Reinforcement Learning for Optimal Control of Network Epidemic Processes

2019

Alec H. Kerrigan
University of Central Florida

REINFORCEMENT LEARNING FOR OPTIMAL CONTROL OF NETWORK EPIDEMIC PROCESSES

by

ALEC KERRIGAN

B.S. University of Central Florida, 2019

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science in the Department of Computer Science in the College of Engineering and Computer Science at the University of Central Florida

Summer Term 2019

Thesis Chair: Chinwendu Enyioha, Ph.D

Abstract

Our society is increasingly interconnected, making it easy for cascades/epidemic (diseases, disinformation etc). Current epidemic control efforts are based on approximate network epidemic models, which often ignore the unique complexity and rich information embedded in the complex interconnections of real-world networks/populations.Deep reinforcement learning (RL) is a powerful tool at learning policies for these nonlinear, complex processes in high-dimension. To control an epidemic outbreak on a Susceptible-Infected-Susceptible network epidemic model, we design a RL framework with a custom reward structure using the *node2vec* embedding technique. Results indicate deep RL is able to determine and converge on an optimal intervention policy in a relatively short time.

Contents

1 Introduction and Background

1.1 Graph Theory

A weighted directed graph (also called a digraph) can be defined as G, which contains three sets (V, E, W). $V \triangleq v_1, v_2....v_n$ is the set of n nodes in the graph. $E \subset V \times V$ denotes the set of all edges connected nodes on a graph. Each edge is associated with a positive real weight $w \in W$. We define the set of neighbors by $i \in V$ as $N_i = j : (i, j) \in E$. The adjacency matrix of a weighted, directly graph G given by $A_G = [a_{ij}]$ is an $n \times n$ matrix where each entry a_{ij} defines the weight of a given edge (v_j, v_i). Adjacency matrices provide many useful functions for performing operations on graphs, but in particular provide efficiency for computationally heavy tasks such as machine learning. For the purpose of this investigation, we only consider graphs with positively weighted edges. Therefore, the adjacency matrix of graphs used are always non-negative.

One useful application for the mathematical representation of graphs is the analysis of social networks. Social networks can be built from a variety of information sources such as text, databases, sensor networks, communication systems, and social media [3]. Of particular interest to this investigation is the utilization of graphs for representation of complex social media relationships. The recent rise in misinformation and rumor propagation on social networks is of critical importance due to its ability to influence elections and international politics [2]. Recent research in social networks and their relation to the spread of misinformation finds that one of the primary spread factors for conspiracy theories is the formation of homogeneous clusters in the network. These central clusters in the graph can crudely be described as 'echo-chambers'. Often, a central source inside an echo-chamber is slowly responsible for propagating misinformation, while other members of the echo-chamber ensure that every other member receives it.

1.2 Epidemiology

The use of mathematics to study and analyze epidemic processes dates back to the 18th century. Bernoulli began studying the age-specific equilibrium prevalence of individuals immune to smallpox. Since then, the focus of study in the analysis of epidemic spread among populations has been in the solutions of differential equations involved in both the Susceptible-Infected-Susceptible (SIS) as well as the Susceptible-Infected-Recovered (SIR) model and their variations (See Figure 1) [1]. For example, the SIR model can be described with the following set of differential equations.

$$\frac{\partial S}{\partial t} = -\beta SI \tag{1}$$

$$\frac{\partial I}{\partial t} = \beta SI - \gamma I \tag{2}$$

$$\frac{\partial R}{\partial t} = \gamma I \tag{3}$$

The following figure outlines exactly how individuals transition from one state to another under different spread models.

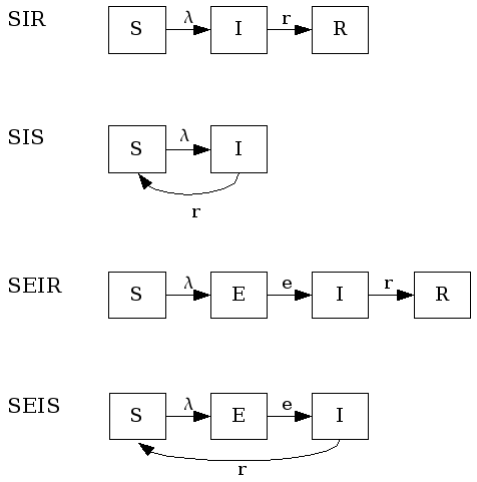

Figure 1: A visualization of various epidemic spreading models. λ represents the probability of transitioning from susceptible to infected. τ represents the probability of transition from infected to recovered, or infected to susceptible in the SIS model. In SEIR and SEIS, an additional state is added: exposed. The rate at which exposed individuals become infected is e.

1.3 Spread Models

There exists an inherent link between the modeling of epidemic spread and network science. Early epidemic models were often based on wide, random-mixing populations, and rather homogeneous distributions of individuals [10]. However, in the real world individuals have a finite number of total connections for which they can transmit the disease. Therefore, networks play an important role in painting a more complete picture of overall epidemic dynamics. Knowing this, it becomes increasing important to understand the difference between spread models, as small differences in how nodes change condition can radically change how different network topology effect overall spread [7].

The most common two epidemic models are the SIR (Susceptible-Infected-Recovered) model and the SIS (Susceptible-Infected-Susceptible) model [9]. In the SIS model, an individual can occupy one of two states: *susceptible* or *infected*. Any susceptible individual i that is connected to an infected neighbor has as probability β_i of entering the infected state. Conversely, an individual i in the infected states has a probability δ_i of recovering and reverting back to the susceptible state. The SIS model is most often used to model the spread of recurring diseases for which an immunity is not built [21]. Of particular interest in these classes of diseases in sexually transmitted diseases, such as chlamydia or gonorrhea, where it is quite common for an individual to re-infected.

In the SIR model, individuals are classified instead in to one of three states susceptible, infected, and recovered. Similar to the SIS model, any susceptible individual i in contact with an infected neighbor has a probability $beta_i$ of entering the infected state. However, in this model, an individual i in the infected state cannot revert back to the susceptible state. Instead, these individuals have a probability $delta_i$ of transitioning from the infected state to the recovered state. It is worth noting that the 'recovered' state does not necessarily correspond to an individual who is safe from the disease, or otherwise healthy. In many real world models, these individuals also represent those who have died or otherwise been removed the population capable of contracting the disease

1.4 Differential Equation Models

Many solutions in the past to the optimal control problem for epidemic processes have focused on use of differential equations to model both the population dynamics as well as the control dynamics. Of this research, much has focused on removal or transfer of populations, rather than vaccination. The quarantine and isolation strategies lend themselves much better to differential equation models as their movement is much more regular than discrete vaccinations.

One analysis by Yan, Zou, and Li [29] examined a model of severe acute respiratory syndrome (SARS) high infection risk ($S_1(t)$)-low infection risk ($S_2(t)$)-asymptomatic (E(t))-quarantined (Q(T))-asymptomatic(I(t))-isolated(J(t))-recovered(R(t))-dead model. The following equations modeled the removal and quarantine process.

$$
\begin{aligned}
S &= -\beta S_1 \frac{I + qE + \epsilon Q + lJ}{N} \\
E &= -\beta(S_i + pS_2)\frac{I + qE + \epsilon Q + lJ}{N} - (u_1(t) + k_1)E \\
Q &= u_1(t)E - \sigma Q \\
I &= -kE - (u_2(t) + \gamma_1 + \delta)I \\
K &= u_2(t)I + \sigma Q - (\gamma_2 + \delta)J \\
R &= \gamma_1 I + \gamma_2 J \\
D &= \delta I + \delta J
\end{aligned}
$$

Using these differential equations, the goal can therefore be formalized to the minimization of the following:

$$J(u_1, u_2) = \int_0^{t_f} [B_1 E(t) + B_2 Q(t) + B_3 I(t) + B_4 J(t) + \frac{C_1}{2} u_1^2(t) + \frac{C_2}{2} u_2^2(t)]dt$$

The results of their investigation found that their existed many optimal control policies that experienced significantly greater results to simple constant control, as well control based on complete investment into their quarantine and isolation strategies.

Recent work by Ruscheul, Pereira, Yanchuk, and Young [23] uses this same approach to a more simple Susceptible-Infected-Quarantined (SIQ) model. For their model, they introduce further variables to assist in the definition of quarantined. If a host does not enter state Q at time τ remain infections until they recover on their own. A host that enters state Q remains in this state for κ. If a a host remains infected with τ units without recovering, it enters Q with a probability p. r is defined as the reproductive number of the disease in the absence of isolation. In the relevant literature, "reproductive number" often refers to some defined threshold that a disease or spread process to meet for its continued growth to be assured. The differential equations are outlined as follows.

$$S(t) = qrS(t)I(t) + I(t) + r\epsilon S(T - \tau\kappa)I(t - \tau - \kappa)$$
$$I(t) = rS(t)I(t) - I(t) - r\epsilon S(t - \tau)I(t - \tau)$$
$$Q(t) = r\epsilon[S(t - \tau)I(t - \tau) - S(t - \tau - \kappa)I(t - \tau - \kappa)]$$

In this context, $epsilon = pe^{-\tau}$ is the effectiveness of identifying infected individuals for quarantining.

2 Problem Statement

2.1 Model

For our approach, we analyze the traditional SIS (susceptible-infected-susceptible) model of network epidemic outbreak. We formally define the following terms.

- Agent state $X_i(t) \in \{0, 1\}$ indicates the health status of node i at time-step t, where $X_i(t) = 0$ implies susceptible, and $X_i(t) = 1$ implies infected.
- Each agent i has an infection rate denoted β_i.
- Each agent i also has a recover rate denoted δ_i.

- The probability of moving from susceptible to infected state is denoted by the following relationship:

- N_i represents all the neighboring nodes in the network G

$$\begin{aligned} \Pr(X_i(t+1) = 1 | X_i(t) = 0) &= \sum_{j \in N_i} \beta_i X_j \\ \Pr(X_i(t+1) = 0 | X_i(t) = 1) &= \delta_i \end{aligned} \tag{4}$$

Given a network of size N, since each node can be in one of two states, the state space is of size 2^N, which grows exponentially in the size of the network and is difficult to analyze. Mean field approximations are use to average out the effect of neighboring nodes on each other. In summary, rather than have nodes switch states as expressed in (4), each node has probability of infection, which quantifies the likelihood of being in the infected state [18].

2.2 Cost

In the trivial case, $\bar{\delta}$ can simply be set to 1 for each agent to ensure that all agents have the maximum possible recovery rate for control of an outbreak. This trivial case, while effective, does not provide useful or interesting information that can develop insight into real world epidemic outbreaks. In real world outbreak scenarios, controllers have significant barriers and limitations that must be considered [16]. Additionally, assuming an ability to arbitrarily vaccinate any disease or epidemic fails consider natural scarcity of corrective matters. In many developing areas, investing an arbitrary amount of money and resources into prevention of outbreaks is simply not possible. Even in situations in which full scale vaccination strategies can be deployed, extreme measures must be avoided due to the possibility of external economic and even social consequences. For example, much of the recent economic losses in Southeast Asia in recent decades can be attributed to losses of poultry due to the excessive government culling of infected birds [13]. Thus, both technical and real-world concerns require the setting of reasonable bounds for infection and recovery rates.

To normalize the costs of vaccinations, we use the following two functions, where β_i and δ_i

represent the infection rate and the recovery rate of the node i, respectively [18]:

$$f_i(\beta_i) = \frac{\beta_i^{-1} - \bar{\beta}_i^{-1}}{\underline{\beta}_i^{-1} - \bar{\beta}_i^{-1}} \quad \text{and} \quad g_i(\delta_i) = \frac{(1-\delta_i)^{-1} - (1-\bar{\delta}_i)^{-1}}{(1-\underline{\delta}_i)^{-1} - (1-\bar{\delta}_i)^{-1}}$$

3 Previous Approaches

3.1 Graph Heuristics

The most common approach to solving the problem of network epidemic control is the use of simple graph heuristics such as centrality. Centrality of a node in a network describes how "important" a single node is compared to other nodes in the network, however there is no single formal definition of this measure [6]. The most simple measure of centrality is degree centrality, which is defined as the sum of all incoming nodes. [20]

$$k_i = \sum_{j=1}^{N} A_{ij} \tag{5}$$

Another common measure of network centrality is betweenness. The betweenness measure of a node describes how likely one is to encounter the node on a random walk of the network. The betweenness centrality is given by the expected number of visits to each node i during a random walk.

$$B_i = \sum_{a=1}^{N} \sum_{b=1}^{N} w(a, i, b) \tag{6}$$

To investigate the link between different measures of centrality and the probability of a node of becoming infected, we performed a simple experiment using a simulation of the epidemic model. A random Erdos-Renyi graph of size $n = 1000$ with an average degree of 5 was generated. Each simulation of the modified SIS epidemic was run for 100 time steps and the total number of times each node was infected was recorded. After 500 total simulations were run, the total number of times each node was infected was tallied and converted to log form. Four measures of centrality were considered: degree, closeness, betweeness, and eigenvector centrality. Because each measure of centrality fundamentally in absolute scale,

we instead mark each node by their standard deviations from the average centrality across all 1000 nodes. This experiment revealed a clear link between a node's level of centrality and that node's propensity to infection, shown in the figure below.

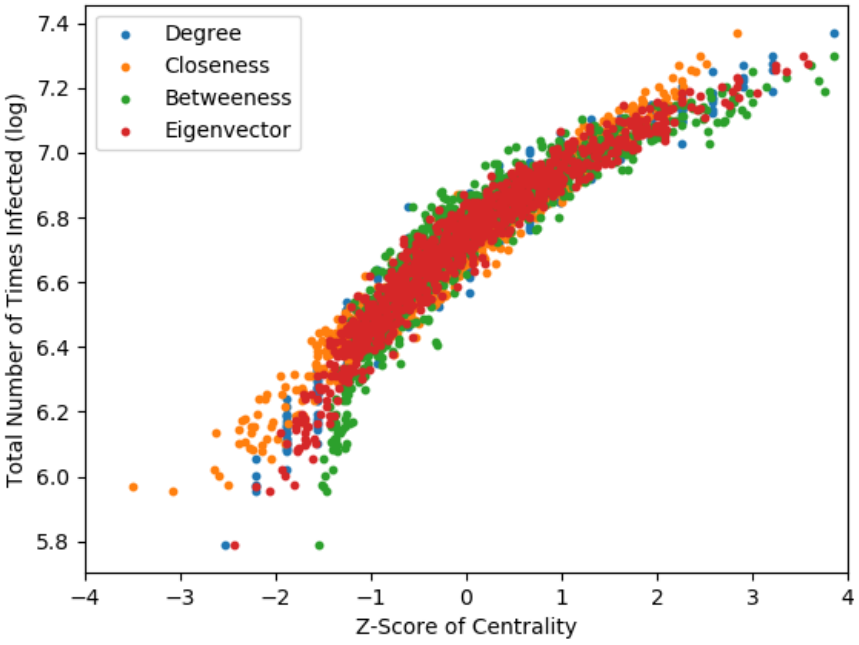

Figure 2: A comparison between the z-score of a node's centrality (based on four measures) on a random Erdos-Renyi graph of n=1000 and the total number of times each node became infected over 500 simulations of 50 timesteps each. The edge probability was 0.2. At each timestep the status of each individual node was recorded, then the log of total number of times infected was plotted against the standard deviations of centrality (z-score). This was done rather than recording the number of times nodes transitioned, as the dynamics of the spread can effect whether a node stays infected as well as transitions.

These results show that while there are slight variations in the different centrality meaures, overall they average out to nearly the same trend. In recent experiments, Khansari and Kaveh [11] found that continually removing the most central nodes in a network contributed the most to the reduction in overall rate of spread. Their research measured two heuristics for rating the strength of some control measure: epidemic threshold and largest-connected-component size. Epidemic threshold of a network is the minimum number of nodes that must be infected for a network to reach an epidemic level. A larger epidemic threshold indi-

cates a smaller chance for the network to reach epidemic status. This threshold can be found by taking the reciprocal of the largest eigenvalue of the adjacency matrix of the network [4]. The Largest-Connected-Component size simply refers to the largest fully connected path in the network. While our approach is primarily concerned with reducing epidemic threshold, understanding the largest path an infection can travel is crucial to analyzing the path of spread. Of all centrality-based targeting approaches, PageRank-Degree and radiality-Degree centrality were found to be most effective at reducing the epidemic threshold of a network. Interestingly, targeting nodes for vaccination based on eccentricity produced worse than random results.

While a centrality based removal approach presents an important benchmark for understanding how individual nodes in a network contribute to epidemic threshold, it fails to capture the more dynamic features involved in epidemic spread. Nonetheless, these contributions provide important estimations for oftentimes highly complex systems.

3.2 Optimization

Estimation techniques explained above have further uses in providing average control policies. The problem of optimal control of network spread processes can be reduced to one of two optimization problems.

Rate Constrained Problem: Given a maximum rate of infection, find the minimum total cost of vaccination
Budget Constrained Problem: Given a maximum cost of vaccination, find the minimum rate of infection possible

Both of these perspectives allow us to formalize the optimal control problem. Previous work as shown that eigenvalues of graphs are useful in estimating stability of various network processes [19]. Further work, specifically with the process of epidemic spread, has found that the overall epidemic stability of a network can be approximated by the following eigenvalue

[17].

$$\lambda_1(BA_G - D) \tag{7}$$

Where A_G represents the adjacency matrix of the network. B represents the diagonal matrix of infection rates, while D represents the diagonal matrix of recovery rates such that

$$B_{ii} = \beta_i \quad \text{and} \quad D_{ii} = \delta_i. \tag{8}$$

Preciado et al. [18] further finds that that this mean field approximation of the modified SIS model determines the disease-free equilibrium of the network in the following way:
If the real part of the principle eigenvalue of $BA_G - D$ satisfies

$$\Re[\lambda_1(BA_G - D])] \leqslant -\epsilon \tag{9}$$

for some $\epsilon \geqslant 0$, we say that the disease-free equilibrium is globally exponentially stable. This epsilon can either be set for the purpose of replicating some real world epidemic process, or empirically based on experimental results. They therefore formalize both problems with the following geometric programs.

The rate constrained problem can be formalized as

$$\begin{aligned} \underset{\{\beta_i,\ \delta_i\}_1^n}{\text{minimize}} \quad & \sum_{i=1}^{n} f_i(\beta_i) + g_i(\delta_i) \\ \text{subject to} \quad & \Re[\lambda_1(BA_G - D])] \leqslant -\epsilon \end{aligned} \tag{10}$$

with appropriate bounds on β_i and δ_i, where f and g are the cost functions for correction and prevention, respectively.
The budget constrained problem can be formalized as

$$\begin{aligned} \underset{q_1,\ldots,q_N}{\text{Minimize}} \quad & \sum_{i=1}^{n} f_i(\beta_i) + g_i(\delta_i) \\ \text{subject to} \quad & \Re[\lambda_1(BA_G - D])] \leqslant -\epsilon \end{aligned} \tag{11}$$

where C is a predetermined "budget" for an overall vaccination strategy. Similar to *epsilon*, this budget can either be determined based on real world data or from experimental results based on solutions to problem 1. Both of these two optimization problems are solved via Geometric Programming, and allow for optimal control of network epidemics on weighted and directed networks in polynomial time.

However, these findings are still dependant on mean-field approximations of the spread process, and do not take into account the complex spread dynamics present in true epidemic models. These findings, using the approximations, put great weight on the correlation between the investment on a node and its centrality (based on multiple measures of centrality). In fact, based a certain threshold, the geometric optimization technique yields an almost perfect linear relationship between the centrality of nodes and their overall investment on vaccination. However, further research shows that the value of peripheral nodes is greatly underestimated when using approximation techniques [26].

4 Reinforcement Learning

4.1 Background

Reinforcement learning is a subset of machine learning that attempts to map action states to actions for the purpose of maximizing some type of reward. Reinforcement learning is unique to other forms of learning as it does not rely on labeled training data, but instead learns by doing, similar to how humans train [27]. In most reinforcement learning models, we define abstract actor learning and doing actions as the "agent". The system that the agent acts in is known as the "environment". In a well defined reinforcement learning system, every action that an agent can take in some environment can be objectively rated or judged in some way. As we require the agent to have some metric to optimize, we call this rating the 'reward'.

However, in most situations, a static, unchanging environment is not particularly useful. In

our model's case, the state of the dynamic spread process is ever-changing, and therefore our reinforcement learning approach much account for this. Therefore, in most reinforcement learning models, rewards are mapped from some pair of action and state. In our case, any given node on the network can either be in the infected or susceptible state. Due to this, the total number of possible states in the environment of an arbitrarily sized network in combinatorially large. Learning the optimal action for each of these states, or even learning the expected reward for each of these states, is therefore not feasible.

4.2 Models

The most basic of deep reinforcement learning models is "Deep Q-Learning" a modification of the basic reward signal mode that underlines traditional reinforcement learning using neural networks. In Q-Learning, we build a mapping of states (s) and action (a) pairs to a representation of maximal possible future reward (Q). We can formalize this process with the Bellman equation.

$$Q^*(s,a) = \mathbb{E}_{s' \ \varepsilon}[r + \gamma \max_a Q^*(s',a')|s,a] \tag{12}$$

Over multiple iterations of the environment, for any state, we take the action with the highest potential Q value (with some small chance of random action, determined by a separate function) [15]. Additionally, future reward is discounted by some percentage γ. This is due to the inherent stochastic nature of the environments we wish these agents to learn. Q-Learning is valuable for solving small scale problems with both a small state space and action space, however it is difficult to use with environments with a very large state space. In fact, it is impossible to perform with a continuous action space. The solution to the state space problem is instead of learning a function that maps every single state, action pair to a expected reward value (Q-value), we learn a function to estimate the reward values for each possible action in some state. Neural networks have been proven in experiments to be extremely good at performing this specific operation [15].

For environments containing a continuous action space, representing the entire action space is impossible, as clearly a neural network could not represent a Q value for every single real value between zero and one. Therefore, "actor critic"[12] methods have instead opted to not only use a neural network for estimating the value of each action, but also which action has the greatest Q value. In these models, the "actor" network attempts to learn the policy itself, while a seconds network called the "critic" is deployed to estimate the Q value of the action that the first network outputted. By performing gradient ascent on the critic network, the loss from backpropagation can be transferred over the critic, allowing for a direct policy to be learned.

While for this research many different types of reinforcement learning models were tested, the results presented in this research utilized Proximal Policy Optimization (PPO)[24]. PPO receives its namesake from its "clipping" procedure, which allows the model to sample sub optimal actions to find possible new global optima. However, clipping prevents the PPO model from searching too far outside what it has already determined to be a relatively optimal policy. The model samples different actions on a Gaussian distribution centered around the current most optimal policy, then uses that distribution to choose which actions to take. This allows the model to continually search for new policies, which always taking a relatively optimal action.

4.3 Reward Model

Our reward model is built to provide a balance between containing the epidemic outbreak and doing so at minimal cost. This proves to be a difficult task, as there is no inherently clear way to weigh the value of preventing spread against minimal use of preventative and corrective resources. It is a non-trivial task to ascertain exactly how much any given corrective resource ought to be worth when compared to the survival of any given timestep. Therefore, we can not simply use a flat reward amount for the survival of a timestep, and instead design it as a controllable variable. This way, our model can be tuned to allow for many different types of networks, without having to rework the entire model when testing

it on different data. For example, when using flat reward amounts for each different type of graph, we found experimentally that the reinforcement learning model would rarely attempt to balance survival and correction, and would exclusively favor one over the other except in rare circumstances. However, it is worth noting that in real life circumstance it is likely we would favor survival over optimal correction, massive loss of life from disease spread can often have incalculable external cost. Despite this fact, it is still important to consider hard constraints that communities may have for intercepting outbreaks, and therefore the importance of optimizing the cost of correction cannot be ignored.

We define ϵ as the allowable epidemic threshold, as a percentage of the total nodes on the network. For most experiments, This value is set to 0.5.

We define as a constant representing the base reward given to the model for surviving one time step (preventing the total number of infected nodes to surpass $N\kappa$. Due of the complexity of large networks, the value will often need to be manually adjusted based on the individual networks to prevent unintended results. This parameter is especially important as it can tip the model to favoring survival over optimally when set great enough. In our experiments, we found that if this parameter was not set great enough, the model would often ignore looking for paths of survival, and would simply optimize for whatever time step the epidemic would often overrun at.

We define B as the "budget" allotted to the model. This also functions as a percentage of the total number of nodes in the network. Intuitively, we expect much larger networks to have a smaller percentage of their nodes being "critical" nodes. For example, consider a map of the United States with the capital of each state being what can be considered "critical" nodes with respect to some viral infection spreading over airlines. If we add more cities into the airline network, we are simply making the network more dense, the capitals remain the most central and important nodes. Therefore, it is important to force the model to try and spend a smaller percentage of the overall possible investment in order to "push" the model

toward faster learning. for the purpose of controlling epidemic processes. Because the cost of investment into correction and prevention is normalized between 0 and 1, the maximum cost of any given time step in the model is N, where N is the size of the network.

$$R_t = \begin{cases} \omega + (n_G * B) - \sum_{i \in G}, & \text{if } \sum_{i \ inG} X_i < \epsilon \\ -150, & \text{otherwise} \end{cases} \tag{13}$$

where R_t refers to the reward given at time-step t.

To demonstrate the susceptibility of results of the model to the individual graphs, we perform a simple experiment comparing the results of a trivial vaccination strategy on three different small graphs: the Karate-Club social graph [30], Watts-Strogatz Small World Graph [28], and an Erdos-Renyi Random Network [5]. We define the *Trivial Solution* as simply setting β to $\underline{\beta}$ and δ to $\bar{\delta}$ at all time-steps in the simulation, therefore forcing maximal investment into both correction and prevention. (Note that for these tests, a simplified method of measuring rewards were used, so the absolute values differ from the experimental results presented later. This is because the PPO model we used relies on batch sampling to measure progress, which is much more resource intensive then a simple consecutive run of the epidemic simulation).

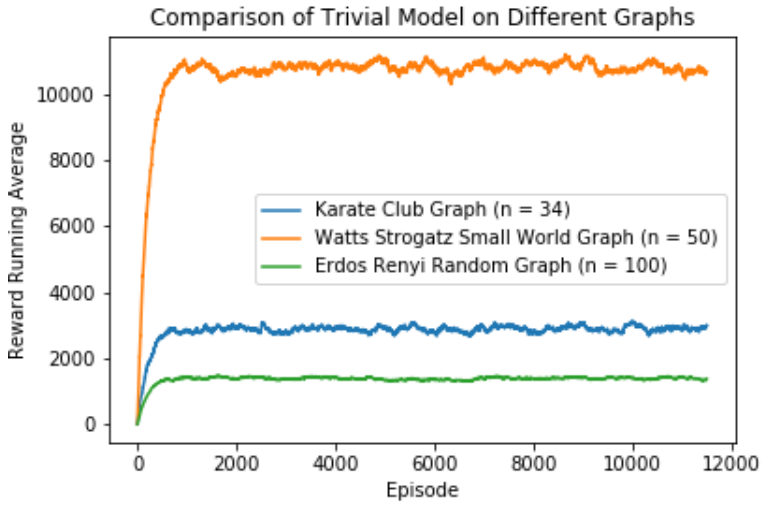

Figure 3: A comparison of the effect on a trivial vaccination strategy on different small random graphs. The following parameters were used: $\bar{\beta} = 0.3$, $\underline{\beta} = 0.1$, $\bar{\delta} = 0.4$, $\underline{\delta} = 0.2$, $\kappa = 0.5$, $\omega = 6$, $B = 0.5$

While the Karate-Club graph and the Erdos-Renyi graph performed relatively close, we find that the Watts-Strogatz Small World graph seems to outperform the other two by a factor of 5 when the trivial vaccination strategy is performed. A further experiment confirms that this discrepancy can be attributed to the average degree of the network. Because the degree of all nodes in a Watts-Strogatz network (defined as k) are predefined and static, we can compare how tuning the We ran the same simulation with the trivial vaccination policy on three different Watts-Strogatz networks with varying k values.

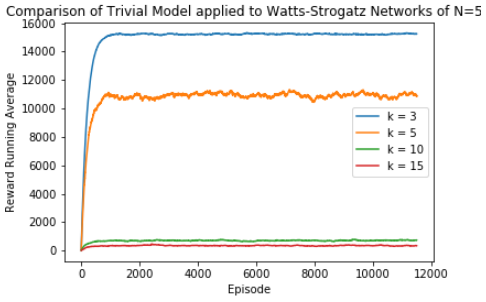

Figure 4: Comparison of different k-values of a Watts-Strogatz network of n=50

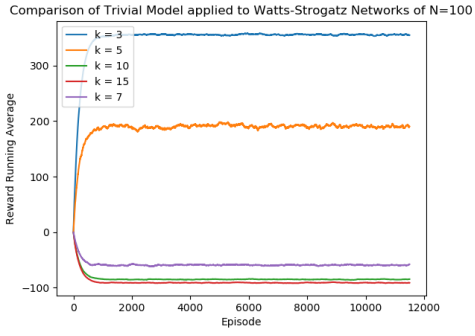

Figure 5: Comparison of different k-values of a Watts-Strogatz network of n=100

We can note a fairly clear drop in performance from the $k = 5$ simulation to the $k = 10$ simulation. This suggests that the average degree of the network is heavily influential on the difficulty of optimally controlling an epidemic process. We can hypothesize that after a certain average degree threshold is passed, it becomes impossible to vaccinate a network, even with maximal investment. It is therefore imperative for experiments that a balance is struck between the bounds of the infection, and the overall connectivity of the network.

5 Node Embedding and Clustering

While we have been able to formalize a model of the epidemic spread problem from which baseline reinforcement learning algorithms can be applied, the action space of our model grows linearly with the size of the network, as we assume the ability to manually tune each infection and recovery rate at each node. We found that on sufficiently large networks (x ¿ 200), convergence on an optimal policy takes an non-feasible amount of time. This is because in artificial neural networks, the number of parameters to train increase exponentially with the size of the output space. The difference can be compared to trying to learn to walk by trial and error when there are 10 joins to consider vs 500 joints. Therefore, we must reduce the action space of our model by choosing smaller clusters to act upon.

Instead of clustering according to inherent features of the network, it can often be more effective to use a learned embedding scheme to determine a more effective method of clustering nodes. In order to best perform the clustering task, we turn to algorithms that can extract hidden features from networks that aid in understanding what makes one node similar to another.

Feature learning for words in some natural language has been a heavily explored field of research for some time. *word2vec* [22] by utilizing one of two models: a continuous bag of words (CBOW) or a skip-gram, coupled with traditional optimization techniques.

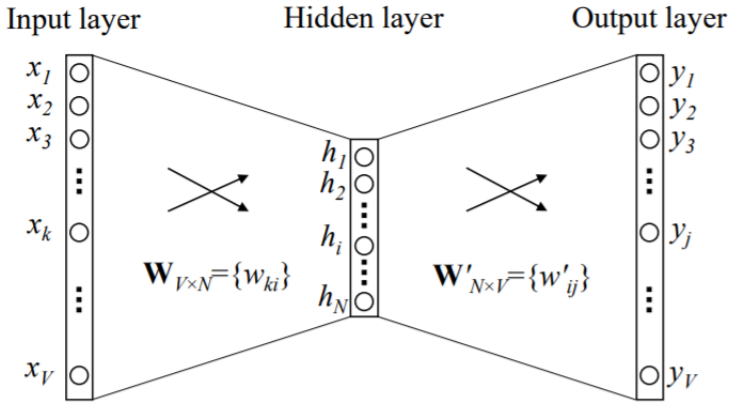

Figure 6: A single word context CBOW. V denotes the size of the vocabulary, with $x_i \in 0, 1$ denoting the presence of a word in the input layer.

The goal of the CBOW is to maximize the conditional probability of observing the target word W_O given a set of words in similar context W_I. u_j represents the score that each word in the vocabulary is given based on the hidden weights of the neural network. y_j represents the output for the jth word in the vocabulary.

$$\max p(w_O|w_I) = \max yj* = \max logy_{j*} = u_{j*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) \equiv -E \tag{14}$$

The skip-gram model is essentially the opposite of the CBOW model [14]. Instead of optimizing to maximize the probability of observing the target word W_O given the set of context words W_I, we instead optimize the network to maximize the probability of observing the context words $W_{O,1}, W_{O,2}...W_{O,C}$ (with C being the number of context words sampled) given the the target word W_I. The loss function is therefore changed to the following

$$E = -\log \Pr(W_{O,1}, W_{O,2}...W_{O,C}|W_I) = -\sum_{c=1}^{C} u_{j*_c} + C \times \log \sum_{j'=1}^{V} \exp(u_{j'}) \tag{15}$$

Both of these models are often used for the application of *word2vec*. *word2vec* has proven to be an incredibly powerful tool in the realm of word classification tasks. It has shown to beat previous cutting edge techniques in the fields of sentiment analysis [31], named entity

recognition [25], and even musical analysis [8].

Due to the wide range of utilization for word12vec, the algorithm can just as easily be extended for the purposes of graph embedding. In order to execute *word2vec*, a corpus of "sentences" must be built for the algorithm to learn. The sentences can simply be a sequence of nodes, with any given node in the graph appearing only once. Once an embedding of the network has been established, a simple clustering algorithm such as k-means is used to determine groups on which the reinforcement learning model is permitted to act upon.

For *node2vec* to build these sequences that are fed into *word2vec*, biased random walks are used. Let c_i be the i^{th} step of a walk, with $c_1 = u$, where u is the source node []. First, we generate a series of biases on each edge $\in E$. This ensures that the resulting vectors are not too normally distributed, as this prevents us from effectively clustering the network. One simple way to produce these random walks would be to use difference of centrality, or just randomly generate using a random distribution. We denote the bias from v to x as π_{vx}. A random constant chosen for each walk in the set of random walks used to produce variation is denoted by Z.

$$\Pr(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{xv}}{Z} \text{ if } (v, x) \in E \\ 0 \text{ otherwise} \end{cases} \tag{16}$$

Using this model, we can produce a series of walks that serve as the "corpus" for the network we are trying to embed. Traditionally, *node2vec* utilizes the skip-gram *word2vec* model, which performs better with a smaller corpus size as well as smaller corpus scope.

6 Results

We fine that baseline reinforcement learning models, particular Promixal Policy Optimization, are effective at learning control of network epidemic processes. On most small to medium sized networks, PPO is able to converge on a policy that successfully prevents an epidemic from spreading beyond an arbitrary threshold ϵ, given reasonable bounds of the

infection.

The following small Watts-Strogatz small world network was used for initial testing of models.

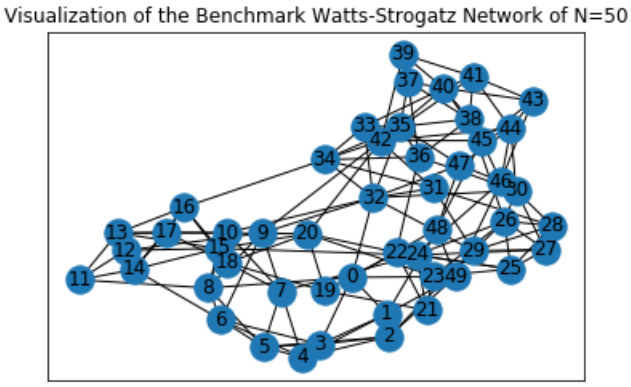

Figure 7: Networkx drawing of Watts-Strogatz small world network used for small scale testing, with $n = 50,\ k = 7,$ and $p = 0.2$.

To verify the ability of the PPO model to learn a solution to the epidemic spread problem, we first ran unclustered (with a action space equal to the number of nodes in the network) for 7000 episodes. We observe clear evidence that proximal policy optimization can converge on an efficient vaccination policy in a relatively short training time.

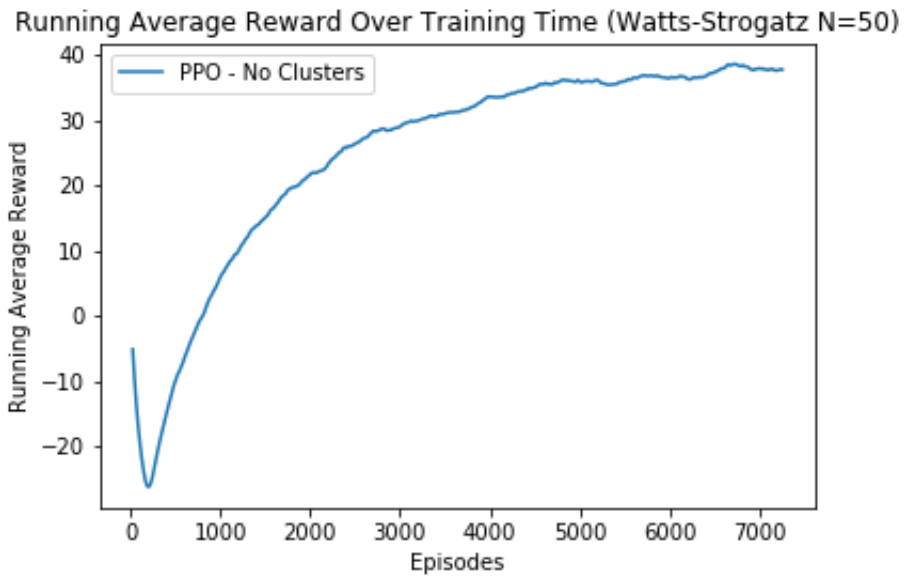

Figure 8: The results of 7 hours of training of the PPO model with no clustering on the benchmark Watts-Strogatz network of n = 50. The benchmark for the trivial policy is a reward of -340.

This model takes approximately 3 to 6 hour to train on a Nvdia GTX 1060 with PyTorch. However, this relatively short training time grows exponentially as the size of the network grows. This suggests that while the complex relationships between nodes in a larger network can be learned, information embedded in the network becomes increasing noisy as its size increases.

To compare the effectiveness of our clustering scheme on the ability of the model to learn, we run the model on two different standardized Watts-Strogatz networks, one of n = 50 and one of n = 100. The model for n = 50 ran for approximately 8 hours before converging, while the model for n = 100 took approximately 3 hours to converge.

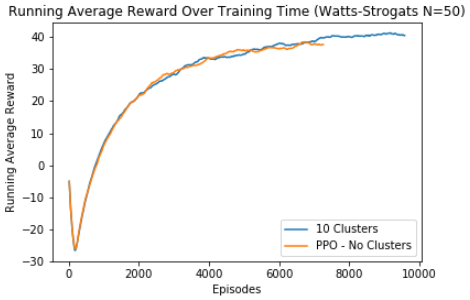

Figure 9: Training results of a network of size n=50. $\bar{\beta} = 0.25$, $\underline{\beta} = 0.04$, $\bar{\delta} = 0.25$, $\underline{\delta} = 0.1$, $\kappa = 0.5$, $\omega = 6$, $\bar{B} = 0.5$

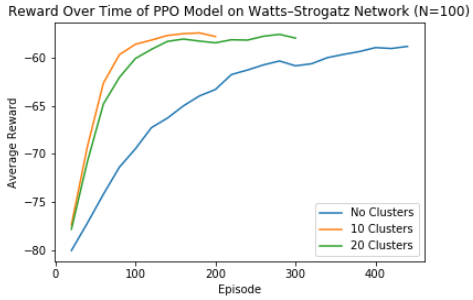

Figure 10: Training results of a network of size n=100. $\bar{\beta} = 0.265$, $\underline{\beta} = 0.085$, $\bar{\delta} = 0.275$, $\underline{\delta} = 0.125$, $\kappa = 0.5$, $\omega = 6$, $B = 0.5$

7 Conclusion

We can conclude on face that reinforcement learning, particularly proximal policy optimization, is an effective means of solving the optimal control problem for network epidemic processes. Additionally, we find that the use of clustering via a combination of *node2vec* for embedding of nodes, as k-means to create clusters after an embedding has been performed marginally decreases the time for the model to converge on a vaccination strategy capable of preventing an outbreak scenario. We particularly observe that the benefit of cluster targeting over targeted the entire network increases as the size of the network increases. This can be easily explained by the fact that the number of weights in a densely connected neural network decreases exponentially with reduction in the size of the network.

We can also note by the amount of hyperparmeter tuning in the simulation that must be done to perform experiments that yield useful information that the model in it's current form may be difficult to apply to current favored real-world examples. Clearly, in real world situations, one cannot modify the bounds of infection as well as bounds of vaccinations to such a precise degree. It is also important to note that the model itself can be considered problematic as those in charge of vaccination and health policy do not have the ability to apply continuous amounts of vaccine or preventative medicine. Typically, set discrete doses are allowed for any given corrective measure. Nonetheless, the results from our model show

that reinforcement learning can be applied broadly to problems involving network epidemics.

8 Future Work

In particular, we would like to explore the how well the model scales, as many of the networks tested for this work were relatively small. Unlike many reinforcement learning environments, the action space of our SIS environment scales linearly as the complexity of the network grows. We found that after a certain network size threshhold, usually after n=300, the training time of the model became too large to be viable. In many experience with extremely large networks ($n \geqslant 500$), we found that even with clustering, the epidemic simulations themselves would become quite CPU intensive. Interestingly, the training of the neural networks themselves rarely presented a resource problem. Future iterations of the epidemic simulation used ought to attempt to make use of parallel processing and the use of multithreading.

Next, we would like to further explore and analyze the various hyperparameters of both the epidemic model, as well as the reinforcement learning environment. For our experiments, we found each different network required careful tuning of parameters related to the specific epidemic. In particular, the simulation is quite sensitive to the careful tuning of $\bar{\beta}$, $\underline{\beta}$, $\bar{\delta}$, and $\underline{\delta}$. In tuning these variables, we were looking for thresholds such that eliminating all investment lead to near instant epidemic outbreak, while maximizing investment across all nodes led to just barely barely surviving all timesteps. In future work, we would like to more deeply analyze the relationship between these variables and the structure of the networks. Finally, while it is clear that the reinforcement learning method is effective at learning a policy and converging on an optimal solution, we would like to further explore how optimal this policy is compared to other possible methods, such as targeting nodes in proportion to their centrality, across multiple measures. Due to the nature of deep learning, the reinforcement learning model is effective at determining a locally optimal control policy, but cannot guarantee a globally optimal policy. We suspect that much of what determines which

local policy the model first converges on depends greatly on how rewarding each individual timestep is, versus how rewarding less investment into vaccination is. Therefore, in future work, we would seek to strike a more careful balance between these two reward structures to ensure that the model does not overly prioritize one over the other. In many experiments, when ω is set too low, we find that the model prioritizes cost savings, and rarely attempts to control the outbreak. On the other hand, when omega is set too high, the model does not attempt to optimize cost at all. Therefore, future work ought to focus on precisely determining what these parameters are set to.

References

[1] Bailey, N. T., et al. *The mathematical theory of infectious diseases and its applications*. No. 2nd ediition. Charles Griffin & Company Ltd 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

[2] Budak, C., Agrawal, D., and El Abbadi, A. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web* (2011), ACM, pp. 665–674.

[3] Campbell, W. M., Dagli, C. K., and Weinstein, C. J. Social network analysis with content and graphs. *Lincoln Laboratory Journal 20*, 1 (2013), 61–81.

[4] Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., and Faloutsos, C. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC) 10*, 4 (2008), 1.

[5] ERDdS, P., and R&wi, A. On random graphs i. *Publ. Math. Debrecen 6* (1959), 290–297.

[6] Freeman, L. C. Centrality in social networks conceptual clarification. *Social networks 1*, 3 (1978), 215–239.

[7] Ganesh, A., Massoulié, L., and Towsley, D. The effect of network topology on the spread of epidemics. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* (2005), vol. 2, IEEE, pp. 1455–1466.

[8] Herremans, D., and Chuan, C.-H. Modeling musical context with word2vec. *arXiv preprint arXiv:1706.09088* (2017).

[9] Hethcote, H. W. A thousand and one epidemic models. In *Frontiers in mathematical biology*. Springer, 1994, pp. 504–515.

[10] Keeling, M. J., and Eames, K. T. Networks and epidemic models. *Journal of the Royal Society Interface 2*, 4 (2005), 295–307.

[11] Khansari, M., Kaveh, A., Heshmati, Z., and Motlaq, M. A. Centrality measures for immunization of weighted networks. *Network Biology 6*, 1 (2016), 12–27.

[12] Konda, V. R., and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in neural information processing systems* (2000), pp. 1008–1014.

[13] Mack, A., Choffnes, E. R., Sparling, P. F., Hamburg, M. A., Lemon, S. M., et al. *Ethical and legal considerations in mitigating pandemic disease: workshop summary*. National Academies Press, 2007.

[14] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[15] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[16] Murray, C. J., Lopez, A. D., Chin, B., Feehan, D., and Hill, K. H. Estimation of potential global pandemic influenza mortality on the basis of vital registry data from the 1918–20 pandemic: a quantitative analysis. *The Lancet 368*, 9554 (2006), 2211–2218.

[17] Preciado, V. M., Zargham, M., Enyioha, C., Jadbabaie, A., and Pappas, G. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *52nd IEEE conference on decision and control* (2013), IEEE, pp. 7486–7491.

[18] Preciado, V. M., Zargham, M., Enyioha, C., Jadbabaie, A., and Pappas, G. J. Optimal resource allocation for network protection against spreading processes. *IEEE Transactions on Control of Network Systems 1*, 1 (2014), 99–108.

[19] Restrepo, J. G., Ott, E., and Hunt, B. R. Approximating the largest eigenvalue of network adjacency matrices. *Physical Review E 76*, 5 (2007), 056119.

[20] Rodrigues, F. A. Network centrality: an introduction. In *A Mathematical Modeling Approach from Nonlinear Dynamics to Complex Systems*. Springer, 2019, pp. 177–196.

[21] Rohani, P., Earn, D. J., and Grenfell, B. T. Impact of immunisation on pertussis transmission in england and wales. *The Lancet 355*, 9200 (2000), 285–286.

[22] Rong, X. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738* (2014).

[23] Ruschel, S., Pereira, T., Yanchuk, S., and Young, L.-S. An siq delay differential equations model for disease control via isolation. *Journal of mathematical biology* (2019), 1–31.

[24] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[25] Sienčnik, S. K. Adapting word2vec to named entity recognition. In *Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania* (2015), no. 109, Linköping University Electronic Press, pp. 239–243.

[26] Šikić, M., Lančić, A., Antulov-Fantulin, N., and Štefančić, H. Epidemic centrality—is there an underestimated epidemic impact of network peripheral nodes? *The European Physical Journal B 86*, 10 (2013), 440.

[27] Sutton, R. S., and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

[28] Watts, D. J., and Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature 393*, 6684 (1998), 440.

[29] Yan, X., Zou, Y., and Li, J. Optimal quarantine and isolation strategies in epidemics control. *World Journal of Modelling and Simulation 3*, 3 (2007), 202–211.

[30] Zachary, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research 33*, 4 (1977), 452–473.

[31] Zhang, D., Xu, H., Su, Z., and Xu, Y. Chinese comments sentiment classification based on word2vec and svmperf. *Expert Systems with Applications 42*, 4 (2015), 1857–1863.