Research Journal of Applied Sciences, Engineering and Technology 5(2): 370-373, 2013
ISSN: 2040-7459; E-ISSN: 2040-7467

Submitted: April 17, 2012 Accepted: May 14, 2012 Published: January 11, 2013

Algorithm Researching in Infectious Diseases Outbreak Detection

[1]Manxiang Miao and [2]Yijin Gang
[1]Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou, 450015, China
[2]Intelligent and Information Institute, Sippr, Zhengzhou, 450000, China

Abstract: Today's world, disease outbreaks influence seriously on people's normal life. But how we can find the infectious diseases source in social network at short time to avoid more people affected. This problem can be as outbreak detection which can be modeled as selecting people in a social network. This study uses a new methodology which improved from normal greed algorithm for detecting this problem in this and related problems, exhibiting the property of "sub modularity". This efficient algorithm scales to large problems, simulation results achieving near optimal solution.

Keywords: Greed algorithm, infectious diseases outbreak detection, social network, sub modularity

INTRODUCTION

When we explore the problem of detecting outbreaks in networks, we are given a network and a dynamic process spreading over this network and our goal is to select a set of nodes to detect the process as effectively as possible. In the process of disease spread, the people consider as a social network, we want to monitor few people to detect the disease as soon as possible, in order to avoid more people affected.

In the domain of disease spread, some people effected diseases and through contact and air to affected other people. We can observe the spread of information in the social net work through time stamped. In this case, we want to select a set of people to monitor who are most up to date to get more information, the simple way is to select the big affected areas, but these areas contain many people, so it's time consuming to monitor them. And the advisable method is to monitor small area but high quality information; this is our goal this study will present. There are several possible criteria one may want to optimize in outbreak detection. Such as minimize detection time, minimize number of people affected and so on. In algorithm, these criteria defined as objective functions. Optimizing these objective functions is NP-hard and this methodology our study present can get a nearly optimal solution in practice. 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.

(Each layer shows an information cascade, circles correspond to people and all people at 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)

ALGORITHM CAPTION

In infectious diseases outbreak, we want to select a subset A of people in a graph G = (v, ε), which detect outbreaks (spreading of a virus/information) quickly.

The links point to the destination of information and the cascades grow (information spreads) in the direction of the edges. Figure 2 presents an example of such a graph for social network. Each of the six areas consists of a set of posts. Connections between people represent "hyper-links" and labels show the time difference between the source and destination people (e.g., person P23 linked P31 two days demonstrate that P23 would be affected 2 days later than P31 was affected). Outbreaks (e.g., information cascades) initiate from a single node of the network (e.g., P11, P12 and P31) and spread over the graph, such that the traversal of every edge (s, t) ∈ε takes a certain amount of time (indicated by the edge labels).

As soon as the event reaches selected node, alarm is triggered. E.g., selecting area A6 would detect the

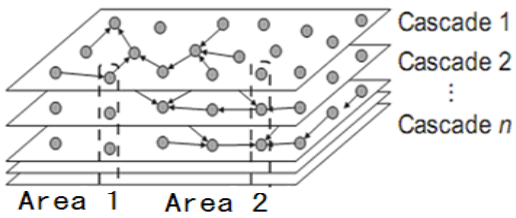


Fig. 1: Spread of disease between areas

Corresponding Author: Manxiang Miao, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou, 450015, China

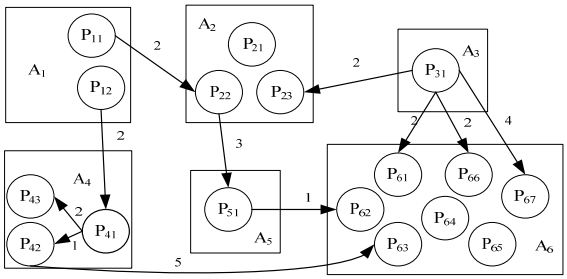


Fig. 2: Time stamped links between the posts in blog graph

cascades originating from person P11, P12 and P31, after 6, 8 and 2 days after the start of the respective cascades.

Depending on which nodes we select, we achieve a certain placement score. Figure 2 illustrates several criteria one may want to optimize. If we only want to detect as many stories as possible, then supervising just area A6 is best. (Supervising only area A6 captures all cascades, but late.) However, supervising A 1 would only miss one cascade (P31), but would detect the other cascades immediately. In general, this placement score is a set function R, mapping every placement A to a real number R (A) (our reward), which we intend to maximize.

We associate a nonnegative cost c (s) with every node (person) s, we also associate a cost c (A) with every placement A and require, that this cost does not exceed a specified budget B which we can spend. And define the cost of placement A:

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

So, our goal is to solve the optimization problem:

$$\begin{aligned} &MAX \quad R(A) \ \ (A\subset V)\\ &s.t \quad c(A)\le B \end{aligned} \tag{2}$$

where, B is a budget we can spend for selecting the posts

An event i ∈ Γ from set Γ of scenarios originates from a node s'∈ V of a network G = (v, ε), and spreads through the network, affecting other nodes through citations. Eventually, it reaches a monitored node s ∈ A ⊆ V (people we read) and gets detected. Depending on the time of detection t = T (i, s) and the impact on the network before the detection (e.g., the size of the cascades missed), we incur penalty πi(t). Note that the penalty function πi(t) depends on the scenario. Our goal is to minimize the expected.

This penalty over all possible scenarios i:

$$\pi(A)=\sum_{i} P(i)\pi(T(i,A)) \tag{3}$$

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

where, for a placement A ⊆ V, T (i, A) is the time until event i is detected by one of the sensors in A and P is a (given) probability distribution over the events. We assume πi(t) to be monotonically non decreasing in t. We also set T(I, Φ) = ∞ and set πi(∞) to some maximum penalty incurred for not detecting event i. So instead of minimizing the penalty π(A), we can consider the scenario specific penalty reduction Ri(A) = πi(∞) (T(i, A)) and the expected penalty reduction:

$$R(A)=\sum_{i} P(i)R_i(A)=\pi(\Phi)-\pi(A) \tag{5}$$

This alternative formulation has crucial properties (sub modular), which exhibits a diminishing returns property that is to say reading a blog when we have only read a few people provides more new information than reading it after we have read many people. The formulation expression of the properties as following:

$$R(B\cup\{S\})-R(B)\le R(A\cup\{S\})-R(A)\ \ (A\subseteq B\subseteq V) \tag{6}$$

Improved algorithm:

Greed algorithm: having the constant cost function (usually c (s) = 1) and iteratively in step k, adds the location sk which maximizes the marginal gain:

$$S_k=\underset{s\in v\setminus A_{k-1}}{\arg\max} R(A_{k-1}\cup\{s\})-R(A_{k-1}) \tag{7}$$

The algorithm stops, once it has selected B elements, the greedy algorithm is guaranteed to find a solution which achieves at least a constant fraction 63% of the optimal score (Nemhauser *et al*., 1978).

When cost function is non-constant:

$$S_k=\underset{s\in v\setminus A_{k-1}}{\arg\max} \frac{R(A_{k-1}\cup\{s\})-R(A_{k-1})}{c(s)} \tag{8}$$

The reference (Jure and Reas, 2007) have proved the algorithm perform arbitrarily worse than the optimal solution. We can add the following steps to normal

greed algorithm to get the new algorithm this study demonstrated.

- Use Eq. (7) to get the people set A1
- Use Eq. (8) to get the people set A2
- Max $\{R(A_1), R(A_2)\}$ (Krause and Guestrin, 2007)

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

Assume the marginal increments δs (A) = R (A∪{S}) - R(A(or δs (A)/c(s)) for all s ∈V\A. The key idea is to realize that, as our node selection A grows, the marginal increments can never increase. For A⊆B⊆V, it holds that δs (A) ≥δs (B). So instead of re-computing δs = δs (A) for every node after adding s' (and hence requiring |V| - |A| evaluations of R).

We perform lazy evaluations: Initially, we mark all δs as invalid. When finding the next location to a node, we go through the nodes in decreasing order of their δs. If the δs for the top node s is invalid, we re-compute it and insert it into the existing order of the δs (e.g., by using a priority queue). In many cases, the re-computation of δs will lead to a new value which is not much smaller and hence often, the top element will stay the top element even after re-computation In this case, we found a new node (person) to add, without having reevaluated δs for every person.

The inverted index is the main data structure we use in optimization algorithms. In the social networks, we need to consider several millions of persons, which make up the cascades. However, most outbreaks are sparse. Hence, most nodes s do not reduce the penalty incurred by an outbreak (i.e., Ri ({s}) = 0)). So we can get the R(A) without having to scan the entire data set.

ALGORITHM PSEUDOCODE AND SIMULATION RESULTS

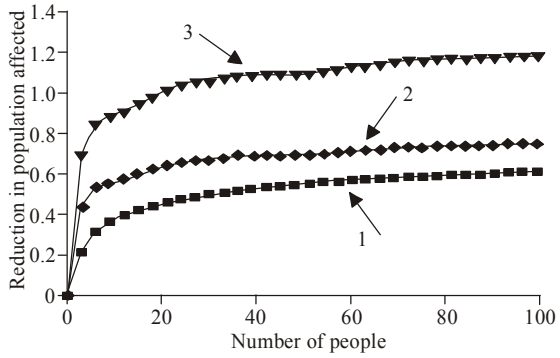
Using data set from reference (Glance *et al*., 2005) (3.5 million people, at least 3 in-links), extracting 100 datas for simulation and the MATLAB simulation result as Fig. 3.

In Fig. 3a, 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:

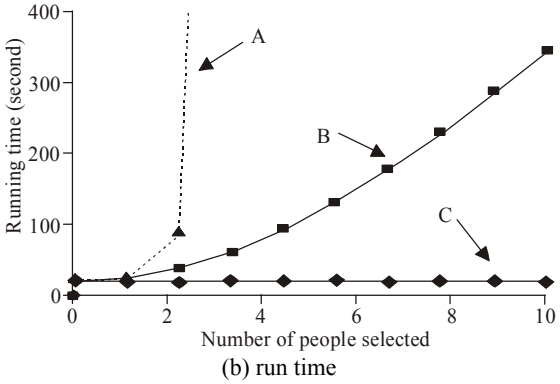
- That the first bound is much tighter than the traditional third bound.
- The proposed algorithm performs very close to the optimum (Huo, 2005).

Figure 3b plots the running time of selecting k people. (A represent exhaustive search, B represent the naive greed algorithm and C is the algorithm this study 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 Figure shows the performance of this algorithm. For example, for selecting 200 people, greedy algorithm runs 9.0 h, while this algorithm takes 50 sec (700 times faster). Algorithm pseudo code as following:

Function: F (g = (v, ε), R, c B)
A ← ∅; for each s ∈ v do $\delta_s \leftarrow +\infty$;
while $\exists s \in v\backslash A; c/(A \cup \{s\}) \leq B$ do
for each $\exists \delta \in v\backslash A$ do
flag s ← false;
while true do
if type = 1 then



(a) The performance of proposed algorithm



(b) run time

Fig. 3: MATLAB simulation result

$$s^* \leftarrow \underset{s \in v \setminus A, c(A \cup \{s\}) \le B}{\arg\max} \delta_s$$

if type = 2 then

$$s^* \leftarrow \underset{s \in v \setminus A, c(A \cup \{s\}) \le B}{\arg\max} \frac{\delta_s}{c(s)}$$

if flags then A ← A ∪ S*; break; else δ_s ←R(A ∪{S}) – R(A);
flags ← ture
return A;
then: A1 = F (g = (v, ε), R, c, B, 1;
A2 = F (g = (v, ε), R, c, B, 2);
return argmax {R(A1), R(A2)};

CONCLUSION

Infectious diseases outbreak detection is one of the important application in outbreak detection, the researching of diseases outbreak can give us an important indicator of some diseases, so we can do some really preventive steps to reduction loss. In this study, using improved greed algorithm to detect the infectious diseases outbreak. And give multi-objective function, transform way also be used to decrease the number of objective function to make the algorithm easier. The application of inverted index technology enhances the efficient of algorithm. And the simulation result proves to be nearly optimal solution and can give us an ideally prediction.

ACKNOWLEDGMENT

This study is supported by the science research project fund of Henan province (0624220069).

REFERENCES

Glance, N.S., M. Hurst, K. Nigam, M. Siegler, R. Stockton and T. Tomokiyo, 2005. Deriving marketing intelligence from online discussion. Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05, NY, pp: 419-428.

Huo, H., 2005. Algorithm Design and Analysis. Xidian University Press, Xi'an.

Jure, L. and K. Reas, 2007. Cost-effective Outbreak Detection in Networks. MU-ML-07-111.

Krause, A. and C. Guestrin, 2007. A note on the budgeted maximization of submodular functions. Technical Report, CMU-CALD-05-103.

Nemhauser, G., L. Wolsey and M. Fisher, 1978. An Analysis of the Approximations for Maximizing Submodular Set Functions. Mathematical Rogramming, pp: 14.