A Concise Presentation of AI

Richard S. Sutton

February, 2012

Artificial intelligence (AI) is the study and design of agents that interact with environments to achieve goals. Their major task is to predict and control their input stream, in particular, their reward, which formalizes the idea of goal. Thus, the first step in AI is usefully thought of as the study and design of algorithms for finding policies that maximize a scalar reward signal received from the environment, which can be well thought of as a finite Markov decision process (MDP). We proceed now to formally define these terms.

1 Finite Markov decision processes

Definition 1. *A* finite Markov decision process (MDP) *(discounted continuing case) consists of any $\mathcal{S}$, $\mathcal{A}$, $\mathcal{R}$, γ, and P, where*

$\mathcal{S}$ *is a finite* state *set,*

$\mathcal{A}$ *is a finite* action *set,*

$\mathcal{R}$ *is a finite* reward *set,*

$\gamma \in [0, 1)$ *is a* discount-rate parameter*, and*

$P : \mathcal{S} \times \mathcal{A} \times \mathcal{R} \times \mathcal{S} \to [0, 1]$ *is a mapping, characterizing the MDP's dynamics, such that* $\sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} P(s, a, r, s') = 1, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$.

The MDP's dynamics function P characterizes how rewards and next states are generated from previous states and actions, but does not specify how the actions are generated. The actions of course come from the agent, but more formally they come from *policies*, which may be deterministic or, more generally, stochastic.

From here on we consider there to be a given, arbitrary MDP, and definitions are theorems are implicitly with respect to that MDP and its components.

Definition 2. *A* deterministic policy *is any mapping $\pi : \mathcal{S} \to \mathcal{A}$. We write $\pi(s)$ for the action to take in state s.*

Note: There is a natural generalization to state-dependent action sets, which we omit to avoid complicating the notation.

Definition 3. *A* stochastic policy *is any mapping $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ such that $\sum_{a \in \mathcal{A}} \pi(s, a) = 1, \forall s \in \mathcal{S}$. We write $\pi(s, a)$ for the probability that action a is taken when in state s.*

Definition 4. *A* history *$h = s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_k$ is a finite sequence of states $s_i \in \mathcal{S}$, actions $a_i \in \mathcal{A}$, and rewards $r_i \in \mathcal{R}$, beginning and ending in a state, for finite $k \geq 0$.*
The set of all histories of length k is denoted $\mathcal{H}_k = \mathcal{S} \times (\mathcal{A} \times \mathcal{R} \times \mathcal{S})^k$,
and the set of all histories of length 0 or more is denoted $\mathcal{H} = \mathcal{S} \times (\mathcal{A} \times \mathcal{R} \times \mathcal{S})^*$.

Definition 5. *An infinite-length history is called a* trajectory.

Definition 6 (Probability measure over histories)**.** *Given a stochastic policy π and an initial state distribution $\mu : \mathcal{S} \to [0, 1]$, such that $\sum_{s \in \mathcal{S}} \mu(s) = 1$, we define a* probability measure over histories *$p : \mathcal{H} \to [0, 1]$ as:*

$$p(s_0, a_0, r_1, s_1, a_1, \ldots, s_k) = \\ \mu(s_0)\pi(s_0, a_0)P(s_0, a_0, r_1, s_1)\pi(s_1, a_1) \cdots P(s_{k-1}, a_{k-1}, r_k, s_k). \tag{1}$$

Note: There is an obvious generalization to deterministic policies, which we omit.

Theorem 1. $\sum_{h \in \mathcal{H}_k} p(h) = 1, \forall k = 0, 1, 2, \ldots$.

Notation: A random history, generated for example according to p, is denoted $H = S_0, A_0, R_1, S_1, \ldots, S_k$, where H and each of its components are random variables. In this context, we can explicitly say how policies determine actions, that is, that $\pi(s, a) = \mathbb{P}\{A_t = a \mid S_t = s\}$ (where $\mathbb{P}\{|\}$ denotes conditional probability), and how P determines rewards and next states, that is, that $P(s, a, r, s') = \mathbb{P}\{R_{t+1} = r, S_{t+1} = s' \mid S_t = s, A_t = a\}$.

2 State-value functions

So far we have formally described the interaction between an agent, in the form of a policy, and its environment, in the form of an MDP, but we have not yet discussed

the agent's goal. Informally, the agent's goal is to find a policy that maximizes the rewards in the trajectory, but there are many rewards in a trajectory, and exactly how they should be combined and traded-off must be formally defined. Critical to this definition, and to algorithms for finding good policies, is the discount-rate parameter, γ (which so far we have defined but not used), and the notion of value functions. We turn now to formalizing the first version of these concepts.

Recall that these formalizations are all with respect to a particular but arbitrary MDP that may be left implicit.

Definition 7 (Discounted return)**.** *For any trajectory $H = S_0, A_0, R_1, S_1, A_1, R_2, \ldots$, the* returns *$G_t$, for $t = 0, 1, 2, \ldots$, are random variables defined by*

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \\ &= \sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \qquad (2) \\ &= R_{t+1} + \gamma G_{t+1}, \qquad (3) \end{aligned}$$

where γ is the discount-rate parameter of the MDP.

Definition 8. *A* state-value function *is any function from $\mathcal{S}$ to $\mathbb{R}$ (where $\mathbb{R}$ is the set of real numbers).*

Definition 9. *The* state-value function for policy π *is defined by*

$$V^\pi(s) = \mathbb{E}_\pi\{G_t \mid S_t = s\},$$

where the expectation is conditional on the actions $A_t, A_{t+1}, A_{t+2}, \ldots$ being generated according to π.

Theorem 2. *V^π is the only state-value function satisfying the system of linear equations:*

$$V^\pi(s) = \sum_{a,r,s'} \pi(s,a) P(s,a,r,s')[r + \gamma V^\pi(s')], \quad \forall s \in \mathcal{S}. \qquad (4)$$

This is known as the Bellman equation *for π.*

Theorem 3 (Policy evaluation)**.** *For an arbitrary initial state-value function v_0, the sequence of state-value functions defined by*

$$v_{k+1}(s) = \sum_{a,r,s'} \pi(s,a) P(s,a,r,s')[r + \gamma v_k(s')], \ \ \forall s \in \mathcal{S} \qquad (5)$$

converges to the state-value function for policy π: $\lim_{k\to\infty} v_k = V^\pi$.

Definition 10. *The* optimal state-value function*,* $V^* : \mathcal{S} \to \mathbb{R}$*, is*

$$V^*(s) = \max_\pi V^\pi(s), \tag{6}$$

where the maximization is over all policies, stochastic or deterministic.

Theorem 4. *V^* is the only state-value function satisfying the system of nonlinear equations:*

$$V^*(s) = \max_a \sum_{r,s'} P(s,a,r,s')[r + \gamma V^*(s')], \quad \forall s \in \mathcal{S}. \tag{7}$$

This is known as the Bellman optimality equation*.*

Theorem 5 (Value iteration)**.** *For an arbitrary initial state-value function v_0, the sequence of state-value functions defined by*

$$v_{k+1}(s) = \max_a \sum_{r,s'} P(s,a,r,s')[r + \gamma v_k(s')], \ \ \forall s \in \mathcal{S} \tag{8}$$

converges to the optimal state-value function: $\lim_{k\to\infty} v_k = V^*$.

3 Better and best policies

So far we have defined and shown how to compute policy-specific and optimal state-value functions, V^π and V^*. Next we define optimal policies and show several ways to find them using these value functions.

Definition 11 (Partial order on policies)**.** *A policy π is said to be* better than or equal to *another policy, π', denoted $\pi \geq \pi'$, if and only if $V^\pi(s) \geq V^{\pi'}(s)\ \forall s \in \mathcal{S}$.*

Theorem 6 (Existence of optimal policies)**.** *There is at least one policy, π^*, that is better than or equal to all the others, that is, for which $\pi^* \geq \pi$, $\forall \pi$. Any such policy is called an* optimal policy*.*

Theorem 7. *For any optimal policy π^*, $V^{\pi^*} = V^*$. In other words, the optimal state-value function is common to all optimal policies.*

Definition 12 (Greedy policies)**.** *For any state-value function v, the corresponding deterministic* greedy policy*, denoted* $\text{greedy}(v) : \mathcal{S} \to \mathcal{A}$*, is defined by*

$$\text{greedy}(v)(s) = \arg\max_a \sum_{r,s'} P(s,a,r,s')[r + \gamma v(s')]. \tag{9}$$

Theorem 8. *The greedy policy of the optimal state-value function,* greedy(V^*) *is optimal.*

The above theorem gives us our first way of finding an optimal policy: we compute V^* by value iteration (Theorem 5) and then compute greedy(V^*) by (9). The next two theorems provide a second way of finding an optimal policy.

Theorem 9 (Policy improvement)**.** *For any policy* π*,* greedy$(V^\pi) \geq \pi$.

Theorem 10 (Policy iteration)**.** *For an arbitrary initial policy* π_0*, define a sequence of successor policies by*

$$\pi_{k+1} = \text{greedy}(V^{\pi_k}). \tag{10}$$

Then each successive policy is an improvement on the previous, $\pi_{k+1} \geq \pi_k$*, and the sequence converges to an optimal policy in a finite number of steps.*

4 Action-value functions

This section is left as an exercise. Repeat everything in the previous two sections, from Definition 8 on, appropriately changed and generalized to apply to action-value functions as defined below.

Definition 13. *The* action-value function for policy π *(and a given finite MDP), denoted* $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$*, is defined by*

$$Q^\pi(s, a) = \mathbb{E}_\pi\{G_t \mid S_t = s, A_t = a\}. \tag{11}$$

5 Episodic MDPs

Here we describe the changes needed to handle episodic MDPs, in which there are multiple, finite-length trajectories with restarts, and in which the discount-rate parameter, γ, may be 1. Conditions must be given on P in order to ensure finite values. We introduce a special terminal state, $\perp$, and a notation $\mathcal{S}^+$ for the state set augmented by $\perp$. We define a square matrix γP_π that combines γ, P, and π and whose powers must converge to zero. We introduce a starting-state distribution μ as an intrinsic part of the episodic MDP.

6 Episodic Markov reward processes

Here we describe episodic Markov reward processes, as will be used in project 2. These are basically episodic MDPs with the actions removed, or equivalently, for a fixed, given policy, so they should be straightforward.