporal dimension, our model can provide insights into the spread of diseases, enabling us to intervene more effectively and proactively in the early stages of an outbreak, thereby mitigating their potential to escalate into large-scale crises.

Moreover, our novel GNN architecture holds great promise in shedding light on network cascades and the prevention of future multi-source outbreaks. By accurately identifying the sources of infections at different time steps, we gain a deeper understanding of how diseases propagate through networks and how to disrupt these pathways. This knowledge can be harnessed for devising more effective prevention and containment strategies, ultimately contributing to enhanced public health preparedness and response.

In summary, our semester-long endeavor aims to advance the field of disease outbreak modeling and prediction by developing a cutting-edge GNN architecture that considers both time and the number of initial infections. This innovative approach has the potential to transform our ability to tackle multi-source pandemics, understand network cascades, and bolster our capacity to prevent future outbreaks.

7 SCHEDULE

As we only have two team members, we want to evenly divide the work as follows. We plan that this research will involve the following steps: further literature review, graph neural network architecture planning, simulation creation, graph neural network creation, data visualization, research paper writing, research paper revision. We want this literature review to be conducted together, as we both want to understand the current methodologies and problem space. The graph neural network architecture will also be created together, mainly based on the literature we read and best practices we gathered together. We plan on the simulation creation being a joint task (although can be done asynchronously).

We would create different subtasks for the simulation and assign them to each other. As for data visualization and research paper writing, we will assign a lead for each of these tasks who would be in charge of either the plots/visuals and the main writing pieces. However, we will both be a part of these tasks and ensure each other's work is quality. Lastly, the revision will be a joint task, and we will even reach out to others in the class or researchers to vet our work. As for timeline, we have agreed upon a week-by-week action plan as follows:

Week 1 (9/25-10/2): We will read a couple more papers centering around source localization to frame our understanding.

Week 2 (10/2-10/9): We will fill out some portions of our literature review and read more papers on implementations of graph neural networks in epidemiological contexts.

Week 3 (10/9-10/16): We will begin the simulation creation process using Python notebooks mimicking some of the simulations used in existing papers.

Week 4 (10/16-10/23): We will get an initial version of the simulation up-and-running and create thousands of different networks for our model to train on.

Week 5 (10/23-10/30): We will complete the architecture of our graph neural network based on existing papers and amend the architecture using the changes we feel will improve the capability/accuracy.

Week 6 (10/30-11/6): We will code up the graph neural network and train the model using the data we created in our simulation.

Week 7 (11/6-11/14): We will visualize all the data and create a variety of different visuals for use in the paper. We will select the most relevant ones and concurrently begin drafting the core of the paper (approach, methods, etc.).

Week 8 (11/14-11/21): We will continue drafting the paper together and include the data, analysis, and visuals.

Week 9 (11/21-11/28): We will complete the paper and begin the revision process. We will get additional reviews of the paper and integrate the feedback accordingly. We will also draft the presentation to the class based on our research.

Finalization of Project: We will submit the paper and present to the class.

ACKNOWLEDGMENTS

To Dr. Aditya Prakash for his guidance in the initial research.

REFERENCES

[1] Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. 2022. Source Localization of Graph Diffusion via Variational Autoencoders for Graph Inverse Problems. (Aug. 2022). https://doi.org/10.1145/3534678.3539288

[2] Andrey Y. Lokhov, Marc Mezard, Hiroki Ohta, and Lenka Zdeborova. 2014. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E* 90, 1 (July 2014). https://doi.org/10.48550

[3] Chintan Shah, Nima Dehmamy, Nicola Perra, Matteo Chinazzi, Albert-Laszlo Barabasi, Alessandro Vespignani, and Rose Yu. 2020. Finding Patient Zero: Learning Contagion Source with Graph Neural Networks. (June 2020). https://doi.org/10.48550

[4] Devavrat Shah and Tauhid Zaman. 2009. Rumors in a Network: Who's the Culprit? (Sept. 2009). https://doi.org/10.48550

[5] Sushila Shelke and Vahida Attar. 2019. Source detection of rumor in social network – A review. 9 (Jan. 2019). https://doi.org/10.1016

[6] Kyungwoo Song, Hojun Park, Junggu Lee, Arim Kim, and Jaehun Jung. 2023. COVID-19 infection inference with graph neural networks. (2023). https://doi.org/10.1038/s41598-023-38314-3

[7] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. 2018. Multiple Source Detection without Knowing the Underlying Propagation Model. (Dec. 2018). https://doi.org/10.48550/arXiv.1812.08434

[8] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2017. Graph Neural Networks: A Review of Methods and Applications. (Feb. 2017). https://doi.org/10.1609/aaai.v31i1.10477