Afterstate Formulation

PANG Kaicheng

1 Formulation

At each time step t, the agent's environmental states is $S_t \in \mathcal{S}^+$, and selects an action, $A_t \in \mathcal{A}(s)$. After the action, the agent's state is transformed to the afterstate $H_t \in \mathcal{S}^+$. This is a deterministic process, $h = f(s, a)$. The reward received is split into two parts: from action R^a_{t+1} and from environment R^e_{t+1}. In the problem which can be formed as afterstate one, the action reward is deterministic while the environmental process can be stochastic. The probability of next state $s' \in \mathcal{S}^+$ and $r^e \in \mathcal{R} \subset \mathbb{R}$ accurring at time t, given the current afterstate h:

$$p(s', r^e \mid h) \doteq \Pr\{S_{t+1} = s', R^e_{t+1} = r^e \mid H_t = h\} \tag{1}$$

The state-transition probabilities can be written as:

$$p(s' \mid h) \doteq \Pr\{S_{t+1} = s' \mid H_t = h\} = \sum_{r^e \in \mathcal{R}} p(s', r^e \mid h) \tag{2}$$

The expected environmental rewards as a function of afterstate $\rho_e : S \to \mathcal{R}$:

$$\rho_e(h) \doteq \mathbb{E}[R^e_{t+1} \mid H_t = h] = \sum_{r_e \in \mathcal{R}} r_e \sum_{s' \in \mathcal{S}} p(s', r^e \mid h) \tag{3}$$

The value function and action function of an afterstate h under policy π is:

$$\begin{aligned}
v_\pi(h) &\doteq \mathbb{E}[R^e_{t+1} + \gamma G_{t+1} \mid H_t = h] && (4)\\
&= \sum_{r_e} r_e \sum_{s'} p(s', r^e \mid h) + \gamma \mathbb{E}[v_\pi(s') \mid H_t = h] && (5)\\
&= \sum_{r_e} r_e \sum_{s'} p(s', r^e \mid h) + \gamma \sum_{s'} p(s' \mid h) v_\pi(s') && (6)\\
&= \sum_{r_e} r_e \sum_{s'} p(s', r^e \mid h) + \gamma \sum_{s'} \sum_{r_e} p(s', r^e \mid h) v_\pi(s') && (7)\\
&= \sum_{s'} \sum_{r_e} p(s', r^e \mid h)[r_e + \gamma v_\pi(s')] && (8)\\
&= \sum_{s'} \sum_{r_e} p(s', r^e \mid h)[r_e + \gamma \sum_{a'} \pi(a' \mid s')(v_\pi(h') + r'_a)] && (9)\\
q_\pi(h) &= q_\pi(f(s, a)) = v_\pi(h) + r_a && (10)
\end{aligned}$$

where $r_a = \rho_a(s, a)$ is the deterministic action reward.

2 Algorithm

Algorithm 1: 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 $V(h)$, for all $h \in \mathcal{S}^+$, arbitrarily except that $V(\text{terminal}) = 0$
repeat
  $\Delta \leftarrow 0$;
  foreach $h \in \mathcal{S}$ **do**
    $h \leftarrow f(s, a)$ $v \leftarrow V(h)$
    $V(h) \leftarrow \sum_{s', r_e} p(s', r^e \mid h)[r_e + \gamma \sum_{a'} \pi(a' \mid s')(V(f(s', a')) + r'_a)]$
    $\Delta \leftarrow \max(\Delta, |v - V(h)|)$
  end
until $\Delta < \theta$;

Algorithm 2: Policy Iteration for estimating $\pi \approx \pi*$

1. Initialization
$V(h) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$
2. Policy Evaluation
repeat
  $\Delta \leftarrow 0$
  foreach $h \in \mathcal{S}$ **do**
    $h \leftarrow f(s, a)$
    $v \leftarrow V(h)$
    $V(h) \leftarrow \sum_{s', r_e} p(s', r^e \mid h)[r_e + \gamma(V(f(s', \pi(s'))) + r'_a)]$
    $\Delta \leftarrow \max(\Delta, |v - V(h)|)$
  end
until $\Delta < \theta$;
3. Policy Improvement
policy-stable $\leftarrow$ *true*
foreach $s \in$ **do**
  old-action $\leftarrow \pi(s)$
  $\pi(s) \leftarrow \text{argmax}_a V(f(s, a)) + \rho_a(s, a)$
  If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
end
If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2