Off-Policy Knowledge Maintenance for Robots

Joseph Modayil, Patrick M. Pilarski, Adam White, Thomas Degris, and Richard S. Sutton
Dept. of Computing Science, University of Alberta, Edmonton, Canada; Email: jmodayil@cs.ualberta.ca

***Abstract*— A fundamental difficulty in robotics arises from changes in the experienced environment—periods when the robot's current situation differs from past experience. We present an architecture whereby many independent reinforcement learning agents (or *demons*) observe the behaviour of a single robot. Each demon learns one piece of world knowledge represented with a generalized value function. This architecture allows the demons to update their knowledge online and off-policy from the robot's behaviour. We present one approach to active exploration using curiosity—an internal measure of learning progress—and conclude with a preliminary result showing how a robot can adapt its prediction of the time needed to come to a full stop.**

I. Introduction

Adapting to unanticipated situations presents a considerable challenge for robots. When robots are trained extensively in one particular environment or distribution of states (for example, driving on a hardwood floor), many traditional learning and control approaches will fail to explore and update their world knowledge when faced with a new distribution—e.g. driving on thick carpet, or in more extreme situations, ice or mud. The key challenge is to detect that modelling errors exist and to learn about the change. One insight is that to detect modelling errors, the robot's knowledge must be in a form that is *verifiable*, so that the robot can test the knowledge directly from experience. When the knowledge has errors *and the robot can learn about it*, then it is worthwhile for the robot to spend time learning about a new phenomenon.

There are two common approaches for generating models. One approach generates models in the lab, carefully describing all expected phenomena. When this approach is feasible, it will often result in models that are overly cautious, and these models still will not adapt to unanticipated changes. Another approach is to assume that the expected distribution is well-modelled by the robot's initial experience or training distribution. A dataset is gathered for the training distribution, with the hope that all the real-world experience will match the training distribution. These approaches will work to a point, but will often fail dramatically when environmental conditions change, for example when the robot veers off a paved road and encounters a new kind of terrain (dirt and grass). Standard methods, such as on-policy reinforcement learning, attempt to correct the behaviour but do not acquire knowledge about other policies on this new road condition.

We propose off-policy knowledge maintenance as a viable approach to this problem of learning about (and adapting to) changing environmental conditions. It would be advantageous for a robot to acquire knowledge about a new environment while still performing and learning about its original behaviour. Additional benefits might arise if some measure of off-policy learning progress was used to influence behaviour in a way that causes the robot to explore changing dynamics—in essence, becoming "curious" about its new environment. The notion that curiosity-like behaviour of this kind arises from rewarding learning progress—and can be used to shape learning—has a rich history[1, 2, 3] and has been applied successfully to a robotic task [4]. Curiosity will reward actions that induce changes to the learned knowledge. In some sense, this is akin to *detecting* that the current knowledge has errors, and *responding* by performing the same action in similar circumstances to learn more about it.

II. Formulation

Using the recently introduced GQ(λ) reinforcement learning algorithm of Maei et al. [5], it is now possible for an agent to behave using one policy and at the same time update its knowledge about other policies. The GQ(λ) algorithm is online, off-policy, stable under function approximation, and efficient with a time complexity that is linear in the size of the representation. It is therefore well suited to agents that must adapt and explore complex real-world environments, where function approximation is required to make tractable the essentially infinite variability of real-world dynamics.

Using the GQ(λ) algorithm as a framework, we propose that learned world knowledge can be effectively represented as answers to questions about following some policy. For example, one question could be "If I slammed on the brakes, then how long will it be before I come to a complete stop?" The answer is learned in the form of a generalized value function, represented as a linear function of the state representation. As the learning algorithm is off-policy, the system can learn answers even when the robot's behaviour does not match the exact sequence of actions (policy) being explored by the question.

The learning architecture presented in this work has the robot's single sensory-motor stream being used to train multiple learners (termed "demons") in parallel. Each demon is an independent reinforcement learning agent updating a generalized value function that estimates the expected return from following a policy π. The return comes from an instantaneous reward function r, continuing at each time step with probability γ until termination, and finally collecting the outcome reward of z. The question q asked by a demon is thus represented by a tuple: $q =< r, z, \gamma, \pi >$. This formulation provides a unified approach to modelling both discounted and episodic tasks.

The generalized value function computes the expected return from following π for a duration T governed by γ:

$$Q^q(s,a) = E[r_{t+1} + \ldots + r_{t+T} + z_{t+T} | s_t = s, a_t = a]$$

Each demon approximates its answer using a linear function, namely the dot product of a parameter vector w with the state-action feature representation $\phi(s,a)$.

$$Q^w(s,a) = w^{\mathsf{T}} \cdot \phi(s,a) \approx Q^q(s,a).$$

Thus the demon d can be represented in the form of a question and an answer: $d = <q, w>$. This representation allows many bits of knowledge to be represented, and the algorithms allow the internal models to updated online.

In this framework, a curiosity driven behaviour can detect the demons are learning and actively guide the behaviour to learn more. We define an instantaneous curiosity reward r_c^t for learning progress as the sum over all demons of changes in parameter weight w_d at time t (Eq. 1). This reward function can be used alone for curiosity, or it can be combined with external reward to provide a blended behaviour.

$$r_c^t = \sum_d ||w_d^t - w_d^{t-1}||. \tag{1}$$

III. Example

We describe a small but concrete example to demonstrate how an off-policy demon updates a robot's knowledge. We use a physical three-wheeled robot as our experimental platform. For simplicity, we define a single demon that is predicting the time required for the robot's wheels to come to a complete stop under current environmental conditions. Initially the robot is driving on wood, but at some point the robot is suspended in the air and the demon must update its predictions.

The robot has two possible actions, specified in terms of applied motor voltages: ROTATE at full speed and STOP. The robot's state is represented by a 16 bit vector, combining eight discretized velocity levels for one wheel (v_{wheel}) with a Boolean variable $m_{spinning}$ for the desired command (ROTATE or STOP). The desired command is provided by the environment. At each timestep, one state bit is active, indicating the current combination of desired mode and observed velocity.

The robot's behaviour is not influenced by the demon. The desired mode $m_{spinning}$ is set as a square wave, with periods of maximum and zero wheel velocity each lasting 5 seconds. The desired mode is used to define a task reward, and a learned robot behaviour quickly converges to the optimal policy of spinning and stopping according to the desired mode. We define a single demon that learns the number of time steps required for the wheels to come to a complete stop.

$$q = <r = 1, z = 0, \gamma = 1_{\{v_{wheel} \neq 0\}}, \pi = STOP> .$$

Note the demon's policy differs from the robot's behaviour policy. However, the behaviour regularly provides relevant experience for the demon to learn accurate predictions.

For the experiment, the value function for the demon was initialized to zero, and the robot was placed on a wood floor. As the robot followed its optimal task policy, the demon learned until value function convergence. The robot was then suspended in the air—where stopping characteristics are different due to greatly reduced friction and inertia—and run again until demon convergence. During the robot's task-based behaviour, the demon was maintaining its knowledge of the world, in this case updating its estimates of the time needed for this wheel to come to a complete stop from different internal states. The learned value function $Q^w(s, a)$ represents the demon's stopping time estimates. The time-varying progress of the value function entry for stopping from maximum wheel speed in its current environment is shown in Fig. 1. While this study was conducted for a single question, an off-policy demon framework allows many such questions to be answered in parallel, and a study of this will be presented in future work.

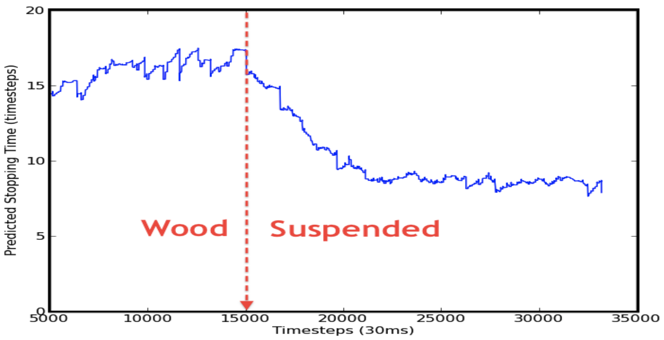

Fig. 1. Plot of the predicted stopping time for two different environments, *wood floor* and *suspended*, measured from the value function $Q^w(s, a)$. While the robot is following its behaviour policy, the off-policy learner updates its predictions about the time needed to come to a full stop. True times were empirically acquired, and agreed with the demon's estimates once converged.

Experiments with curiosity-based learning are ongoing, to explore how active learning can improve the rate of demon convergence. For this problem, there are behaviours that provide faster demon convergence (rapid alternation between ROTATE and STOP). We expect that curiosity should be able to find learning behaviours that are at least as efficient as these.

IV. Discussion

By using off-policy demons to maintain knowledge, the robot is continually learning about its current environment. Models that are defined in the lab and not subject to revision from experience are fundamentally not correctable by the system. A key aspect of this approach is that knowledge must be in a form that can be updated directly from the robot's experience. This requires that knowledge about the world is expressed in terms of a robot's subjective experience.

In summary, we present a new architecture that allows the robot to continually update its knowledge online using off-policy demons. We also describe a method whereby a robot can actively adapt its behaviour by incorporating a curiosity-based reward to examine those aspects of world that are changing and learnable. We expect these ideas will lead to flexible, continually adapting robot learning systems.

References

[1] J. Schmidhuber, "Curious model-building control systems," *1991 IEEE International Joint Conference on Neural Networks, 1991*, pp. 1458–1463, 1991.

[2] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically motivated reinforcement learning," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1281–1288.

[3] Ö. Şimşek and A. Barto, "An intrinsic reward mechanism for efficient exploration," *Proceedings of the 23rd international conference on Machine learning*, p. 840, 2006.

[4] P. Oudeyer, F. Kaplan, and V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[5] H. Maei and R. Sutton, "GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces," in *Artifical General Intelligence 2010*, 2010.