0.1 Common AIDA notation

$\mathbb{R}_{\geq 0}, \mathbb{Z}_{\geq 0}$	Set of nonnegative reals, resp., integers
$\mathbb{P}[\bullet]$	Probability measure
$\mathbb{E}_f[\bullet]$	Expectation under distribution f
$\|\bullet\|$	Norm (context-dependent)

0.1.1 Reinforcement learning and control notation.

Q	Quality function
r, R	Running objective (reward or cost) as definite, resp., random value

Conditional notation. When `mlnotation` toggle is set `true`. You should set the toggle in `PREAMBLE` section right after loading the preamble.

π	Policy, a law that generates actions from observations
s, S	State, as definite, resp., random value
a, A	Action, as definite, resp., random value
o, O	Observation, as definite, resp., random value
$\mathbb{S}$	State space
$\mathbb{A}$	Action space
$\mathbb{O}$	Observation space
Π	Policy space
p	State dynamics law (function or probability distribution)
V^π	Total objective (value or cost) of policy π
V^*	Optimum total objective (value or cost) under the optimal policy π^*
$\mathcal{A}^{\pi,\pi'}$	Advantage of policy π relative to policy π'
π^θ	Actor network with weights θ
$\hat{V}^w$	Critic network (state-valued) with weights w

0.2 Introduction

Reinforcement learning commonly addresses the following infinite-horizon optimal control and/or decision problem:

$$\begin{aligned} &\max_{\pi\in\Pi} V^{\pi}(s) = \\ &\max_{\pi\in\Pi} \mathbb{E}_{A_t\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(S_t,A_t)|S_0=s\right], \end{aligned} \tag{1}$$

where $S_t \in \mathbb{S}$ at a time $t \in \mathcal{T} := \mathbb{Z}_{\geq 0}$ is the environment's state with values in the state-space $\mathbb{S}$, r is the reward (in general, running objective) rate, γ is the discount factor, π is the agent's policy of some function class Π. The running objective may be taken as a random variable R_t whose probability distribution depends on the state and action. The agent-environment loop dynamics are commonly modeled via the following Markov chain:

$$S_{t+1} \sim p(\bullet|s_t, a_t),\ t \in \mathcal{T}. \tag{2}$$

For the problem (1), one can state an important recursive property of the objective optimum $V^*(s)$ in the form of the Hamilton-Jacobi-Bellman (HJB) equation as follows:

$$\max_{a\in\mathbb{A}}\{\mathcal{D}^a V^*(s) + r(s,a) - \gamma V^*(s)\} = 0, \forall s \in \mathbb{S}, \tag{3}$$

where $\mathcal{D}^a V^*(s) := \mathbb{E}_{S_+\sim p(\bullet|s,a)}\left[V^*((S_+))\right] - V^*(s)$.

The common approaches to (1) are dynamic programming [Ber19; LV09] and model-predictive control [GPM89; BBM11; DN12; May14]. The latter cuts the infinite horizon to some finite value $T > 0$ thus considering effectively a finite-time optimal control problem. Dynamic programming aims directly at the HJB (3) and solves it iteratively over a mesh in the state space $\mathbb{S}$ and thus belongs to the category of tabular methods. The most significant problem with such a discretization is the curse of dimensionality, since the number of nodes in the said mesh grows exponentially with the dimension of the state space. Evidently, dynamic programming is in general only applicable when the state-space is compact. Furthermore, state-

space discretization should be fine enough to avoid undesirable effects that may lead to a loss of stability of the agent-environment closed loop. Reinforcement learning essentially approximates the optimum objective V^* via a (deep) neural network.

Input: θ_0

for Learning iteration $i := 0 \dots \mathcal{I}$ **do**

Policy weight update

$\theta_{i+1} \leftarrow \theta_i - \alpha_i \hat{\mathbb{E}}_{\rho^{\pi^{\theta_i}}} \left[C^{\gamma}_{0:T} \sum_{t=0}^{T-1} \nabla_\theta \log \pi^{\theta_i}(Z_t) \right]$

$\alpha_i > 0$, learning rate

end for

return Near-optimal policy $\pi^{\theta_{\mathcal{I}}}$

Bibliography

[Ber19] D. P. Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[May14] D. Q. Mayne. "Model predictive control: Recent developments and future promise". In: *Automatica* 50.12 (2014), pp. 2967–2986.

[DN12] M. L. Darby and M. Nikolaou. "MPC: Current practice and challenges". In: *Control Engineering Practice* 20.4 (2012). Special Section: IFAC Symposium on Advanced Control of Chemical Processes - ADCHEM 2009, pp. 328–342.

[BBM11] F. Borrelli, A. Bemporad, and M. Morari. "Predictive Control For Linear And Hybrid Systems". In: *Cambridge February* 20 (2011), p. 2011.

[LV09] F. L. Lewis and D. Vrabie. "Reinforcement learning and adaptive dynamic programming for feedback control". In: *IEEE circuits and systems magazine* 9.3 (2009), pp. 32–50.

[GPM89] C. E. Garcia, D. M. Prett, and M. Morari. "Model predictive control: theory and practice–a survey". In: *Automatica* 25.3 (1989), pp. 335–348.