Reward-Machine-Guided, Self-Paced Reinforcement Learning

Extended Abstract

Cevahir Koprulu
The University of Texas at Austin
Austin, TX, USA
cevahir.koprulu@utexas.edu

Ufuk Topcu
The University of Texas at Austin
Austin, TX, USA
utopcu@utexas.edu

ABSTRACT

Self-paced reinforcement learning (RL) aims to improve the sample efficiency of RL by automatically creating sequences, i.e., *curricula*, of probability distributions over *contexts*. However, existing self-paced RL methods fail in tasks that involve temporally extended behaviors. As a remedy, we exploit prior knowledge about the underlying task structure and develop a self-paced RL algorithm guided by reward machines, i.e., a finite-state machine that encodes such structure. The proposed algorithm integrates reward machines in the updates of 1) the policy and value functions obtained by an RL algorithm, and 2) the automated curriculum that generates context distributions. Our empirical results evidence that the proposed algorithm achieves optimal behavior in cases where existing methods fail, and also reduces curriculum length and variance.

KEYWORDS

Reinforcement Learning; Curriculum Learning; Reward Machines

ACM Reference Format:
Cevahir Koprulu and Ufuk Topcu. 2023. Reward-Machine-Guided, Self-Paced Reinforcement Learning: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

The design of task sequences, i.e., curricula, aims to increase the data efficiency of reinforcement learning (RL) by beginning with *easier* tasks and gradually increasing the difficulty [8]. To replace manual curriculum design, Klink et al. [6] develop *self-paced RL*, which automatically creates a sequence of probability distributions over *contexts* [3], parameterizing dynamics, reward function, and initial state distribution of an environment. Although self-paced RL [5] outperforms the state-of-the-art curriculum learning methods [2, 10], existing self-paced RL approaches work poorly in long-horizon planning tasks with non-Markovian reward functions. A remedy is to exploit high-level structural relationships [9, 11], e.g., via a type of finite-state machine, called *reward machines* (RMs) [4].

We study self-paced RL for long-horizon planning tasks in which RMs are available a priori to the agent. Our contribution is threefold. 1) We propose an *intermediate self-paced RL* algorithm that updates the policy and value functions via an RM. 2) We establish a *reward-machine-context mapping* that, given a transition in the RM, outputs the smallest set of context parameters, that determine if a high-level event occurs. 3) We develop a *reward-machine-guided, self-paced RL* algorithm that extends intermediate self-paced RL by navigating curricula via the mapping (see Figure 1). Our empirical results show that proposed methods accomplish tasks in our use cases, whereas state-of-the-art methods fail. Guiding curriculum generation also avoids inefficient exploration of curriculum space.

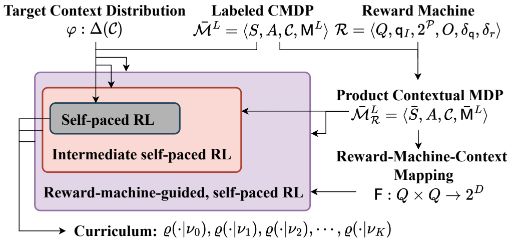
Target Context Distribution
$\varphi : \Delta(\mathcal{C})$
Labeled CMDP
$\bar{\mathcal{M}}^L = \langle S, A, \mathcal{C}, \mathsf{M}^L \rangle$
Reward Machine
$\mathcal{R} = \langle Q, \mathsf{q}_I, 2^{\mathcal{P}}, O, \delta_\mathsf{q}, \delta_r \rangle$
Self-paced RL
Intermediate self-paced RL
Reward-machine-guided, self-paced RL
Product Contextual MDP
$\bar{\mathcal{M}}^L_{\mathcal{R}} = \langle \bar{S}, A, \mathcal{C}, \bar{\mathsf{M}}^L \rangle$
Reward-Machine-Context Mapping
$\mathsf{F} : Q \times Q \to 2^D$
Curriculum: $\varrho(\cdot|\nu_0), \varrho(\cdot|\nu_1), \varrho(\cdot|\nu_2), \cdots, \varrho(\cdot|\nu_K)$

Figure 1: Workflow diagram of proposed approaches.

2 PROBLEM

We focus on long-horizon planning tasks, modeled as labeled contextual Markov decision processes (CMDPs) $\bar{\mathcal{M}}^L$, where RMs $\mathcal{R}$ encode their non-Markovian reward functions. Consider a two-door environment (see Figure 2), where an agent has to pass the first door $d1$, then get the key in the box b, unlock the second door $d2$, and reach the goal g, without hitting the walls w. A labeled CMDP $\bar{\mathcal{M}}^L = \langle S, A, C, \mathsf{M}^L \rangle$ consists of state, action, and context spaces S, A, C, respectively, and a mapping M^L from C to a labeled MDP [12] $\mathsf{M}^L(c) = \langle S, A, p_c, R^L_c, \phi_c, \gamma, \mathcal{P}, L_c \rangle$. Parameterized by $c \in C$, a labeled MDP $\mathsf{M}^L(c)$ combines a probabilistic transition function p_c, an initial state distribution ϕ_c, a non-Markovian reward function R^L_c, a discount factor γ, a finite set $\mathcal{P}$ of propositional variables, and a labeling function $L_c : S \times A \times S \to 2^{\mathcal{P}}$. In two-door, c determines the door positions, hence affecting p_c and R^L_c. A reward machine [4] $\mathcal{R} = \langle Q, \mathsf{q}_I, 2^{\mathcal{P}}, O, \delta_\mathsf{q}, \delta_r \rangle$ consists of a finite set Q of states, an initial state $\mathsf{q}_I \in Q$, an input alphabet $2^{\mathcal{P}}$, an output alphabet O, a deterministic transition function δ_q, and an output function δ_r. In Figure 2, the transition from q_0 to $\mathsf{q}_1 = \delta_\mathsf{q}(\mathsf{q}_0, \ell)$ occurs when the agent gets label $\ell = \{d1\}$, yielding reward $\delta_r(\mathsf{q}_0, \ell) = 1$.

Problem Statement: Given a labeled CMDP $\bar{\mathcal{M}}^L$, its RM $\mathcal{R}$, and a target context distribution φ, obtain an optimal policy that solves $\max_\pi \mathbb{E}_{\varphi(c),\phi_c(s),\pi}[\sum_{t=0}^{\infty} \gamma^t R^L_c(s_0 a_0 \cdots s_t a_t s_{t+1})]$, where $a_t \sim \pi$.

3 INTERMEDIATE SELF-PACED RL

Given initial and target context distributions $\varrho(\cdot|\nu_0)$ and φ, respectively, self-paced RL [5] addresses the problem we describe by generating a sequence of context distributions $\{\varrho(\cdot|\nu_k)\}_{k=1}^K$. The agent updates π using trajectories collected in contexts drawn from

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

$\varrho(\cdot|\nu_k)$, parameterized by ν_k, that self-paced RL obtains by solving

$$\begin{aligned} \max_{\nu'_k} \quad & \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T_i-1}\gamma^t g(c_i) r_{i,t} - \alpha D_{\mathrm{KL}}(\varrho(c|\nu'_k) \,||\, \varphi(c)) \\ \text{s.t.} \quad & D_{\mathrm{KL}}(\varrho(c|\nu'_k) \,||\, \varrho(c|\nu_{k-1})) \leq \epsilon, \end{aligned} \tag{1}$$

where $g(c_i) = \frac{\varrho(c_i|\nu_k)}{\varrho(c_i|\nu_{k-1})}$ is the importance sampling weight used to estimate the value of state $s_{i,0}$ in context c_i with respect to the current context distribution $\varrho(\cdot|\nu_k)$, as c_i is sampled from $\varrho(\cdot|\nu_{k-1})$. We propose *intermediate self-paced RL*, that extends self-paced RL by running an RL agent on a product CMDP $\bar{\mathcal{M}}^L_{\mathcal{R}} = \langle \bar{S}, A, C, \bar{\mathrm{M}}^L \rangle$, which differs from $\bar{\mathcal{M}}^L$ due to its product state space $\bar{S} = S \times Q$, combining the states of $\bar{\mathcal{M}}^L$ and $\mathcal{R}$. On $\bar{\mathrm{M}}^L(c)$, an agent follows policy π to collect trajectory $\bar{\tau} = \{(\bar{s}_{t-1}, a_{t-1}, \bar{r}_t, \bar{s}_t)\}_{t=1}^{T}$, with state $\bar{s}_t \in \bar{S}$, action $a_t \sim \pi$, and reward $\bar{r}_t = \bar{R}^L_c(\bar{s}_{t-1}, a_{t-1}, \bar{s}_t)$ in c.

4 FROM REWARD MACHINES TO CONTEXTS

In two-door, we observe that the first context parameter $c[1]$, i.e., the position of the first door, determines which $\bar{\mathcal{M}}^L_{\mathcal{R}}$ transitions enable the agent to pass the first door, yielding label $\{d1\}$. Changing $c[1]$ causes different $\bar{\mathcal{M}}^L_{\mathcal{R}}$ transitions to yield label $\{d1\}$. However, such change does not impact the transitions that enable the agent to pass the second door, i.e., to obtain label $\{d2\}$. Taking this observation into account, we define a *reward-machine-context mapping* $\mathsf{F} : Q \times Q \rightarrow 2^D$, which outputs the smallest set of *identifier* context parameters that determines if a transition in $\mathcal{R}$ happens, where D is the dimensions of C. In two-door, D is $\{1, 2\}$ and F outputs $\mathsf{F}(\mathsf{q}_0, \mathsf{q}_1) = \{1\}$, $\mathsf{F}(\mathsf{q}_1, \mathsf{q}_2) = \emptyset$, and $\mathsf{F}(\mathsf{q}_2, \mathsf{q}_5) = \{1, 2\}$, etc.

5 RM-GUIDED, SELF-PACED RL

Klink et al. [7]'s self-paced RL algorithm uses an importance weight $g(c_i) = \frac{\varrho(c_i|\nu_k)}{\varrho(c_i|\nu_{k-1})}$ by assuming that every context parameter has an effect on the reward that an environment interaction yields, e.g., the position of the second door, $c[2]$, affects which interaction allows the agent to pass the first door, obtaining a reward of 1. We remove this assumption by proposing F to compute the importance sampling weight of a reward received in transition $(\mathsf{q}_{t-1}, \mathsf{q}_t)$. *RM-guided, self-paced RL* achieves this in (1) by utilizing the marginal context distributions for the identifier context parameters $\mathsf{f}_t = \mathsf{F}(\mathsf{q}_{t-1}, \mathsf{q}_t)$ to compute $g(c_i) = \frac{\varrho_{\mathsf{f}_t}(c_i|\nu_k)}{\varrho_{\mathsf{f}_t}(c_i|\nu_{k-1})}$ where $\varrho_{\mathsf{f}_t}(\cdot|\nu_k)$ and $\varrho_{\mathsf{f}_t}(\cdot|\nu_{k-1})$ are the current and previous marginal context distributions with respect to f_t, respectively.

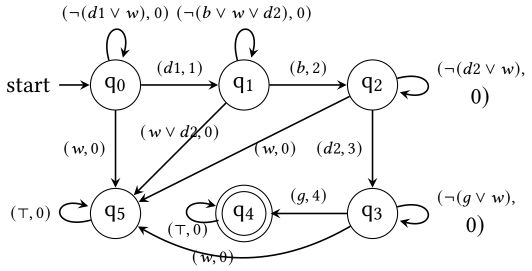

Figure 2: Reward machine of the two-door environment.

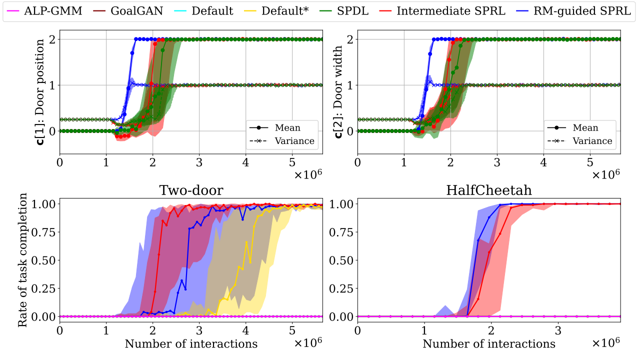

Figure 3: Progression of (Top) context distributions generated by self-paced RL methods in two-door, and (Bottom) the rate of task completion in contexts drawn from φ in two-door (15 runs) and HalfCheetah (10 runs). Bold lines indicate median values, and shaded regions cover the first and third quartiles.

6 EMPIRICAL RESULTS

We evaluate intermediate and RM-guided self-paced RL algorithms against a self-paced RL method, SPDL [7], two state-of-the-art automated curriculum generation methods, GoalGAN [2] and ALP-GMM [10], as well as two baselines, Default and Default*. Default draws contexts from φ without generating a curriculum, and Default* extends it by running an RL algorithm on $\bar{\mathcal{M}}^L_{\mathcal{R}}$. The top row of Figure 3 demonstrates the progression of statistics (mean and variance) of Gaussian context distributions generated by self-paced RL algorithms in the two-door environment. RM-guided, self-paced RL's curricula converge to φ faster, by one-fourth of the curriculum updates of others, and have smaller curricula variance, see the narrow blue region around the median, with statistical significance of $p < 0.0001$ according to a Welch's t-test. The bottom-left figure shows the progression of the rate of task completion in the same environment. Intermediate and RM-guided self-paced RL algorithms converge before Default*, which indicates that curriculum generation boosts learning, whereas the rest of the algorithms fail to accomplish the task, concluding that incorporating reward machines in learning allows the agent to capture the temporal structure of the task. We also test the algorithms in a customized HalfCheetah-v3 environment [1]. The bottom-right figure reports that only the proposed methods are able to accomplish the task, where RM-guided, self-paced RL converges faster than its intermediate counterpart.

7 CONCLUSION

We develop two self-paced RL algorithms for long-horizon planning tasks: 1) intermediate self-paced RL, which uses reward machines to update the value function and policy of RL agents, and 2) RM-guided, self-paced RL, which extends the first by navigating curriculum generation via a reward-machine-context mapping that we propose. Our empirical evaluations conclude that the proposed algorithms achieve optimal behavior in long-horizon planning tasks we use, whereas baselines and state-of-the-art methods fail. In addition, guiding curriculum generation reduces curricula variance, avoiding inefficient exploration of the curriculum space.

ACKNOWLEDGMENTS

This work is supported by the Office of Naval Research (ONR) under grant number N00014-22-1-2254, and the National Science Foundation (NSF) under grant number 1646522.

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:1606.01540

[2] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. 2018. Automatic goal generation for reinforcement learning agents. In *PMLR*. 1515–1528.

[3] Assaf Hallak, Dotan Di Castro, and Shie Mannor. 2015. Contextual Markov Decision Processes. arXiv:1502.02259

[4] Rodrigo Toro Icarte, Toryn Klassen, Richard Valenzano, and Sheila McIlraith. 2018. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *PMLR*. 2107–2116.

[5] Pascal Klink, Hany Abdulsamad, Boris Belousov, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. 2021. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *JMLR* 22 (2021), 182–1.

[6] Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. 2020. Self-paced contextual reinforcement learning. In *CoRL*. 513–529.

[7] Pascal Klink, Carlo D' Eramo, Jan R Peters, and Joni Pajarinen. 2020. Self-paced deep reinforcement learning. In *NeurIPS*. 9216–9227.

[8] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *JMLR* (2020), 1–50.

[9] Ronald Parr and Stuart Russell. 1997. Reinforcement learning with hierarchies of machines. *NeurIPS* (1997).

[10] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *CoRL*. 835–853.

[11] Satinder P Singh. 1992. Reinforcement learning with a hierarchy of abstract models. In *National Conference on Artificial Intelligence*. 202–207.

[12] Zhe Xu, Ivan Gavran, Yousef Ahmad, Rupak Majumdar, Daniel Neider, Ufuk Topcu, and Bo Wu. 2020. Joint inference of reward machines and policies for reinforcement learning. In *ICAPS*. 590–598.