FPT Education
FPT UNIVERSITY

Optimal Policies and Value Function

Objectives

- Define an optimal policy
- Understand an optimal value function
- Understand how to use optimal value function to get optimal policies

Optimal Policies

- The role of policies in Reinforcement Learning (RL) is pivotal as they dictate the behavior of an agent within its environment.
  - Action Selection: Policies determine how an agent selects actions in different states of the environment.
  - Learning Objective: Policies define the learning objective for the RL agent.
  - Exploration vs. Exploitation: Policies balance exploration and exploitation

Optimal Policies

- The role of policies is pivotal as they dictate the behavior of an agent within its environment.
  - Evaluation of States and Actions: Policies implicitly or explicitly evaluate states and actions based on their expected rewards.
  - Adaptation to Changing Environments: Policies allow RL agents to adapt to changes in the environment.
  - Generalization: Policies can facilitate generalization across similar states.

Optimal Policies

- The role of policies is pivotal as they dictate the behavior of an agent within its environment.
  - Representation of Knowledge: In some RL algorithms, policies serve as a representation of the agent's knowledge about the environment.
  - Policy Improvement: In iterative RL algorithms like policy iteration or actor-critic methods, policies are updated iteratively to improve performance.

Policies Types

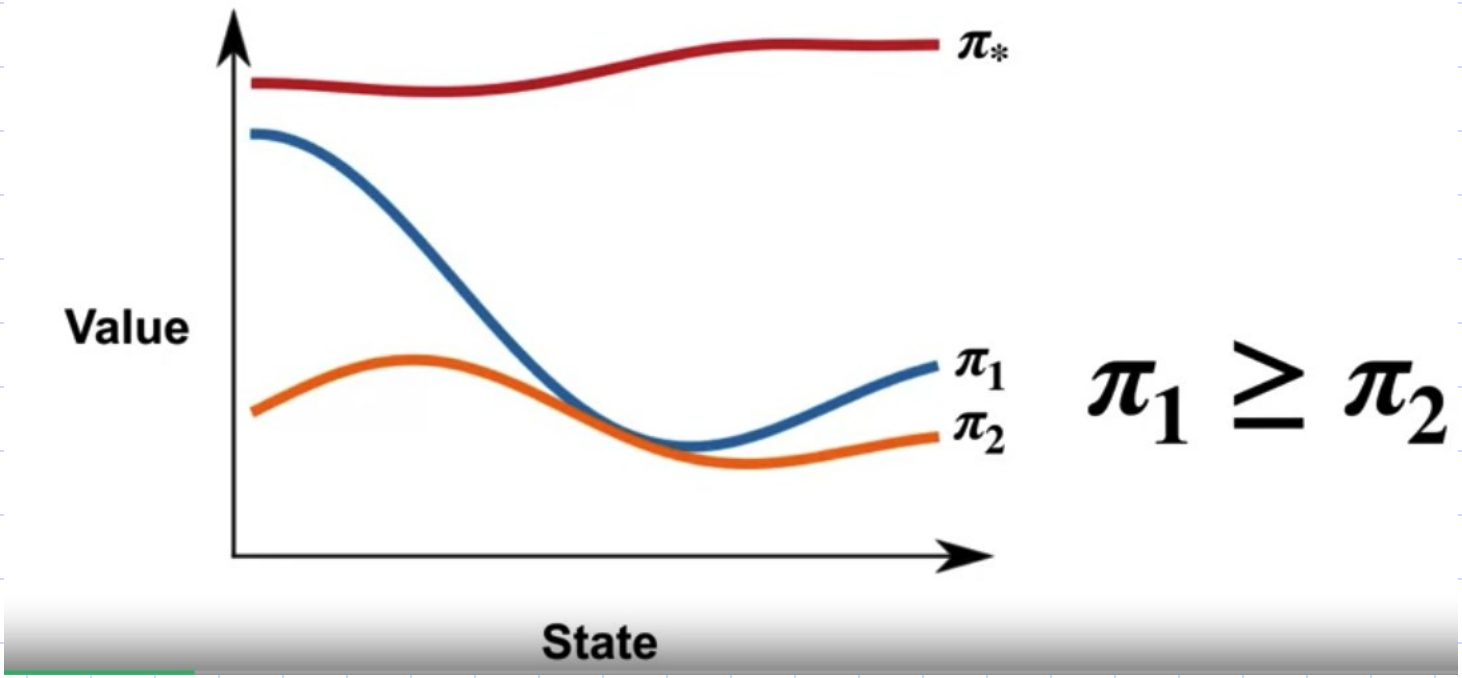
- **Deterministic Policy**: for each state, the agent selects a single action with certainty.
- **Stochastic Policy**: the agent selects actions based on a probability distribution over possible actions for each state.
- **Optimal Policy**: the policy that yields the highest expected cumulative reward over time.

Optimal Policies

- It refers to the policy that maximizes the expected cumulative reward over time.
- It is the policy that the RL agent should follow in order to achieve the best possible performance in the given environment.
- The optimal policy can be learned through various RL algorithms: Q-learning, policy gradient methods, and deep reinforcement learning methods.

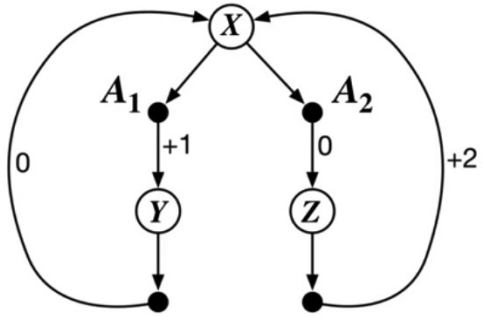
Optimal Policies

- An optimal policy Pi star is as good as or better than all the other policies



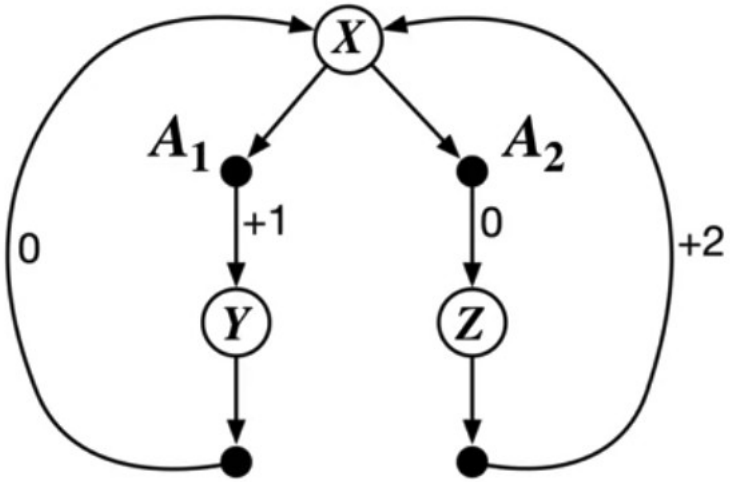
Optimal Policies

- Example to build some intuition about optimal policies.
  - The two choice MDP
  - From state X: action A1→ the agent to state Y. In state Y, only action A1 → agent back to state X.
  - From state X: action A2→ the agent to state Z. From state Z : action A1 → back to state X.
  - The numbers show the rewards after each action.



Optimal Policies

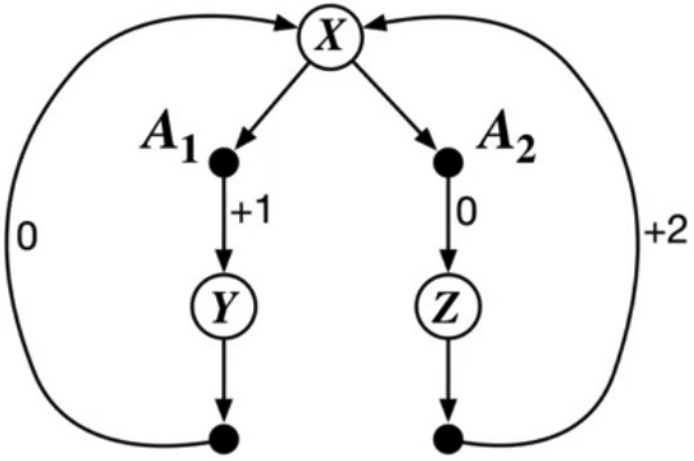
- Two deterministic policies: Pi1 and Pi2
- Which of these two policies is optimal?



$\pi_1(X) = A_1 \quad \pi_2(X) = A_2$

Optimal Policies

- Depends on the discount factor Gamma.



$\pi_1(X) = A_1 \quad \pi_2(X) = A_2$

$\gamma = 0$

$v_{\pi_1}(X) = 1$

$v_{\pi_2}(X) = 0$

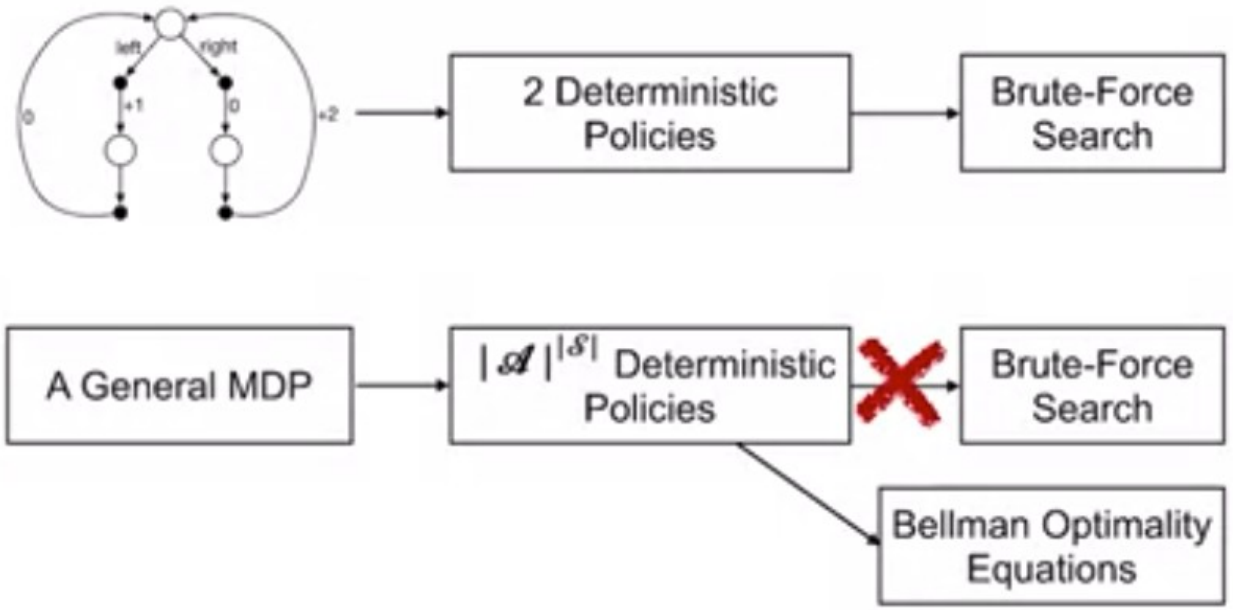
$\gamma = 0.9$

$$v_{\pi_1}(X) = \sum_{k=0}^{\infty} (0.9)^{2k} = \frac{1}{1 - 0.9^2} \approx 5.3$$

$$v_{\pi_2}(X) = \sum_{k=0}^{\infty} (0.9)^{2k+1} * 2 = \frac{0.9}{1 - 0.9^2} * 2 \approx 9.5$$

Optimal Policies

- However we can only directly solve small MDPs



Optimal Value Function

- The value function for the optimal policy thus has the greatest value possible: V star

Recall that

$\pi_1 \geq \pi_2$ if and only if $v_{\pi_1}(s) \geq v_{\pi_2}(s)$ for all $s \in \mathcal{S}$

v_* $v_{\pi_*}(s) \doteq \mathbb{E}_{\pi_*}[G_t \mid S_t = s] = \max_{\pi} v_\pi(s)$ for all $s \in \mathcal{S}$

Optimal Value Function

- The shared action value function by q star.

Recall that

$\pi_1 \geq \pi_2$ if and only if $v_{\pi_1}(s) \geq v_{\pi_2}(s)$ for all $s \in \mathcal{S}$

v_* $v_{\pi_*}(s) \doteq \mathbb{E}_{\pi_*}[G_t \mid S_t = s] = \max_{\pi} v_{\pi}(s)$ for all $s \in \mathcal{S}$

q_* $q_{\pi_*}(s,a) = \max_{\pi} q_{\pi}(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$

Optimal Value Function

- We can rewrite the equation in a special form, which doesn't reference the policy itself. This is Bellman Optimality Equation for V star

Recall that

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)\left[r + \gamma v_\pi(s')\right]$$

$$v_*(s) = \sum_a \pi_*(a|s) \sum_{s'} \sum_r p(s', r|s, a)\left[r + \gamma v_*(s')\right]$$

$$v_*(s) = \max_a \sum_{s'} \sum_r p(s', r|s, a)\left[r + \gamma v_*(s')\right]$$

Optimal Value Function

- The Bellman Optimality Equation for q star

Recall that

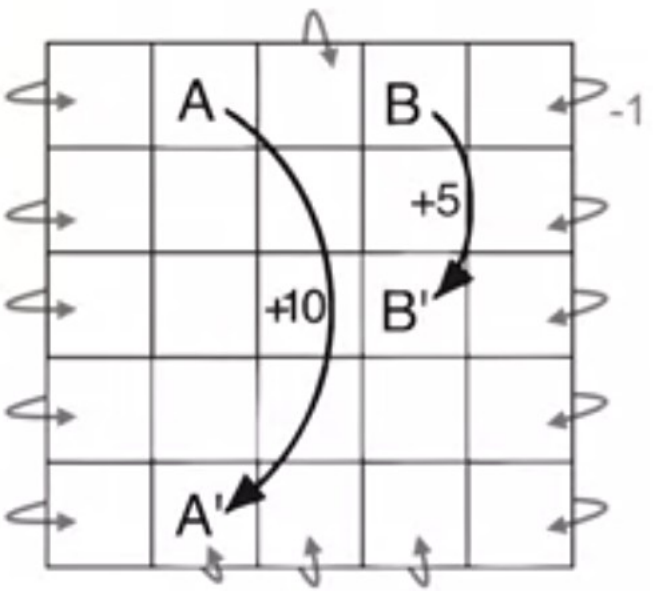
$$q_\pi(s,a) = \sum_{s'} \sum_{r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right]$$

$$q_*(s,a) = \sum_{s'} \sum_{r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi_*(a'|s') q_*(s',a') \right]$$

$$q_*(s,a) = \sum_{s'} \sum_{r} p(s',r|s,a) \left[r + \gamma \max_{a'} q_*(s',a') \right]$$

Relationship

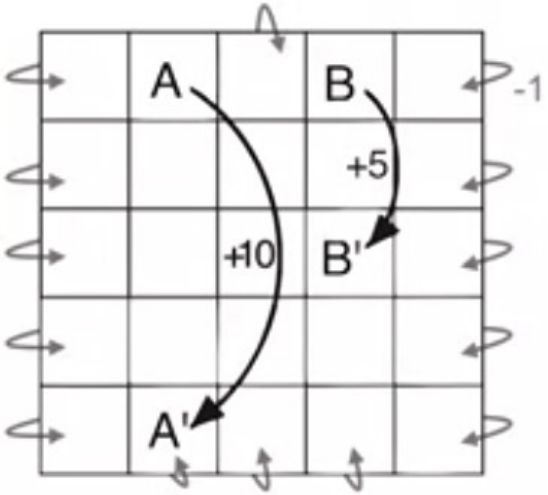
- Example



- All actions in state A transition to state A prime with a reward of +10.
- State B, all actions transition to B prime with a reward of +5.
- The reward is zero everywhere else except for -1, for bumping into the walls

Relationship

- The discount factor is 0.9
- The associated optimal values for each state.



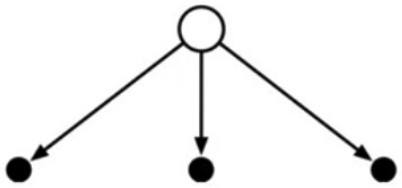
$\gamma = 0.9$

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*

Relationship

- v star is equal to the maximum of the boxed term over all actions.
- Pi star is the argmax, which simply means the particular action which achieves this maximum.

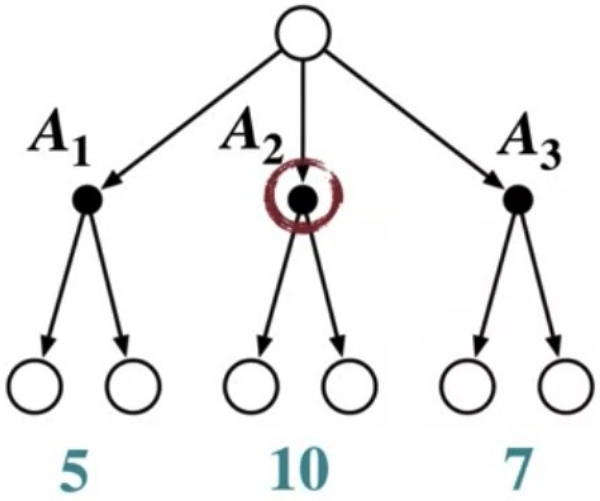


$$v_*(s) = \max_a \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

$$\pi_*(s) = \operatorname{argmax}_a \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

Relationship

- We can determine an optimal policy



$$v_*(s) = \max_a \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

$$\pi_*(s) = \operatorname{argmax}_a \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

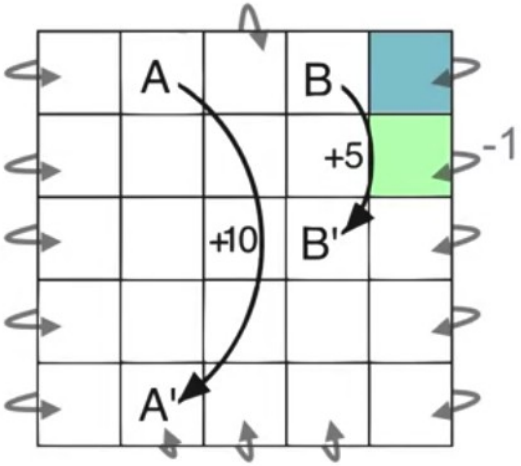
Relationship

- Calculate

$$\pi_*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

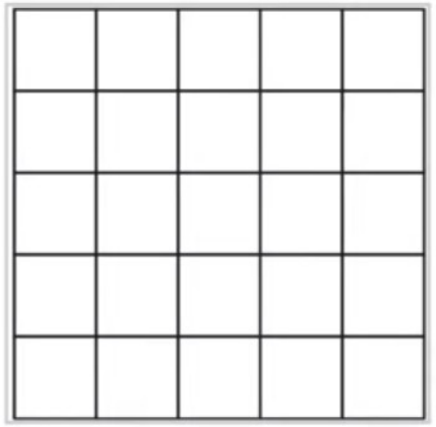
$\gamma = 0.9$

$0 + 0.9 * 17.5 = 14.0$



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

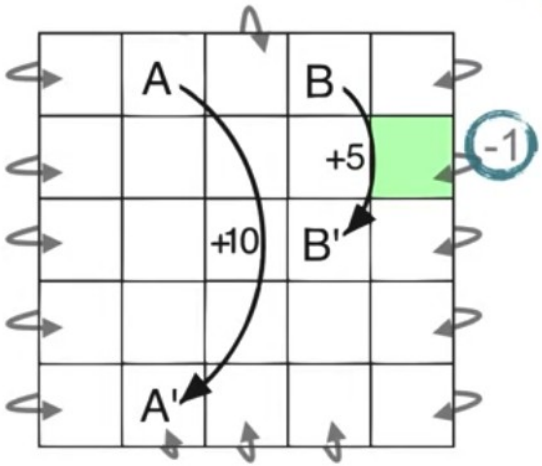
Relationship

- Calculate

$$\pi_*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

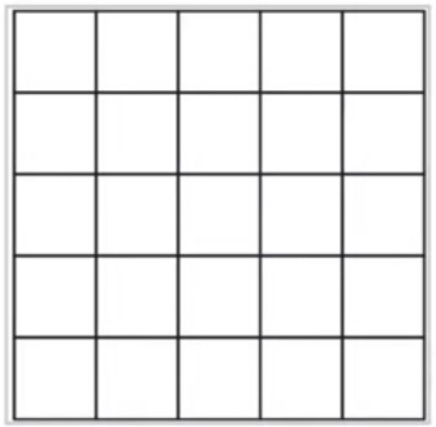
$\gamma = 0.9$

$-1 + 0.9 * 16.0 = 13.4$



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

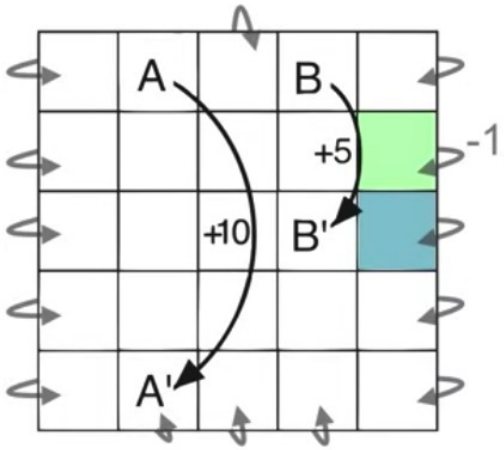
Relationship

- Calculate

$$\pi_*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

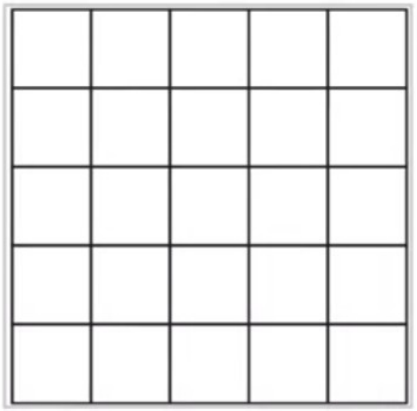
$$\gamma = 0.9$$

$$0 + 0.9 * 14.4 = 13.0$$



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



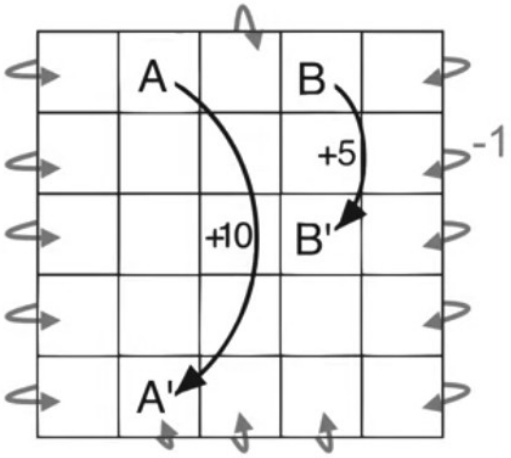
π_*

Relationship

- Finding the optimal policies

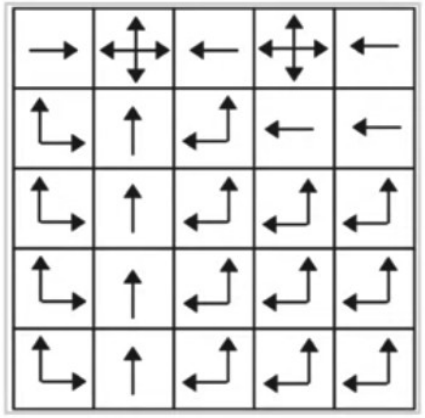
$$\pi_*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

$\gamma = 0.9$



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

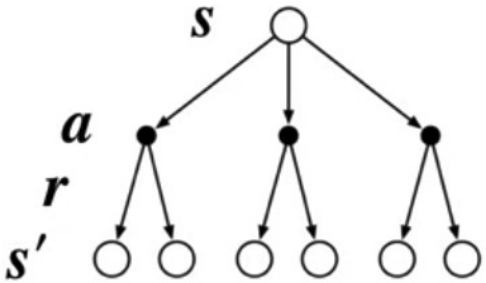
v_*



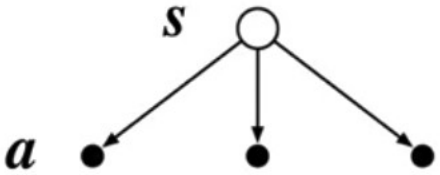
π_*

Relationship

- Determining an optimal policy



$$\pi_*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$



$$\pi_*(s) = \underset{a}{\operatorname{argmax}}\, q_*(s, a)$$

Relationship

- Relationship between Value Function and Optimal Policies
  - The value function provides information about the expected return that an agent can achieve from different states
  - Optimal policies dictate the best actions to take in each state to maximize the expected return.
  - *The value function provides critical information about the expected return from different states, which guides the selection of actions in order to derive optimal policies that maximize the cumulative reward over time.*

Summary

- Define an optimal policy
- Understand an optimal value function
- Understand how to use optimal value function to get optimal policies

FPT Education
FPT UNIVERSITY

Q & A