CSE 8803 EPI: Data Science for Epidemiology, Fall 2023

Lecturer: B. Aditya Prakash
Scribe: Amish Saini, Atticus Rex

September 21, 2023
Lecture # : 9 Inference II

1 Summary

As data collection will never be a perfect process, we continue to discuss ways that missing data can be inferred from what data we do have. In the first half of the class, we wrapped up the Inference I powerpoint by discussing the GT Wifi Mobility Project that was used during the pandemic to track spread. Then we discussed several possible ways to calibrate agent-based models in general.

Then we moved onto the Inference II powerpoint and discussed two main topic: how to find Patient Zero and how to infer missing infections. In the former, several methods are given to inituitively track down the center of an infection graph. Then in the latter, we discussed 4 methods in detail for finding the infections that have slipped through the crack. Using these methods in tandem, we can observe a more complete view of an epidemiological history and better understand how to contain current and future epidemics.

2 Finding Patient Zero

Data collection is not perfect and will miss some cases. In some real-world disease analyses, such as with Tuberculosis (CDC, 2007) or AIDS, the source was able to be predicted through reconstructing the likely transmission path to the first case, Patient Zero [1]. Finding Patient Zero not only tells us the original causes of an epidemic, but can also tell us how to intervene.

Humans can pick up on trends, such as figuring out from past experience where the likely origin of these two clusters are:

Obviously, the patient zeros of these two clusters are in the centers, however, the question of how we implement this algorithmically on a graph remains. The answer is almost always MLE (Maximum Likelihood Estimation) [2, 3].

The best source is the one that maximizes the data likelihood in the SI model.

$$\hat{v} = argmax P(G_N | v* = v)$$

where G_N is the infected subgraph and v is an observed node in G_N.

For any given infection subgraph, there can be a multitude of starting positions. This means that calculating the probability of a given cascade is stochastic and non-trivial.

2.1 Rumor Centrality

This process sums up the likelihood of all ripples across different time snapshots of a network [7, 8].

$$P(G_N | v* = v) = \sum_{\text{all ripples}} R_i$$

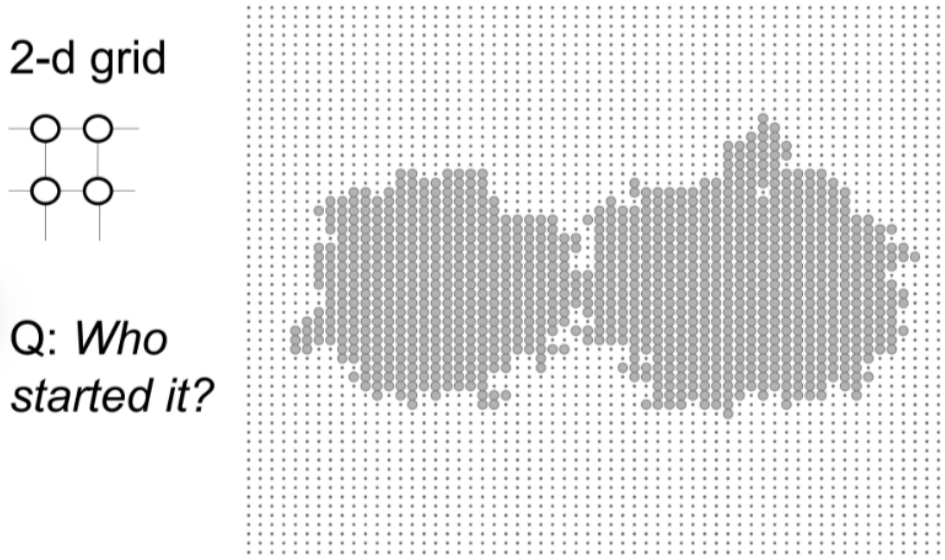


Figure 1: *This shows an example spread of infection. From simply looking at these two blobs, it seems obvious that the centers of the blobs should be the most likely sources of the patient zero. But how do we represent this algorithmically?*

These are difficult to compute: NP-hard (Due to the fact that there are so many different ways that the disease can propagate)! However, probabilities of different cascades can be computed efficiently for k-regular trees assuming it's an SI model. Effectively, we can figure out the most likely Patient Zero from trees, but then we have to extend this idea to regular graphs.

To extend for general graphs, extract a tree from the graph such as a Minimum spanning tree that covers all nodes in the graph or use breadth-first search to generate a subgraph. This can serve as a quick and dirty heuristic.

2.2 Finding Number and Identity of Sources

When looking back at Figure 1, we notice that it may seem likely that there are 2 sources at the centers of each cluster. However, how for more complicated scenarios, how can we tell not just who the sources are, but how many there are?

When infected nodes are surrounded by a lot of uninfected nodes, this reduces the chance that the former is the originator. This is called exoneration. Since the original infected has the longest time frame to infect neighbors, one that still has uninfected neighbors is less likely to be the first.

We arrive at a two part solution: use minimum description length for various numbers of seeds, and for that seed number, calculate exoneration as the sum of centrality and penalty. The first part, MDL uses bit compression to figure out which source is the simplest, and therefore according to Occam's Razor, the most likely explanation of the spread. The running time of this method is linear and can be optimized using NetSleuth. Seedset Scoring: $L(S) = L_N(|S|) + log\binom{N}{|S|}$ where $|S|$ is the encoding integer and the log term is the number of possible $|S|$-sized sets. Big O Runtime: $O(k * (E_1 + E_F + V_I))$, meaning it is linear!

The method to optimize the score is first we identify the high-quality node set given k,

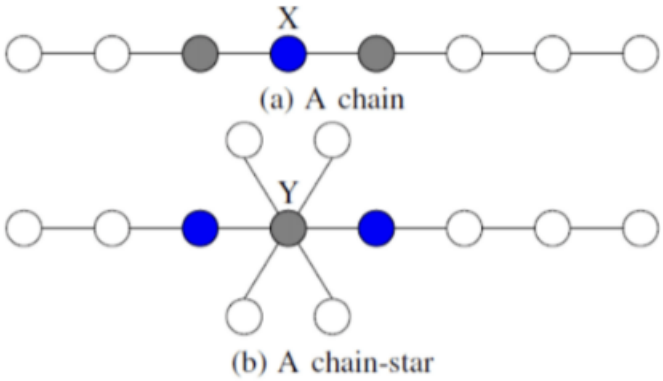


Figure 2: *This shows how the network geometry greatly affects the Maximum Likelihood Estimation. Because X is in the middle of two nodes and has infected both nodes, it is highly likely to be the source of the infections. Because Y has six neighbors and only two of them are infected, it is much less likely to have caused the outbreak. As we can see, having more uninfected neighbors makes Y trickier.*

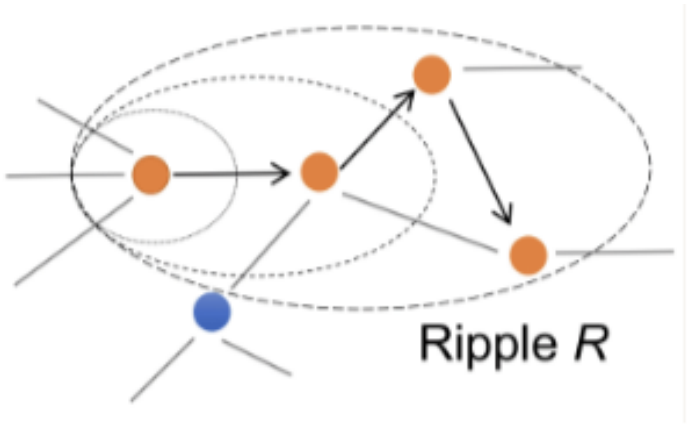


Figure 3: *This visualizes the propagation of an example ripple of infections within a network.*

and then given the nodes we must optimize the ripple R (Figure 3). The ripple cost of each additional one added is:

$$L(R|S) = L_N(T) + \sum_{t}^{T} L(F^t)$$

where $L_N(T)$ represents how long a ripple is and the summed term describes how the "frontier" advances.

Overall, the total MDL cost is

$$L(G_I, S, R) = L(S) + L(R|S)$$

Some extensions of The MDL Algorithm that have been published include:

- Different models

- Temporal networks
- More rigorous results on specific graphs
- Noisy input
- Using graph neural networks

2.2.1 GNNs

A final method to detect patient zero is using graph neural networks. We can use a deep generative model to estimate the distribution of diffusion sources. We can use the graph neural networks to approximate the forward direction of the epidemic and discover the diffusion probabilities. It can both learn the model parameters by iterating forward and guess the source through iterating in reverse.

2.3 Reconstructing the COVID-19 Pandemic Epicenter

How was Hunan Seafood Wholesale Market in Wuhan guessed as COVID ground zero? Researchers used spatial relative risk analysis done by compiling data from many sources: Weibo cases, CCDC sequencing PCR report, population density data, animal sales records, and mobility data. Investigators were then able to pinpoint the source to vendors selling live mammals as seen in Figure 4 [9].

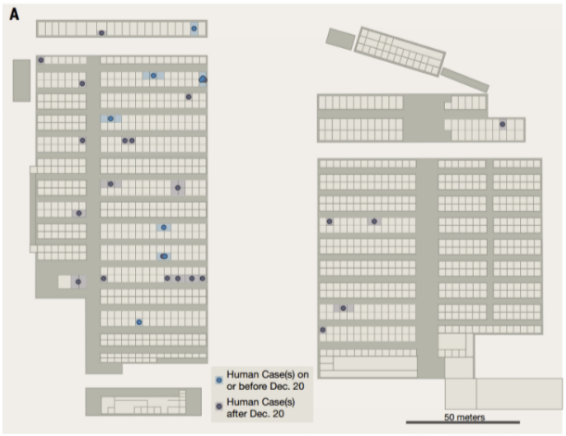


Figure 4: *This visual model shows how we can use physical spaces to model real-world interactions and make inferences about the kind of underlying networks that might be at play. The red dots show that new infections mostly happen within the vicinity of the old infection, indicating the efficacy of some kind of MLE model.*

3 Finding Missing Infections

A large number of COVID-19 infections went and still are continuing to go unreported. When there was only 23 reported infections in 5 major American cities in March 1st, there may have been greater than 28,000 in actuality. The difficulty in estimating the unreported infections allowed it to spread more quickly in the US and worldwide, and when severity is

slowed so too is the remedial action that follows. Potential reasons for missing infections vary from a lack of testing to asymptomatic infections.

To identify the true reported rate, consider a serological methodology. This is considered the gold standard for disease reporting, It tests if the person's body has the required antibodies for the disease, indicating that they contracted the disease. Hence, it can give a very accurate picture of the people who contracted the infection. However, this is an expensive method and is also a delayed process and we might not obtain the information for analysis in time. Comparing COVID to influenza and using the ILI system suffers from ad-hoc corrections and confounding factors.

3.1 Approach 1: Real-World Calibration

The idea is to use real-world data to provide some insight on the proportion of cases that aren't being reported. Then fit I_r to reported infections to estimate the unknown parameters such as the rate of reporting α. However, they suffer greatly from ad-hoc modeling assumptions. This means altering α slightly can drastically alter forecasts.

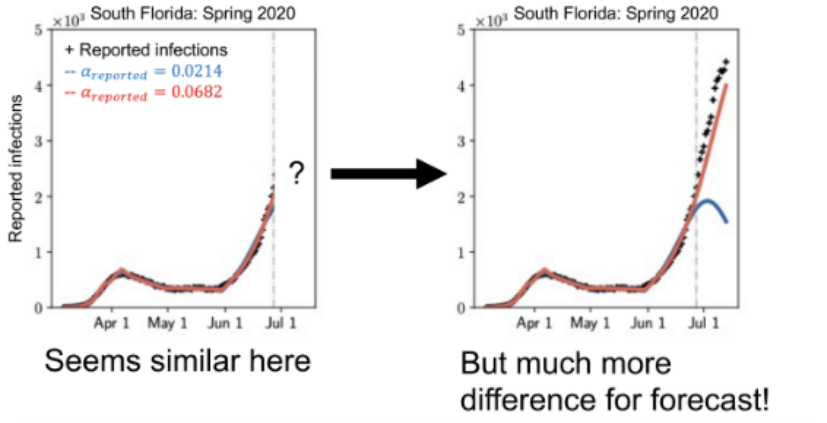


Figure 5: *With two different values of α, the model matches the training data exceedingly well. However, one model performs very well on holdout data, while the other performs very poorly. This illustrates the need for parameter optimization that goes beyond how how well a model matches training data.*

Overparameterization is a big problem in epidemiology and you can have many combinations of parameters that will fit your data.

3.2 Approach 2: MDLInfer

In this approach the number of reported infections is known: $D_{reported}$. If we are then given the correct count of total infections D, the model can be calibrated with both to find a better fit of $D_{reported}$. Put simply, we want to find the $D*$ that fits the $D_{reported}$ curve the best. Now we use MDL for a clear problem formulation:

$$D^* = \underset{D}{\operatorname{argmin}} L\left(D_{\text{reported}} \mid D\right) + L(D)$$

Governing Concepts:

- **Minimum Description Length (MDL):** This is largely driven by concepts from Information Theory that aim to find the best set of parameters to a model that both minimizes the amount of information needed to both describe the model and the residual errors between observed and model-predicted values.
- **Likelihood Function**: This is key in quantifying how well a model describes the observed data. In the context of epidemiological models, this function often involves terms for susceptible, exposed, infected, and recovered individuals (S, E, I, R).
- **Regularization**: A penalty term is introduced to the likelihood function to prevent overfitting. The MDLInfer algorithm favors simpler models unless a more complex model significantly improves the fit to the data. This is achieved by encoding the number of bits used in the model. A neural network would have a *lot* of bits because there are so many parameters, but an SIR model would only have a few because there are so few parameters.

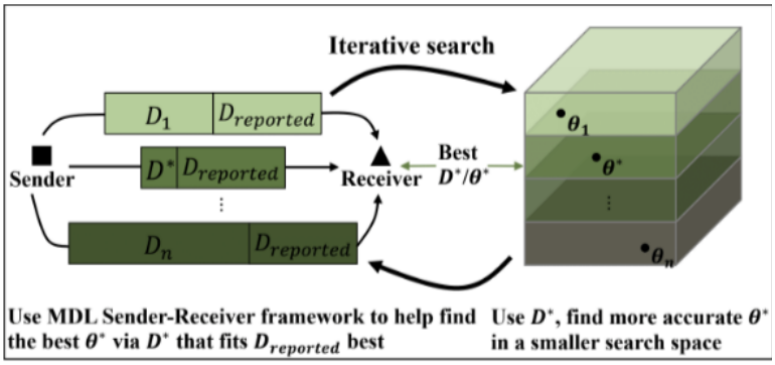


Figure 6: *This illustrates how the MDLInfer algorithm works; it relies on an iterative search through the parameter space and communication between a sender and receiver of information. This helps penalize the information of the data given the model and the information encoded in the model.*

This says that the information of the data given the model plus the information actually encoded in the model is the quantity of bits encoded [3]. This is the idea of Occam's razor which says that the best solution is the simplest solution because we can find many models that seem to fit well (Figure 5). MDLInfer was found to have great performance when predicting the future.

3.3 Approach 3: Steiner Trees

This is a datamining method that learns the tree with minimum weight in the graph that connects all of the given terminals. In our case, the terminals would be known cases and the minimum tree built to include each of those would implicate many other nodes that may be the unreported infections. These are determined with dynamic solvers [4].

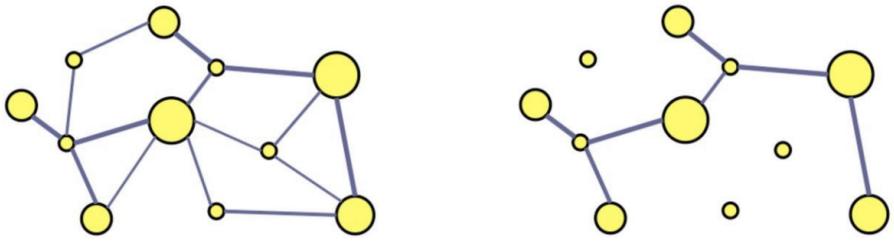


Figure 7: *Where larger circles are infected and smaller circles are uninfected, we can clearly see the construction of a minimum spanning tree requires uninfected nodes to connect infected ones. This gives us a direction to look in to find likely unreported infections.*

The Steiner tree method can be extended to temporal networks by converting it to one static graph. It can also be used to learn node weights from features, thus allowing us to estimate high-danger patients, for instance.

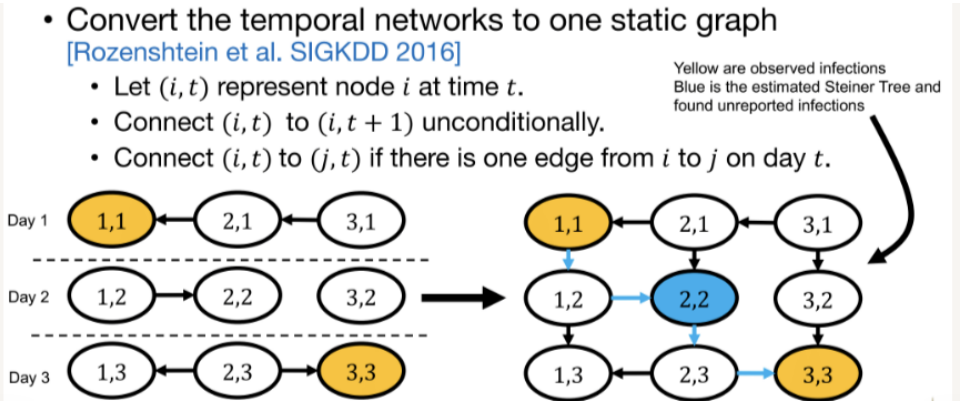


Figure 8: *Steiner Trees can be extended to temporal networks to find unreported infections.*

Steiner Trees allow epidemiologists to recognize patterns and identify what kinds of people are being unreported. Such a problem is common in hospitals.

3.4 Approach 4: NetFill

Missing infections can sometimes be derived based on seeds as shown in Figure 9. Because of this, it intuitively makes sense that our seeds are impacted by missing infections and can depend on whether we can recognize missing infections.

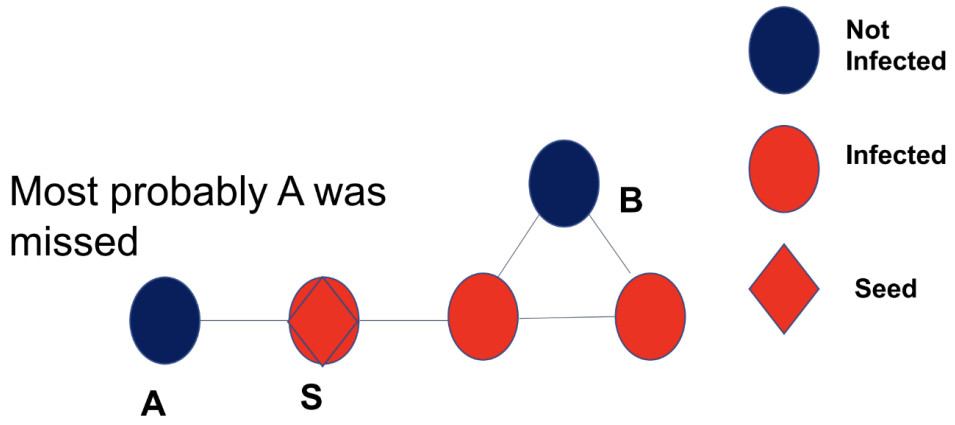
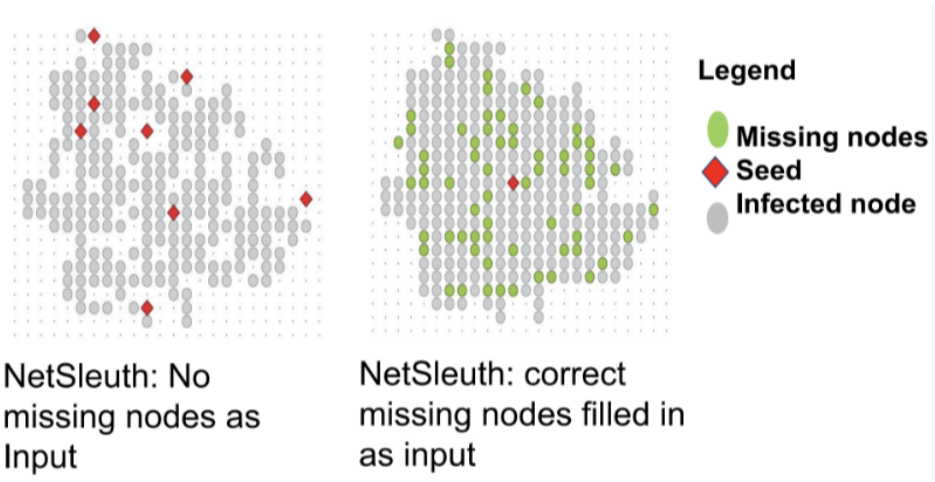


Figure 9: *We can see that missing infections can be detected based on their relationship to seeds. The seed S is directly connected to A and was able to infect both neighbors of B. Therefore, it is likely that A was a missed infection.*

NetFill is the idea of filling in gaps within a network to accurately trace back the seed of a network. This is done in the following steps:

1. Find starting points given missing nodes
2. Find missing nodes given starting points
3. Iterate above steps until convergence

The two ideas directly feed into one another and can be done in the NetSleuth algorithm [6].



4 Approach 5: Using Graphical Methods

Graphical methods may serve as a robust tool for modeling the dynamics of an epidemic, especially in situations where there exists a high probability of missing infections or "latent

spreaders" within a population. Essentially, these models use graphical representations such as networks or graphs to model the interactions between various entities to infer infection, recovery, exposure and susceptibility rates.

Latent Spreaders and Activation Probability

One of the key hurdles in epidemiological modeling is accounting for latent spreaders, or people who are infected but asymptomatic or do not get tested and, hence, are not included in official counts. In graphical models, latent spreaders can be represented as nodes with particular attributes that indicate their potential for spreading the infection [5].

The probability of activation in the presence of latent spreaders can be examined by analyzing the edges between nodes in the graph. By doing this, we can gather key insights into how a susceptible individual became infected when interacting with a latent spreader.

References

[1] N. Bock, P. Jensen, B. Miller, and E. Nardell. Tuberculosis infection control in resource-limited settings in the era of expanding HIV care and treatment. 196:S108–S113.

[2] S. R. Cole, H. Chu, and S. Greenland. Maximum likelihood, profile likelihood, and penalized likelihood: A primer. 179(2):252–260.

[3] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. 96(454):746–774. Publisher: Taylor & Francis _eprint: https://doi.org/10.1198/016214501753168398.

[4] F. K. Hwang and D. S. Richards. Steiner tree problems. 22(1):55–89. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.3230220105.

[5] M. Makar, J. Guttag, and J. Wiens. Learning the probability of activation in the presence of latent spreaders. 32(1). Number: 1.

[6] B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: How many and which ones? In *2012 IEEE 12th International Conference on Data Mining*, pages 11–20. ISSN: 2374-8486.

[7] D. Shah and T. Zaman. Rumor centrality: a universal source detector. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 199–210. Association for Computing Machinery.

[8] D. Shah and T. Zaman. Rumors in a network: Who's the culprit? 57(8):5163–5181. Conference Name: IEEE Transactions on Information Theory.

[9] M. Worobey, J. I. Levy, L. Malpica Serrano, A. Crits-Christoph, J. E. Pekar, S. A. Goldstein, A. L. Rasmussen, M. U. G. Kraemer, C. Newman, M. P. G. Koopmans, M. A. Suchard, J. O. Wertheim, P. Lemey, D. L. Robertson, R. F. Garry, E. C. Holmes, A. Rambaut, and K. G. Andersen. The huanan seafood wholesale market in wuhan was the early epicenter of the COVID-19 pandemic. 377(6609):951–959. Publisher: American Association for the Advancement of Science.