CH3. Finite Markov Decision Process

* MDP → 즉각적 rewards (r_t), 연속적 situations (x_t), state (s_t), 미래의 rewards (r_{t+1})

(bandit 문제 : 각 a에 대해 $q_*(a)$ 값 예측

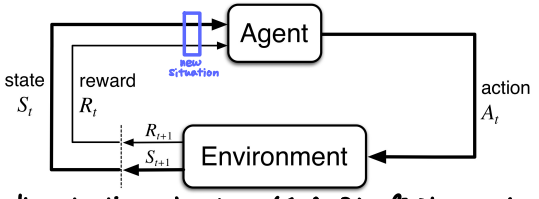
MDP : 각 s에서 각 a에 대해 $q_*(s,a)$ 값 예측

주어진 최적의 a 선택에 대해 각 s의 $v_*(s)$ 값 예측) s dependent → credit할당

* Agent - Enviroment Interface

(agent : learner나 decision maker

environment : agent를 둘러싼, interaction하는 모든것



discrete time step t , (S, A, R) : finite number

R_t, S_t : discrete probability distributions
↓ depends only on S, a

$p(s', r \mid s, a)$: dynamics of MDP (t t $t+1$ $t-1$)

$S \times R \times S \times A \to [0,1]$ deterministic

$\sum_{s'} \sum_{r} p(s', r \mid s, a) = 1$

$p(s' \mid s, a)$: state-transition probabilities

$S \times S \times A \to [0,1]$

$= \sum_{r} p(s', r \mid s, a)$

$r(s, a)$: expected rewards for state-action pairs

$S \times A \to \mathbb{R}$

$= \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r} r\, p(r \mid s, a) = \sum_{r} r \sum_{s'} p(s', r \mid s, a)$

$r(s, a, s')$: expected rewards for state-action-next-state triples

$S \times A \times S \to \mathbb{R}$

$= \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r} r\, p(r \mid s, a, s') = \sum_{r} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)}$

* Goals and Rewards

Goal = maximize total reward

= maximize the expected value of the cumulative sum of a received reward

return $G_t = R_{t+1} + R_{t+2} \cdots + R_T$ (T : final time step)

episode : terminal state에서 끝

($T \neq \infty$: episodic task

$T = \infty$: continuing task

↓ discounting

discounted return $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ ($0 \le \gamma \le 1$: discount rate)

(γ 0 : myopic 즉각적만 → γ 1 : 미래에서)

$= R_{t+1} + \gamma G_{t+1}$

$= \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ ($T \neq \infty$: episodic, $T = \infty$: continuing)

$R_{k+1} = 0$이 나오기 시작하면 $S_k = 0$

* Policies and Value Functions

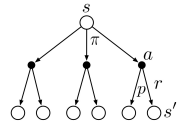
policy : mapping state → probabilities of selecting action

$= \pi(a|s)$

value function for policy π

- state-value function : $v_\pi(s) = \mathbb{E}_\pi\{G_t | S_t = s\} = \mathbb{E}_\pi\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\}$
- action-value function : $q_\pi(s,a) = \mathbb{E}_\pi\{G_t | S_t = s, A_t = a\} = \mathbb{E}_\pi\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\}$

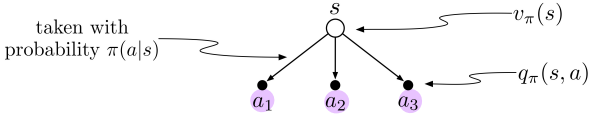
→ estimate w/ Monte-Carlo methods



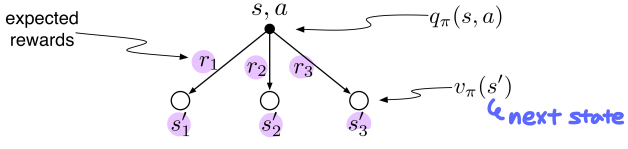
Backup diagram for v_π

$$v_\pi(s) = \mathbb{E}\{G_t | s\} = \mathbb{E}\{R_{t+1} + \gamma G_{t+1} | s\}$$
$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a)(r + \gamma \mathbb{E}_\pi\{G_{t+1} | S_{t+1} = s'\})$$
$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a)(r + \gamma v_\pi(s')) \quad : \text{Bellman equation for } v_\pi$$

$q_\pi(s,a)$



$$q_\pi(s,a) = \sum_{s'} \sum_r p(s', r | s, a)(r + \gamma v_\pi(s'))$$



* Optimal Policies and Optimal Value Functions

$\pi \geq \pi' \longleftrightarrow v_\pi(s) \geq v_{\pi'}(s)$

모든 다른 policies보다 크거나 같은 최소 하나의 policy = optimal policy

$\pi^* \geq$ all other $\pi \longrightarrow v_{\pi^*}(s) \geq v_\pi(s)$

$\therefore v_*(s) = \max_\pi v_\pi(s)$: optimal state-value function

$q_*(s,a) = \max_\pi q_\pi(s,a)$: optimal action-value function

$$= \mathbb{E}\{R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\}$$

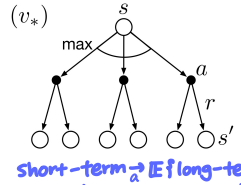
greedy : 확률을 고려하지 않고 다른 길 보는 것

Bellman optimality equation

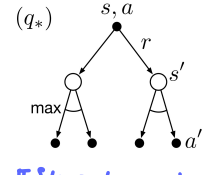
$$v_*(s) = \max_a q_{\pi_*}(s,a) = \max_a \mathbb{E}_{\pi_*}\{G_t | S_t = s, A_t = a\}$$
$$= \max_a \mathbb{E}_{\pi_*}\{R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a\}$$
$$= \max_a \mathbb{E}\{R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\}$$
$$= \max_a \sum_{s',r} p(s', r | s, a)(r + \gamma v_*(s'))$$

$$q_*(s,a) = \mathbb{E}\{R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a\}$$
$$= \sum_{s',r} p(s', r | s, a)\{r + \gamma \max_{a'} q_*(s', a')\}$$

Unique solution



Short-term $\xrightarrow{a}$ $\mathbb{E}\{$long-term$\}$
One-step-ahead search



$\mathbb{E}\{$long-term return$\}$

CH4. Dynamic Programming

env (finite MDP) → DP → continuous S, A

S, A, R dynamics = $p(s', r \mid s, a)$

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

* env를 완전히 안다면 approximation for v_π

: $v_{k+1}(s) = \mathbb{E}_\pi\{R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s\} = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\,(r + \gamma v_k(s'))$: Iterative policy evaluation

(v_π; 가능한 ∀next state에 대해 expected update)

$k \to \infty$: $v_k \to v_\pi$

in place · old를 new로 즉시 overwrite → sweep → v_π로 빠르게 수렴

$\max_s |v_{k+1}(s) - v_k(s)|$가 충분히 작아지면 stop

* policy Improvement

$$v_\pi(s) \le q_\pi(s, \pi'(s)) = \mathbb{E}_{\pi'}\{R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\}$$

(π'이 더 좋아야함 ($v_{\pi'}(s) \ge v_\pi(s)$))

$$\le \mathbb{E}_{\pi'}\{R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s\} \quad (\because \pi' \ge \pi)$$
$$= \mathbb{E}_{\pi'}\{R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \mid S_t = s\}$$
$$\le \cdots \le \mathbb{E}_{\pi'}\{R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s\} = v_{\pi'}(s)$$

$$\pi'(s) = \arg\max_a q_\pi(s,a) = \arg\max_a \mathbb{E}\{R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a\}$$

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

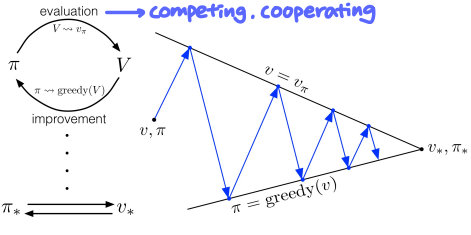
E: evaluation : $\max_s |v_{k+1}(s) - v_k(s)|$가 충분히 작을때까지

I: Improvement : $\pi' = \pi$일때까지

* value Iteration

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

* generalized policy iteration (GPI)



competing, cooperating