Dynamic Programming (cf Model based)

- Optimal Policy

- policy π a distribution over action given states.

$$\pi(a|s) = P[A_t = a \mid S_t = s]$$

stationary. time-independent.

$$\overset{s \to a}{} \qquad \overset{s \to a \to s'}{}$$

$$P^{\pi}_{ss'} = \sum_{a \in A} \pi(a|s) P^{a}_{ss'}$$

$$R^{\pi}_{s} = \sum_{a \in A} \pi(a|s) R^{a}_{s}$$ 예상 reward

- Optimal Value function

state-value $v_*(s) = \max_{\pi} v_{\pi}(s)$ $\leftarrow$ $\pi \geq \pi'$ if $v_{\pi}(s) \geq v_{\pi'}(s)$, $\forall s$

action-value $q_*(s,a) = \max_{\pi} q_{\pi}(s,a)$.

- Optimal policy can be found by maximising over $q_*(s,a)$

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in A}{\operatorname{argmax}}\, q_*(s,a) \\ 0 & \end{cases}$$

- DP in MDP

- 대략구조.

Prediction - Policy Evaluation

Control - Policy Improvement, value iteration

Policy Iteration

Policy Iteration Bellman Expectation

Value Iteration Bellman Optimality

Synchronous

Asynchronous 바로바로 업데이트

- Policy Evaluation

: evaluate a given policy π $v_1 \to v_2 \to v_3 \to \cdot \quad \cdot \to v_{\pi}$

$$v_{k+1}(s) = \sum_{a \in A} \pi(a|s) \left(R^{a}_{s} + \gamma \sum_{s' \in S} P^{a}_{ss'} v_k(s') \right)$$

$$v^{k+1} = R^{\pi} + \gamma P^{\pi} v^{k}$$

처음에는 random policy 배정, k(횟수) 거듭할수록 optimal policy로.

Control

- Policy Improvement : action 고르기

↳ Greedy Policy Improvement

앞에 $V_{k+1}(s)$를 통해 state value 구하기 완료

현재 state → 다음 state

action 고르기

$$\pi' = \arg\max_{a\in A} q_\pi(s,a)$$

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s\in S} \underline{P_{ss'}^a V_\pi(s')} \text{ 확률에 의존}$$

- Value Iteration

evaluation 에 모든 이동가능한 state 에 대한 state value 구하기 완료

이 중 max를 취해 greedy하게 value function을 구하자.

$$V_{k+1}(s) = \max_{a\in A} \left(R_s^a + \gamma \sum_{s\in S} \underline{P_{ss'}^a V_k(s')}\right) \text{ deterministic.}$$