CSE 8803 EPI: Data Science for Epidemiology, Fall 2022

Lecturer: B. Aditya Prakash
Scribe: Gupta Shreyash, Zachary Wang

September 27, 2022
Lecture 10 : Outbreak Detection - 1

1 Introduction

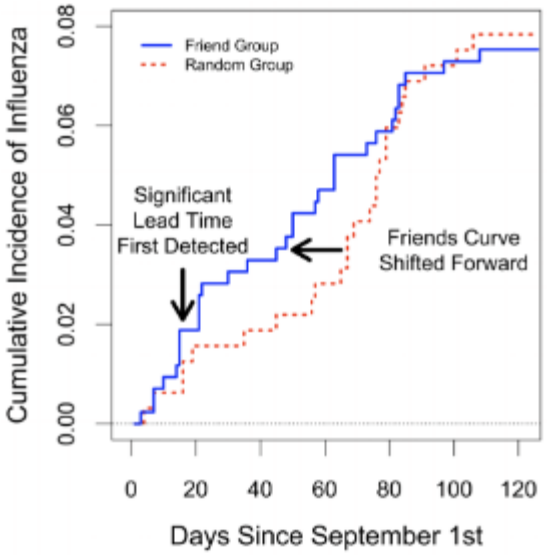
A critical question in epidemiology is how would one detect an outbreak detection. To answer this, one major idea is to think about how we effectively track only a subset of people from a population and figure out if there is an outbreak of a disease. In practice, such an approach would be difficult due to the fact that we lack the resources to track a significant subset to check if there has been an outbreak. With that in mind, we need to consider how to choose the best candidates for sensors.

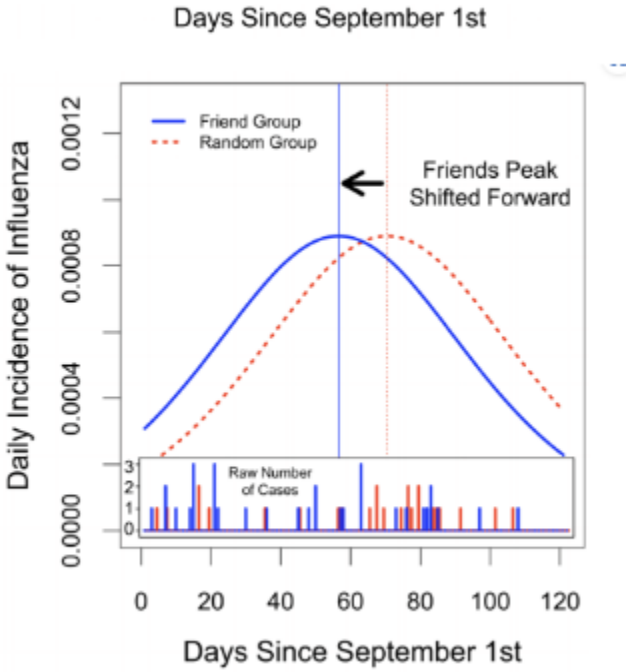
We can think of a population as a social network, in which case we can use the friends-of-friends paradox we have discussed to determine that taking sensors from friends of a random sample is a good way to obtain lead time compared to just sampling randomly, supported by a study on Harvard students. Another way to pick sensors is the idea of dominator trees, where nodes that are present along the shortest paths between other nodes are often good choices for sensors. Finally, we consider the problem of detecting outbreaks in a cascade, in which there is a submodular function that can provide a fast and effective approximation for the optimal set of nodes to select in graph G.

2 Idea of Social Network Sensors

2.1 Social Network Study

This brings light to the idea of social network sensors that was studied in a study [**?**], where 774 undergraduate students from Harvard College were tracked from Sept 1^{st} to Dec 31^{st} 2009 for when they got the flu during the fall. They further took two major classes of data based on the known friends, adding 425 samples to the already established sample network and a network with 319 samples selected randomly from 6650 undergraduates.



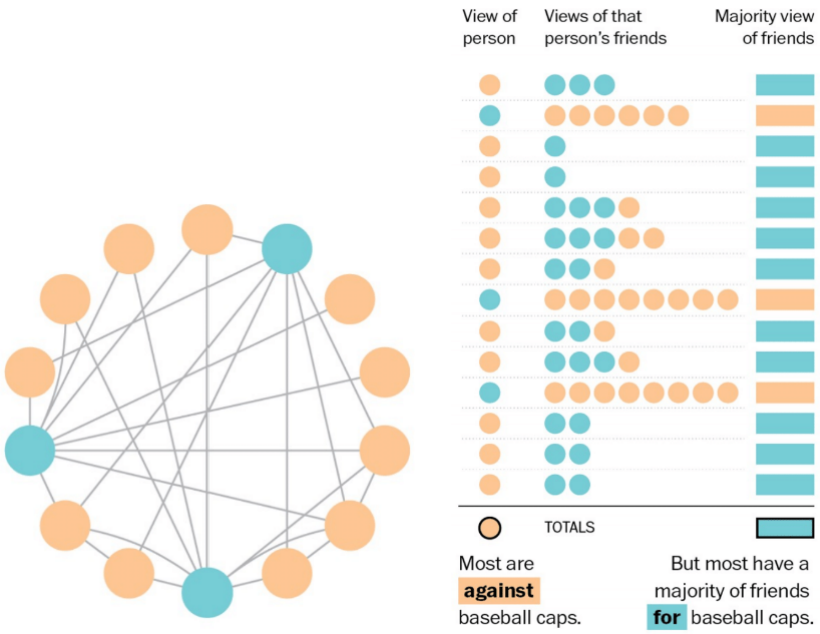


From the figures, there was an observed trend shift between the "Friends" set and the "Random" set. This helped in the detection of a significant lead time that was observed based on the plots. This is further corroborated in the depiction of the same figure in the daily incidence trend which shows the lead time that the "Friend" set had over the "Random" group. The lead time advantage "Friend" had over "Random" was of 5 days.

This observation can be scaled to identify an outbreak of contagion and applied generally, where we can track a subset of people which can give us a lead time advantage. In order to identify the right folks for the job, one should recall the friendship paradox's key statement regarding the average variance of friends of friends one has will always be greater than the average variance of friends one has. In this case study context, random samples will be less essential than the friends of random samples in outbreak detection.

2.2 Majority Illusion

The idea of the "Friendship Paradox" can also create an issue of majority illusion as exhibited by Lerman et al 2015 [**?**]. This is widely observed in political science scenarios where the senate might have an apparent majority in raw numbers. In the illustration described in the class, we can visually attribute the majority of oranges who are against baseball caps, while blues who advocate for baseball caps are in minority. But upon analyzing the catalogue for the friends of every senator we see a shift from the majority political stance. This is attributed to the perception influenced by the blues. A real-world example of this is the support for same-sex marriage. Once you get to know someone who advocates that opinion, your own viewpoint changes.



In public health scenarios, the majority are susceptible population groups and the minority are the infected populations which can infect others.

2.3 Formal Definition for Selecting Sensors

We can formally define the problem for sensor selection with two main methods: PLTM, where we maximize the lead time for the predicted peak, or MAIT, where we are trying to minimize the time to detection for infected nodes.

(ϵ, k)-Peak Lead Time Maximization (PLTM)
Given: Parameters ϵ and k, network G, and the epidemic model
Find: A set of nodes S from G such that

$$S = \underset{S}{\operatorname{argmax}}\, E[t_{pk} - t_{pk}(S)]$$
$$\text{s.t. } f(S) \geq \epsilon,\ |S| = k$$

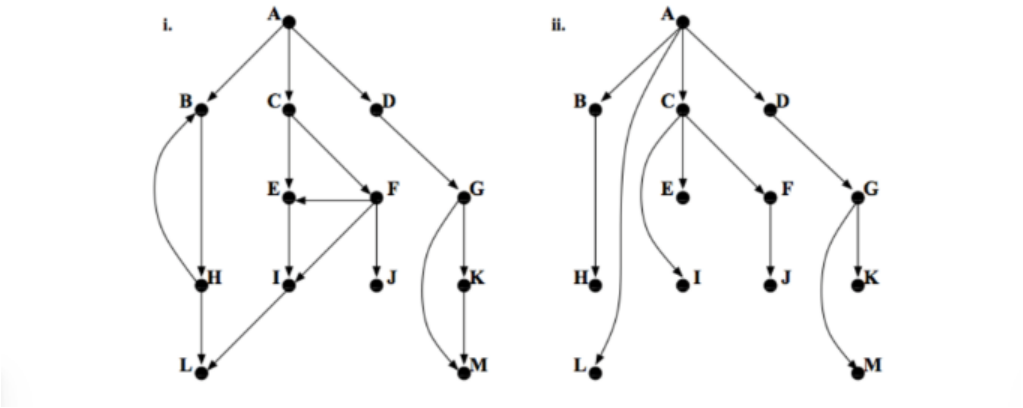
(ϵ, k)-Minimum Average Infection Time (MAIT)
Given: Parameters ϵ and k, network G, and the epidemic model
Find: A set S of nodes such that

$$S = \underset{S}{\operatorname{argmin}} \sum_{v \in S} t_{inf}(v)/|S|$$
$$\text{s.t. } f(S) \geq \epsilon,\ |S| = k$$

2.4 Dominator Trees

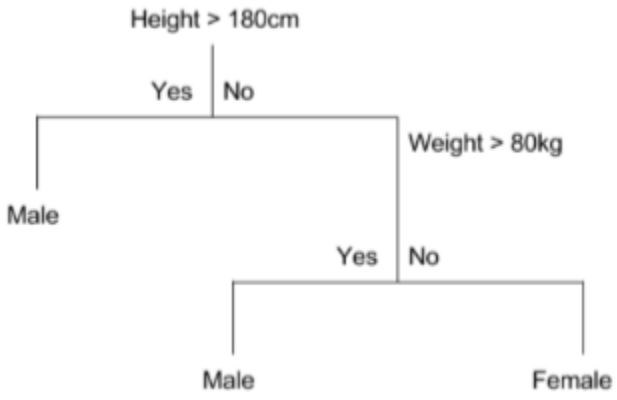
Another method for selecting effective sensor nodes is to use dominator trees. The idea is that nodes that are present on many of the shortest paths between other nodes are more likely to be infected when an epidemic spreads throughout the graph. Following this idea, we can generate dominator trees for dendrograms on a graph, and the top k nodes in that tree will become our sensor set. While it has limitations, this algorithm has the merit of being especially fast, running in linear time over a graph.

1. generate dominator trees corresponding to each dendrogram;
2. compute the average depth of each node v in the dominator trees (as in the transmission tree heuristic);
3. discard nodes whose average depth is smaller than ϵ_0;
4. we order nodes based on their average depth in the dominator tree, and pick S to be the set of the first k nodes.



2.5 Surrogates

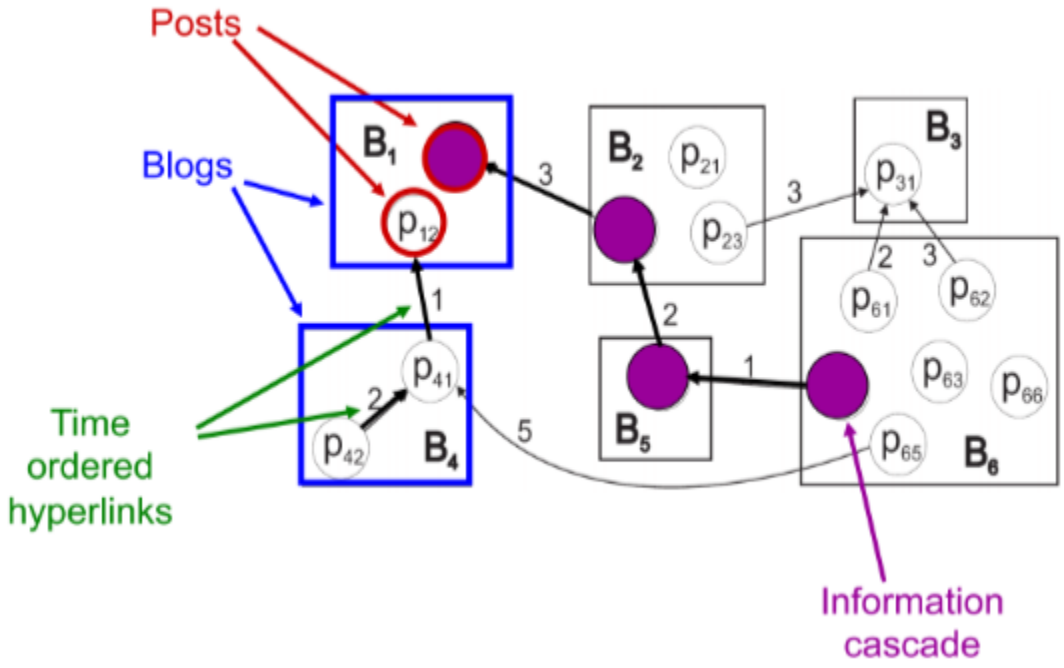
We know there are methods for finding the best theoretical nodes to be sensors in a graph, but how can apply this knowledge to the real world? In other words, how can we use the information from these methods to determine who in the real world is an effective sensor? One way is a decision tree that determines if a person is a good sensor candidate.



3 Cascades in Blogs

3.1 Problem and Formulation

In this problem, instead of placing sensors to detect outbreaks before they happen, we are given the cascade of an outbreak beforehand and want to place sensors to detect all possible infected nodes.



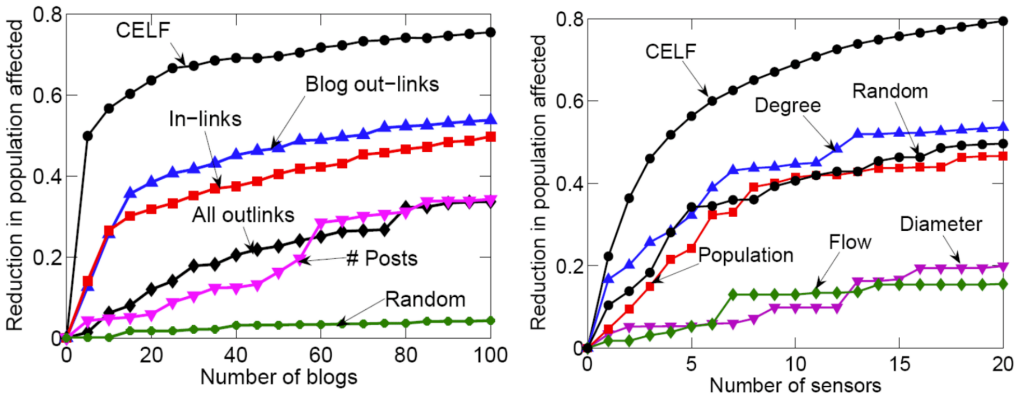
Given a series of cascades over a network, place sensors to detect the outbreaks of those cascades. The problem is formulated as follows:

- Given:
  - a graph G(V,E)
  - a budget B for sensors
  - Cascades
- Select a subset of nodes A that maximize the expected reward R:

$$R = \sum_i P(i) R_i(A) = \pi(\emptyset) - \pi(A)$$

s.t. cost(A)<B

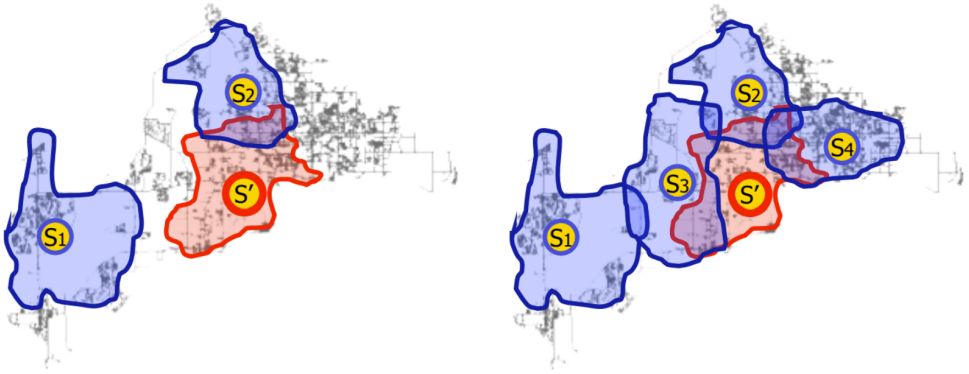
To put it simply, we simply trying to pick nodes up to a budget B such that we maximize the expected reward R of those nodes. We calculated the reward of a set of nodes by looking over all i cascades and summing the reward of the sensors in each of those cascades.
An important property of this function is that $R(A)$ is submodular, meaning it can be approximated in a reasonable amount of time. This is crucial because trying to solve this problem by brute force is would be an impossibly expensive task on large graphs.
Thus in such a case, greedy algorithms are implemented. One of them, CELF performs well as proven by the figures given below.



For a greedy algorithm to work, the function should be submodular in nature. A function $f(S)$ is submodular when

- Non-negative
- Monotone $f(S + v) \geq f(S)$
- Has diminishing returns property where for all $S \in T$

$$f(S+v) - f(S) \geq F(T+v) - f(T)$$



(a) *Adding s' to set $\{s_1, s_2\}$* (b) *Adding s' to superset $\{s_1, \ldots, s_4\}$*

In other words, if the gain of adding a node to a smaller set is greater than the gain of adding a node to a bigger set, a non-negative and monotonous function will be submodular function.
For a NP-Hard problem to calculate submodular functions, one can use greedy algorithms to get an approximation.

References

[1] F. Christakis. Social network sensors for early detection of contagious outbreak. In *PLoS One*, 2010.

[2] Z. W. Kristina Lerman, Xiaoran Yan Xin. The majority illusion in social networks. In *arXiv:1506.03022*, 2015.