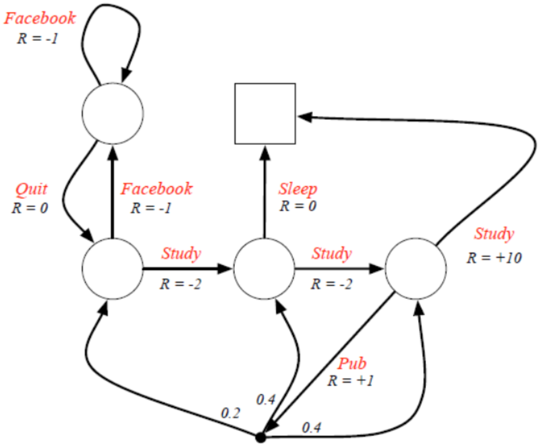
Finite Markov Decision Process

Hello!

I am Dido Choi

Brake Control Logic Developer

MDP(Markov Decision Process)



[1] MDP Graph

This problem involves evaluative feedback, as in bandits, but also an **associative aspect**—choosing different actions in different situations → MAB(Multi-Arm Bandit) has no state-transition probability)

MDPs are **a mathematically idealized form** of the reinforcement learning problem for which precise theoretical statements can be made.

[1] https://dnddnjs.gitbooks.io/rl/content/mdp.html

MDP(Markov Decision Process) VS MAB

	Actions don't change state of the world	Actions change state of the world
Learn model of outcomes	Multi-armed bandits	Reinforcement Learning
Given model of stochastic outcomes	Decision theory	Markov Decision Process

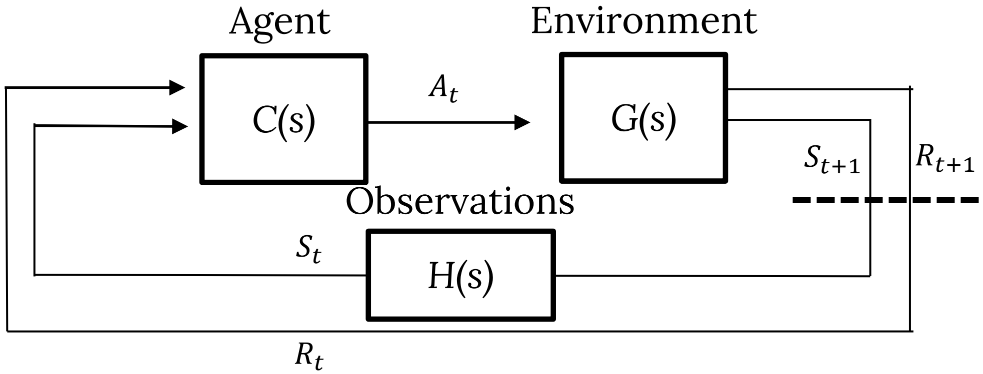
[Zhou, 2015]

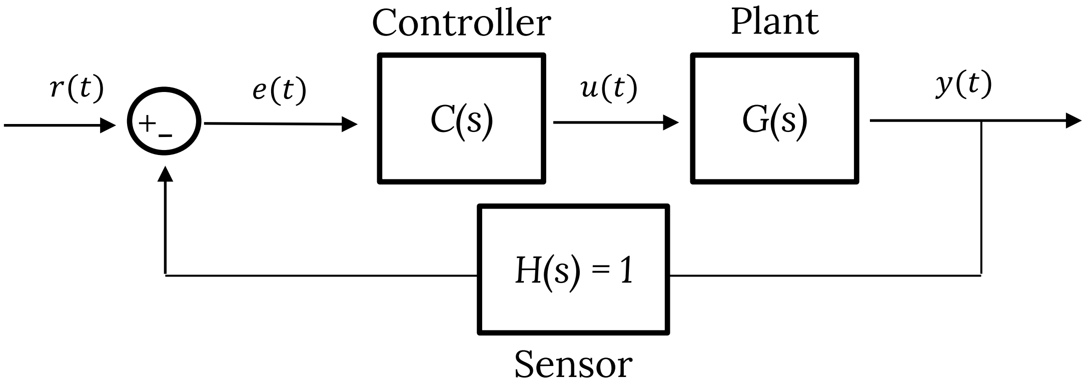
[1] MAB vs MDP

[2] MDP Diagram

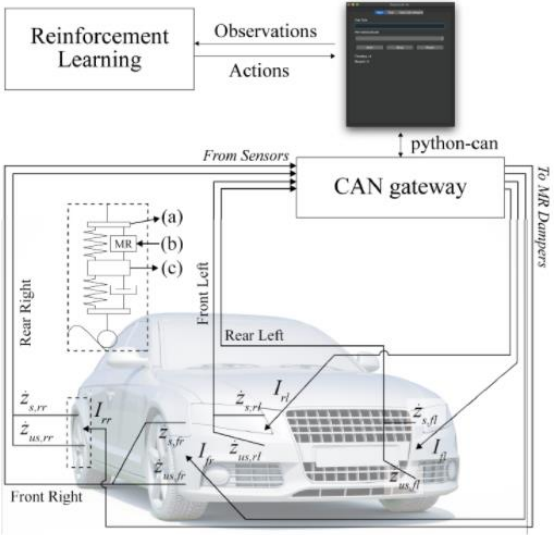
[1] Zhou, Li. "A survey on contextual multi-armed bandits." CoRR, abs/1508.03326 (2015)
[2] https://boliu68.github.io/2017/Reinforcement-Learning-versus-Bandit/

Reinforcement Learning Interface: MDP





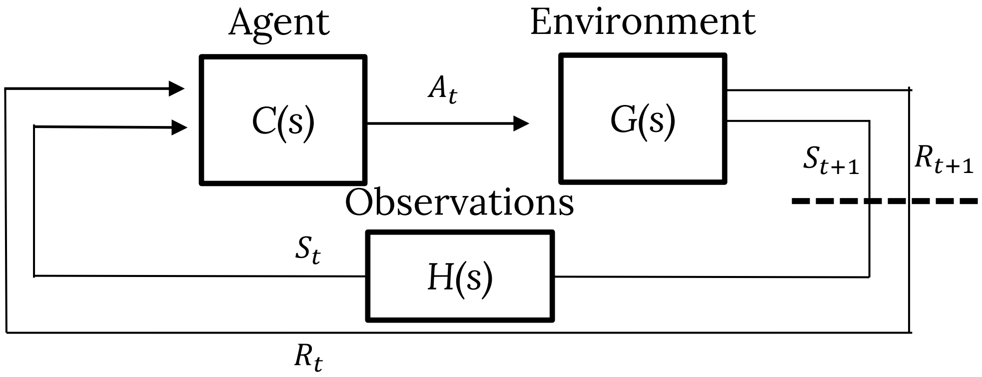
Reinforcement Learning Interface: MDP



[1] Suspension Control using RL

[1] H. Yong, J. Seo, J. Kim, M. Kim and J. Choi, "Suspension Control Strategies Using Switched Soft Actor-Critic Models for Real Roads," in *IEEE Transactions on Industrial Electronics*, doi: 10.1109/TIE.2022.3153805.

Reinforcement Learning Interface: MDP



State $S_t \in \mathcal{S}$

Action $A_t \in \mathcal{A}(s)$

Reward $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

Conditional Probability

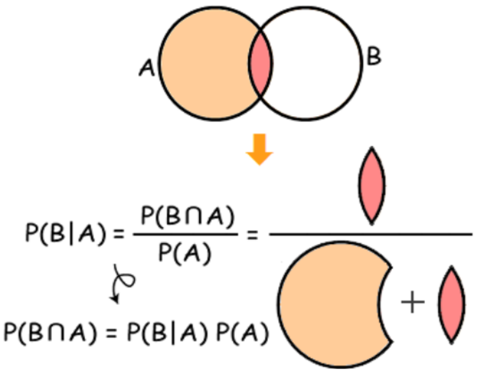
$A \in \mathcal{F}$

$\Pr(A) > 0$

이 주어졌다고 하자. 임의의 사건 $B \in \mathcal{F}$에 대하여, A에 대한 B의 **조건부 확률**은 다음과 같다.

$$\Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)}$$

[1] Conditional Probability Definition



[2] Conditional Probability

[1] https://ko.wikipedia.org/wiki/%EC%A1%B0%EA%B1%B4%EB%B6%80_%ED%99%95%EB%A5%A0
[2] https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=alwaysneoi&logNo=100148922781&view=img_13

Reinforcement Learning Interface: MDP

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

Dynamics of MDP
$$p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$$
$$\text{for all } s', s \in \mathcal{S},\ r \in \mathcal{R}, \text{ and } a \in \mathcal{A}(s)$$

We also know that
$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s).$$

State-transition Probability
$$p(s' \mid s, a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r \mid s, a).$$

a three-argument function $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$

$$P_{ss'}^{a} = \boldsymbol{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

Reinforcement Learning Interface: MDP

Expected Rewards: State-Action

$$r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}:$$

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

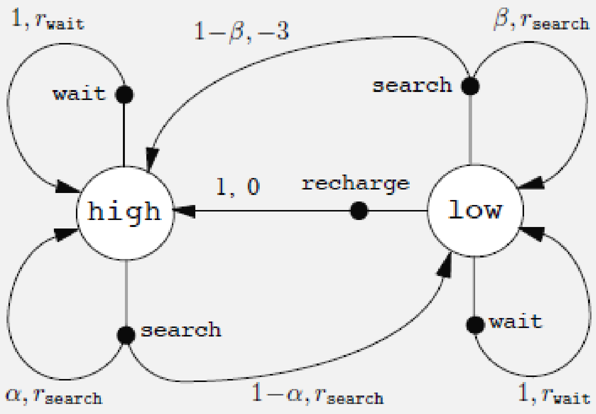
Expected Rewards: State-Action-Next State

$$r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R},$$

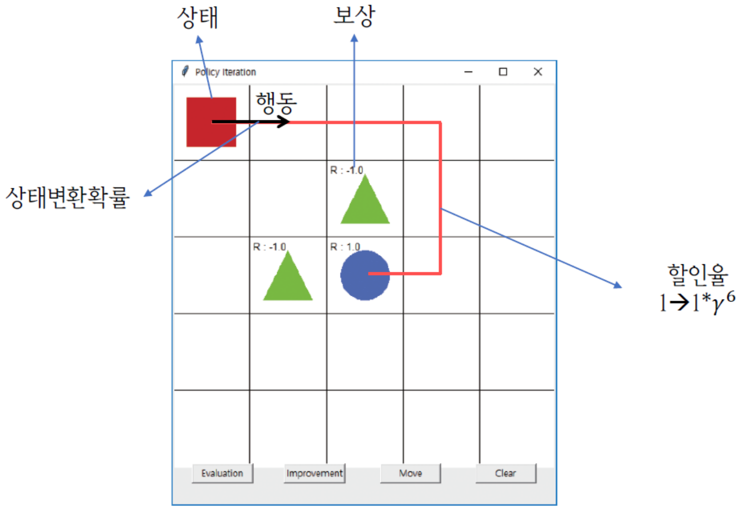
$$r(s,a,s') \doteq \mathbb{E}[R_t \mid S_{t-1}=s, A_{t-1}=a, S_t = s'] = \sum_{r\in\mathcal{R}} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)}.$$

MDP: Recycling Robot

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	$r_{\texttt{search}}$
high	search	low	$1-\alpha$	$r_{\texttt{search}}$
low	search	high	$1-\beta$	-3
low	search	low	β	$r_{\texttt{search}}$
high	wait	high	1	$r_{\texttt{wait}}$
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	$r_{\texttt{wait}}$
low	recharge	high	1	0
low	recharge	low	0	-



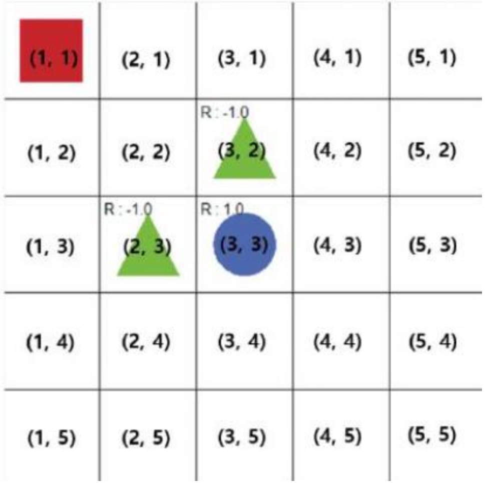
MDP: Grid World



그리드월드 문제에서의 MDP

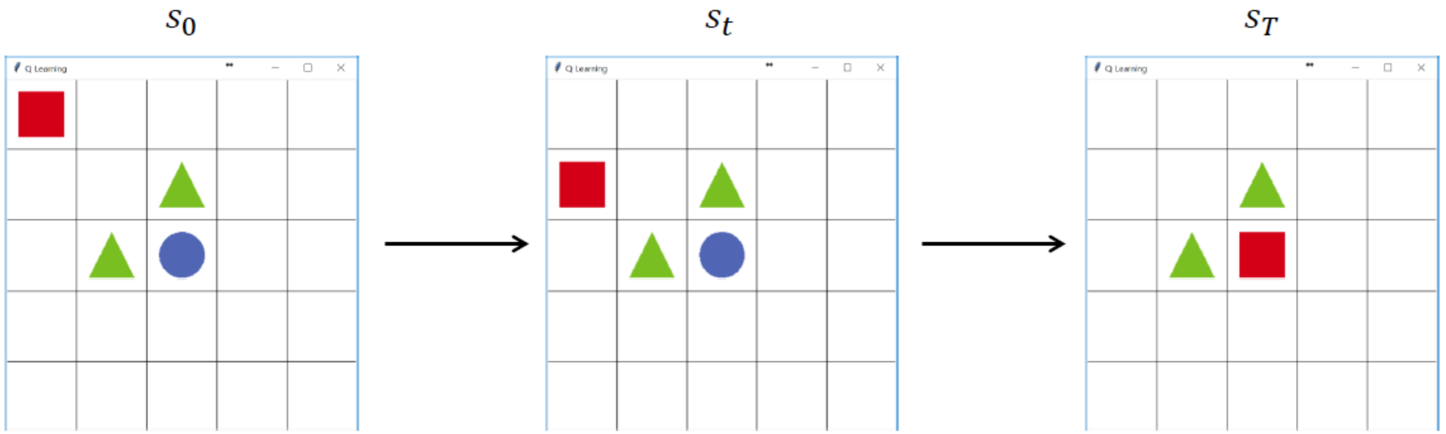
[1] Grid World, Woongwon Lee, Seminar, https://tykimos.github.io/2018/02/07/ISS_Near_and_Far_DeepRL/

MDP: Grid World



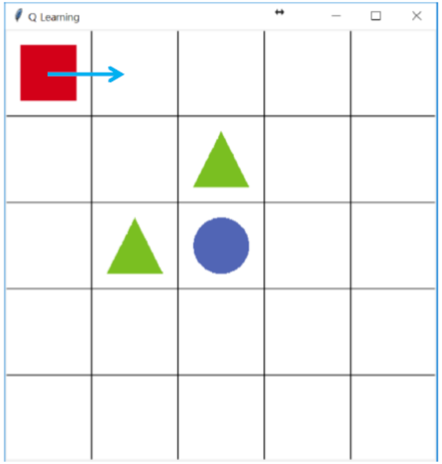
[1] Grid World, Woongwon Lee, Seminar, https://tykimos.github.io/2018/02/07/ISS_Near_and_Far_DeepRL/

MDP: Grid World



[1] Grid World, Woongwon Lee, Seminar, https://tykimos.github.io/2018/02/07/ISS_Near_and_Far_DeepRL/

MDP: Grid World

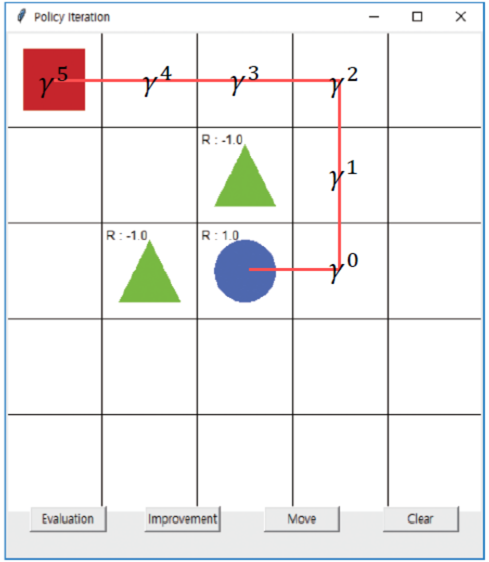


상태 (1, 1)에서 행동 “우”를 했을 경우

1. 상태 (2, 1)에 갈 확률은 0.8
2. 상태 (1, 2)에 갈 확률은 0.2

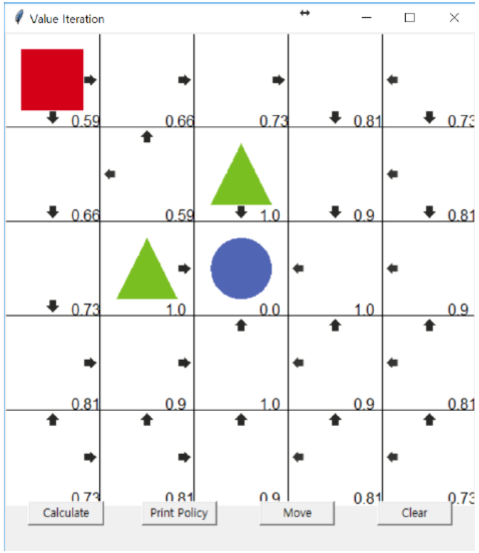
[1] Grid World, Woongwon Lee, Seminar, https://tykimos.github.io/2018/02/07/ISS_Near_and_Far_DeepRL/

MDP: Grid World



[1] Grid World, Woongwon Lee, Seminar, https://tykimos.github.io/2018/02/07/ISS_Near_and_Far_DeepRL/

MDP: Grid World



[1] Grid World, Woongwon Lee, Seminar, https://tykimos.github.io/2018/02/07/ISS_Near_and_Far_DeepRL/

Purposes and Rewards

Problem of Short-term Reward

Sparse Reward, Delayed Reward

The agent's goal is to **maximize the total amount of reward** it receives

Reward Hypothesis

That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

Returns and Episodes

Discounted Return(Episodic Tasks)

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

Discounted Return(Continuing Tasks), **Discount Rate**

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called the *discount rate*.

Discounted Return(Continuing Tasks) → Recurrence Relation

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\ &= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots \right) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned} \qquad t < T$$

Discounted Return(Continuing Tasks) → All rewards are 1

$$G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}.$$

Unified Notation for Episodic and Continuing Tasks

Unified Discounted Return

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k,$$

including the possibility that $T = \infty$ or $\gamma = 1$ (but not both).

Policy and Value Functions

A policy is a **mapping from states to probabilities** of selecting each possible action

State-value Function

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S},$$

Action-value Function

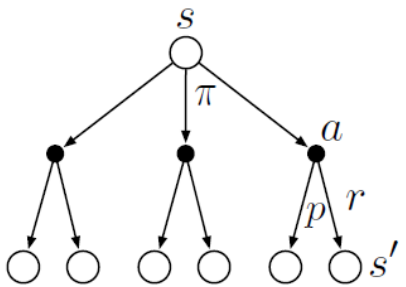
$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

Bellman Equation

Bellman Expectation Equation of State-value Function

$$\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)\Big[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']\Big] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_\pi(s')\Big], \quad \text{for all } s \in \mathcal{S},
\end{aligned}$$

Backup Diagram



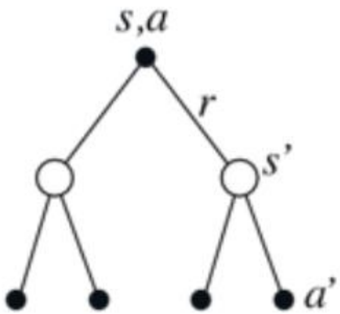
Bellman Equation

Bellman Expectation Equation of Action-value Function

$$q_\pi(s, a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

$$q_\pi(s, a) = \sum_{s', r} p(s', r \mid s, a)\left[r + \sum_{a' \in \mathcal{A}} \pi(a' \mid s') \cdot \gamma q_\pi(s', a')\right]$$

Backup Diagram



Optimal Policy and Optimal Value Functions

Optimal Policy

Solving a reinforcement learning task means, roughly, finding a policy that achieves a lot of reward over the long run. For finite MDPs, we can precisely define an optimal policy in the following way. Value functions define a partial ordering over policies. A policy π is defined to be better than or equal to a policy π' if its expected return is greater than or equal to that of π' for all states. In other words, $\pi \geq \pi'$ if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in \mathcal{S}$. There is always at least one policy that is better than or equal to all other policies. This is an *optimal policy*.

Optimal policy is also called 'Deterministic Policy'

- Deterministic Policy: $a = \pi(s)$
- Stochastic Policy: $\pi(a|s) = P[A_t = a|S_t = s]$

Optimal Policy and Optimal Value Functions

Optimal State-value Function

$$v_*(s) \doteq \max_\pi v_\pi(s),$$

for all $s \in \mathcal{S}$.

Optimal Action-value Function

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a),$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$.

Between Optimal State-value Function and Optimal State-action Function

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a].$$

Optimal Policy and Optimal Value Functions

Bellman Optimality Equation of State-value Function

$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big].
\end{aligned}
$$

Bellman Optimality Equation of Action-value Function

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \,\Big|\, S_t = s, A_t = a\Big] \\
&= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma \max_{a'} q_*(s',a')\Big].
\end{aligned}
$$

Optimal Policy and Optimal Value Functions

Backup Diagram

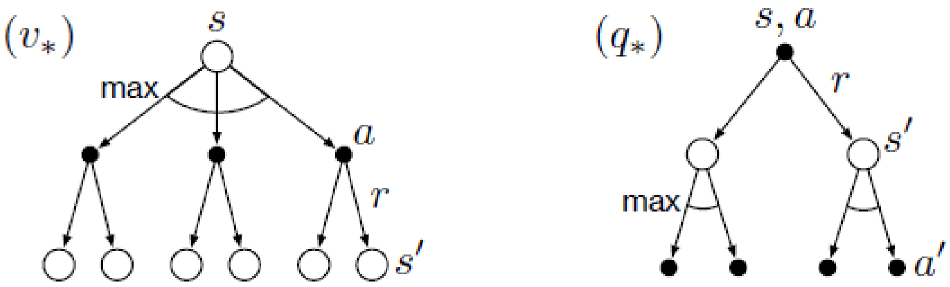


Figure 3.4: Backup diagrams for v_* and q_*

Thanks!

Any ***questions*** *?*