Classifying Latent Infection States in Complex Networks

Yeon-sup Lim
University of Massachusetts Amherst
ylim@cs.umass.edu

Bruno Ribeiro
Carnegie Mellon University
ribeiro@cs.cmu.edu

Don Towsley
University of Massachusetts Amherst
towsley@cs.umass.edu

ABSTRACT

In this work, we develop techniques to identify the latent infected nodes in the presence of missing infection time-and-state data. Based on the likely epidemic paths predicted by the simple susceptible-infected epidemic model, we propose a measure (*Infection Betweenness Centrality*) for uncovering unknown infection states. Our experimental results using machine learning algorithms show that *Infection Betweenness Centrality* is the most effective feature for identifying latent infected nodes.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Network problems, Graph algorithms

Keywords

Epidemics, Information cascades, Information diffusion

1. INTRODUCTION

Networks are underlying mediums for the spread of epidemics such as diseases, rumors, and computer viruses. Determining the infection state of network nodes is the first step to taking corrective or preventive action to stop or slow the spread of an epidemic. Unfortunately, the infection state of network nodes is often unknown; for example: in the spread of computer malware (say, a contaminated email attachment) over a large organization, IT specialist will likely only inspect the computers of users that open trouble tickets; a similar problem occurs with the spread of rumors over online social networks. Hence, the problem of effectively identifying the infection state of unobserved nodes given a set of observed nodes is of central importance in the study of infection cascades.

Our research question is: *Given a set of nodes with known infection states and the network topology can we correctly uncover the unknown infection state of the remaining nodes?* In this work, we consider a network where an epidemic starts from a single source. Each node appears in one of two states:(i) susceptible, capable of being infected, (ii) infected, able to spread the epidemic further. We also assume that the infection state of a subset of nodes is known and the full network structure (adjacency matrix) is available.

Let $G(V, E)$ be an undirected graph where V is a set of nodes and $E \subseteq V^2$ is a set of edges. Suppose that an epidemic starts at a single node (denoted "source") and propagates to neighbors in $G(V, E)$. Let $X_i \in \{0, 1\}$ denote the state of node $i \in V$ where $X_i = 0$ means node i is susceptible and $X_i = 1$ that it is infected. Assume that an infected node contaminates neighbors at rate λ. Then,

$$X_i \quad : \quad 0 \to 1 \quad \text{at rate } \lambda \sum_{j \in n(i)} X_j,$$

where $n(i)$ is the set of neighborhood of i.

Assume that there are l nodes with observed infection state $L = \{(1, X_1), ..., (l, X_l))\}$. There are also $u = |V| - l$ nodes with unknown infection state, $U = \{x_{l+1} ..., x_{l+u}\}$; l is typically much smaller than u. Given the set of observed nodes L and the adjacency matrix $\mathbf{A}$ of the network, our goal is to correctly assign an infection state X_i to node $i = l + 1, \ldots, l + u$.

2. MEASURING INFECTION STATE

2.1 Propagation Properties

Under the assumption that an epidemic propagates from a single source to neighboring nodes following the Susceptible-Infected (SI) model [5], we identify the following properties.

Let S_o denote the set of observed susceptible nodes and I_o the set of observed infected nodes.

Property 1: If removing all nodes in S_o from the network disconnects the network, then one of the disconnected components contains all of the infected nodes

Property 2: Let $S \in V$ be a cut set that divides I_o into multiple components, then at least one node in S is infected

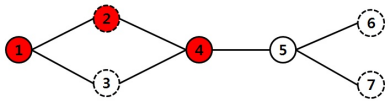


Figure 1: Red nodes and white nodes represent infected and susceptible nodes, respectively. Dotted circles (nodes 2, 3, 6, and 7) show nodes with unknown infection state and full circles (nodes 1, 4, and 5) show nodes with known infection state.

Consider the topology shown in Figure 1. Removal of node 5, which is observed and susceptible divides the graph into two components, $\{1, 2, 3, 4\}$ and $\{6, 7\}$. Only the component $\{1, 2, 3, 4\}$ contains infected nodes (Property 1). Since there is no propagation path from infected nodes without node 5, we can determine that nodes 6 and 7 are susceptible (deterministic susceptible nodes). Observed infected nodes $\{1, 4\}$ divide into two components by removing nodes 2 and 3, which are not observed. Because the removal of nodes 2 and 3 places infected nodes 1 and 4 in distinct components,

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2579361 .

Table 1: Topologies

Topology	Type	n	m	c	σ	s	d [2]	Description
YEAST	Biological	1870	2277	0.0672	3.1374	6.5044	19	Yeast Protein Interaction Network [2]
GRQC	Collaboration	5242	28980	0.5296	7.9179	3.8317	17	Collaboration networks from ArXiv General Relativity and Quantum Cosmology [4]
HEPTH	Collaboration	9877	51971	0.4714	6.1864	3.0213	18	Collaboration networks from ArXiv High Energy Physics [4]
POWER	Device	4941	6594	0.0801	1.7913	2.1898	46	Topology of the Western States Power Grid of the United States [8]
OREGON	Device	11174	23409	0.2964	33.0948	46.4017	10	Topology of Autonomous Systems (AS) peering information inferred from Oregon route-views between March 31 2001 and May 26 2001 [3]

[1] n, m, c, σ, s, and d are the number of nodes, the number of edges, clustering coefficient, standard deviation of degree distribution, skewness of degree distribution [9], and diameter of network, respectively
[2] d is calculated with the largest connected component if a network has multiple connected components

node 2 and/or 3 must be infected (Property 2). Using Property 1, we can reduce the number of nodes whose state is unknown by ignoring nodes in components that can be isolated by healthy nodes. In the rest of this paper, we focus on the reduced graph in which observed and deterministic susceptible nodes are excluded from the original graph. Even though Property 2 does not provide a direct way for determining the unknown infection state, it points to the importance of a particular node in possibly infecting known infected nodes.

2.2 Infection Betweenness Centrality

Let G' be a subgraph constructed by removing all nodes that must be healthy according to Property 1. The number of paths of length $r \geq 0$ between a pair of nodes (i, j) in G', N_{ij}, is

$$N_{ij}^{(r)} = (\mathbf{A}^r)_{ij},$$

where $\mathbf{A}$ is the adjacency matrix of G'.

Suppose that each path of length r is given a weight $\alpha > 0$; then

$$N_{ij} = \sum_{r=0}^{\infty} \alpha N_{ij}^{(r)} = \sum_{r=0}^{\infty} (\alpha^r \mathbf{A}^r)_{ij}.$$

is the weighted sum of paths from i to j. We can write this expression in matrix notation

$$\mathbf{N} = \sum_{r=0}^{\infty} \alpha^r \mathbf{A}^r = (\mathbf{I} - \alpha\mathbf{A})^{-1}.$$

Let $N_u(i, j)$ denote the weighted sum of paths from node i to j through node u. Given $G'' = G' - \{u\}$, we can calculate $N_u(i, j)$ by subtracting the weighted sum of paths from i to j in G'' from the sum in G'; however, constructing G'' and performing the inverse operation for $\mathbf{N}$ of each G'' requires additional computation. Therefore, we resort to simple approximation $N_u(i, j) \approx \mathbf{N}_{iu} \times \mathbf{N}_{uj}$. Summing over all possible nodes $u \in V$ yields

$$\mathbf{M}_{ij} = \sum_{u \in V} N_u(i, j) = \sum_{u \in V} \mathbf{N}_{iu}\mathbf{N}_{uj} = (\mathbf{N}^2)_{ij}.$$

We define the Infection Betweenness of node u between two infected nodes i and j as:

$$B_u(i, j) = \frac{N_u(i, j)}{\mathbf{M}_{ij}},$$

which is the fraction of the weighted sum of path from i to j through u over the total weighted sum of paths from i to j; thus, node u is more likely to be infected by node i or j as $B_u(i, j)$ increases. Assuming that $B_u(i, j)$ is the probability that node u is contaminated by node i or j, we define *Infection Betweenness Centrality* of node u given the set of observed infected nodes, which is the measure that a node u is infected, as:

$$P(u) = 1 - \prod_{i,j \in I_o,\ i \neq j} (1 - B_u(i, j)), \quad (1)$$

where I_o is the set of observed infected nodes.

3. RESULTS

To test our approach, we use datasets from several real world networks. We classify the datasets into three categories - biological, collaboration, and device networks: YEAST (biological), GRQC, HEPTH (collaboration), POWER and OREGON (device) as described in Table 1. We run batches of simulations for each network while varying the fraction of observed nodes from 5% to 25%. In each run, we simulate an SI cascade starting at a randomly selected seed node with infection rate $\lambda = 0.5$ until 10% of nodes are infected. The parameter weight α of *Infection Betweenness Centrality* is set to 0.01 to guarantee to be less than the reciprocal of largest eigenvalue of adjacency matrix of the reduced graph (the condition that α must satisfy for the sum $\mathbf{N}$ in the equation of infection betweenness to converge). If a network has multiple connected components as does YEAST, we assume that an epidemic starts at a node in the largest connected component. In order to evaluate accuracy, we use three metrics: precision, recall, and F-measure [11].

3.1 Incorporating Infection Betweenness Centrality into Machine Learning Algorithms

Now, we introduce a classification method using *Infection Betweenness Centrality* and other node features based on machine learning (ML) algorithms such as Naive Bayes (**NB**), Naive Bayes with kernel density estimation (**NBK**), and C4.5 Decision Tree (**C4.5**). To apply these ML algorithms to experiments, we use the WEKA machine learning software suite [11].

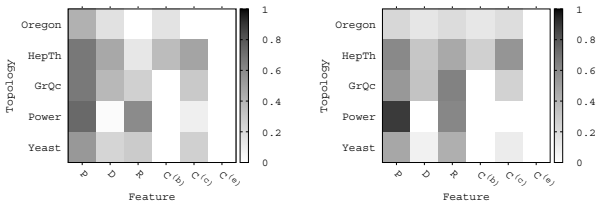
3.1.1 Node features

We consider six node characteristics that are available using information regarding network topology and the observed nodes, as features for building ML-based classifiers. The first five features are: degree normalized by the maximum degree in the network D, observed infected neigh-

bor ratio R, betweenness centrality $C^{(b)}$, closeness centrality $C^{(c)}$, and eigenvector centrality $C^{(e)}$. We also include *Infection Betweenness Centrality* P as a feature, defined as the measure that a node is infected shown in Eq. (1).

3.1.2 Predictive Features

To examine which features provide meaningful information for identifying latent infected nodes, we investigate the performance of ML-based classifiers with each feature when we create cascades that infect approximately 10% of the nodes in the network and then reveal the infection state of 15% of the nodes (randomly selected). Figure 2 shows the average F-measure of **NB** and **C4.5** with each feature for all the networks. The best feature will have an F-measure close to one (darker squares). We observe that the *Infection Betweenness Centrality* produces the darkest column showing it to be the best predictive feature in both **NB** and **C4.5** algorithms over nearly all networks. In the case of **C4.5**, R yields similar performance to P. We also see that D and $C^{(c)}$ are also meaningful features in several networks although not as good as P. However, except for P, the effectiveness of other features differs significantly depending on the network and the ML algorithm.



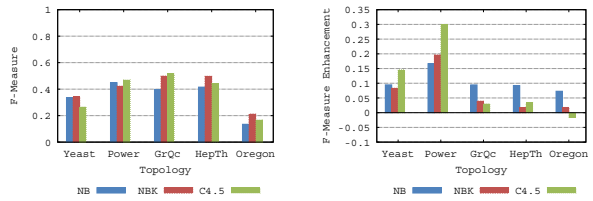
(a) NB (b) C4.5

Figure 2: Predictive power of each feature.

3.2 ML-based Infection State Prediction

3.2.1 Effect of Infection Betweenness Centrality

Figure 3(a) shows the F-measure of each classifier using all features except for P. In all the considered networks, the classifiers yield F-measure of less than 0.5. We observe that the best classifier differs according to the network, but there is no significant difference between the classifiers for each network. Note the significant low performance of the classifiers in Oregon: the F-measure of even the best classifier, **NBK**, is around 0.2. This is because the predictive power of each feature is quite weaker in Oregon than in the other networks as shown in Figure 2. Next, we compare the ML-based classifiers using all of the features to those excluding P in order to check whether P can improve the performance of the classifiers.



(a) F-measure when using all features except for P (b) F-measure Enhancement by including P

Figure 3: Performance of ML algorithms

Figure 3(b) shows the F-measure of each classifier using all six features minus the F-measure of the same classifiers using five features (which excludes P). All classifiers using all features see performance improvements in all the networks except for **C4.5** in Oregon compared to excluding P. This shows that we can improve the performance of a particular classifier by combining the infection betweenness centrality P with the other node features. In particular, the classifiers with all features including P yield comparatively large performance enhancement of the classifiers in Yeast and Power, e.g., using all features increases the F-measure of **C4.5** applied to in Yeast and Power by around 0.15 and 0.3, respectively. In the case of **NB**, adding P enhances performance by almost the same amount (around 0.3) regardless of the network. This is because the predictive power of P for **NB** does not significantly differs across the networks as shown in Figure 2(a). Note that **C4.5** with all features exhibits the F-measure enhancement except for Oregon. Even in Oregon, the performance degradation of **C4.5** by using all features is not noteworthy. We observe then that for **C4.5**, *Infection Betweenness Centrality* is by far the most important feature as adding P to the feature set in most cases increases classification accuracy.

3.2.2 Prediction v.s. fraction of observed nodes

Figure 4 compares the average precision and recall of each classifier according to the fraction of observed nodes. Again, the epidemic infects 10% of nodes. Here, we also compare our classifiers against random-guessing (Random), which tosses a biased coin and with probability 0.1 (0.1 is the fraction of infected nodes) declares the node to be infected. As shown in Figure 4, our classifiers outperform random-guessing both in precision and recall. Also, the precision and recall of our classifiers increases with the fraction of observed nodes; as expected, increasing the fraction of observed nodes provides more information about the infection state of the unobserved nodes. In a closer look **C4.5** exhibits the best precision over all classifiers on almost of all the networks: the only exception is Power, where **NBK** yields slightly better precision performance than **C4.5**. Comparing the precisions of each network, we observe that our classifiers show the best precision in Power followed by GrQc, HepTh, Yeast, and Oregon; Power is almost planar, likely making the classification task easier. In the next section, we also explore how network characteristics affect the performance of our classifiers.

Figure 4 shows that **NBK** yields the best recall performance over all the networks except Power. Note that the precision of **NBK** is lower than that of **C4.5** except for Power. It means that **NBK** is more likely to classify unknown node states to infected, resulting in the higher recall, but those classifications are not as accurate as **C4.5**. All classifiers yield better recall performance when applied to Power than the other networks. Also, Oregon remains the most difficult network within which to correctly find the infected nodes. Even though all classifiers yield relatively high precisions (greater than 0.5) in Oregon, their recall performance in Oregon is less than 0.2, which is similar to that of random-guessing. That is, in Oregon, our classifiers make correct decisions when they classify unknown states to infected, but many infected nodes are classified as healthy. In future work, we will explore a method to improve the recall performance of these classifiers.

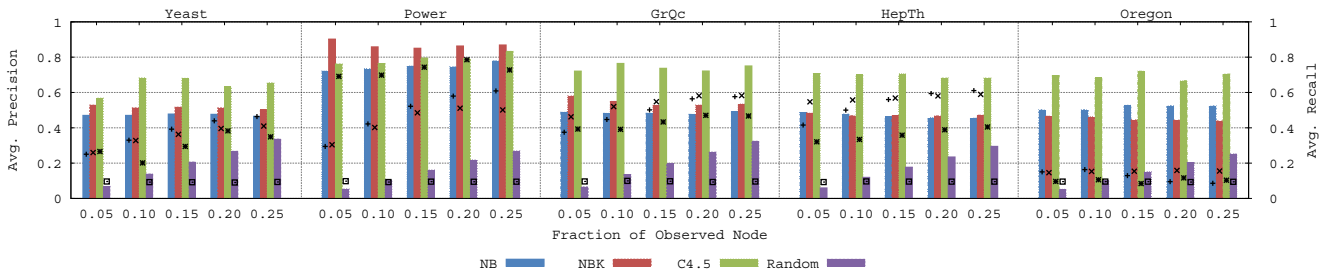


Figure 4: Accuracy for varying fraction of nodes with observed state (Bar: Precision, Dot: Recall)

3.2.3 *Impact of Network Characteristics*

We now investigate the impact of network characteristics on the performance of our classifiers (using all six features). To this end, we investigate the correlation coefficient between the F-measure performance ranks and ranks of network characteristics for each network; for instance, **NBK** yields the worst performance in OREGON (the fifth rank in terms of **NBK**'s F-measure performance among the five networks) and OREGON has the largest degree skewness (the first rank in terms of degree skewness). Table 2 presents the Pearson's correlation coefficient [10] between the ranks of network characteristics and F-measure performance.

Table 2: Correlation Coefficient between Ranks according to F-measure and Network Characteristics

Characteristic	Correlation	
	NB	**NBK & C4.5**
Clustering Coefficient	0.1	0.2
Standard Deviation of Degree	-0.7	-0.6
Degree Skewness	-1.0	-0.9

As shown in Table 2, the performance of the classifiers is strongly negatively correlated with degree skewness and the degree standard deviation. As the degree skewness and the degree standard deviation decrease, the classifiers become more accurate. Interestingly, there is a little correlation between clustering coefficient and classification performance even though an epidemic is more likely to propagate to nodes in a same cluster. A validation with extensive experiment using more networks is a part of our future work.

4. RELATED WORK

Shah and Zaman [7] studied the problem of finding the source of a computer virus in a network. They focused on how to find the source among the set of infected nodes that are observed, which is different from our goal. Based on their metric called *rumor centrality*, they constructed a machine-learning estimator that finds the source exactly or within a few hops in networks. They also analyzed the asymptotic behavior of their virus source estimator for regular trees and geometric trees. Sadikov et al. [6] present an estimation method of network properties, such as the number of weakly connected components, given a sampled network. By formulating a simple k-tree model and approximating it to the original network, their method can estimate the properties of original networks; they showed that their method can accurately estimate properties of the original network even when 90% of nodes are not sampled. Closely related to our work is that of Gomez et al. [1], who develop an algorithm for inferring the network over which a diffusion propagates. Given the observed times when nodes become infected, they determine paths through which the diffusion most likely took, i.e., a directed graph where a contagion passed through. In contrast, our work tries to identify the infection state of each unobserved node given a limited number of nodes with known infection state and no infection timestamps.

5. CONCLUSION

In this paper, we studied how to identify the infected nodes without individually inspecting all nodes in the network. Based on the well known SI model, we defined the *Infection Betweenness Centrality* for identifying the latent infection status of nodes. Our empirical results show that the machine learning classifiers using the *Infection Betweenness Centrality* along with other network-wide features outperform random-guessing and the same classifiers without it. We also analyzed the impact of the amount of missing data as well as the impact of network characteristics on the effectiveness of the algorithms.

6. ACKNOWLEDGEMENT

This work was supported by the NSF grant CNS-1065133, ARL Cooperative Agreement W911NF-09-2-0053, and ARO under MURI W911NF-08-1-0233.

7. REFERENCES

[1] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. ACM KDD '10.
[2] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, May 2001.
[3] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. ACM KDD '05.
[4] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1, March 2007.
[5] M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.
[6] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. ACM WSDM'11.
[7] D. Shah and T. Zaman. Detecting sources of computer viruses in networks: theory and experiment. ACM SIGMETRICS'10.
[8] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.
[9] Wikipedia. Skewness — Wikipedia, the free encyclopedia. [Online; accessed 13-Jan-2014].
[10] Wikipedia. Spearman's rank correlation coefficient — Wikipedia, the free encyclopedia. [Online; accessed 13-Jan-2014].
[11] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, June 2005.