Researching of Improved Greed Algorithm used in COVID-19 Burst Detection

Gang Yijin [1,2], Wu Guozeng[1,2], Li Tao[1,2], Hu Mingyang[1], Wang Zhiwei[3]

[1] SIPPR Engineering Group Co., Ltd, Zhengzhou, China

[2] School of Human Settlements and Civil Engineering (HSCE) of Xi'an Jiaotong University, Xi'an, China

[3] Faculty of Science of Melbourne University, Australia

Abstract

Nowadays, the outbreak of COVID-19 has severely affected people's normal lives. How to detect the source of infectious diseases as soon as possible by observing as few people as possible before the outbreak of the epidemic, to prevent more people from being infected, is a research problem of great practical significance. This problem is a burst detection, and we need to select one of the cases in many hospitals in our country for detecting. This article uses a new method improved from the conventional greedy algorithm to detect this problem and related problems, thus showing the characteristics of "sub-modularity". This algorithm is suitable for large-scale problems, and the simulation results are close to the optimal solution.

Keywords*: COVID-19; greed algorithm; infectious diseases burst detection; sub-modularity*

1. Introduction

Due to the suddenness of outbreaks of infectious diseases, epidemics often occur in a short time and in local areas, and the number of patients suddenly increases. As we all know, the early stages of an infectious disease outbreak are the best time for epidemic control and patient treatment. Therefore, in the early stages of the epidemic, the impact of an epidemic can be effectively controlled by observing as few people as possible in the shortest possible time to detect the occurrence of an epidemic. Take the COVID-19 as an example. This infectious disease can be spread through respiratory droplets, close contact and aerosols, and people are generally susceptible to this disease. The early symptoms of COVID-19 are similar to influenza, so it is not easy to detect in the early stages of the epidemic.

Under normal circumstances, the hospital writes medical records based on patients' visits and enters them into its electronic database. Therefore, we can observe the spread of the disease by checking the time and place of entry of the medical record. In this case, we hope to check a recently updated electronic medical record to obtain more information. The simplest and most intuitive method is to select the big and well-known hospitals. The medical record databases of these hospitals contain more information, but because these databases contain more medical records, it takes a long time to detect. Another method is to choose a regular clinic in the city center. These clinics are trusted and convenient places for office workers and elderly people to see a doctor. Therefore, there will be many medical records of conventional diseases in their database. This is the goal of the algorithm described in this article. This is the goal of the algorithm described in this article. There is more than one criterion for optimizing targets in burst detection, such as the shortest time and the least affected people. In the algorithm, these criteria are described by objective functions. Optimizing these objective functions is NP-hard, and the result of the algorithm described in this article is close to optimal. The transmission of disease information is shown in Figure 1 below.

Figure 1 give the spread of disease among areas (two areas for example), We want to pick a few people quickly to capture most cascades.

ISSN: 0010-8189

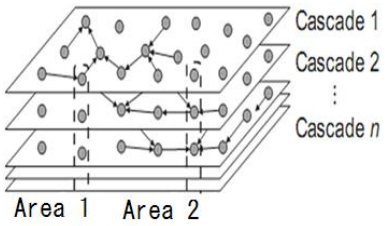

Figure 1: Spread of disease between areas

(Each layer shows an information cascade, each circle represents a person and all people on the same vertical column belong to the same area. Edges represent the flow of information. The cascade starts at top-left circle of the top layer)

2. Algorithm caption

During the spread of the epidemic, we want to select a subset of people, A, in a graph $G = (v, \varepsilon)$, which detect outbreaks (spreading of a virus/information) quickly.

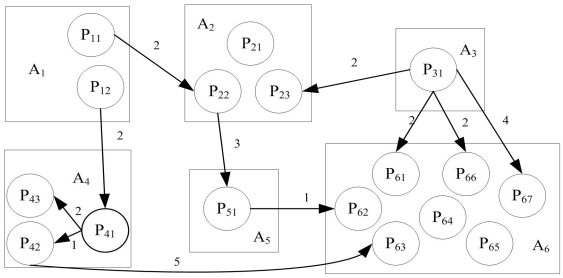

Figure 2: Human-to-human virus transmission

The arrows point to the destination of information and the cascades grow (information spreads) in the direction of the edges. Fig.2 shows an example of the spread of the virus in the population. Each of the six area consists of a group of people. Arrows between people represent the spread of the virus, and the label next to an arrow shows the time difference between the person carrying the virus and the infected person (e.g., person P_{31} points to P_{23} and labeled with number 2, which demonstrate that P_{23} will be infected two days after P_{31} is infected). The outbreak of the epidemic (e.g., information cascades) starts from several individual nodes (e.g., P_{11}, P_{12} and P_{31}) and gradually spreads across the entire graph, so it takes a certain amount of time to traverse each transmission path $(s, t) \in \varepsilon$ (determined by the label of the arrow).

Once the virus reaches the selected node, an alarm is triggered. For example, area A6 will detect cascades derived from person P_{11}, P_{12} and P_{31}, respectively 6, 8, 2, and 4 days after the start of each cascade.

According to the node we choose, we can get a certain placement score. Figure 2 shows several criteria that may be optimized. If we only want to detect as many people as possible, it is better to only monitor area A6. (Only monitoring area A6 can capture all cascades, but it takes more time.) However, monitoring A1 will only miss one cascade (i.e., P_{31}), but it will take a short time to detect other cascades. In general, this placement score is a set of functions R that maps each placement *A* to the real number *R(A)* (our reward) that we intend to maximize.

We associate the non-negative cost *c(s)* with each node (person) s, we also associate the cost *c(A)* with each placement *A* and require that the cost does not exceed the specified budget B that we can spend. And define the cost of placement *A*:

$$c(A) = \sum_{S \in A} c(s) \qquad (1)$$

So, our goal is to solve the optimization problem.

$$MAX \quad R(A) \quad (A \subset V) \qquad (2)$$
$$s.t \quad c(A) \leq B$$

(Where *B* is the budget, which we can spend on detecting people.)

A virus, *i,* originates from a node $s' \in V$ in a community $G = (v, \varepsilon)$ (where $i \in \Gamma, \Gamma$ is a set of diseases,) and spreads among people, affecting other nodes through contact. Eventually, it reaches a monitored node $s \in A \subseteq V$ (i.e., the person we detected) and is detected. Based on the detection time $t = T(i, s)$ and the impact of the virus on the community before the detection (e.g., the size of the missing cascade), we will produce a penalty $\pi_i(t)$. Note that the penalty function$\pi_i(t)$depends on the diseases. Our goal is to minimize expectations.

This is the penalty for all possible diseases:

$$\pi(A) = \sum_i P(i)\, \pi\big(T(i, A)\big) \qquad (3)$$

$$A \subseteq V, T(i, A) = \underset{s \in A}{MIN} T(i, s) \qquad (4)$$

where, for a placement A ⊆ V, *T(i, A)* is the time it takes for the detection system in A to detect virus *i*, and P is a (given) probability distribution of the virus. We assume that $\pi_i(t)$ is monotonically constant in t. We also let $T(i, \Phi) = \infty$, and set $\pi_i(\infty)$ as the maximum penalty incurred for not detecting virus i. Therefore, in addition to minimizing the penalty π(A), we can also consider reducing the disease-specific penalty and reducing the expected penalty.

$$R(A) = \sum_i P(i)\, R_i(A) = \pi(\Phi) - \pi(A) \qquad (5)$$

This alternative formula has a key attribute, that is, sub-modularity. This attribute shows diminishing returns. That is to say, when we only examine the medical records of a few people, we can obtain more information than when we detect many people. The formula expression of this attribute is as follows.

$$R(B \cup \{S\}) - R(B) \leq R(A \cup \{S\}) - R(A) \quad (A \subseteq B \subseteq V) \qquad (6)$$

3. Improved algorithm

Greed algorithm: having a constant cost function (usually$c(s) = 1$), and iterative k steps, adds the location s_k which maximizes the marginal gain.

$$S_k = \underset{s \in v \backslash A_{k-1}}{argmax} (A_{k-1} \cup \{s\})(A_{k-1}) \qquad (7)$$

Once B elements are selected, the algorithm ends the operation, thus ensuring that the greedy algorithm can find a solution that can achieve at least a constant fraction 63% of the optimal score. [1]
When cost function is non-constant:

$$S_k = \underset{s \in v \backslash A_{k-1}}{argmax} \frac{R(A_{k-1} \cup \{s\}) - R(A_{k-1})}{c(s)} \qquad (8)$$

The reference [2] proves that the performance of this algorithm worse than the optimal solution. We can add the following steps to the ordinary greed algorithm to get the new algorithm demonstrated in this article.

1) Use equation (7) to get a set of people, A_1
2) Use equation (8) to get a set of people, A_2
3) Max $\{R(A_1), R(A_2)\}$[3]

$$max\{R(A_1), R(A_2)\} \geq 1/2(1 - 1/e) \underset{A,\, c(A) \leq B(A)}{max} \qquad (9)$$

Assume the marginal increments is $\delta_s(A) = R(A \cup \{S\}) - R(A)$ (or$\delta_s(A)/c(s)$) for all $s \in V \backslash A$. The key point we need to pay attention to is that as A increases, the marginal increment will never increase. For$A \subseteq B \subseteq V$, it holds that $\delta_s(A) \geq \delta_s(B)$. So instead of re-computing $\delta_s = \delta_s(A)$ for every node after adding s' (and hence requiring $|V| - |A|$ evaluations of R).

4) Perform lazy evaluations: Initially, we mark all δ_s as invalid. When the location of the next node is found, we go through the nodes in decreasing order of their δ_s. If the δ_s of the top node s is invalid, we will recalculate it and then insert it into the existing order of δ_s (e.g., by using a priority queue). In many cases, the recalculated δ_s is basically unchanged, so even after recalculation, the top element remains unchanged. In this case, we found a new node (person) that can be added, and there is no need to re-evaluate δ_s for everyone.

The inverted index is the main data structure we use in the optimization algorithm. In the spread of disease, we need to consider millions of people, who form a cascade. However, in the epidemic, not everyone will be exposed to the virus. Therefore, most nodes s will not cause a penalty (i.e., $R_i(\{s\}) = 0$). Therefore, we can get $R(A)$ without scanning the entire data set.

4. Algorithm pseudocode and simulation result

Using the data set in reference [4] (3.5 million people, at least 3 links), extracting 100 data for simulation, the MATLAB simulation result are shown in Figure 3.

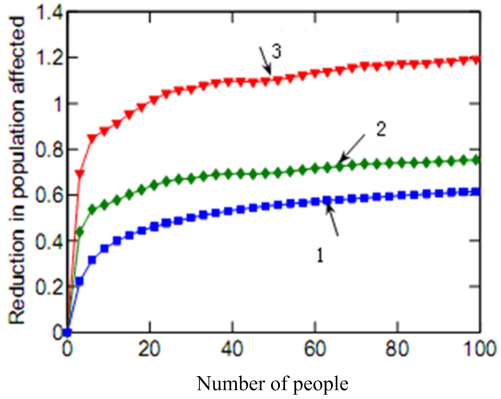

Figure 3(a): The performance of proposed algorithm

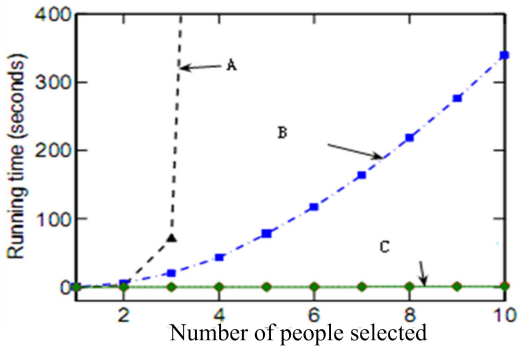

Figure 3(b): Running time

In figure 3(a), the third bound shows that the unknown optimal solution lies between our solution bottom line and the bound (top line). Notice the discrepancy between the lines is big, which means the bound is very loose. On the other hand, the middle line shows the second bound, which again tells us that the optimal solution is somewhere between our current solution and the bound. Notice, the gap is much smaller. This means (a) that the first bound is much tighter than the traditional third bound. (b) the proposed algorithm performs very close to the optimum.

Figure 3(b) plots the running time of selecting k people. (A represent exhaustive search, B represent the naive greed algorithm, and C is the algorithm this paper present) We see that exhaustively enumerating all possible

subsets of k elements is infeasible (the line jumps out of the plot for k = 3). The simple greedy algorithm scales as Ω(k|V|), since for every increment of k we need to consider selecting all remaining |V| − k people. The bottom line overlapping the x-axis of Fig. shows the performance of this algorithm. For example, for selecting 200 people, greedy algorithm runs 9.0h, while this algorithm takes 50 seconds (700 times faster). Algorithm pseudocode as following.

Function: F $(g = (v, \varepsilon), R, c, B)$

$A \leftarrow \varepsilon$; for each $s \in v$ do $\delta_s \leftarrow +\infty$;

while $\exists s \in v \backslash A; c(A \cup \{s\}) \leq B$ do

for each $\exists s \in v \backslash A$ do

flag s $\leftarrow false$;

while true do

if type=1 then $s^* \leftarrow \underset{s \in v \backslash A, c(A \cup \{s\}) \leq B}{argmax} \delta_s$;

if type=2 then $s^* \leftarrow \underset{s \in v \backslash A, c(A \cup \{s\}) \leq B}{argmax} \frac{\delta_s}{c(s)}$;

if flag s then $A \leftarrow A \cup S^*$;break;

else $\delta_s \leftarrow R(A \cup \{S\}) - R(A)$; flag s $\leftarrow ture$;

return A;

then :$A1 = \mathrm{F}(g = (v, \varepsilon), R, c, B, 1)$;

$A2 = \mathrm{F}(g = (v, \varepsilon), R, c, B, 2)$;

return argmax{ $R(A1), R(A2)$ }.

5. Conclusion

Infectious disease detection is an important application of the burst detection, the study of disease outbreaks may provide some important instructions about the disease for us, so we can take some real precautions to reduce losses. This paper uses an improved greedy algorithm to detect outbreaks of infectious diseases. Multi-objective functions are given, and transformation methods are used to reduce the number of objective functions, making the algorithm easier. The application of inverted index technology improves the efficiency of the algorithm. The simulation results prove that it is almost the optimal solution and can give us ideal predictions.

Reference

[1] G. Nemhauser, L. Wolsey, and M. Fisher. *An analysis of the approximations for maximizing submodular set functions*[J]. Mathematical rogramming,14, 1978.
[2] Jure Leskovec and reas Krause.*Cost-effective Outbreak Detection in Networks*[R]. MU-ML-07-111,2007
[3] A. Krause, C. Guestrin. *A Note on the Budgeted Maximization of Submodular Functions*. Technical Report, CMU-CALD-05-103, 2007.
[4] Glance, Natalie, Hurst, et al. *Deriving marketing intelligence from online discussion*[C]. Proceeding of the eleventh ACM SIGKDD international conference. ACM, 2005.
[5] Wang Jie, Gang Yijin, Li Fengguang, Wu Weiwei. *Research of improved greed algorithm used in blogsphere outbreak detection.* Computer Engineering & Applications, 2008. DOI：10.3901/JME.2008.11.304
[6] Miao Manxiang, Gang Yijin. *Algorithm Researching in Infectious Diseases Outbreak Detection*[J]. Research Journal of Applied Sciences, Engineering and Technology, 2013, 5(2):370-373. DOI：10.19026/rjaset.5.4960