Reinforcement Learning: Applications to epidemiological models

Albanese, Federico[1,2,†] and Ferreyra, Emanuel Javier[1,‡]

[1]Instituto de Cálculo (IC), Universidad de Buenos Aires - CONICET
[2]Instituto de Ciencias de la Computación (ICC), Universidad de Buenos Aires - CONICET
[†]ffalbanese@gmail.com
[‡]emanueljf@gmail.com

Agosto 2019

We attempt to analyse the optimal vaccination strategy in a population with individuals susceptible to contracting a disease in a contact process that we consider markovian, using reinforcement learning techniques. In connotation with theoretical results, our simulation shows that the intention of vaccinating depends on the proportion of infected in the population being as high as possible when the infection is somehow probable, and with no vaccination after that phase.

1 Introduction

1.1 Reinforcement learning

Reinforcement learning problems are described by an agent who take actions or decisions in order to optimize a given utility function and interacts with an environment which can observe uncertainly [1]. The actions taken can modify the environment and the state of the agent, who can incur in some instant costs or benefit. This could restrict the future decisions and possible states, implying the necessity of designing a strategy or plan to get the optimum.

These methods are based on the Markov Decision Problems theory, where a policy is optimal if the maximum benefit is attainable from any state when the correct action is selected. Thus, there is defined a value function containing the information of the expected returns from each moment in time depending on the current state and the possible actions to be taken. In reinforcement learning methods, the value function is estimated with the Q-function.

1.1.1 Reinforcement Learning: Monte Carlo

If an agent follows a policy for many episodes, using Monte-Carlo Prediction, we can construct the Q-table from the results from these episodes. We learn value functions from sample returns of the MDP taking an average, then we improve our existing policy by greedily choosing the best action at each state as per our knowledge [1].

1.1.2 Reinforcement Leraning: Temporal difference Methods

Instead of updating the Q-function at the end of the episode, one can implement temporal difference methods, as SARSA or Q-learning, that boot strap, meaning that they make this update at the next time step [2]. As in dynamic programming, they involve the Bellman equation to update. We apply the both methods mentioned, being the difference between them how the q-function is updated.

In SARSA, the update equation is

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \lambda Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)) \quad (1)$$

where Q is the estimation of the value function, S_t, A_t are the state and action at time t respectively, α is an hyperparameter that regulate the importance of the latter experiences over the starting ones, and λ is the discount factor.

The update equation in the Q-learning method is the following

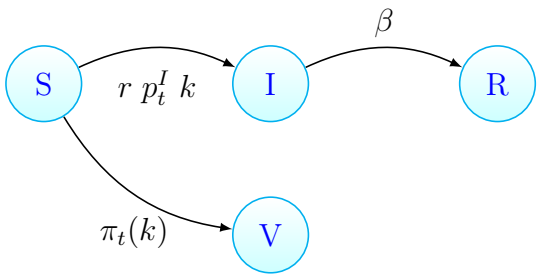
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \lambda \max_a Q(S_{t+1}, a) - Q(S_t, A_t)) \tag{2}$$

1.2 Epidemiological models

The state of the art concerning epidemic models with or without immunisation is very extensive, being able to find models where individuals are susceptible to a certain disease and before a period of infection they return to the initial state (SIS model), or they immunise before the infection and considered removed from the epidemic propagation (SIR model) [3]. There are also models in which there is a first stage in which some individuals get vaccinated, and then the dynamics are the classical, and others in which some information is propagated simultaneously with the infection. This models are supported over fully connected populations, or more complex networks, finite or not, with heterogeneous individuals or homogeneous, and the list can continue. For a complete review of epidemic models in complex networks we refer the reader to [4] or [5].

1.2.1 SIRV

We are interested in a model in which individuals could be in four possible states: Susceptible (S), Infected (I), Recovered (R) or Vaccinated (V). For a Susceptible individual we consider several exponential clocks, one for each edge with an Infected alter with parameter r. It will describe the contact process: if this clock rings the Susceptible transitions to state Infected and remain infectious during an exponential time with mean $1/\beta$, whereupon it will not longer infect any node, going to Recovered state. Also for the Susceptible, we consider another exponential clock with parameter $\pi_t(k)$ depending on the degree k of the node, which is the time dependent control variable and represents the rate at which the individual becomes Vaccinated. From the definition, the system results a Markov Decision Process with bounded rates. This is the schematic of transitions:



1.3 Analytical results

The model we analyze is very similar to the proposed in [6] mainly differing the time setting, but we know from [7] that continuous time Markov decision process with uniformly bounded transition rates is equivalent to a simpler discrete time Markov decision process, which is the case we consider. In the mentioned stochastic model, authors show a closed system of equation for the dynamics we are regarding, and after that they describe the optimal vaccination policy. They focus on the perspective of a particular individual immersed in the population, who will take the optimal decision in order to minimize her cost regarding the behaviour of the population correspondent to a global vaccination policy, which evolves according to a fixed vaccination strategy π_P. Since it is theoretically assumed a population with infinite individuals, the behaviour of this new individual will not affect the evolution of hole population.

It is shown that the optimal vaccination behaves as a bang-bang control, namely, the solution is to vaccinate with a higher rate at the beginning, and after some threshold, that depends on the connectivity of the network and the costs, not to vaccinate [6]. Consequently, the strategy indicates to get vaccinated, when the probability of infection is also high. After this phase, the rational individual will not get vaccinated. This can be explained by the fact that the risk of infection after the threshold is low since the disease has been almost eradicated, and is wise to potentially pay the cost of contracting the disease, which is less probable, than the cost of getting vaccinated.

1.4 Motivation

This type of epidemiological models are well studied in physics, biology and mathematics works, and we can also find papers of Machine Learning on epidemics like [8] or [9] about clinical decision making, but, as the best or our knowledge this is the first attempt to apply Reinforcement Learning techniques to an epidemic scenario. Consequently, we imple-

mented these algorithms in order to find the optimal vaccination policy simultaneously with the spread of a disease and compared the simulation results with the analytical results.

2 Source Code

In recent years, many tools have been developed to simulate reinforcement learning environments. Among the best known, it can be cite Open AI Gym [10]. However, to the best of our knowledge, there is no implementation of any epidemiological reinforcement learning environment. Therefore, we designed and implemented an open source epidemiological environment fully compatible with OpenAI Gym's toolkit and any numerical computation library, like TensorFlow [11] or PyTorch [12].

In this environment, the state of the agent is a tuple, where its first value is its disease state: S (Susceptible), I (Infected), R (Recovered) or V (Vaccinated); and its second value is the percentage of infected persons in the society (PI).

The transition probabilities between states are hyperparameters that the user can set its value. In each iteration, the agent can modify its vaccination rate, secondly could change its state depending on these hyperparameters and finally PI is actualised. The experiment finishes when the agent reach the recovered or the vaccinated state.

The agent learns to modifies its own vaccination rate in order to reduce the cost of getting infected, the cost of staying susceptible and the cost of getting vaccinated. The vaccination rate is also a function of the PI, since the agent can observe how the infection is spreading. On the other hand, the user provides to the society a policy as input. Consequently, an iterative simulation can be done setting the society's policy to the agent previous iteration policy. Running till convergence, an equilibrium can be found where neither the agent nor the society has anything to gain by changing only their own strategy.

The full code can be seen in our GitHub [13].

3 Results

In order to understand the underlying dynamics of the system and find the best agent's policy, we calculate the Q-function (the action-value function) using Monte Carlo Control Algorithm and Temporal Difference methods such as SARSA and Q-learning (described in the previous section).

As early stated, the agent's policy depends on the PI (the percentage of infected persons in the society). Intuitively, the lower the PI the fewer the chances of getting infected and, consequently, the vaccination rate policy should converge to a smaller value. On the contrary, the higher the PI, the bigger vaccination rate. Therefore, in our first experiment, we seek to find the best policy for a range of values of PI, which can be observed in Figure 1. This plot is in log scale in order to appreciate the transition. As it can be seen, bigger values of PI correspond to a bigger vaccination rate, and smaller values of PI correspond to smaller values of the vaccination rate. In particular, when the percentage of infected persons tends to 0, the rate is also 0 because there are no possibilities of getting infected. Three sections can be seen in this figure: a first one where the infected population does not present a real threat to the agent, so the vaccination rate is small; a transition section where the optimal policy goes from a small value to a bigger one; and a third section where the optimal vaccination rate stops increasing with PI and reach a maximum. This maximum value of the vaccination rate depends on the hyperparemetes of the environment β (recuperation rate), r (infection rate), ν (an environment parameter that sets a maximum value for the vaccination rate) and the costs of getting infected and getting vaccinated.

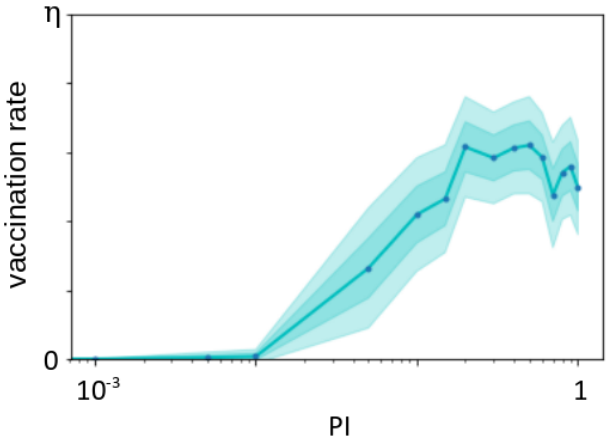


Figure 1: **Vaccination rate vs PI**: The average (over 1000 simulation) learnt vaccination rate policy of an agent in different environments, each with a constant percentage of infected persons. The horizontal axis is in log scale and the fill area correspond to the standard deviation.

In addition, it would be interesting to analyse the time dependency of the policy. For this experiment, we use an iterative method that learn the agent's policy. The society's policy is randomly ini-

tialise and, in each iteration, the society's policy is updated to the agent's best policy of the last iteration. This method is run until convergence, where a Nash equilibrium can be found [14] and the agent does not change its policy. Since in these experiments we allow Pi to change its value though time, we would like to observe if these results match the analytical results mention in the first section.

Interestingly, Reinforcement Learning results does not match the analytical results. In contrast, the vaccination rate of the best policy is constant through time and its value is between 0 and ν and depends on all the parameters that the user sets to the environment. In the machine learning approach, the agent has gone through the epidemic 15 million times (there are 0.5 million episodes in each iteration and it last 30 iterations till converges). Therefore, the agent already knows if the percentage of infected persons will grow and become a real threat or will decrease in time. The agent learnt the best policy for an epidemic and acts beforehand. For this reason, the agent policy is logically constant through time. On the other hand, the analytical approach does not consider this, and the agent does change its vaccination rate, since the PI and the probability of getting infected varies through time.

4 Discussions

Reinforcement Learning become a successful area of Machine Learning by solving complex tasks and has gained a lot of popularity in recent years. However, there are no previous works or publications that use Reinforcement Learning with an epidemiological environment or Reinforcement Learning with vaccination campaigns, to the best of our knowledge.

Consequently, we propose an open source epidemiological reinforcement learning environment implementation fully compatible with previous reinforcement learning tool-kits, such as Open AI Gym and other machine learning libraries like TensorFlow and PyTorch. This framework allows the user to simulate the spreading of a disease and find the best policy for an agent.

Moreover, first experiments with the environment shows that the agent efficiently learnt to affront a small cost and get vaccinated if there is a real threat of getting infected, which would lead to a bigger penalty. On the contrary, when the probability of getting infected is small, the agent does not get vaccinated in order to reduce the cost in the susceptible state.

Further analysis of the reinforcement learning approach exhibit a mismatch with analytical results. Considering the agent has experience multiple times the same epidemic, the agent has a deeper knowledge of the disease and knows what is going to happen. Therefore, the agent can learn to act beforehand, whereas, in the second scenario, the agent just react to the percentage of infected persons in the society and the probability of getting infected. Thereby, this new machine learning approach gives new and intuitive insight on mathematical modelling of infectious disease.

References

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

[2] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[3] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[4] I. Kiss, J. Miller, and P. Simon, *Mathematics of Epidemics on Networks: From Exact to Approximate Models*. Interdisciplinary Applied Mathematics, Springer International Publishing, 2017.

[5] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Rev. Mod. Phys.*, vol. 87, pp. 925–979, Aug 2015.

[6] I. Masaaki, "Optimal strategies for vaccination using the stochastic sirv model," *Transactions of the Institute of Systems, Control and Information Engineers*, vol. 25, no. 12, pp. 343–348, 2012.

[7] R. F. Serfozo, "An equivalence between continuous and discrete time markov decision processes," *Operations Research*, vol. 27, no. 3, pp. 616–620, 1979.

[8] J. Wiens and E. S. Shenoy, "Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology," *Clinical Infectious Diseases*, vol. 66, no. 1, pp. 149–153, 2017.

[9] S. M. Shortreed, E. Laber, D. J. Lizotte, T. S. Stroup, J. Pineau, and S. A. Murphy, "Informing sequential clinical decision-making through reinforcement learning: an empirical study," *Machine learning*, vol. 84, no. 1-2, pp. 109–136, 2011.

[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[11] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[12] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[13] F. Albanese and E. J. Ferreyra, "Epidemiological reinforcement learning environment." https://github.com/fedealbanese/Reinforcement_Learning_Environments.

[14] E. Maskin, "Nash equilibrium and welfare optimality," *The Review of Economic Studies*, vol. 66, no. 1, pp. 23–38, 1999.