Models and Algorithms for Network Immunization

George Giakkoupis
University of Toronto

Aristides Gionis, Evimaria Terzi and Panayiotis Tsaparas
University of Helsinki

Abstract

Recently, there has been significant research activity in the algorithmic analysis of complex networks, such as social networks, or information networks. A problem of great practical importance is that of network immunization against virus spread. Given a network, a virus-propagation model, and an immunization cost function, we are interested in containing the spread of the virus while minimizing the immunization cost. In this paper, we consider two virus-propagation models and we propose immunization algorithms for each model. The experimental evaluation shows that our algorithms perform well on both synthetic and real graphs. Furthermore, it reveals the following interesting facts (a) the simple heuristic of immunizing the nodes with the highest degree is not optimal, and (b) our algorithms perform significantly better in small-world networks.

1 Introduction

It is often the case that natural or man-made systems are organized in networks. Examples include the Internet, the Web, social networks, or networks of proteins. The normal operation of such networks is threatened by the diffusion of harmful information that is propagated through the links. Examples include cascading failures in an electrical grid, sexually transmitted diseases in a sexual network, computer viruses on the Internet, harmful gossip or panic in a social network. We will collectively refer to the harmful information that is propagated as a *virus*, and we will refer to the process of impeding the spread of the virus as the immunization of the network. This is a problem of obvious practical importance. We want to prevent disease spreads, protect computer networks from viruses, and control the leakage of sensitive information and unpleasant gossip. At the same time our resources (vaccination, anti-virus software, influence) are costly and limited, so we are interested in achieving the best possible effect, while allocating the minimum possible resources.

The problem of immunization is defined informally as follows. Given a network, and a virus-propagation model, assume that an adversary places a number of viruses in the network. We are interested in immunizing the minimum set of nodes, such that in the resulting network the spread of the virus is contained. Alternatively, given a fixed budget, we are interested in containing the spread of the virus as much as possible without exceeding our budget.

The immunization process depends obviously on the virus propagation model. Models for diffusion of information in a network have been studied extensively in various disciplines, including computer science, sociology, physics, and epidemiology. In this paper, we consider two different types of models: the *independent-cascade model*, considered by Kempe et al. [10] for the propagation of gossip, and *dynamic-propagation models* similar well known SIS model [14] for epidemic spread. Each of these models, defines a different objective for the immunization algorithm, and thus requires different immunization strategies.

Our contributions can be summarized as follows.

- We define a general framework that allows us to formally define the immunization problem.
- We consider two different virus propagation models, and we propose immunization algorithms for these models.
- We study the algorithms experimentally on both real and synthetic networks. We observe that contrary to popular belief immunizing the node with the highest degree does not yield the best results. The benefits of our algorithms are especially pronounced in graphs with high clustering coefficient, a case that is encountered in many real-life networks.

The rest of the paper is structured as follows. In Section 2 we review some related work. In Section 3 we define the general framework for the problem of network immunization. In Section 4 we we discuss in detail the virus-propagation models we consider, and we define immunization problems for these models. Our immunization algorithms are presented in Section 5, and our experiments in Section 6. Section 7 is a short conclusion.

2 Related Work

The study of mathematical models for epidemic spread has a long history in biological epidemiology, as well as in the study of computer viruses [11]. The pioneering work of Kermack and McKendrick [12] establishes the first stochastic theory for epidemic spread and proves the existence of an *epidemic threshold*, which determines whether the epidemic will spread, or die out. A large amount of recent work in mathematical epidemiology focuses on providing analytic expressions for epidemic thresholds for different propagation models and different families of networks [3–6, 14].

In homogeneous networks, immunizing random nodes in the network is an effective mechanism for preventing the epidemic spread [1, 14]. However, the method of uniform immunization breaks down for scale-free networks due to the existence of highly connected nodes. For such networks it can be shown that there is epidemic threshold [14]. However, in these cases immunizing highly connected nodes appears to be highly effective [8,14]. In the case that the topology of underlying network is unknown Cohen et al. [7] show that immunizing random acquaintances of random nodes is more effective than immunizing random nodes. Finally, Aspnes et al. [2], assume that nodes in the graph act selfishly and they study inoculation strategies from a game-theoretic point of view. They also consider centralized versions of the problem, and they introduce the sum-of-squares partition problem, for which they obtain a polynomial-time $O(\log^2 n)$-approximation algorithm.

For a more complete review on virus propagation models and immunization algorithms we refer the reader to [13, 14].

3 The general framework

The immunization problem has the following components.

- The network, over which the virus propagates. This is modeled as a graph $G = (V, E)$. We will consider only *undirected* graphs, although most of our results apply for directed graphs as well. It is possible to assume that the graphs are drawn from a specific family. In the definitions and algorithms we consider all possible graphs. In our experiments we investigate various popular families of graphs.

- The virus propagation model, that determines how the virus spreads in the network.

- The immunization algorithm, which has the power to immunize a set of nodes in the network in order to minimize the spread of the virus. An immunized node cannot receive or transfer the virus. Conceptually, for the purpose of the virus propagation we can think of the immunized nodes as being removed from the network. The cost of the immunization algorithm is the number of nodes that are immunized.

- The adversary, who has knowledge of the virus propagation model, and she plants r copies of the virus in the network so as to maximize the spread of the virus. We will use A_r to denote such an adversary. The adversary may also have knowledge of the choices made by the immunization algorithm. We call such an adversary an *adaptive* adversary. We also consider a *randomized* adversary who places virus copies uniformly at random.

4 Virus Propagation and Epidemic Spread

Modeling virus propagation is a problem with long history. The most popular models are the SIR (Susceptible-Infected-Removed) model, and SIS (Susceptible-Infected-Removed) model. In the SIR model, a node may be in any of the following three states: Susceptible, in the case it does not have the virus, but it can become infected if exposed to it; Infected, in the case that it has the virus and can pass it on; Removed (or Recovered), in the case that it used to have the virus, but it recovered (or died), and now it is permanently immunized and it no longer participates in the virus propagation process. In the SIS model, we assume that a node may be cured from the virus, but it is not immunized, and thus it can become infected again. Therefore, a node alternates between the susceptible and immunized states.

The two virus propagation models we consider are special cases of these two models. We now describe them in detail.

4.1 The Independent Cascade Model

The first model is a discrete-time special case of the SIR model. At time $t = 0$ the adversary plants r viruses to some nodes of the graph. Then, if a node i becomes infected for first time at time t it is given a single chance to infect each of its neighbors j that is currently uninfected. The probability that node i succeeds in infecting node j is p_{ij}. If node i succeeds in infecting j, then j becomes infected at time $t + 1$; otherwise node i never attempts to infect node j again in the future (eventhough j might eventually get infected by some of its other neighbors). The virus-propagation process continues until no more infections are possible — clearly the process stops after at most n steps.

This model is a special case of the SIR model, where we assume that time proceeds in discrete time steps and we require that nodes stay infected for exactly one time step. We refer to it as independent cascade model, following the terminology of Kempe et al. [10].

Now, let G be a graph of size n, and let N_r be a subset of r nodes of the graph where r copies of the virus are placed. Assume now that the propagation process is completed. Let $S(N_r, G)$ denote the expected number of infected nodes in G. The expectation is taken over all random choices made by the propagation model. Also, let

$$S_r(G) = \max_{N_r} S(N_r, G),$$

denote the maximum expected number of infected nodes, where the maximum is taken over all possible initial virus placements. The subset $A_r = \arg\max_{N_r} S(N_r, G)$ corresponds to choices of the adaptive adversary. We call $S_r(G)$ the *epidemic spread* in G. We can give similar definitions for the epidemic spread in the case of a randomized adversary. In this case we define

$$\widehat{S}_r(G) = E_{N_r}[S(N_r, G)],$$

to be the *expected epidemic spread*, where now the expectation is taken over all possible positions of the r viruses.

We now define the following immunization problems.

Problem 1 (Epidemic Spread Minimization**)** *Given a graph G, a number r of initial viruses, and a number k, immunize k nodes in G, such that the epidemic spread $S_r(G')$ in the immunized graph G' is minimized.*

Problem 2 (Expected Epidemic Spread Minimization**)** *Given a graph G, a number r of initial viruses, and a number k, immunize k nodes in G, such that the expected epidemic spread $\widehat{S}_r(G')$ in the immunized graph is minimized.*

We note that for problem Epidemic Spread Minimization (Problem 1), the role of the adaptive adversary in our framework is played by the influence-maximization algorithm in [10]. Problem 1 is NP-hard (the proof is omitted due to space constraints).

The case of Expected Epidemic Spread Minimization (Problem 2) is different, since an algorithm for that problem attempts to immunize the graph against a random strategy for influence spread. This problem is more closely related with the sum-of-squares partition problem, which was studied recently by Aspnes et al. [2], even though our formulation is more general since it involves a stochastic virus-propagation model. Aspnes et al. [2], prove that the sum-of-squares partition problem is NP-hard, which implies immediately that Problem 2 is also NP-hard.

4.2 Dynamic Propagation Models

The models we consider in this section are special cases of the SIS model. We view virus propagation as a dynamical birth-death process that evolves over time. Viruses are continuously propagated in the network, but they may also die. More precisely, an infected node i propagates the virus to a node j in a single step with *propagation probability* β, while at the same time an infected node may recover with *recovery probability* δ. The ratio β/δ defines the *infection rate* of the virus.

Let M denote the adjacency matrix of graph G, and $\lambda_1(M)$ be the largest eigenvalue of M. Then, condition $\beta/\delta < 1/\lambda_1(M)$ is sufficient for quick recovery of the system. More precisely the following theorem can be proven [9, 15].

Theorem 1 *Given a graph G with adjacency matrix M, and infection rate β/δ, if $\beta/\delta < 1/\lambda_1(M)$ then the expected time until the virus dies out is logarithmic in the number of nodes in the system, against an adaptive adversary.*

Moreover, for many interesting families of graphs, the above condition is also necessary for quick recovery, i.e., if $\beta/\delta > 1/\lambda_1(M)$, the expected time until the virus dies out is exponential in the system size [9]. Thus, it makes sense to talk about an *epidemic threshold* of the network, which determines whether an epidemic will spread, or die out quickly.

A rigorous analysis of the dynamical model encounters the problem of dealing with a non-linear system, which is hard to solve analytically. So, we also consider a *multiple copies* model, which is easier to analyze. In this model we assume that each node may hold multiple *copies* of the virus. More precisely, let $\boldsymbol{v}^t$ be an n-dimensional vector that describes the state of the network at time step t, where v_i^t is the number of copies of the virus at node i at step t. At $t = 0$, v_i^0 is the number of copies of the virus planted by the adversary at node i. At step t, the system evolves as follows. For every node i in the network, and for each of the v_i^t copies of the virus at node i, a copy of the virus is propagated to node j with probability β. Then, the virus copy dies with probability $1 - \delta$. If $\Delta = \beta M + \mathrm{diag}(1 - \delta, \ldots, 1 - \delta)$, and $\hat{\boldsymbol{v}}^t$ is the expected state of the system at step t, then $\hat{\boldsymbol{v}}^t = \Delta \hat{\boldsymbol{v}}^{t-1}$. Therefore, the system is completely linear and we can prove the following theorem.

Theorem 2 *Given a graph G with adjacency matrix M, and infection rate β/δ, the expected time until the virus dies out is logarithmic in the number of nodes in the system if $\beta/\delta < 1/\lambda_1(M)$, and it is unbounded if $\beta/\delta > 1/\lambda_1(M)$, against an adaptive adversary.*

We are now ready to define the following immunization problem for the dynamic model.

Problem 3 (ThresholdMaximization) *Given a graph G, and an infection rate β/δ, immunize the minimum number of nodes in G, such that $\beta/\delta < 1/\lambda_1(M')$, where M' is the adjacency matrix of the immunized graph.*

5 Immunization Strategies

5.1 The Independent Cascade Model

Minimizing the epidemic spread: In this section we discuss the proposed algorithms for network immunization and epidemic-spread containment. We first describe our algorithm for Problem 1, EPIDEMIC SPREAD MINIMIZATION.

For the rest of this section we consider the case that k nodes are immunized against an adversary who places only one virus ($r = 1$). The basic ingredient of the algorithm, based on the observation of Kempe et al. [10], is to view the probabilistic process of virus propagation as sampling over all $2^{|E|}$ possible graphs according to the distribution defined by the probabilities p_{ij} and then run the deterministic-cascade model on the sampled graph. For completeness, we repeat the argument here: when a node i becomes infected, each currently uninfected neighbor j of i becomes also infected with probability p_{ij}. This process is equivalent with a process where each edge (i, j) is *live* in the graph with probability p_{ij}, and the virus is propagating only over the live edges. In turn, this is equivalent to sampling a graph X from the set of all subgraphs of G, where each edge (i, j) of G is present in X with probability p_{ij}, and then propagating the virus *deterministically* on the graph X.

Consider a graph X sampled from G as described in the previous paragraph. If the adversary places a virus at node u, then the number of infected nodes, $s(\{u\}, X)$, is the size of the connected component of X that contains u. Then, the expected epidemic spread $S(\{u\}, G)$ in G with initial placement of the virus at node u can be expressed as

$$S(\{u\}, G) = \sum_{X} Pr[X]s(\{u\}, X), \qquad (1)$$

where $Pr[X]$ is the probability of obtaining graph X from G when sampling according to edge probabilities p_{ij}. Therefore, given G, we can estimate $S(\{u\}, G)$ using Equation (1): sample graphs X from G and compute the expected size of the connected component that u belongs to.

Let $G|_{w_1,w_2,...}$ be the graph resulting after immunizing the nodes $w_1, w_2, \dots$ of G. Assume first that we want to immunize only one node ($k = 1$). For all candidate nodes w_1 to be immunized, we can compute the value of $S(G|_{w_1}) = \max_u S(\{u\}, G|_{w_1})$, which is the worst-case (over all possible initial virus placements) expected epidemic spread if the node w_1 is chosen to be immunized. Then we choose to immunize the node w that minimizes the epidemic spread $S(G|_w)$. In the case that we want to immunize $k > 1$ nodes we proceed in a *greedy* fashion: we first immunize the node w_1 that minimizes $S(G|_{w_1})$. Then we find the best node w_2 to be immunized in the graph $G|_{w_1}$, that is, we find the node w_2 that minimizes $S(G|_{w_1,w_2})$ given the choice of w_1 in the previous step, and we continue until we select k nodes. We call this algorithm GREEDY.

It is instructive to discuss the above GREEDY algorithm in the light of other possible immunization strategies. One such strategy, which we call MAXDEGREE and which we evaluate in our experimental section, is to immunize the nodes with the highest degree in the graph. Intuitively, the best nodes to immunize are the nodes that disconnect the graph in small-size connected components, since such dispersed configurations contain the virus as much as possible. The drawback of the MAXDEGREE strategy is that the nodes with the highest degree do not necessarily disconnect the graph in small connected components. As a simple example consider a chain graph: except for the two side nodes all other nodes have degree 2, so the strategy of selecting the maximum-degree node cannot distinguish among any node. On the other hand, it is clear that the best node to immunize is the node in middle of the chain. The GREEDY will correctly identify the middle of the chain as the best node to immunize, since this is the node that minimizes $S(G|_w)$.

Running time: The overall running time of GREEDY, for a graph of n nodes and m edges is $O(Q(n^2 + nm)k)$, where k is the number of nodes to immunize and Q is the number of samples per iteration. We omit the detailed analysis due to lack of space.

Minimizing the expected epidemic spread: We now discuss the modifications needed in the above GREEDY algorithm in order to address the problem of minimizing the expected epidemic spread in the network. Consider a sample graph X with c connected components of sizes $n_1, \dots, n_c$, such that $n_1 + \ldots + n_c = n$, and let $f_i = n_i/n$ for $i = 1, \dots, c$. Let $\widehat{s}(X)$ denote the expected epidemic spread on X, where expectation is taken over the adversary's placements. Assuming again that the adversary places one virus in the graph, the virus will infect the whole i-th connected component with probability f_i and the size of spread will be precisely n_i. Therefore,

$$\widehat{s}(X) = \sum_{i=1}^{c} f_i n_i = \frac{1}{n} \sum_{i=1}^{c} n_i^2.$$

From the above equation, it is immediately obvious why this case is related with the sum-of-squares partition problem [2], as we mentioned in Section 4.1.

The GREEDY algorithm for EXPECTED EPIDEMIC SPREAD MINIMIZATION is similar to the one we described before. Nodes are immunized one at a time until a total of k nodes are selected. To select a node to immunize, we use the equation $\widehat{S}(G'|_w) = \sum Pr[X]\widehat{s}(X|_w)$, which is analogous to Equation (1). We estimate $\widehat{S}(G'|_w)$ for all nodes w and select the one that yields the smallest value. The overall running time of the algorithm is $O(Q(n + m)k)$.

5.2 Dynamic Propagation Models

Following the discussion in Section 4 the epidemic threshold for the dynamic propagation model is equal to the inverse of the largest eigenvalue of the adjacency matrix M. Therefore, the objective of the immunization algorithm is to decrease this eigenvalue, while incurring the minimum possible damage to the network.

Now, let λ_1, and $\boldsymbol{w}_1$ denote the largest eigenvalue of matrix M and the corresponding eigenvector. Also let $\boldsymbol{a}_i$ denote the i-th row vector of M. The value $\lambda_1 w_1(i)$ is the projection length of the vector $\boldsymbol{a}_i$ on the eigenvector $\boldsymbol{w}_1$. The value λ_1 captures the collective strength of the alignment of the row vectors with vector $\boldsymbol{w}_1$. The eigenvector $\boldsymbol{w}_1$ is the vector with which the points are most strongly aligned. Node i with the maximum $w_1(i)$ value corresponds to the row vector that is most strongly aligned with the vector $\boldsymbol{w}_1$. Therefore, removing i, we expect a large disturbance in the alignment with $\boldsymbol{w}_1$ and thus a large decrease in the eigenvalue λ_1. In the multiple copies model the value $w_1(i)$ determines the rate at which the node i accumulates virus copies. After enough time steps, the node with the maximum value $w_1(i)$ will be the node with the largest number of virus copies.

The algorithm we propose, named EIG, is the following. Proceed in iterations, where each iteration takes as input a matrix B. For the first iteration, $B = M$ the system matrix. Compute the largest eigenvalue λ_1 and the corresponding eigenvector $\boldsymbol{w}_1$ of B. Let β/δ be the epidemic threshold. If $\beta/\delta < \lambda_1$ the algorithm stops. Otherwise, find the node i with the maximum value in the eigenvector $\boldsymbol{w}_1$, and remove it from the graph, that is, remove the corresponding row and column from B. The resulting matrix will be given as input to the next iteration. The running time of the algorithm is $O(kT)$, where k is the number of nodes removed, and T is the time to compute the first eigenvalue and eigenvector. If the graph is sparse this can usually be done in time proportional to the edges of the graph.

We make one more observation about the qualitative properties of our algorithm. The principal eigenvalue of a graph gives us also an indication about the connectivity of the graph. Large eigenvalue corresponds to a graph that is densely connected. The nodes with the maximum value in the first eigenvector are the ones that are most tightly interconnected. Removing these nodes causes the connectivity of the graph to drop. Note that the eigenvector values provide information about the global structure of the graph. This is one of the reasons why our algorithm, as it will become obvious in the experiments, performs in general better than the simple MAXDEGREE heuristic that removes the node with the maximum degree, which takes into account only local information.

6 Experiments

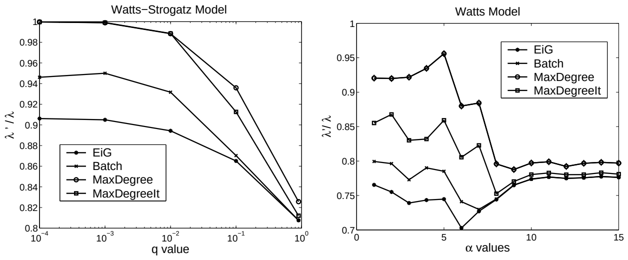

Figure 4. Drop of the first eigenvalue for different values of the parameters q and α.

In this section we compare the performance of the proposed immunization strategies against other natural heuristic strategies. We also explore how the underlying graph structure affects the relative performance of the various algorithms. For the experimental evaluation we use both synthetic and real datasets. We will demonstrate that our algorithms out-perform the other heuristics. We will also demonstrate that our algorithms perform better for graphs with high clustering coefficient, thus being more appropriate for small-world networks.

6.1 Datasets

For the synthetic datasets, we generate two different graph types: *scale-free* and *small-world* graphs.

Scale-free graphs: In scale free-graphs. the probability that a node of the network has degree k is proportional to $k^{-\gamma}$, with $\gamma > 1$. We generate scale free graphs using the generating model proposed in by Barabasi and Albers [3]. The graph generation process proceeds by inserting nodes sequentially. Each new node to be inserted in the graph is linked to one existing node, which is chosen with probability proportional to its current degree. This process simulates the “rich get richer” effect, and generates scale-free graphs with exponent $\gamma = 3$. We use $\mathcal{G}_B$ to denote the family of graphs generated by this model.

Small-world graphs: We use the term *small-world* graphs [16, 17] to describe graphs with small *characteristic path length* and large *clustering coefficient*. The characteristic path length L is defined as the average shortest path between any pair of vertices. The clustering coefficient C is defined as the average fraction of pairs of neighbors of a node that are also connected to each other. We generate small-world graphs using the generating models proposed in [16] and [17]. We use $\mathcal{G}_W$ to denote the family of graphs generated by the former model and $\mathcal{G}_{WS}$ for the family of

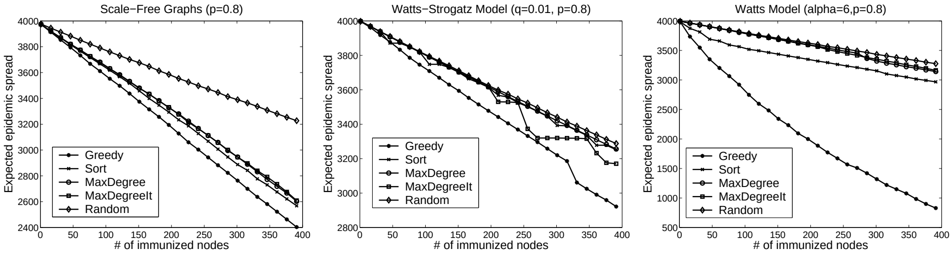

Figure 1. Expected epidemic spread for $\mathcal{G}_B$ and $\mathcal{G}_{WS}$ ($q = 0.01$) and $\mathcal{G}_W$ ($\alpha = 3$).

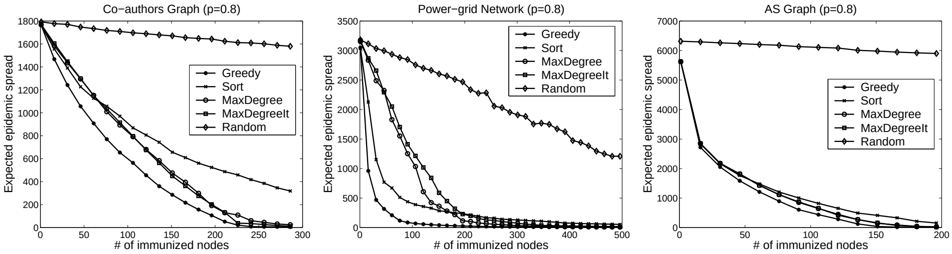

Figure 2. Expected epidemic spread for real graphs.

graphs generated by the latter. For graphs in $\mathcal{G}_W$, the generation process is governed by a parameter α. Intuitively, α determines the probability that two nodes will be connected, given the number of their common neighbors, thus, it controls to which extent the graph will contain communities. For small values of α the graph has small and densely connected components. As α approaches infinity, the generated graphs become random graphs. For the $\mathcal{G}_{WS}$ graphs on the other hand, the generation process is governed by the parameter q. Initially, all nodes are on a ring lattice with each node having degree k. The parameter q determines the probability that an edge from the initial lattice is rewired to connect to another random node in the network. Small values of q entail in graphs that have high clustering coefficient and large average path length, while large values of q create random graphs. For values of q close to 0.01 the generated graphs are small-world graphs.

We note that the families $\mathcal{G}_W$, $\mathcal{G}_{WS}$ and $\mathcal{G}_B$ are quite distinct. We do not observe power law degree distributions for the graphs in $\mathcal{G}_W$ and in $\mathcal{G}_{WS}$, while we observe very low clustering coefficient for the graphs in $\mathcal{G}_B$. However, this is not the case in real life, where we observe networks with scale free and small world properties. Such an example is the co-authors graph we describe next.

In addition to the synthetic datasets we also experiment on real graphs: the *co-authors* graph, the *power-grid* graph and the autonomous systems *AS-graphs*.

Co-author graph: The co-authors dataset consists of 8000 authors of papers in VLDB, PODS and SIGMOD available at the Collection of Computer Science Bibliographies[1]. The co-author graph is constructed by creating undirected edges between authors that have been co-authors in the same paper. We think of the co-authors dataset, as representative of a social network.

Autonomous Systems (AS) graphs: These graphs represent the Autonomous Systems topology of the Internet. Every vertex represents an autonomous system, and two vertices are connected if there is at least one physical link between the two corresponding Autonomous Systems. We considered 8 such different datasets.[2]

Power-grid graph: In this graph the vertices represent generators, transformers and substations, and edges represent high-voltage transmission lines between them.

6.2 The Independent Cascade Model

For the independent-cascade model, we compare the GREEDY algorithms with strategies MAXDEGREE and MAXDEGREEIT, which select the nodes to immunize based on their degrees. The MAXDEGREE algorithm immunizes the nodes in decreasing order of their degree in the original graph. The MAXDEGREEIT algorithm, is similar, but after each step it updates the degrees of the nodes and selects the best in the current graph. We also compare with the SORT algorithm which is defined as follows. Initially, the algorithm computes for each node the gain achieved by removing only this node from the network. It then sorts the nodes

[1] http://liinwww.ira.uka.de/bibliography

[2] Available at `http://www.cs.ucr.edu/ vkrish/`.

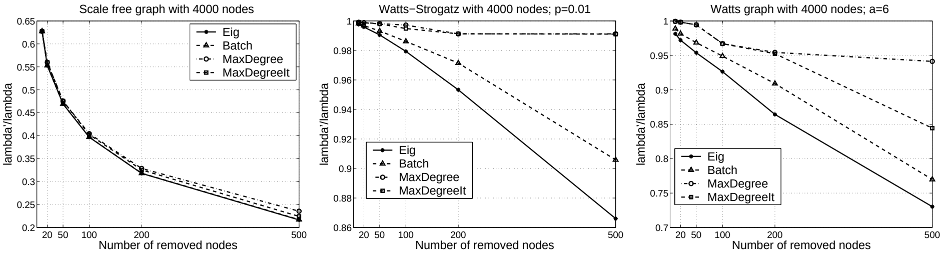

Figure 3. Drop of the first eigenvalue after removing fixed number of nodes for generated graphs

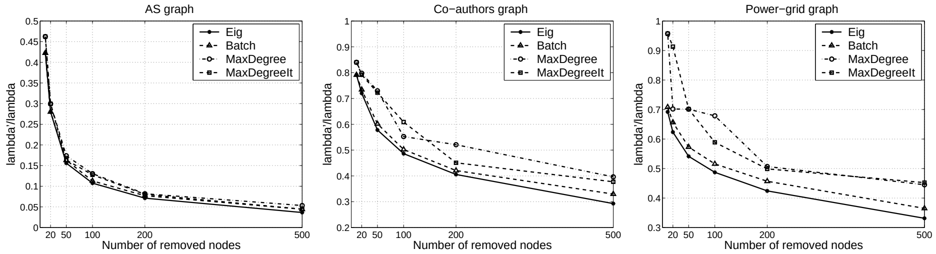

Figure 5. Drop of the first eigenvalue after removing fixed number of nodes for real graphs

in decreasing order of that gain, and proceeds by removing the nodes in that order. Finally, we compare with the algorithm RANDOM, which at every step selects a random node to immunize. Throughout the experimental section if the results for RANDOM are omitted is because the algorithm performs incomparably bad.

For realizing the independent cascade model we assign a uniform probability p to each edge. We experiment with different values for the parameter p. In all cases we consider as measure for the performance the epidemic spread S, or the expected epidemic spread $\widehat{S}$.

Figure 1 shows the expected epidemic spread for the different algorithms as a function of the number of nodes removed from a scale free network. We experiment with a 4000-node graph from the $\mathcal{G}_B$, $\mathcal{G}_{WS}$ and $\mathcal{G}_W$ families. In all cases we have used $p = 0.8$, although our results show similar trends for other values of p of the independent cascade model. Additionally, there graphs from $\mathcal{G}_{WS}$ family were generated with $q = 0.01$. For the graphs from $\mathcal{G}_W$ we used $\alpha = 6$. Both the values of the two parameters result in models where the underlying graphs have high clustering coefficient and low average path length. (We elaborate more on the relationship between the performance of our algorithms and the clustering coefficient in the next subsection.) Although the the GREEDY algorithm performs consistently better than all other strategies, its superiority becomes particularly apparent in small-world graphs. Those are the graphs of the $\mathcal{G}_{WS}$ and $\mathcal{G}_W$ families.

For real graphs we report experiments only for the expected epidemic spread. The results for epidemic spread are similar. In Figure 2 we show results on the co-authors graph and an arbitrarily selected AS graph with 8000 nodes. In all experiments we conducted that the GREEDY algorithm consistently outperformes the other strategies. We also noticed that the SORT algorithm tends to make good choices, although it does not take into account the changes in the underlying graph structure. The same important observation, that the performance of the algorithms is related to the clustering coefficient of the graph, carries over to the real graphs as well. In particular, in the co-authors graph the GREEDY algorithm performs clearly better than the other methods. The difference between the different algorithms is not that striking in the AS graph. In trying to explain this variation, we consider the clustering coefficient of the two graphs and we find out that for the AS graph we have $C_{AS} = 0.42$, which is much smaller than the clustering coefficient of the co-authors graph, $C_{CO} = 0.64$. This indicates that the structural properties of real graphs affect the performance of the algorithms.

6.3 The Dynamic Propagation Model

In this section we evaluate the performance of our immunization strategies for the dynamic propagation model. We again compare our proposed algorithm EIG with MAXDEGREE, MAXDEGREEIT, and RANDOM, which are defined as in the previous section. We also consider the BATCH algorithm, a faster variant of the EIG algorithm that processes nodes in batches. At every step, the BATCH algorithm removes the ℓ nodes with the highest value in the first eigenvector of matrix P. Experimental evidence shows that using $\ell = 2$ does not degrade the performance of the algorithm.

For values $\ell > 2$, we observed a significant drop in the performance.

For comparing the performance of the different algorithms we experiment as follows. For a given graph we evaluate the ratio of the graph's largest eigenvalue after immunizing a fixed number of nodes, to its initial value. The nodes are immunized according to different immunization algorithms. We repeat the test for different number of removed nodes and for all families of generated graphs and real graphs. Again for the generated graphs we experimented with graphs with fixed number of nodes. However, we also did extensive scalability experiments that show that the relative performance of the algorithms is along the same lines for larger graphs as well.

Figure 3 shows the performance of the algorithms for the different families of generated graphs. For graphs in $\mathcal{G}_{WS}$ and $\mathcal{G}_W$, we fix the parameters q and α to values that allow for the generation of small-world graphs. The results shown are averages over 10 runs. From the plots it is obvious that the algorithm EIG outperforms all the other algorithms. Its superiority becomes particularly apparent for the small-world graphs.

For studying the relationship between the algorithms' performance and the structural properties of the graph we proceed as follows. It is known ([17]) that for the $\mathcal{G}_{WS}$ graphs the parameter q has an impact on the clustering coefficient of the generated graphs. For small values of q the clustering coefficient is high, while it gets small as q approaches 1. Figure 4 (left) shows that our algorithm also performs relatively better than the rest of the heuristics for small values of q, while the difference in the performance drops as q increases. More evidence accumulates when we study the performance of the different heuristics on graphs from $\mathcal{G}_W$, that are generated with different values of the parameter α. The relationship between the value of α and the clustering coefficient has been studied in [16]. Starting with $\alpha = 1$ the clustering coefficient of the generated graph is high and increases even more until reaching some maximum value when α takes a value around 7. After that the clustering coefficient drops as α increases. Figure 4 (right) shows that similar is the variation in the relative performance of our algorithm with respect to the rest of the heuristics.

The results for the real datasets demonstrate again the superior performance of EIG algorithm. In all cases the algorithm performs better than the other 3 alternative solutions we compare with. The difference in the performance of the algorithms becomes substantial mainly in the case of the power-grid and the co-authors graph.

7 Conclusions

In this paper, we consider the problem of network immunization against a virus spread. We study the immunization problem under two different models for virus propagation. For the independent-cascade model we propose a greedy algorithm and for the dynamic-propagation models we propose a simple heuristic, which is based on intuition drawn from linear algebra. We experimentally show that our algorithms performs extremely well in practice and much better than degree-related heuristics. Our experimental evaluation shows that our algorithms perform strakingly better than other heuristics when considering graphs with high clustering coefficient. Additionally, their performance is not affected by the average path length of the graph, which makes our algorithms useful for small-world networks.

References

[1] R. M. Anderson and R. M. May. *Infectious diseases in humans*. 1992.

[2] J. Aspnes, K. Chang, and A. Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. In *SODA*, 2005.

[3] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286, 1999.

[4] M. Barthelemy, A. Barrat, R. Pastor-Satorras, and A.Vespignani. Dynamical patterns of epidemic outbreaks in complex heterogeneous networks. *Journal of Theoretical Biology*, 2005.

[5] M. Boguna and R. Pastor-Satorras. Epidemic spreading in correlated complex networks. *Phys. Rev. E 66*, 2002.

[6] M. Boguna, R. Pastor-Satorras, and A. Vespignani. Epidemic spreading in complex networks with degree correlations. *Statistical Mechanics of Complex Networks*, 2003.

[7] R. Cohen, S. Havlin, and D. ben Avraham. Efficient immunization strategies for computer networks and populations. *Phys Rev Lett.*, 2003.

[8] Z. Dezso and A.-L. Barabasi. Halting viruses in scale-free networks. *Phys. Rev. E 66*, 2002.

[9] A. Ganesh, L. Massouli, and D. Towsley. The effect of network topology on the spread of epidemics. In *IEEE INFOCOM*, 2005.

[10] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, 2003.

[11] J. O. Kephart and S. R. White. Measuring and modeling computer virus prevalence. *IEEE Computer Society Symposium on Research in Security and Privacy*, 1993.

[12] W. Kermack and A. McKendrick. A contribution to the mathematical theory of epidemics. *Proc. Roy. Soc. Lond.*, 1927.

[13] M. E. J. Newman. The structure and function of complex networks. *SIAM Reviews*, 45(2):167–256, 2003.

[14] R. Pastor-Satorras and A. Vespignani. Epidemics and immunization in scale-free networks. *Handbook of Graphs and Networks: From the Genome to the Internet*, 2002.

[15] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *SRDS*, 2003.

[16] D. J. Watts. Networks, dynamics and the small world phenomenon. *American Journal of Sociology*, 105, 1999.

[17] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393, 1998.