Proceedings of Machine Learning Research 123:253–261, 2020 NeurIPS 2019 Competition and Demonstration Track

A Global Health Gym Environment for RL Applications

Sekou L. Remy SEKOU@KE.IBM.COM
and **Oliver Bent** OLIVER.BENT2@IBM.COM
IBM Research|Africa
Nairobi, Kenya

Editors: Hugo Jair Escalante and Raia Hadsell

Abstract

This paper presents a platform for engaging in global health challenges, captured in the form of an OpenAI Gym styled environment. This platform has been used in three competitive challenges in 2019, and exposes a novel application domain for RL practitioners, along with the potential for significant social and scientific impact. While the platform has been demonstrated with problem formulations in global health, it is principally designed to facilitate general learning from simulation in a more abstract manner.

Keywords: Competition, Reinforcement Learning, Global Health, Malaria

1. Introduction

Global health is a domain which presents significant and pressing large scale challenges, any incremental improvement may have significant global benefit at scale. While at the same time, it may often be the case that we are not completely assessing all options which may lead to improved outcomes. In cases where the problems are complex and the range of potential interventions large, it is daunting to consider all the options, and accordingly, it's quite possible that good options go ignored. We leverage models developed by practitioners in the domain to perform *what-if* evaluations at scale. To make the problem tractable and repeatable, we capture output from the models into a form which preserves the stochasticity and intervention dimensionality of the domain. This forms the basis for an *environment* which can be used to benchmark the performance of various learning algorithms. The environment is an object which provides a surrogate for the complex domain model, and is a basis for learning the utility of performing given interventions at specific times. As an initial proof of concept, we selected malaria to be the area of focus for preliminary challenge activities, and we demonstrate how to harness data generated from mechanistic compartmental malaria models. Accordingly, in this work we describe activities based on this novel formulation for the systematic exploration of malaria intervention actions. This formulation is one which is well-suited for the application of Artificial Intelligence (AI) agents to learn the most effective intervention strategies for a specific environment.

In this work we present our platform and novel problem formulation, and we present how it was implemented as an OpenAI gym styled environment. In addition, we present a description of activities performed with this environment in 2019 as an example of a new class of Reinforcement Learning challenges. We conclude with a discussion of the evaluation framework, as well as a summary of the participation statistics observed that timeframe.

2. Background

Malaria is a mosquito-borne disease which is endemic in many countries. It was once prevalent over a much wider region of the globe, but there has been significant progress in the prevention and control of the disease resulting in a reduction of the mortality rate and the number of new cases. Many countries in SSA rely heavily on external funding for malaria prevention and control, while in recent years investments have started to level-off (see Winskill et al., 2011). This means that policy makers will have more difficult decisions to ensure continued success in the management of the disease given the available resources. Individual distributed decision makers (e.g., NGOs, governments and charities) must be able to explore the possible set of actions for appropriate malaria interventions within their populations. Such policies include a mix of actions like the distribution of long-lasting insecticide-treated nets (ITNs), indoor residual spraying (IRS), vector larvicide in bodies of water, and malaria vaccinations. The space of possible policies for malaria interventions is daunting and inefficient for human decision makers to explore without adequate decision support tools. We assert that algorithms from many sub-domains of AI are well suited to provide core elements of such tools. To demonstrate the value that Machine Learning can provide, and to present a new, non-trivial problem domain to the AI community, we formulated this decision support process as a challenge or competition activity.

Competition is a theme tightly woven into the advancement and demonstration of AI. Over the past four decades, machine intelligence in games has been used as a key strategy to demonstrate advances in AI. From games like checkers and chess, to more recently Backgammon, Shogi, Jepoardy, and Go, superhuman performance has been demonstrated, and very often this arises by facilitating competition between AI and other AI entrants and human experts. These competitions have extended well beyond games and some of them provide significant funding/investment to raise awareness and stimulate immediate and further research about their respective topics. More recently, there is also an emerging class of problems presented through dedicated Machine Learning Competition Platforms such as Kaggle, CodaLab, Zindi, and Hexagon-ML, inviting a global community of researchers and practitioners to engage in problems.

The AI sub-discipline of Reinforcement Learning (RL) is developing a rich history in health care. It has been applied to optimizing selection of therapies in diseases, tailoring drugs to control side effects, and determining the best approach to managing care (see Gottesman et al., 2019; Yu et al., 2019). According to Whiteson et al. (2010), the first RL challenges were launched in 2006 and were created out of the necessity in the research community to agree upon benchmarks and common environments to support result generation, sharing, and publication. This framing of problems has harnessed a competitive edge to the demonstration and publication of research in the domain, especially in Deep Reinforcement Learning where the Atari Arcade Environment served in parallel to the demonstration of Deep Learning on benchmark datasets like Imagenet. In addition to creating a focus on certain problems, competitions also provide a way to perform unbiased comparisons of RL algorithms. This has turned out to be incredibly important for modern RL research, as ironically direct comparisons between different RL algorithms are inherently difficult to perform since authors of these algorithms often spend a lot of time tuning them to work well in specific types of domains or environments.

While RL is a domain where algorithms are by design supposed to learn in an active manner from zero experience, such tuning can make it difficult to objectively evaluate different contributions. To provide reproducible, reusable, and robust assessment of RL algorithms, simulation environments provide an important role. Whether games or physical systems approximated through computational models, these environments permit RL to borrow a page from traditional Machine Learning evaluation, while compensating for the interactive nature of learning in RL. Specifically, instead of using training, validation, and testing datasets which are common in other Machine Learning disciplines, in RL one can consider the use of training and testing environments. An RL competitor may decide how happy they are with their approach, specifying how much training must occur prior to submitting their approach for evaluation through a test environment. Access to a training environment may be regulated if challenge organizers deem that sample efficiency is a dimension upon which they'd like to assess the RL solution, or simply wishing to democratize submissions to reasonable computational constraints.

Stepping back, in this work we consider the application of RL in the context of global health. The desired aims of the application of RL will thus impact the future health of individuals or communities. While we consider this work as a competition, the context is not taken lightly. Where the existing healthcare applications of RL differ from what we propose in this work, is that the algorithms are learning from models (e.g. for malaria) and not directly from the real world. This means that we avoid manipulating intervention programs for countries in real time, but instead provide a framework to permit learning from domain models and comparing the recommendations provided by RL. We envision that stakeholders will be able to benefit from the attention which is driven by competition, and when trusted domain models are used, the resulting recommendations would be more readily consumed.

3. Approach

In this work we implement the gym paradigm to provide a common interface for development and testing of this new environment. Due to pioneering projects like the OpenAI Gym introduced by Brockman et al. (2016), this paradigm is now well established in the Reinforcement Learning community and provides a ready entry point for newcomers to the problem formulation. Our environment[1] is novel in that it features a client-server architecture which supports sharing of resource intensive elements, as well as providing a mechanism to support transparency and trust. These elements are outside the scope of this manuscript and will not be emphasized further.

3.1. Environment Webservice

To provide access to the environment in a scalable manner we designed a client-server architecture. The web service is written as a modular codebase, and is based on two Python projects *flask* and *blueprint* to provide an extensible platform to concurrently host multiple environments. Each subdirectory contains the code and/or data required to implement the

1. https://github.com/IBM/ushiriki-policy-engine-library

environment, with only a one-line in the base web service code to import that environment object into the common namespace.

Ultimately, the environment web service can be of one of two forms: The first uses a functional representation of the environment defined for the activity, the second loads the parameters defining the environment from disk storage. The contents of this file could be generated by sampling a particular simulation environment at scale, thereby creating enough data to provide a meaningful surrogate of the original environment. We have developed an infrastructure flexible enough for choice of file format for this data to be determined by the environment developer. In either case, the surrogate then becomes a complete representative environment which can be shared with multiple participants providing a single source of ground truth information about the environment which is frozen in time, yet not simplistic. For the challenges we have run in 2019, both styles have been used for different phases.

3.2. Environment API

The environment is coupled with a language specific API which implements a client library for participants to access methods which evaluate an intervention or a sequence of interventions (i.e. a policy). These methods are linked to the step method standard within gym environments. A reward (a scalar value) is provided as a result for either the intervention or the policy. This reward value contains the signal which enables a participant's algorithm to learn the utility in performing certain interventions as they engage in the challenge activity. The API is designed to integrate with an agent class implemented by the end-user allowing their chosen RL algorithm to interface with the competition environments. The principle functionality of an environment is to produce observations and representative transitions based on actions as input from external agents, along with providing a scalar reward signal to these agents. Mapping the required functions for an environment class to our underlying surrogate models in the following manner:

```
observation, reward, done, _ = env.step(env_action)
```

Where the *reward* is as defined previously, the *observation* is the environment's next state based on the evaluated action, and *done* is a boolean value denoting that the terminal state has been reached in the environment. The info value remains unassigned, in the Python nomenclature this is assigned to "_". The first API was implemented in Python, however leveraging the *reticulate* package, it has been used to support R language challenge activities as well. In the future we envision implementations in multiple languages natively. The web service architecture will make such libraries easy to implement and to support as HTTP functionality is pervasive in modern programming languages.

4. Challenges and Competitions

In 2019, culminating with the NeurIPS conference, a series of similarly themed events were performed at conferences with different audiences. The challenges have been framed in the same manner, where participants have been asked to write an algorithm which is given a fixed number of samples to engage with the environment and learn the best policy (the

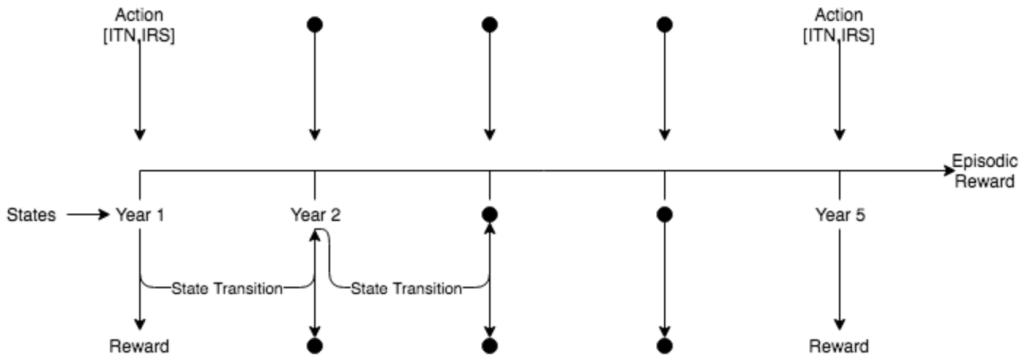

Figure 1: Sequential decision making task competition framing. Actions may be selected each year receiving a reward value or an episodic reward is returned for a complete policy.

sequence of interventions). Each intervention has its own behavior and parameterization in the model as they act on individual simulated humans in the environment.

In the framing, interventions are defined as the coverage of indoor residual spraying (IRS) and distribution of long-lasting or insecticide treated bednets (ITN) at given times in a five year period. The state of the environment changes as a result of the interventions which have been performed, as well as the dynamics which have been implemented. A state signal from the underlying environment is presented as a variable which the algorithm may learn a policy. For the challenges hosted in 2019, time was selected as a common approximation to simulated latent state. Accordingly, the reward derived from interventions were learned as a function of the interventions selected, and the time in which they were selected (See Figure 1). The participant's code was given a fixed compute budget of 2000 samples from the environment which could be used for learning. The problem is framed at a level we assert is accessible to the community. With several possible approaches to the problem, its framing will remain deliberately non-prescriptive. Also, due to our API abstraction for tackling the problem in a machine learning guise, little domain knowledge is required of malaria transmission and control, or its simulation (which is left to the competitor's interests).

Because of the selection of the client-server approach to access the environment and subsequently accessing the remotely hosted environments, it is not considered a distinct advantage to have considerable local compute to run the models in a competitor's considered algorithm. Such model complexity would be discouraged by design as it is unlikely a competitor can generate enough data in the competition time frame to train such a model, with each submission taking hours to run. As all models are learning from surrogates of physics-based simulators developed by epidemiological modelers, the results found should be interpretable by the malaria modeling community. The complexity of a particular strategy should also be confined to real-world feasibility, which is ensured through constraints on timing, cost and number of actions which may be performed. All of which for simplicity were contained in the evaluation metrics, to generate a reward signal. To date, the malaria

scientific community has no clear answer for an optimal strategy in a particular location. We believe such a strategy may efficiently be found through computationally intelligent solutions generated through challenges like ours. As such submitted algorithms may act as a black-box in terms of complexity with regards to this framing. Highly performing algorithms produce results which outperform other strategies in simulation. In continuation from the challenge, these results may be subsequently validated by epidemiological modelers with the long term focus on implementation of the results.

The data used for these competitions is stationary based on the state observation with stochastic returns. Also, due to the high dimensionality of combinations of actions which can comprise an intervention strategy, the possible set of solutions can not be exhaustively searched, so there is little concern for over-fitting. Specifically, competitors will have to grapple with the paradigm of exploration and exploitation of high performing strategies under limited resources in terms of both compute and time. We hope that given the stochastic multi-dimensional nature of the problem, this will discourage over-tuning of any particular model for a simulator due to the noisy results returned.

4.1. NeurIPS and Deep Learning Indaba

Framed as a live, multi-day activity, this challenge format was executed as a single phase event with data from an unidentified mechanistic malaria model used to provide the input loaded by the web service. While synthetic data was used to bootstrap the malaria model for this activity, the model output was a realistic representation of the problem domain. The rewards used for the activity were the parasite prevalence, and a measure of cost effectiveness of the considered intervention policies.

4.2. KDDCup

This activity was executed as a three-phased, four month long activity, featuring multiple distinct environments as well as a single blind submission. During the first two phases, the environments were used for both training and evaluation on a shared leaderboard, while in the final prove phase, submissions were evaluated on an environment unseen by the participants. The reward used for the activity was only the measure of cost effectiveness of the considered intervention policies.

5. Evaluation Framework

Each submission is evaluated based on 10 runs, with each run consisting of multiple episodes, and the greedy policy learnt over a run is submitted along with the episodic reward. This is indicative of the limited experience for any algorithm to learn a high performing policy from direct query of the underlying simulation models. This framing of the problem is novel compared of existing RL benchmarks, which do not so greatly penalize the number of evaluations. There is no sharing of information between runs, so every submission in learning from zero experience during each run. In the live challenge format, only a single environment was used, and participants scores were determined from CSV files generated by their algorithms. In the multi-phase format, in all but that last phase participants

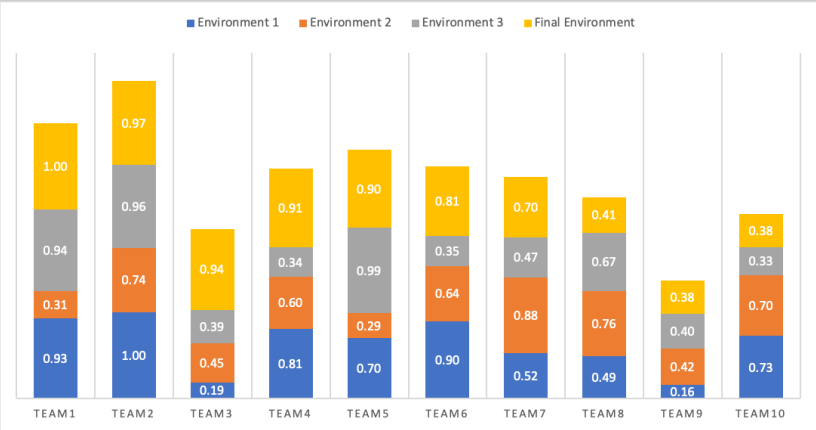

Figure 2: Normalized scores of top 10 teams in multiple environments for one of the 2019 Challenges. The final environment determined the team ranking. The top two teams were consistently high performers, while Team3 improved considerably.

submitted the same type of CSV file. In this last phase, participants submitted a zip file containing their algorithm code for an independent assessment on remote machines.

Code submissions were reasonably constrained to a list of standard Python libraries which could be preinstalled on the infrastructure, limiting the requirement of the machines to be connected to the internet for package installs. If submitted code could not be executed, the team was contacted. If minor remediation or sufficient information was not provided in order to run the code, the submission was ultimately removed, though in 2019 this was only the case for a single submission. The reported score is then the median episodic rewards from the 10 greedy policies learnt over the 10 runs. Figure 2 presents an assessment of the top teams which competed in the multi-phase competition performed via this process.

6. Example Code

This snippet gives an example of the use of an environment in which each *step* evaluates an entire policy. This code demonstrates a very simple version of the type of Python class which is expected to be developed by the competitor. There is another variant of the environment which permits individual actions to be evaluated.

```
import gym
from ushiriki_policy_engine_library.EvaluateSubmission import
    EvaluateAugmentedChallengeGymSubmission

class CustomAgent:
    def __init__(self, environment):
        self.environment = environment
    def generate(self):
        candidates, rewards = [], []
```

```
        # An agent using 10 episodes in each training run
        for i in range(10):
            self.environment.reset()
            policy = {}
            for j in self.environment.observation_space:
                policy[j]=self.environment.action_space.sample()
            _, reward, _, _ = self.environment.step(policy)
            candidates.append(policy)
            rewards.append(sum(reward))

        best_policy = candidates[np.argmax(rewards)]
        best_reward = rewards[np.argmax(rewards)]
        return best_policy, best_reward

env = gym.make("ushiriki_policy_engine_library:ChallengePolicy-v0",
    userID="61122946-1832-11ea-8d71-362b9e155667")
EvaluateAugmentedChallengeGymSubmission(env, CustomAgent, "testsubmission.csv")
```

7. Conclusion

In the health domain, competitions such as Learning to Run and AI for Prosthetics created by the Stanford Neuromuscular Biomechanics Lab, promote the creation of a controller that enables a physiologically-based human model to navigate complex obstacles within a short duration of time and simulate changes in gait that occur after receiving a prosthesis in order to develop approaches that may be generalized to health conditions impacting movement. The Personalized Medicine MSK challenge related to the task of classifying clinically actionable genetic mutations, that is given the genetic mutations of a cancer tumor classify the mutations that accelerate tumor growth from the neutral mutations. In contrast, we developed a novel competition applying machine learning to the global challenge of malaria elimination. Using agent-based models to learn from simulation and demonstrate novel strategies for the elimination of malaria in an endemic location. We believe this challenge builds upon interest across the machine learning community for interpretable, scalable, and repeatable models which solve real world challenges. With higher level motivation provided by the promise of machine learning assisting in the elimination of a disease which for centuries has been missing the medical breakthrough to achieve this goal. Such a challenge applying machine learning to epidemiological models specifically malaria has never been run, to the knowledge of the organizers. For example the KDD Cup alone attracted 248 teams, 296 Competitors and a total 735 Submissions. This applied form of machine learning treating such models as a simulator is particularly novel.

Acknowledgments

We appreciate the generous sponsorship provided by the ACM to contribute to the prizes used for the 2019 KDDCup. We are also indebted to Zindi Data Science Platform for working with us on the Deep Learning Indaba Hackathon Track. And finally, to Hexagon-ML for hosting both the KDDCup activity as well as the NeurIPS Live Challenge.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nat Med*, 25(1):16–18, 2019.

Shimon Whiteson, Brian Tanner, and Adam White. The reinforcement learning competitions. *AI Magazine*, 31(2):81–94, 2010. ISSN 07384602.

Peter Winskill, Mark Rowland, George Mtove, Robert C Malima, and Matthew J Kirby. Malaria risk factors in north-east Tanzania. *Malaria Journal*, 10(1):98, 2011. ISSN 1475-2875. doi: 10.1186/1475-2875-10-98. URL http://malariajournal.biomedcentral.com/articles/10.1186/1475-2875-10-98.

Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: a survey. *arXiv preprint arXiv:1908.08796*, 2019.