Minimizing the Spread of Drug Resistant Malaria using Reinforcement Learning

Auguste Lalande

***Abstract*—Even today, malaria remains a major concern in much of the developing world. However, our ability to deal with the disease has significantly improved over the past decades. One the methods we use to control the spread of the disease is Intermittent Preventive Treatment. With this strategy, antimalarial drugs are administered to vulnerable individuals at regular intervals whether they are sick or not, which creates a negative feedback loop decreasing the prevalence of the disease even in untreated individuals. The downside is that this can lead to an increase in the spread of drug resistant strains of the parasite. It is therefore important to carefully choose the rate at which the drug is administered. This makes both an interesting and challenging reinforcement learning problem due to both a partially observable state, and the necessity for effective transfer learning. In this work, we create a gym environment based on equations developed by Teboh-Ewungkem et al to model the spread of both the resistant and sensitive strains of malaria in function of the preventive treatment rate. We then evaluate the performance of linear function approximator, and a shallow neural net in three distinct learning scenarios. Specifically, we evaluate the performance after offline training on environments with both identical and different parametrizations to the testing environment, and we also evaluate the effect of continued online training on the testing environment. Although our results are limited, we show that both transfer learning, and online learning are possible in the environment.**

I. Introduction

In 2018, malaria remains a major concern in much of Sub-Saharan Africa, Asia, and Latin America, with an estimated 216 million infections, leading to 445 thousand deaths in 2016 [1]. Nevertheless, over the past decades there has been a net decrease in the prevalence of the disease effected by better education and control strategies. One such strategy is the use of Intermittent Preventive Treatment (IPT).

IPT is a malaria control strategy in which a full therapeutic course of antimalarial drugs are administered to vulnerable asymptomatic individuals at regular intervals [2]. This leads to a decrease in the number a human hosts for the malaria parasite, and since the malaria cycle requires both a mosquito and human host to develop [3], this in turn leads to a decrease in the number infected mosquitoes. This creates a negative feedback loop decreasing the prevalence of malaria even in untreated individuals. The downside of such a strategy, is that similarly to the over-prescription of antibiotics this can lead to an increase in the drug resistant form of the disease [4]. This effect can be minimized by carefully tuning the IPT rate.

In this work, we build a reinforcement learning environment based on previous work by Teboh-Ewungkem et al [5] which models the impact of IPT on the prevalence of both a sensitive, and resistant strain of the disease [1]. We explore the effectiveness of two simple reinforcement models, in three scenarios of interest. Specifically, we investigate whether transfer learning is possible by comparing the performance of the models when they are trained on environments with the same parametrization as the testing environment or with different parametrizations. We also investigate the effectiveness of online learning, continuing the training of the models at test time. We find that both tranfer learning and online learning are possible to some extent in the environment.

II. Background

A. Learning Environment

The model used in this work is based on the equations developed in [5]. In their work Teboh-Ewungkem et al. present the necessary differential equations to model the prominence in a population of the sensitive, and resistant strains of malaria over time. Their equations are a function of 25 hidden parameters and 11 partially observable state variables. The rest of this section gives a broad overview of the environment as used for training a reinforcement learning model.

1) State space: One of the factors that makes this environment both difficult and interesting, is that the state space is only partially observable. As can be seen in Table I the model dynamics are dependent on both the symptomatic and asymptomatic infections in the population, however in a realistic setting it may be impractical to detect asymptomatic infections which would require running a blood test on all the population. Additionally, the model differentiates between the sensitive and resistant strains of the disease. However, the difference may not be immediately observable in reality, as it only becomes apparent several days after a treatment has been administered. This leads to a compressed observation space–shown in Table II–in which no difference is made between the sensitive and resistant strains of the disease, or between healthy and asymptomatic individuals.

2) Action space: The action space is limited to choosing a single continuous value at each time step. Specifically, an agent should choose the rate of administration of the IPT. To conform to practical limitations, we limit rate to be between 0 and 0.1 treatments per person per day. Additionally, we limit the agent to changing the rate only once every 50 days.

3) Reward: The reward follows naturally from the goal of the model, which is to minimize the burden of the disease and to balance the prevalence of the sensitive and resistant

[1] https://github.com/augustelalande/gym-mosquitoes

TABLE I
LATENT STATE VARIABLES

Variable	Description of Variable
S	Susceptible population
Is	Infected population (sensitive strain) (symptomatic)
I_a	Infected population (sensitive strain) (asymptomatic)
J_s	Infected population (resistant strain) (symptomatic)
J_a	Infected population (resistant strain) (asymptomatic)
T_s	Treated population (infected symptomatic)
T	IPT treated population (not-infected)
T_a	IPT treated population (infected asymptomatic)
R	Temporarily immune population

TABLE II
OBSERVABLE STATE VARIABLES

Variable	Description of Variable
$I_s + J_s$	Infected population (symptomatic)
T_s	Treated population (symptomatic)
$T + T_a$	IPT Treated population (asymptomatic)
$S + I_a + J_a + R$	Asymptomatic other

strains of malaria through the use of IPT. The reward is therefor simply a penalty for the proportion of the population symptomatically infected with either strain of the disease. Asymptomatic infections are excluded because they are not practically detectable, and because they do not explicitly contribute to the burden of the disease. Explicitly:

$$R = -(I_s + J_s) \tag{1}$$

4) Latent parameters: Like many other environments, the dynamics of this environment are controlled by a series of latent parameters which may not be known to the agent. If these parameters were known at train time, then finding the optimal IPT rate would be a problem of mathematical optimization, and there would be no need for reinforcement learning. However, due to their hidden nature, and because they may be costly to measure, developing a model which can behave optimally simply by observing the variables in Table II is a worthwhile endeavor.

III. METHODS

This section describes the methods used to obtain baseline results in the environment.

A. Continuous Action Space

One of the difficulties of this environment is its continuous action space. Although some methods have been developed to handle these environments–such as Continuous Actor Critic Learning Automatons [6] or the work on Fitted Q-iteration in continuous action space [7]–common practice is still to bypass the issue by discretizing the actions. For both of the following methods, the action space is split into 20 discrete actions at uniform intervals in the range 0 to 0.1.

B. Linear Function Approximation

In addition to having a continuous action space, this environment also has a continuous state space. The simplest method for handling such a space is to use a linear function approximator. With this model, the state-action value function is approximated by the dot product of the input state ($\mathbf{s}$) with some learned weight vector $\mathbf{w}$. Where a different weight vector is learned for each action.

$$Q(\mathbf{s}, a) = \mathbf{s} \cdot \mathbf{w}_a \tag{2}$$

C. Non-Linear Function Approximation

It may be the case that the linear function approximator presented above does not have the capacity to properly model our environment. As an alternative we propose to use a non-linear approximator. In this case we use a shallow neural network with a single hidden layer (20 units), followed by rectifier units. The output layer also has 20 units corresponding to the expected return of each of the actions when taken from the current state.

1) Challenges: Training a non-linear function approximator presents several challenges which may cause it to be unstable or even divergent when trained in a reinforcement learning environment. We follow the recommendations presented in [8] to avoid these issues. First, we utilize experience replay which randomly samples training batches from past data, removing the correlations in the observation sequence. Second, we use two different networks for training and for generating the learning targets, and only periodically update the target network, which reduces the correlations in targets.

D. Feature Space

Finally, to give the model a better chance to learn the latent information behind the environment, we use a feature representation of the state instead of the raw state space. Specifically, we concatenate the current state with previous state, with the intuition that the model should be able to extract environment dynamics by analyzing the transition between states.

E. Training

Both models were trained under the same conditions. First, a memory buffer was initialized with $1,000$ (state, action, reward, next state) tuples using a random policy. Then the models acted with an epsilon greedy policy ($\epsilon = 0.1$) and were trained with randomly sampled batches (batch size = 32) from the memory buffer. New experiences were added to the buffer at each step. The models were trained for 100 episodes, with the episodes being reset after $10,000$ time steps (note as previously specified the models chose a new action every 50 time-step). As specified above, the target model was only updated every 50 updates. Table III summarizes the hyper-parameters used in training.

IV. EVALUATION

In this section we evaluate the performance of the methods presented in Section III in three scenarios: offline training with the same parametrization of the environment, offline training with different parametrizations, and online training.

TABLE III
LIST OF HYPER-PARAMETERS USED IN TRAINING

Hyper-parameter	Value
Batch Size	32
Memory Buffer Size	1000
ϵ	0.1
γ	0.9
Number of Training Episodes	100
Episode Reset Step	10,000
Target Update Freq	50

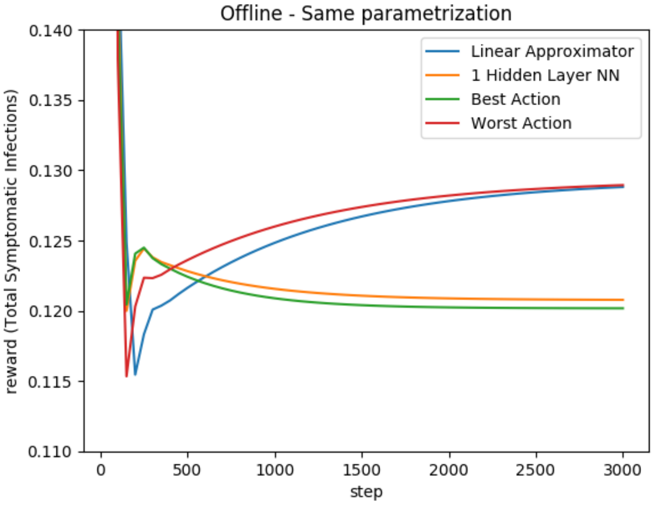

Fig. 1. Model performance when trained offline on test environment

A. *Offline Training (Same Parametrization)*

The simplest scenario for learning is one where the target environment is readily available. In this section we train our models on an environment with parameters chosen for stability (the necessity for this choice is discussed in Section IV-D) and evaluate their performance on the same environment. The results at test time are presented in Figure 1. In addition, to the performance of the two models we also plot the results of using the best and worst constant action policies (evaluated empirically). While the single layer neural net approaches the performance of the best action, the linear approximator is closer to the worst action.

Given that both models converge to a single action after some number of steps it should be possible for the single layer neural net to learn the best action. One possible explanation for this shortcoming is premature termination of training. However, it also possible that the optimal action cannot be learned because it would acting constantly for many time steps to observe its convergence value, and this may not be possible due to the ϵ-greedy action selection. The poor performance of the linear approximator could easily be attributed to a lack of representation capacity, however, it is more likely a result of poor hyper-parameter tuning, or a bad training regimen.

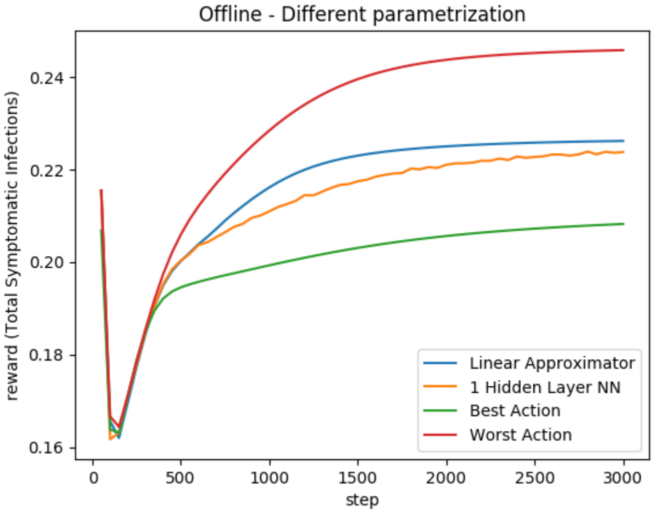

Fig. 2. Model performance when trained offline on different environments

B. *Offline Training (Different Parametrization)*

In this section we train the models on randomly parametrized environments and evaluate their performance on previously unseen parametrizations. We average the testing results over 50 test environments. This training scenario is more readily applicable to real environments since it may not always be possible to correctly model the application domain.

The results of the experiment are presented in Figure 2. Again we also plot the best and worst possible actions evaluated empirically for each test environment. In this case, the single layer neural net still outperforms the linear approximator; although it is further away from the best action than in the previous scenario. The linear approximator performs better than the worst action which contrasts with the previous results. This also supports the claim that its poor performance was due to the specifics of the training conditions, since under new conditions its performance improves. Nevertheless, it seems that some transfer learning is possible for both models.

C. *Online Training*

Finally, we evaluate the results of online training on our models. For comparison purposes we test our models on the same testing environments as the ones used in Section IV-B. However, at test time we continue to let the models learn from their experience.

The results of the experiment are shown in Figure 3. We note a small improvement in the performance of the linear function approximator and an even smaller improvement in the performance of the single layer neural net. Presumably, due to its smaller size the linear approximator is better equipped to learn from small amounts of data. Regardless, the conclusion of the experiment is that some online learning is possible in both scenarios.

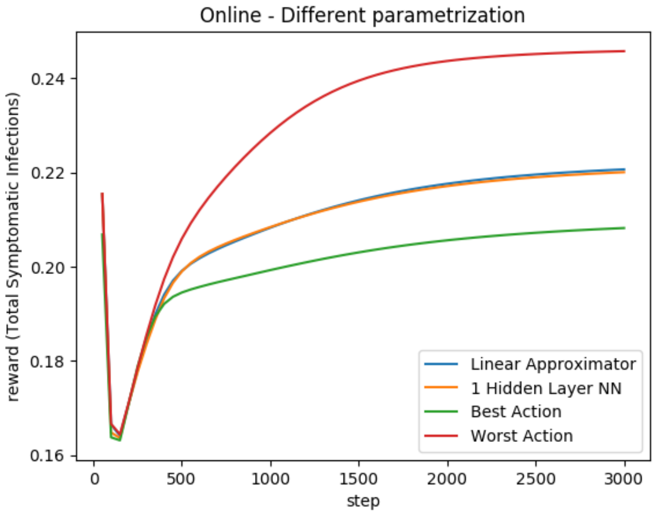

Fig. 3. Model performance when trained online, after being trained offline on a different environment

D. Environment Instability

Finally, we discuss the stability of the learning environment and its effect on the experiments. While experimenting, it was found that many of the randomly parametrized environments were unstable, and would come to be dominated by the resistant form of the parasite even without any IPT. This is partially due to the correlation between some of the hidden environment parameters, which was not taken into account when randomly assigning values. However, even small modifications of some of the parameters from a stable environment could lead to instabilities. This puts into question the realism of the model itself, as any action in such a model could have dramatic consequences, leading to the best action being no-action simply out of precaution, which contrasts with what is observed in the real world (i.e. even with IPT resistant malaria has not become ulta-dominant).

V. Conclusion

In this work, we presented a new reinforcement learning environment which models the prevalence of drug resistant malaria as a function of the use of IPT. We created a publicly available gym environment, to allowed continued research on the problem. We also presented baseline results on the environment, evaluating the use of two models: linear approximator and single layer neural net, in three different application scenarios. These scenarios explore the use of transfer learning and online learning which may be necessary when the latent parameters behind real life environment cannot be obtained. We demonstrated that although the performance of the algorithms was limited, some transfer learning and online learning can be achieved in the environment. Nevertheless there is much room for improvement.

As discussed in the evaluation, the models suffer from a lack of adequate tuning which may have contributed to their underwhelming performance. It follows, that the simplest way to improve the current results is to perform additional tuning experiments. Similarly, the number and variety of models explored in this work is limited to just two function approximators, but the number of models in the reinforcement learning literature is rich and diverse leaving much room for additional experiments. As a first step the use of actor-critic methods should be explored [9]. A more subtle improvement, may also be to changed the feature space (for example concatenating the past 4 states instead of 2), since the model may be too complex to be understood from a single state transition.

An entire different approach may also be to expand the action space. Whereas, the agent is currently limited to controlling the IPT rate, it may also be useful to control the administration of the disease to symptomatic patients as well, since this also contributes to the propagation of drug resistant malaria. We note that there may be some ethical concerns in exploring this option.

Finally, improvements to the environment itself may also be necessary. Although, the environment dynamics are based on previous work on the subject, that work itself made some simplifying assumptions which may not hold in reality, such as assuming a constant population of humans and mosquitoes. Through experimentation, and in the original paper [5] it was noticed that the environment is highly sensitive to the choice of latent parameters and the choice of IPT rate. In fact, it is so sensitive as to create doubt about the viability of the IPT method in the first place, but due to IPT's success in practice the doubt is cast back on the environment itself. It may therefore be necessary to increase the realism of the environment.

References

[1] WHO, "World malaria report 2017," 2017.

[2] J. J. Aponte, D. Schellenberg, A. Egan, A. Breckenridge, I. Carneiro, J. Critchley, I. Danquah, A. Dodoo, R. Kobbe, B. Lell *et al.*, "Efficacy and safety of intermittent preventive treatment with sulfadoxine-pyrimethamine for malaria in african infants: a pooled analysis of six randomised, placebo-controlled trials," *The Lancet*, vol. 374, no. 9700, pp. 1533–1542, 2009.

[3] P. Schlagenhauf-Lawlor, *Travelers' malaria*. PMPH-USA, 2007.

[4] R. D. Gosling, M. E. Cairns, R. M. Chico, and D. Chandramohan, "Intermittent preventive treatment against malaria: an update," *Expert review of anti-infective therapy*, vol. 8, no. 5, pp. 589–606, 2010.

[5] M. I. Teboh-Ewungkem, O. Prosper, K. Gurski, C. A. Manore, A. Peace, and Z. Feng, "Intermittent preventive treatment (ipt) and the spread of drug resistant malaria," in *Applications of Dynamical Systems in Biology and Medicine*. Springer, 2015, pp. 197–233.

[6] H. Van Hasselt and M. A. Wiering, "Reinforcement learning in continuous action spaces," in *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*. IEEE, 2007, pp. 272–279.

[7] A. Antos, C. Szepesvári, and R. Munos, "Fitted q-iteration in continuous action-space mdps," in *Advances in neural information processing systems*, 2008, pp. 9–16.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

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