Dynamic Programming

Rupam Mahmood

February 3, 2020

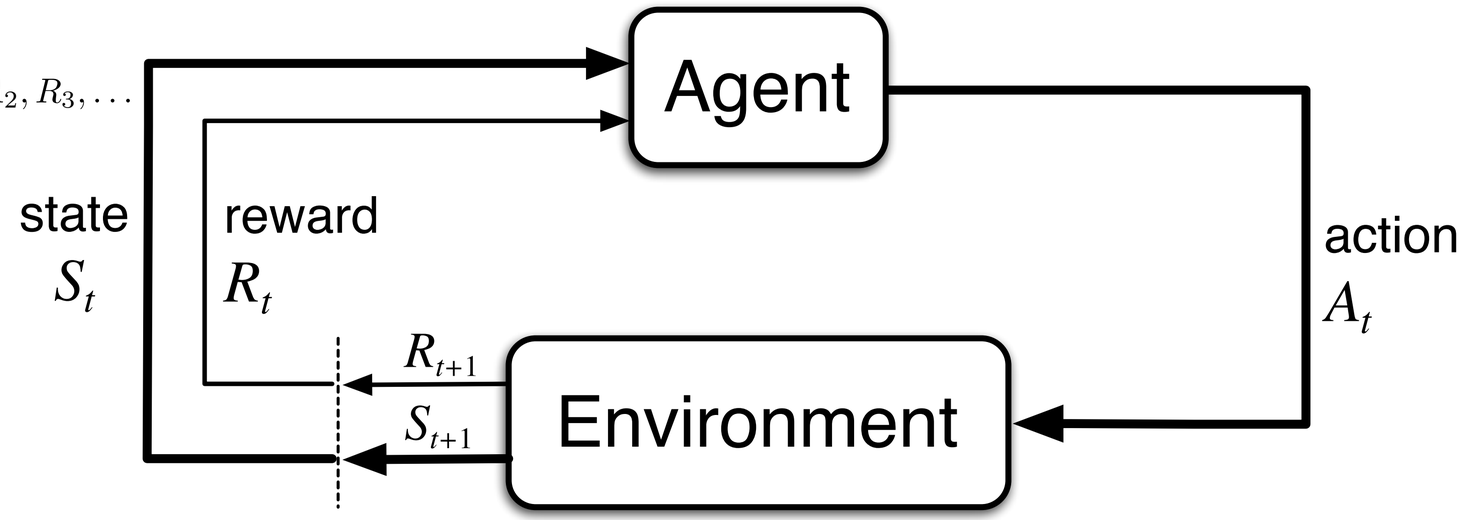
Who knows what?

Learning and other improvement mechanisms

Estimates like *V* or *Q*

$\pi(a|s)$

$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$



$$p(s', r \,|\, s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$$

$r(s,a) \qquad v_\pi(s) \quad q_\pi(s,a) \quad \pi_*$

Dynamic programming

- ✓ Dynamic programming (DP) is a way of knowing values and optimal policies when the model of the environment's dynamics is given
- ✓ If in a problem the model is given to the agent, it can use DP
- ✓ Often the agent is only given an estimate of the model if at all or …
- ✓ The agent estimates the model through experience (model learning)
- ✓ Or a designer tests their agent by comparing agent's performance against true values / optimal policies in a toy MDP
- ✓ Generally, mechanisms that use a given model as opposed to experience to improve performance are known as planning methods

Two key ideas underlying DP methods:

(1) Fixed-point iteration

Bellman equation

$$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

$$v_{k+1}(s) \doteq \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_k(s')]$$

Bellman update

Fixed-point equation

$$x = f(x)$$

$$x_{k+1} \doteq f(x_k)$$

Fixed-point iteration

Converges under some conditions

Iterative policy evaluation

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
    $\Delta \leftarrow 0$
    Loop for each $s \in \mathcal{S}$:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma V(s')\big]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Two key ideas underlying DP methods:
(2) Policy improvement

If for a policy π' we have $q_\pi(s, \pi'(s)) \geq v_\pi(s) \forall s,$ then we also have $v_{\pi'}(s) \geq v_\pi(s), \forall s$

A greedy policy w.r.t. q_π is such a policy

$$\max_a q_\pi(s, a) \geq v_\pi(s) \forall s$$

Policy iteration

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*$$

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
      $\Delta \leftarrow 0$
      Loop for each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s)) \big[r + \gamma V(s')\big]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      old-action $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma V(s')\big]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Value iteration

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

$$v_* = \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_*(s')]$$

$$v_{k+1} \doteq \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_k(s')]$$