



Optimising actions for control objectives

Concept. The idea here is

13.1 States, actions and attributing rewards

Up to this point, we have only considered actions which were either scheduled up front through some fixed process or through user interaction via a game interface. In order to start creating algorithms to act on the system state for us, we now need to develop a formalism which ‘closes the loop’ by feeding information back from the stochastic process to another decision-making process. Note that in most cases, the state of real-world phenomena cannot be measured perfectly. So in order to enable any agent trained on simulated phenomena to potentially act in the real world, we will need to model this measurement process as part of the information retrieval step.

Let’s now define the concept of a ‘measured state’ \mathcal{S}_{t+1} of the system at timestep $t + 1$; this is a new vector that doesn’t have to share the same length as X_{t+1} . We can then say generally that this measured state is ‘observed’ using the following measurement function

$$\mathcal{S}_{t+1}^i = M_{t+1}^i(X', \mathcal{Z}_{t+1}, t), \quad (13.1)$$

where we have also introduced a new vector \mathcal{Z}_{t+1} which we will use to store all of the relevant parameters to the agent¹ at timestep $t + 1$.

If we are now given the conditional probability that an action vector element $\mathcal{A}_{t+1} = a$ is chosen given that state vector $\mathcal{S}_{t+1} = s$ has been measured $\pi(a, s) = p(a|s)$, we can use this to draw new actions for the agent with a newly defined action-generating function

$$\mathcal{A}_{t+1}^i = \Pi_{t+1}^i(\mathcal{S}_{t+1}, \mathcal{Z}_{t+1}). \quad (13.2)$$

¹This vector is intended to include parameters for measurement, policy specification and ultimately the learning algorithm as well.

From this point on we'll call $\pi(a, s)$ the 'policy' adopted by the agent. A Markov Decision Process (MDP) defines an algorithm in which the agent uses a single state measurement vector and its given policy π to draw actions \mathcal{A}_{t+1} at timestep $t + 1$. It then performs these actions in its environment, which we have previously formalised through defining the iteration $X_{t+1} = \mathcal{F}_{t+1}(X', Z_t, \mathcal{A}_{t+1}, t)$.

In order to assess the quality of an agent's actions, we might later attribute a reward value \mathcal{R}_t for actions that were taken at timestep t . Using a series of these rewards, a return value R can also be computed using a future discount factor γ like so

$$R = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}_t. \quad (13.3)$$

A state-value function V_π is defined as the expectation (under policy π) of return R , given state vector $\mathcal{S}_t = s$, i.e.,

$$V_\pi(s) = E_\pi(R|s). \quad (13.4)$$

Similarly, an action-value function Q_π is defined as the expectation (again, under policy π) of return R , given state vector $\mathcal{S}_t = s$ and action vector $\mathcal{A}_t = a$, i.e.,

$$Q_\pi(s, a) = E_\pi(R|s, a). \quad (13.5)$$

Bibliography