FPT Education

FPT UNIVERSITY

Policy Iteration

FPT Education
FPT UNIVERSITY

Objectives

- Understand the policy improvement theorem
- Use a value function for a policy to produce a better policy for a given MDP
- Apply policy iteration to compute optimal policies and optimal value functions

Policy Improvement

- We can find the optimal policy by choosing the Greedy action.
- The Greedy action maximizes the Bellman's optimality equat

Recall that **Greedy action**

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

?

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

$$\pi(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma v_\pi(s')\right] \text{ for all } s \in \mathcal{S}$$

$\rightarrow v_\pi$ **obeys the Bellman optimality equation** $\rightarrow \pi$ **is optimal**

Policy Improvement

- The new policy obtained in this way must be a strict improvement on Pi, unless Pi was already optimal.
- Policy improvement theorem: q Pi: value of a state if you take action A, and then follow policy Pi.

$$q_\pi\left(s, \pi'(s)\right) \geq q_\pi\left(s, \pi(s)\right) \text{ for all } s \in \mathcal{S}$$

Policy Improvement

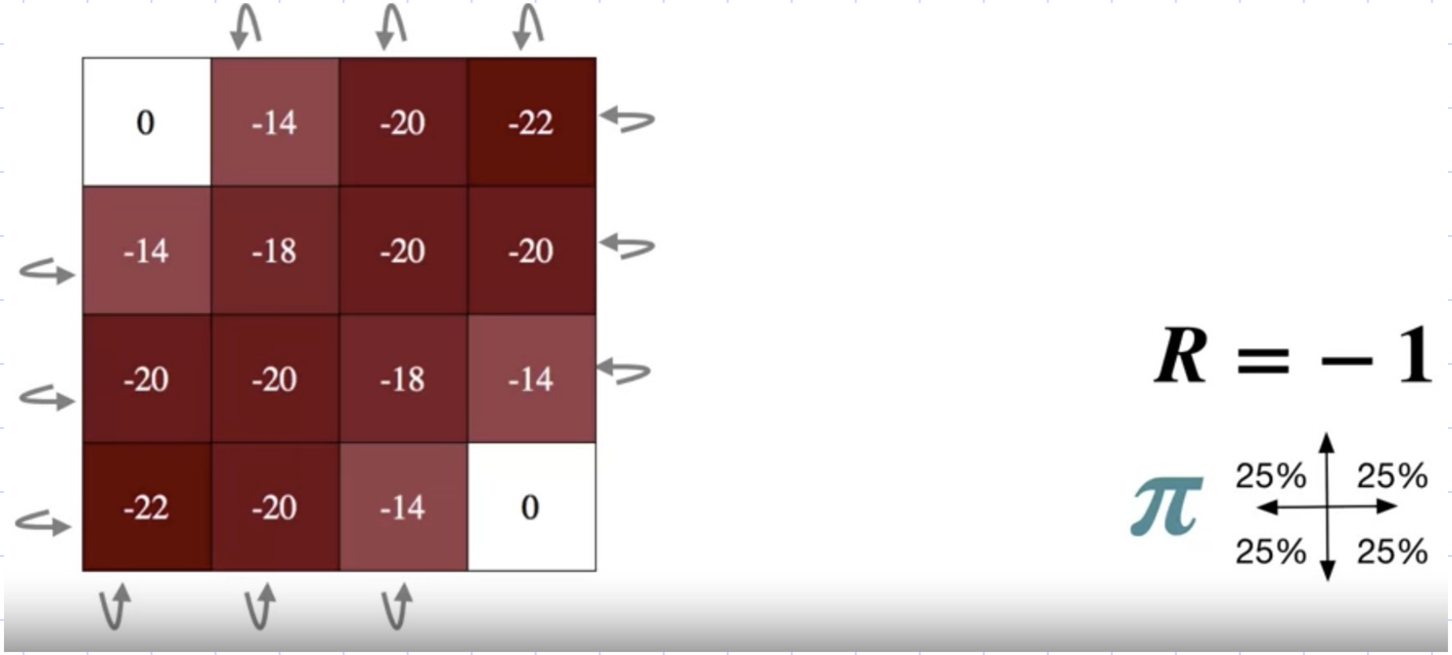
- Policy Pi prime is at least as good as Pi if in each state, the value of the action selected by Pi prime is greater than or equal to the value of the action selected by Pi.
- Policy pi prime is strictly better if the value is strictly greater and at least one state

$$q_\pi(s, \pi'(s)) \geq q_\pi(s, \pi(s)) \text{ for all } s \in \mathcal{S} \rightarrow \pi' \geq \pi$$

$$q_\pi(s, \pi'(s)) > q_\pi(s, \pi(s)) \text{ for at least one } s \in \mathcal{S} \rightarrow \pi' > \pi$$

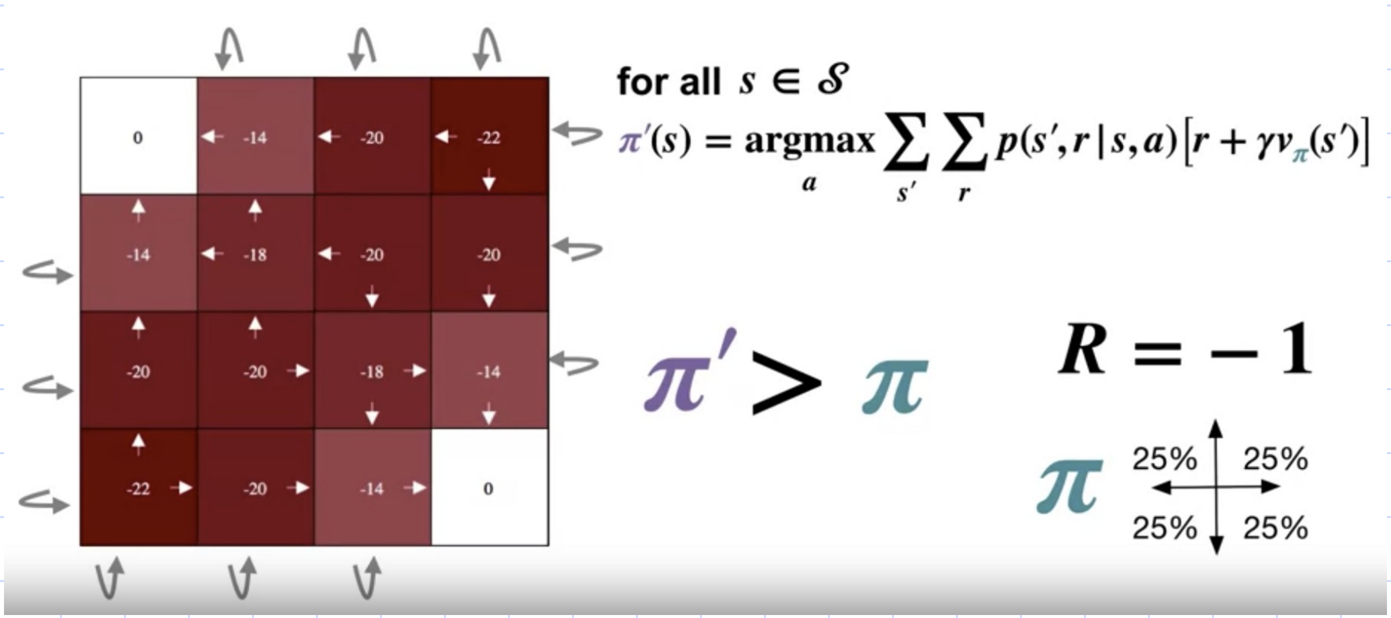
Policy Improvement

- For example the four-by-four grid. The final value function:



Policy Improvement

- The greedy Pi policy



Policy Iteration

- Policy improvement theorem

Greedy action

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

π' is strictly better than π unless π was already optimal

Policy Iteration

- Begin with the policy Pi 1→ evaluate Pi 1 using iterative policy evaluation to obtain the state value, V Pi 1. --> evaluation step.
- Using the results of the policy improvement theorem, we can then greedify with respect to v Pi 1 to obtain a better policy, Pi 2--> We call this the improvement step.
- We can then compute V Pi 2 and use it to obtain an even better pol f better policies.

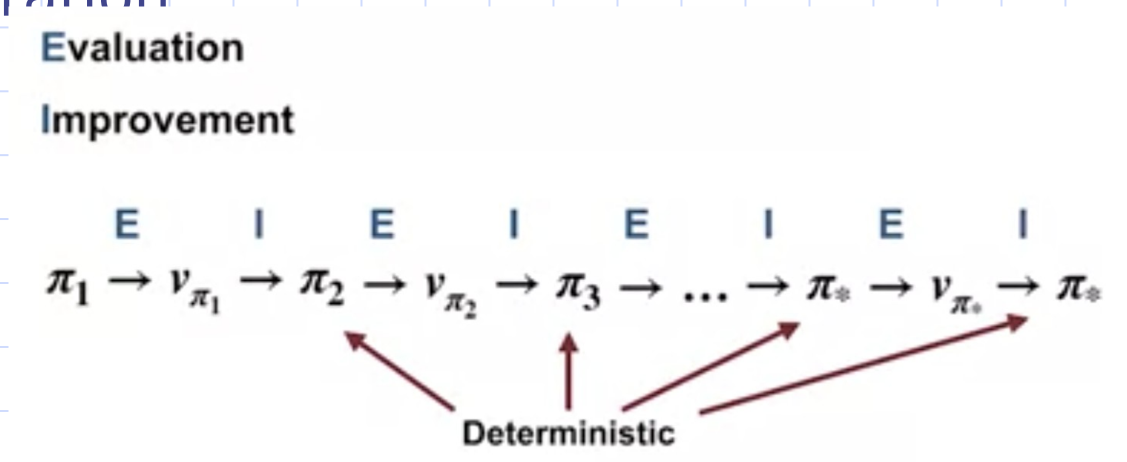
Evaluation

Improvement

$$\pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} v_{\pi_2} \xrightarrow{I} \pi_3 \xrightarrow{E} \dots \xrightarrow{I} \pi_*$$

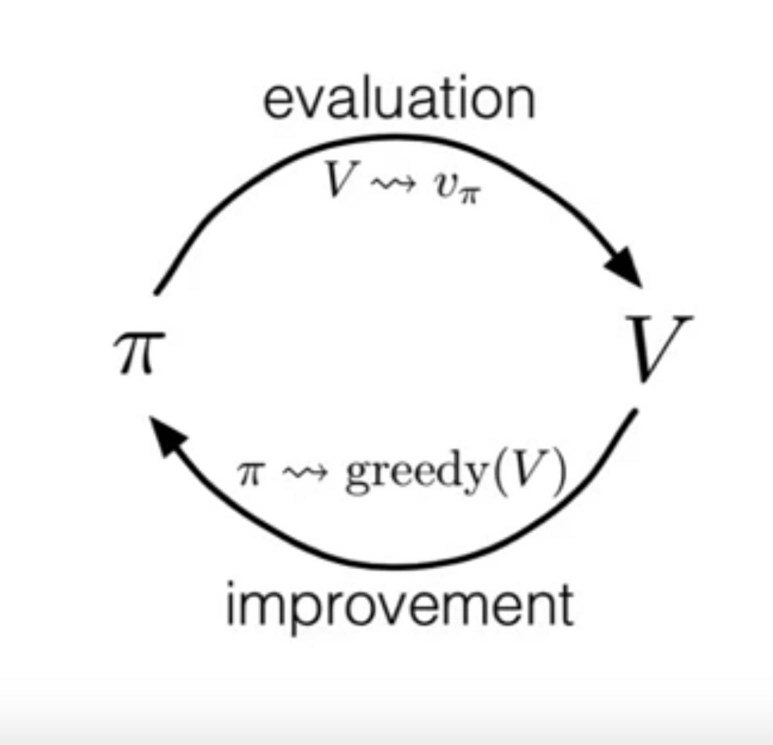
Policy Iteration

- We complete an iteration, and the policy remains unchanged→ found the optimal policy.
- Each policy generated in this way is deterministic.
- This method of finding an optimal policy is called policy iteration



Policy Iteration

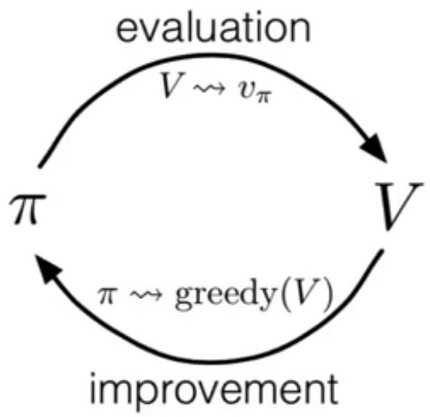
- Policy iteration consists of two distinct steps: evaluation and improvement.



Policy Iteration

- We first evaluate our current policy, Pi 1, which gives us a new value function that accurately reflects the value of Pi 1.

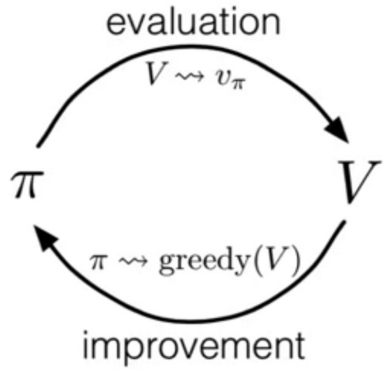
$$\pi_1 \longrightarrow v_{\pi_1}$$



Policy Iteration

- The improvement step then uses V Pi 1 to produce a greedy policy Pi 2. At this point, Pi 2 is greedy with respect to the value function of Pi 1, but V Pi 1 no longer accurately reflects the value of Pi 2.

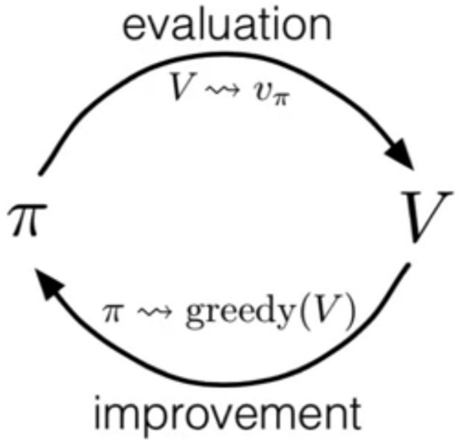
$$\pi_2 \leftarrow v_{\pi_1}$$



Policy Iteration

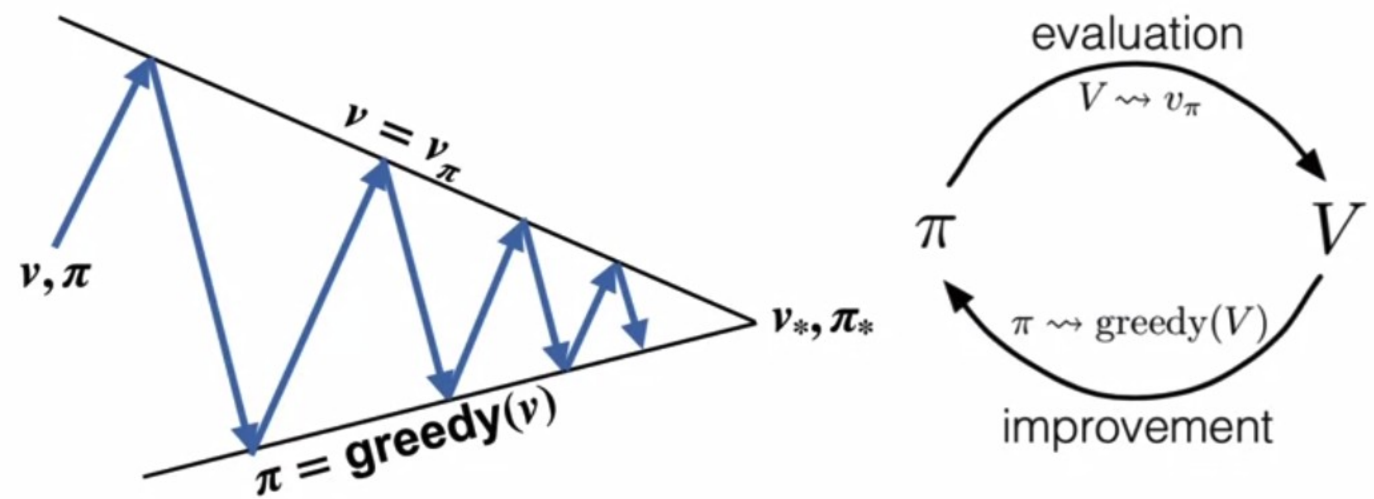
- The policy is greedy and the value function is accurate.

$$\pi_* \longleftrightarrow v_*$$



Policy Iteration

- Visualization of policy iteration



Policy Iteration

- Policy Iteration using iterative policy evaluation for estimate pi ~ pi*

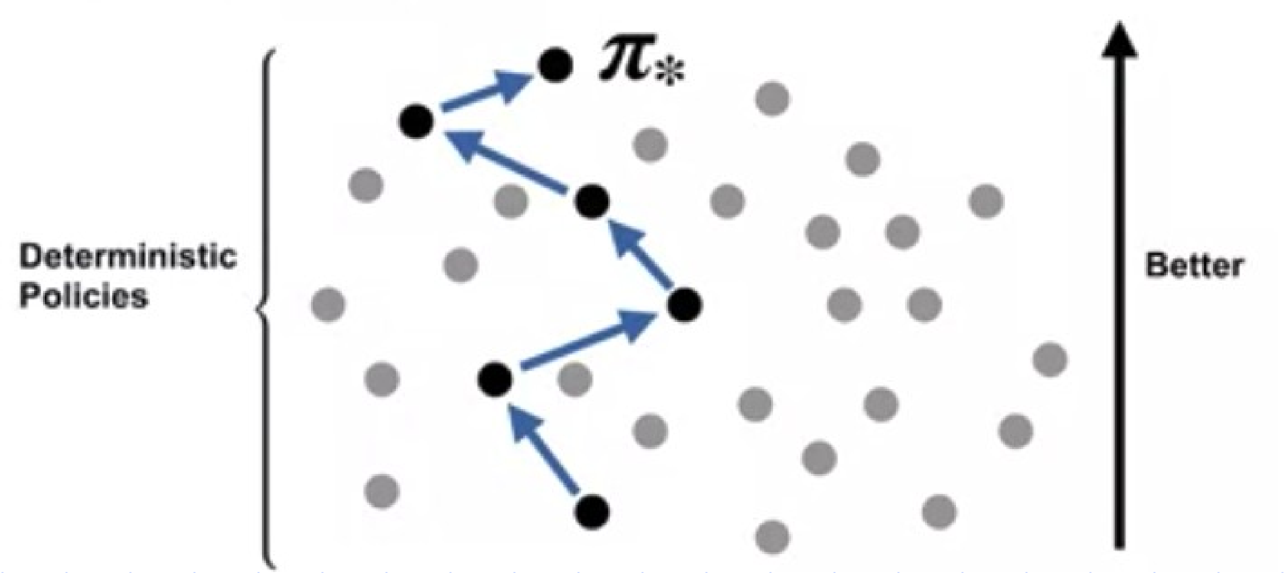
1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\quad \Delta \leftarrow 0$
   $\quad$ Loop for each $s \in \mathcal{S}$:
   $\quad\quad v \leftarrow V(s)$
   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))\big[r + \gamma V(s')\big]$
   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Policy Iteration

- The power of policy iteration



Summary

- Understand the policy improvement theorem
- Use a value function for a policy to produce a better policy for a given MDP
- Apply policy iteration to compute optimal policies and optimal value functions

FPT Education
FPT UNIVERSITY

Q & A