CSE 8803 EPI: Data Science for Epidemiology, Fall 2023

Lecturer: B. Aditya Prakash
Scribe: Nicolas Zacharis, Ishan Nandi

September 26, 2023
Lecture 10 : Outbreak Detection - 1

1 Introduction

A major problem in epidemiology is figuring out how to detect an outbreak of a contagion as early as possible. One answer to this is to track a subset of a population and ascertain if there is an outbreak. However, due to lack of resources, we cannot track a significant enough subset of the population for this method to work. So, we need to find the best candidates within the population, sensors, to track.

There are multiple methodologies for picking sensors. One such methodology is to track the friends of a random sample of the population. This can be more effective than just sampling randomly from a population. Another methodology for picking 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 study of the outbreak of influenza among Harvard students in 2009 [1]. In this study, a social network of 774 undergraduate students was constructed from 6650 undergraduates with two major data groups. One group was a random sample of 319 students and the other group was random sample of 425 of their friends. Both groups were tracked for the spread of influenza and the observations between the two were compared.

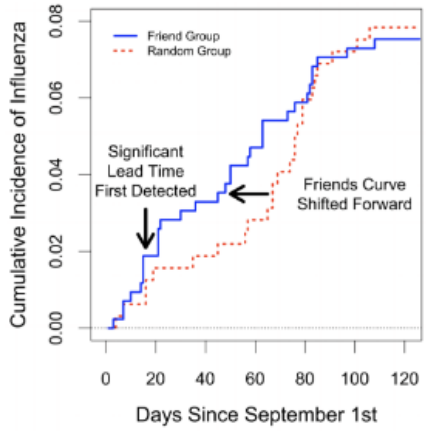


Fig. 1. Cumulative incidence of influenza after September 1st for friend group and random group. Depicts friend group having a significant lead time compared to the random group.

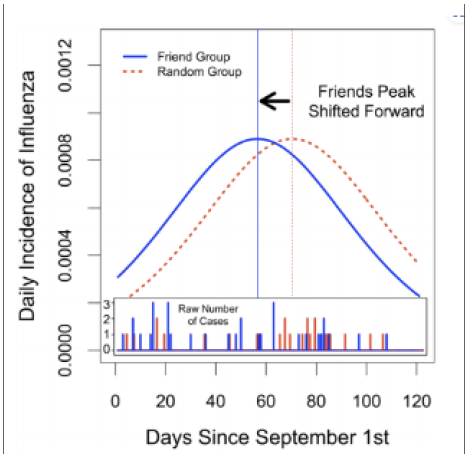


Fig. 2. Daily incidence of influenza after September 1st for friend group and random group. Depicts peak of influenza cases for friend group occurring earlier than random group.

From the figures above, there was an observed trend shift between the friends group and the random group. This trend shift meant that the friend group tended to get influenza before the individuals in the random group. As a result, a significant lead time was observed between the groups as shown in Fig. 1. Due to this significant lead time, the friends peak of daily incidence was shifted forward as shown in Fig. 2.

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 [2]. This is widely observed in political science scenarios where the senate might have an apparent majority in raw numbers. In class, we saw an example illustrating the majority illusion.

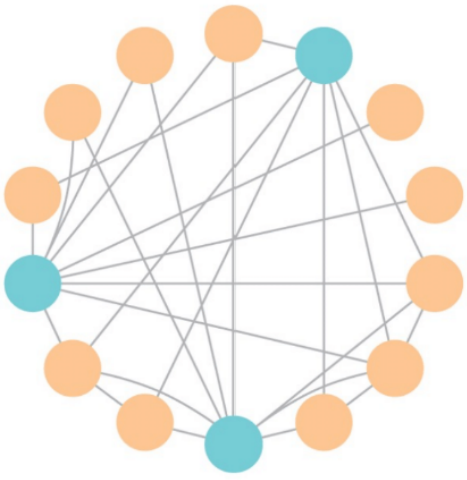


Fig. 3. Network depicting people in favor of baseball caps (blue) and people against baseball caps (orange).

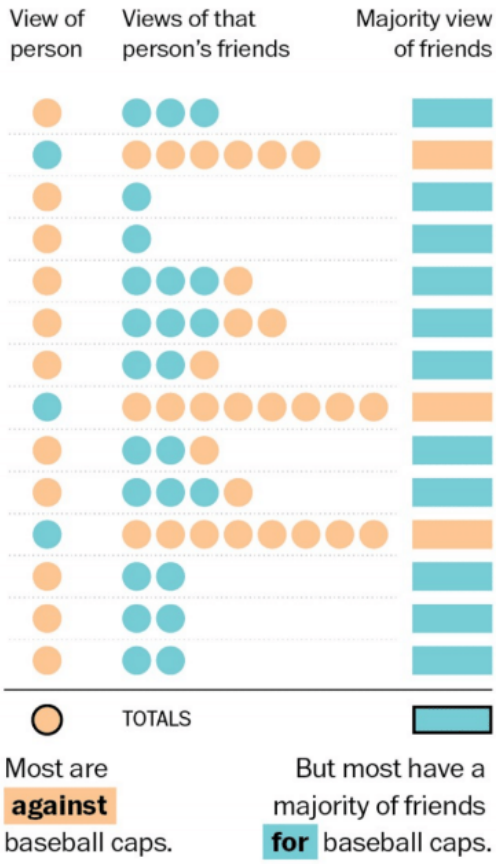


Fig. 4. Local view of the network for each person. Depicts distorted view of network when comparing to the global view.

In Fig. 3 above, we can see the global view of the network in which a majority of people

are against baseball caps (oranges) while the people who are for baseball caps are in the minority (blues). Fig. 4 shows the local view of each person's friends in the network. When analyzing this, we see that people may have a different perception of who is in the majority based on their local view. This perception may be incorrect, as seen in Fig. 4, which is only able to be fixed by disseminating the global view of the network to everybody. 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 notes S from G such that

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

(ϵ, k)-Minimum Average Infection Time (MAIT)

Given: Parameters ϵ and k, network G, and the epidemic model
Find: A set of notes S such that

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

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 tree (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 to the dominator tree, and pick S to be the set of the first k nodes.**

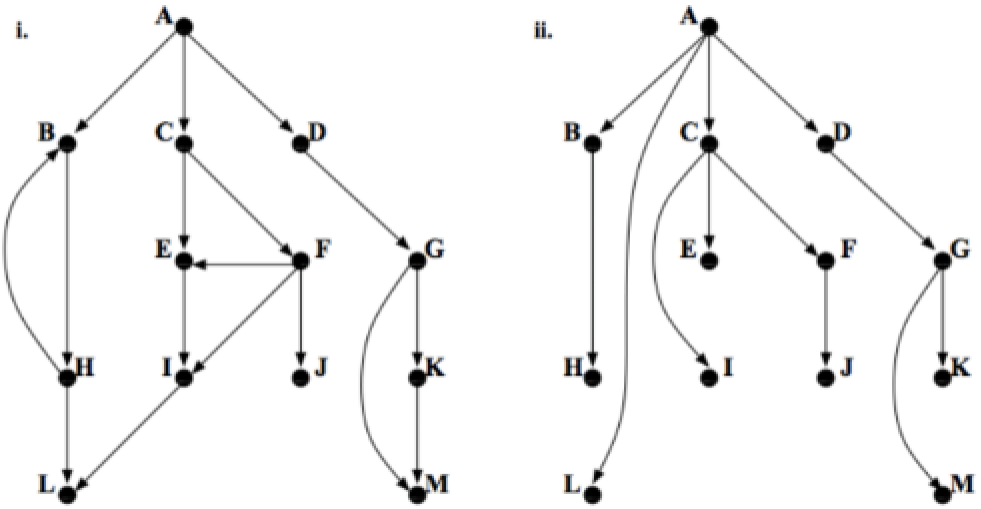


Fig. 5. A graph (i.) and the graph's dominator tree (ii.).

2.5 Surrogates - Redescriptions

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. We can use this to correlate which demographic features correlate to sensors found.

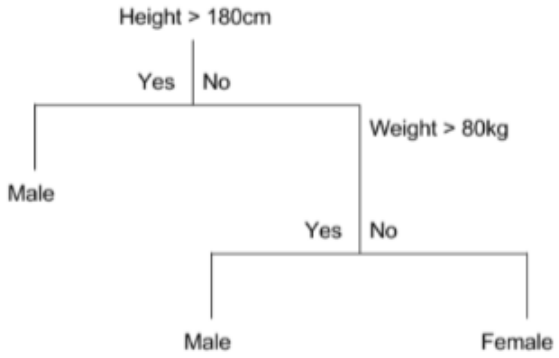


Fig. 6. An example decision tree with demographic features.

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. Lescovec *et al.* investigate an analogous problem domain of information

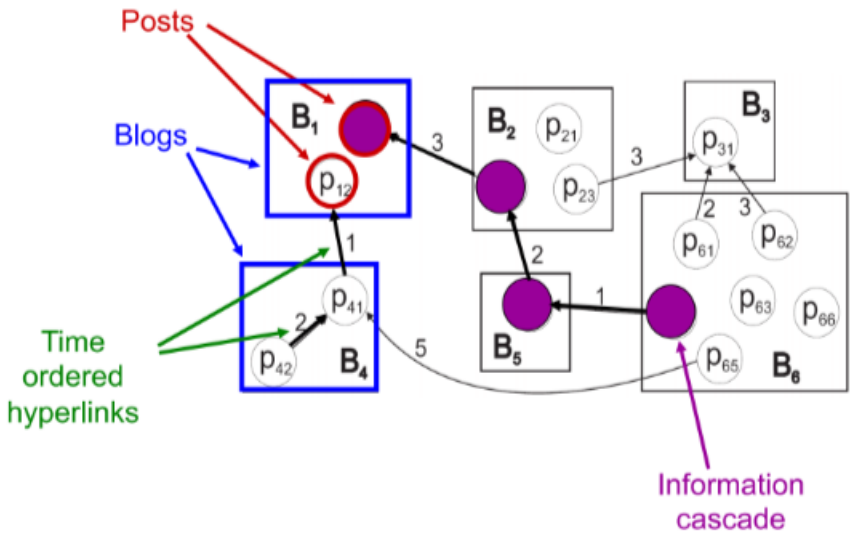


Fig. 7. The time evolution of an information cascade through the posts (p_{ij}, red circles) of blogs (B_i, blue boxes).

propagation between online bloggers' blog posts, for which each story or topic posted about and spread corresponds to an "information cascade" [3]. Selecting the fewest blogs that soonest participate in the most cascades (i.e., the blogs that are the most up-to-date for the greatest number of stories) is analogous to the individuals that, for multiple epidemics, are consistently closest to and soonest infected by the epidemics' patient zeroes. For a solution A to these problems – any set of blog posts, or epidemic patients – Lescovec *et al.* identify multiple criteria to be optimized, each scored and packaged into the vector $R(A)$: (1) *detection likelihood*, the fraction of cascade events any of A's elements participate in; (2) *detection time*, the time elapsed until an element of A becomes involved in a cascade; and (3) *population affected*, those *not* part of a cascade at the moment it is detected by an element of A (in an epidemiological context, those "saved" by detecting an outbreak early).

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, and cascades
Find: A subset A of nodes that maximize the expected reward R, where:

$$R = \sum_i P(i) R_i(A) = \pi(\varnothing) - \pi(A)$$
$$\text{s.t. } \text{cost}(A) < B$$

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.

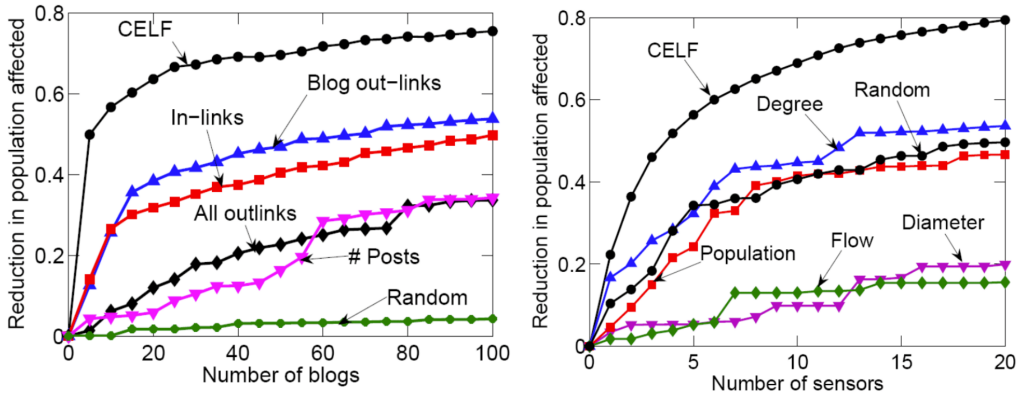
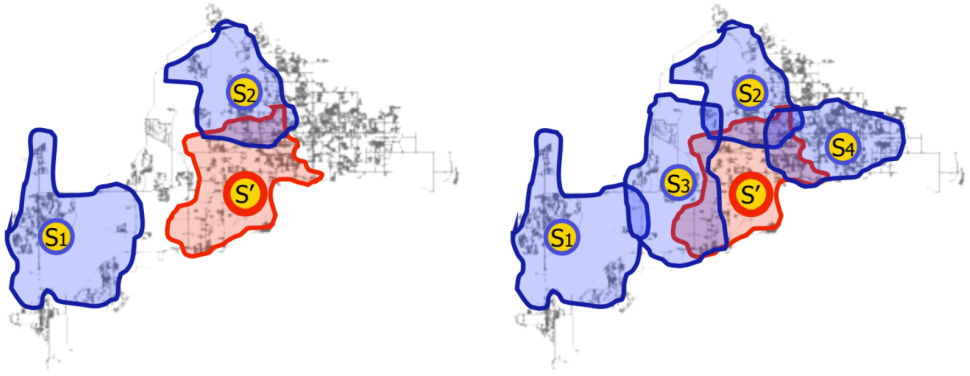


Fig. 8. The diminishing-returns performance of the CELF algorithm in detecting information diffusion in blogs (left) and epidemic outbreaks (right). Because CELF is non-negative, monotonically increasing, and exhibits the diminishing-returns property, we can conclude it to be an acceptable approximation of a submodular function, and thus a valid outbreak detector.

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 outbreak detection objective functions f (such as detection likelihood and detection time for a subset S of nodes) must be submodular, or change more slowly as the size of S increases (*diminishing returns*: every new node added to S should contribute less and less to the value of the objective $f(S)$). $f(S)$ is submodular when:

- Non-negative
- Monotone $f(S+v) \geq f(S)$
- Has diminishing returns property, where $f(S+v) - f(S) \geq f(T+v) - f(T)$ for all $S \subseteq T$ (the gain of adding a node v to a smaller set S is greater than adding v to a larger set T)



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

Though optimizing submodular functions is an NP-Hard problem, designing the outbreak detection reward function R to be submodular permits the use of greedy algorithms to get an *approximation*, such as hill-climbing techniques.

References

[1] N. A. Christakis and J. H. Fowler. Social network sensors for early detection of contagious outbreaks. *PLoS ONE*, 5(9), 2010.

[2] K. Lerman, X. Yan, and X.-Z. Wu. The "majority illusion" in social networks. *PLOS ONE*, 11(2), 2016.

[3] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.