Security on Networks

Bonny Jain, John Wang, Iris Xu

May 14, 2013

1 Simulation Results

In this section, we provide results from computational simulations of our model. The graphs that we simulate on are created using the Erdos-Renyi mechanism with the number of nodes fixed at $n = 100$. Fixing the number of nodes reduces the number of parameters that can vary.

1.1 Simulating the Non-Dynamic Model

To begin, we present results from simulating the non-dynamic model. In this case, each individual chooses his protection $q \in [0, 1)$ before the game begins and is not allowed to change his choice as the game proceeds. The individual is not given any idea of what his position in the graph will be, but he does know the probability of edge creation p.

Solving for a mixed strategy Nash Equilibrium in this context is $PPAD$ complete.[1] This implies that for all practical purposes, solving for a Nash Equilibrium is computationally intractable, although little is known about the actual complexity of $PPAD$ in relation to P and NP.[2]

Because we cannot compute the equilibrium vector of protections $\bar{q}$ where $q_i \in [0, 1)$ is player i's choice of protection, we can sweep along a grid of representative values for $\bar{q}$ and examine the game at each one of these values.

Fortunately, this game is symmetric for each player and Acemoglu et al (2013) have shown that under such situations, the Nash Equilibrium for each player is to play the same protection q. Therefore it suffices to check setting $q_i = q_j$ for each player $i \neq j$ in our simulations. We shall call $q = q_i$ the protection that each player chooses.

To examine the outcome of a contagion spreading on an Erdos Renyi graph with edge probability p, $n = 100$, and protection rate q for each participant, we must be able to compute $\tilde{P}_i(A, q_{-i}, \Phi)$, the probability that the infection reaches i (i.e. the probability that any one of i's neighbors gets infected). However, since computing $\tilde{P}_i$ is NP-hard, we instead use Monte Carlo simulations to approximate $\tilde{P}_i$. We do so in the following manner:

- Generate κ Erdos Renyi graphs with parameters $G(n, p)$.
- For each graph instance and protection value q, we run the following for γ trials: choose a random node to infect and run the infection to completion.
- Compute $\hat{P}$, an estimate of $\tilde{P}_i$ for all i, as the total number of infected nodes after γ trials divided by the total number of nodes. For $n = 100$, if we let I_j be the number of infected nodes on the jth trial, we have:

$$\hat{P} = \frac{1}{n\gamma} \sum_{j=1}^{\gamma} I_j \tag{1}$$

[1] http://www.cs.berkeley.edu/ christos/papers/cacmDGP-2.pdf
[2] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4031362

Figure 1: Average Infection Rates for $np = 0.5 < 1$

Figure 2: Average Infection Rates for $np = 5 > 1$

- Compute the standard deviation of $\hat{P}$ by using standard techniques of computing standard deviations for a sample of random variables from a binomial distribution.

1.2 Simulation Results

We use the above model to simulate results, with $n = 100$, $\gamma = 250$, and $\kappa = 100$. In order words, we generate $\kappa = 100$ random graphs for each set of Erdos Renyi parameters, then on each graph, we run $\gamma = 250$ distinct infections which each start at a random node. The results are then summarized for different protection rates of q.

We begin by presenting the average infection probability $\hat{P}$ when the edge creation probability in Erdos Renyi is very small (we choose $p = 0.005$). Here the Erdos Renyi parameter is $np = 0.5$, which is below the threshold value for the existence of a giant component. Under this regime, the average infection probability $\hat{P}$ is very low for all protection levels (even a protection rate of $q = 0.05$).

The intuition behind this is clear. Since $np = 0.5 < 1$, the average size of a component is small, so the amount that the infection can spread is decreased. Thus, even for low protection rates, the infection cannot spread very far.

When the Erdos Renyi parameter predicts the presence of a giant component, the average infection rate has a strong dependence on the protection rate. We use $np = 5 > 1$ which should produce a giant component with high probability, and simulate the resulting infection probabilities for various values of q. For low q, the average infection rate is incredibly high. For example, when $q = 0.05$, the average infection rate is $\hat{P} = 0.93$ with a standard deviation of 0.0103.

However, as the protection rate increases, the average infection probability decreases almost linearly until about $q = 0.6$. Once $q = 0.6$, the positive externalities start to kick in and the average infection probability drops off more dramatically than the linear trend would predict. This can be attributed to the "network effect", or the extra benefit that everyone obtains from one person increasing their protection. Having higher q will positively impact your neighbors because of the lower probability of you getting infected and thus transmitting to your neighbors.

However, you must have enough connections in order for this network effect to become substantial. This is why the network effect does not manifest itself when there is no giant component. The figure below shows the network effect for different levels of q, where network effect ν is defined as follows:

$$\nu = 1 - q - \hat{P} \tag{2}$$

The network effect starts to increase to something significant by $q = 0.5$, then falls off again after $q = 0.8$. The fall off in the network effect comes from the saturation of protection levels. After a certain point, the marginal benefit of extra protection starts to decrease because of the difficulty of actually spreading the infection.

1.3 Computational Intractibility of $\tilde{P}_i$

Unfortunately, computing $\tilde{P}_i(A, q_{-i}, \Phi)$ in general for a graph which is not a tree is in $\#P$, which is as least as hard as NP. This makes computing $\tilde{P}$ computationally intractable.

The reduction can be seen by introducing a known $\#P$ problem called the Two Terminal Problem.

Figure 3: Network Effect in a Giant Component ($np = 5$)

Definition 1. *Let us define a graph $G = (V, E)$ with some probability labelling $Pr : E \to [0, 1] \cap \mathbb{Q}$. Let $u \in V$ be a source terminal and $v \in V \setminus \{u\}$ be a target terminal. An instance of the* Two Terminal Problem *is to compute $Pr(v|u)$, the probability that there exists a path to node v from node u.*

Note that in the Two Terminal Problem, each edge $e \in E$ has some probability of failure $p_e \in [0, 1]$ as defined by the function Pr.

Showing the difficulty of the computation of $\tilde{P}_i$ in the non-dynamic model is now as simple as showing a reduction of the Two Terminal Problem, since it is $\#P$ complete as shown by Ball (1980).[3]

Theorem 1. *Computing $\tilde{P}_i(A, q_{-i}, \Phi)$ for all $i \in \{1, \dots, n\}$ on an Erdos Renyi graph with edge probability p is $\#P$ hard in the non-dynamic model.*

Proof. We shall show that every instance of the Two Terminal Problem is an instance of the Epidemic Probability Problem (what we shall call the problem of computing $\tilde{P}_i$ in our model).

Suppose we have an instance π_{ttp} of the Two Terminal Problem. We can find a solution as follows:

- Create an instance π_{epp} of the Epidemic Probability Problem using graph G from π_{ttp}.
- Set the protection rates $q_e \in [0, 1)$ for each edge $e \in E$ for the Epidemic Probability Problem as $1 - Pr(e)$, where $Pr(e)$ is the loss probability for edge e in the Two Terminal Problem.
- Solve the Epidemic Probability Problem. The probability $\tilde{P}_v$ for a given source of infection u will be the solution to the Two Terminal Problem.

Thus, we have shown that the Two Terminal Terminal problem can be reduced in polynomial time to the Epidemic Probability Problem, which shows that the Epidemic Probability Problem is at least as hard as the Two Terminal Problem (which is $\#P$ hard). $\square$

The complexity result can be extended to the dynamic case when the protection for each node potentially changes. This means that for both the non-dynamic and the dynamic model, the computation of $\tilde{P}_i$ is computationally intractable.

Corollary 1. *Computing $\tilde{P}_i(A, q_{-i}, \Phi)$ for all $i \in \{1, \dots, n\}$ on an Erdos Renyi graph with edge probability p is $\#P$ hard is the dynamic model with changes in protection in each time step.*

Proof. We can reduce the Two Terminal Problem to the Dynamic Epidemic Probability Problem, where the probabilities are fixed to represent Pr from the Two Terminal Problem. Using the same derivation as before, we can show that the dynamic version of the Epidemic Probability Problem is $\#P$ hard. $\square$

1.4 Computationally Intractable Problems

There are a number of related problems which are computationally hard. Actually, most interesting quantities related to the epidemic problem are NP-hard to compute. For example, computing the expected number of infected nodes is NP-hard.

Theorem 2. *Computing the expected number of infected nodes on the an epidemic network with protections q_i for $i \in \{1, \dots, n\}$ is $\#P$ hard.*

[3]http://onlinelibrary.wiley.com/doi/10.1002/net.3230100206/pdf

Proof. We shall reduce the Epidemic Probability Problem to the Epidemic Expected Infection Problem (the problem of computing the expected number of infected nodes in an epidemic network).

Suppose we have an instance of the Epidemic Probability Problem. Now select a node $u \in V$ and create a new node $v \notin V$ which will be attached to the graph. Now, create an edge $e(u, v)$ between u and v and set v's protection to $q = 0$. Now, if u is ever infected, then v will also be infected.

It is clear that the expected number of infected nodes in this new graph increases by $P_i(u)$, the probability of u becoming infected. Thus, finding the expected number of infected nodes is at least as hard as finding the probability of infection of a node. Thus, we see that the Epidemic Expected Infection Problem is harder than the Epidemic Probability Problem, which is in $\#P$. $\square$