Dynamic Programming

Assumption

- env. is a finite MDP
- S, A, R
- dynamics : $p(s', r \mid s, a)$ is given for all $s \in S$, $a \in A(s)$, $r \in R$, $s' \in S^+$

Key Idea

$$v_*(s) = \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \max_a \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_*(s')]$$

Policy Evaluation (Prediction)

for arbitrary policy π, compute (evaluate) state-value function

$$v_\pi(s) = E_\pi[G_t \mid S_t = s]$$
$$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$
$$= E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$
$$= \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_\pi(s')]$$

Strategy : Iteration

$$v_{k+1}(s) = E_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s]$$
$$= \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_k(s')]$$

iterative policy Iteration

Input : π, to be evaluated

$v_0(s) = 0$. $k = 0$.

Repeat

for each all $s \in S$

$$v_{k+1}(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_k(s')]$$
$$\Delta v = \max_s |v_{k+1}(s) - v_k(s)|$$
$$k = k + 1$$

until $\Delta v < \varepsilon$.

$v_\pi = v_{k+1}$

Policy Improvement

How to improve policy? Try new policy and compare.

∴ If $q_\pi(s, \pi'(s)) \geq V_\pi(s)$, $\pi(s) \to \pi'(s)$ is better.

(proof) $V_\pi(s) \leq q_\pi(s, \pi'(s))$

$$= E[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)]$$

$$= E_{\pi'}[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s]$$

$$\leq E_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s]$$

$$= E_{\pi'}[R_{t+1} + \gamma E_{\pi'}[R_{t+2} + \gamma V_\pi(S_{t+2})] \mid S_t = s]$$

$$= E_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 V_\pi(S_{t+2}) \mid S_t = s]$$

$$\leq E_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V_\pi(S_{t+3}) \mid S_t = s]$$

$$\vdots$$

$$\leq E_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s]$$

$$= V_{\pi'}(s)$$

Intuitively, if one step try other policy $\pi'(s)$ for all states is better then $\pi(s)$, π' is better for whole process.

$$\pi'(s) = \arg\max_a q_\pi(s, a) \quad \to \text{greedy policy (simple solution)}$$

$$= \arg\max_a E[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = a]$$

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

Policy Iteration

$\pi_0 \xrightarrow{E} V_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V_{\pi_1} \to \cdots$

greedy policy.

1. Initialization : $V(s)$, $\pi(s)$ arbitrarily
2. Policy Evaluation

   Iterate $V_{k+1}(s) = \sum_{s', r} p(s', r \mid s, a = \pi(s))[r + \gamma V_k(s')]$

   until $\Delta V < \varepsilon$
3. Policy Improvement

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

   If $\pi(s) =$ old policy, Stop

$V \approx V_*$, $\pi \approx \pi_*$.

Value Iteration

One drawback of policy iteration

- too many iteration in policy evaluation step
- especially, initial iterations meaningless because policy itself is meaningless

policy evaluation + policy improvement = value iteration

$$v_{k+1}(s) = \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_k(s')].$$

Algorithm

$v_0(s) = 0$, $k = 0$.

Repeat

For each $s \in S$:

$$v_{k+1}(s) = \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_k(s')]$$
$$\Delta v = \max_s |v_{k+1}(s) - v_k(s)|, \quad k = k+1,$$

until $\Delta v < \varepsilon$.

~~TT~~

$$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_k(s')].$$