FPT Education

FPT UNIVERSITY

Policy Evaluation

Objectives

- Understand the distinction between policy evaluation and control
- Explain the setting in which dynamic programming can be applied, as well as its limitations
- Apply iterative policy evaluation to compute value functions

Policy Evaluation vs. Control

- Policy evaluation is the task of determining the value function for a specific policy.
- Control is the task of finding a policy to obtain as much reward as possible: finding a policy which maximizes the value function.

Policy Evaluation vs. Control

- Imagine someone hands you a policy and your job is to determine how good that policy is. Policy evaluation is the task of determining the state value function v pi for a particular policy pi.

$$\pi \longrightarrow v_\pi$$

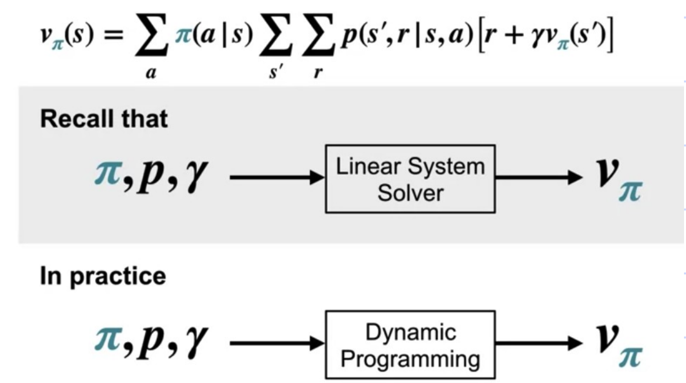
Recall that

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

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

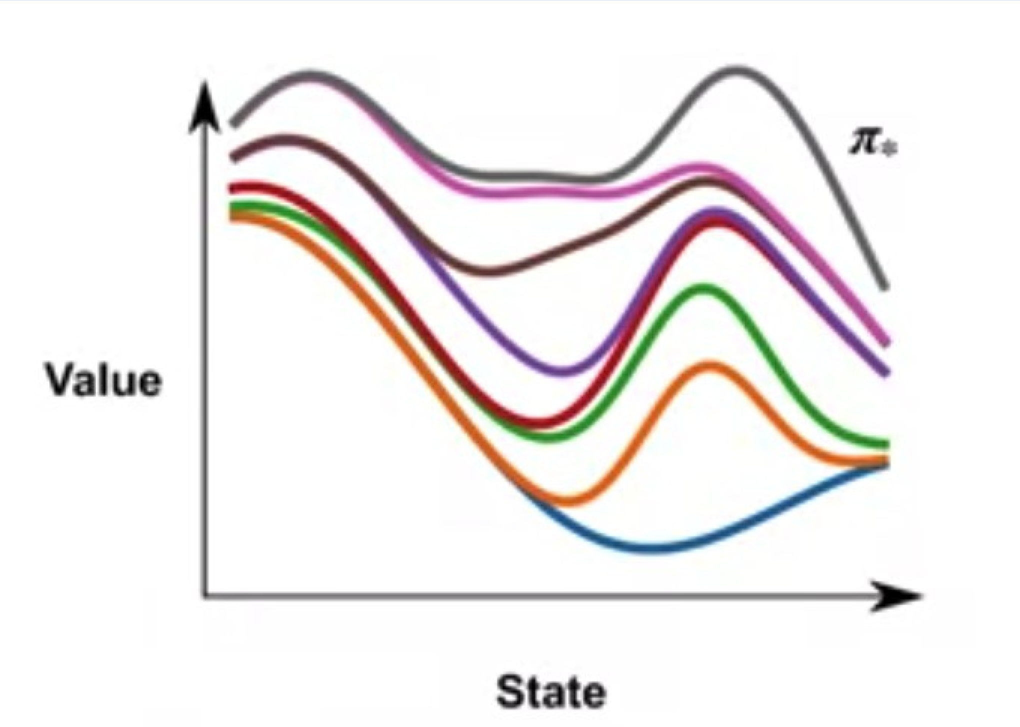
Policy Evaluation vs. Control

- The Bellman equation reduces the problem of finding v pi to a system of linear equations
- The problem of policy evaluation reduces to solving this system of linear equations.
- The iterative solution methods of dynamic programming are more suitab



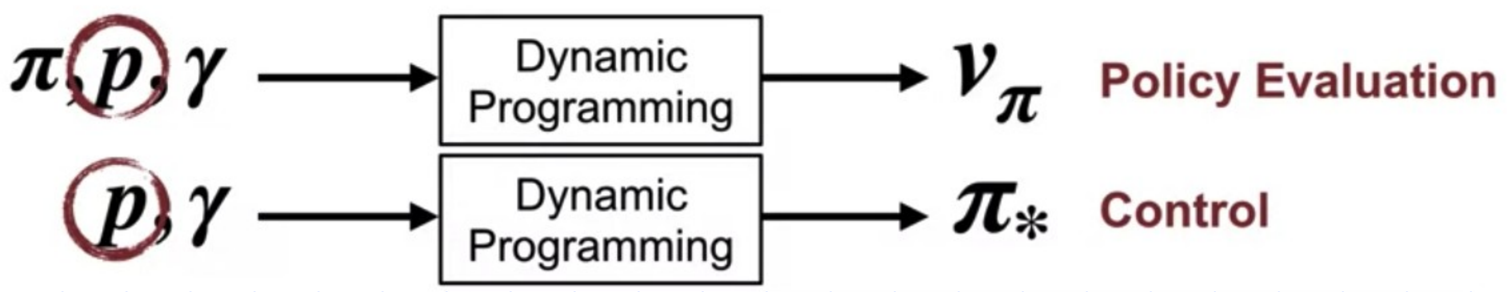
Policy Evaluation vs. Control

- Control is the task of improving a policy..



Policy Evaluation vs. Control

- The dynamics of the environment: p
- We use dynamic programming methods to compute value functions and optimal policies given a model of the MDP..



Iterative Policy Evaluation

- The Bellman equation gives us a recursive expression for V Pi.
- Iteratively refine our estimate of the value function.

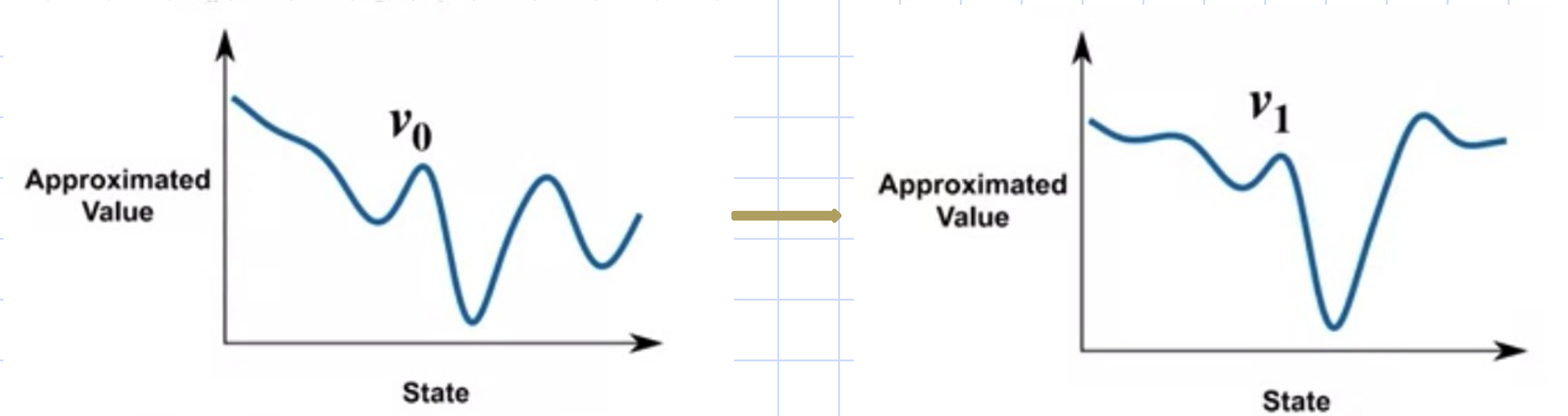
$$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_\pi(s')\right]$$

$$\downarrow$$

$$v_{k+1}(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_k(s')\right]$$

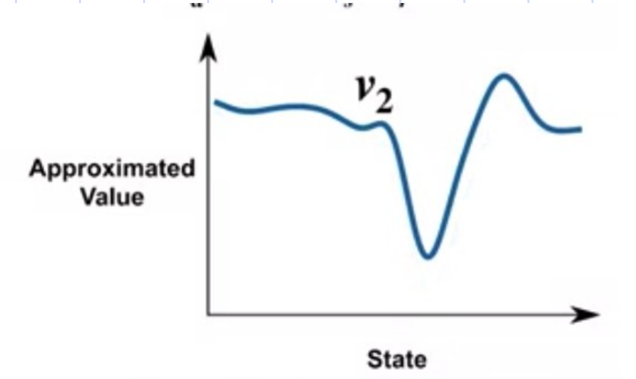
Iterative Policy Evaluation

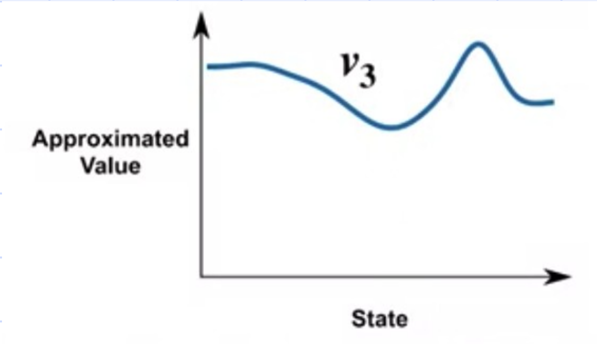
- An arbitrary initialization for our approximate value function- v_0
- Using update rule: $v_{k+1}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \sum_r p(s',r|s,a)[r+\gamma v_k(s')]$

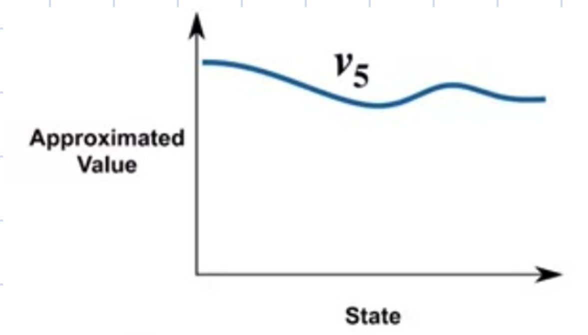


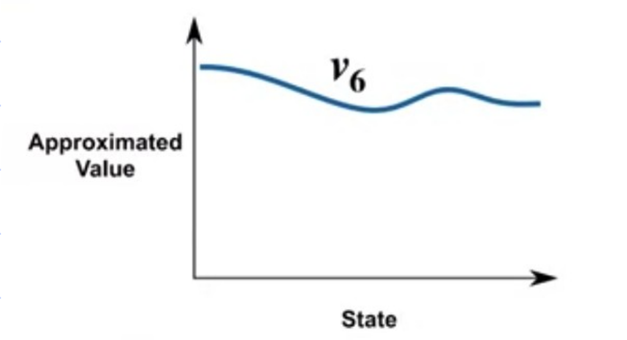
Iterative Policy Evaluation

- Update V using interactive





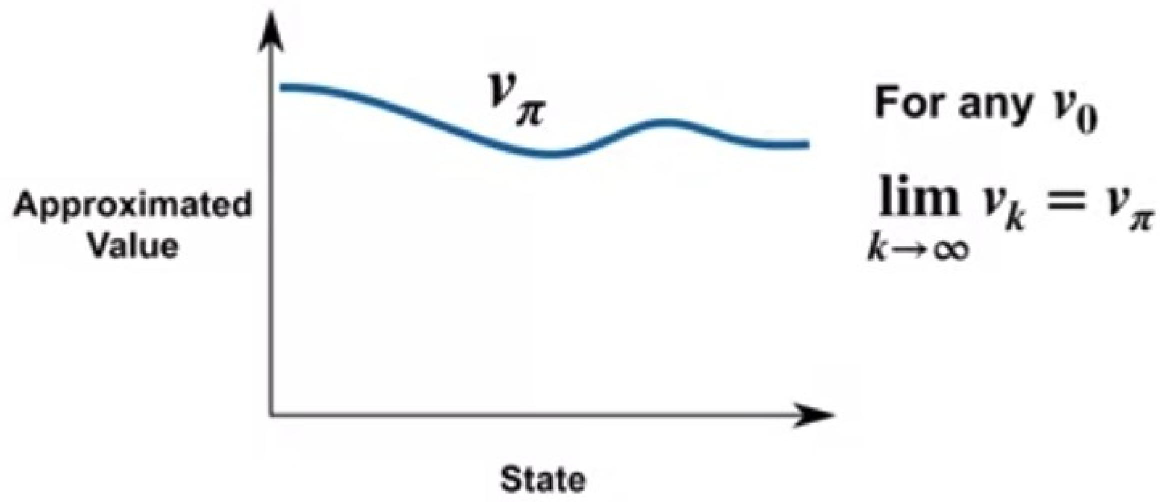




Iterative Policy Evaluation

- For any choice of v_0, v_k will converge to v Pi in the limit as k approaches infinity.

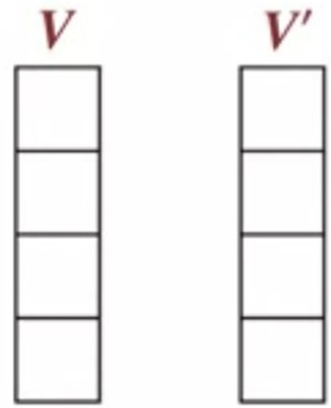
$$v_k(s) = \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma v_k(s')\right]$$



Iterative Policy Evaluation

- To implement iterative policy evaluation, we store two arrays.
  - One array- V stores the current approximate value function.
  - Another array, V prime, stores the updated values.

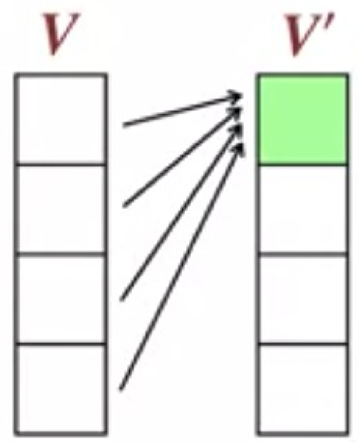
$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

- By using two arrays, we can compute the new values from the old one state at a time without the old values being changed in the process.

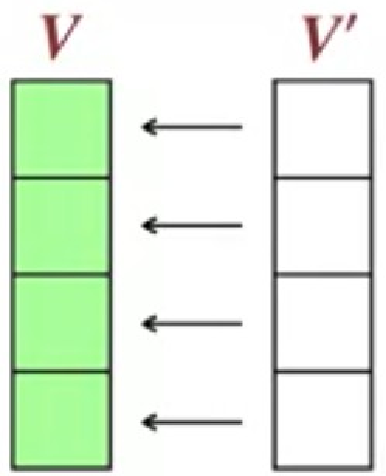
$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

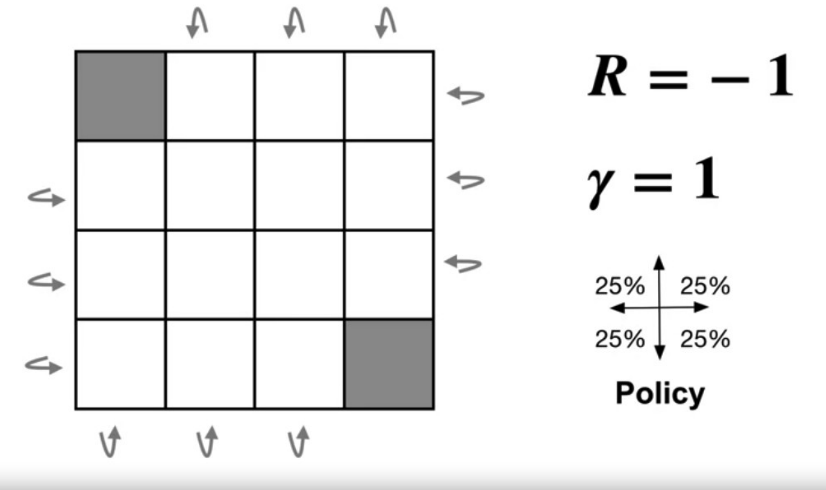
- At the end of a full sweep, we can write all the new values into V; then we do the next iteration..

$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

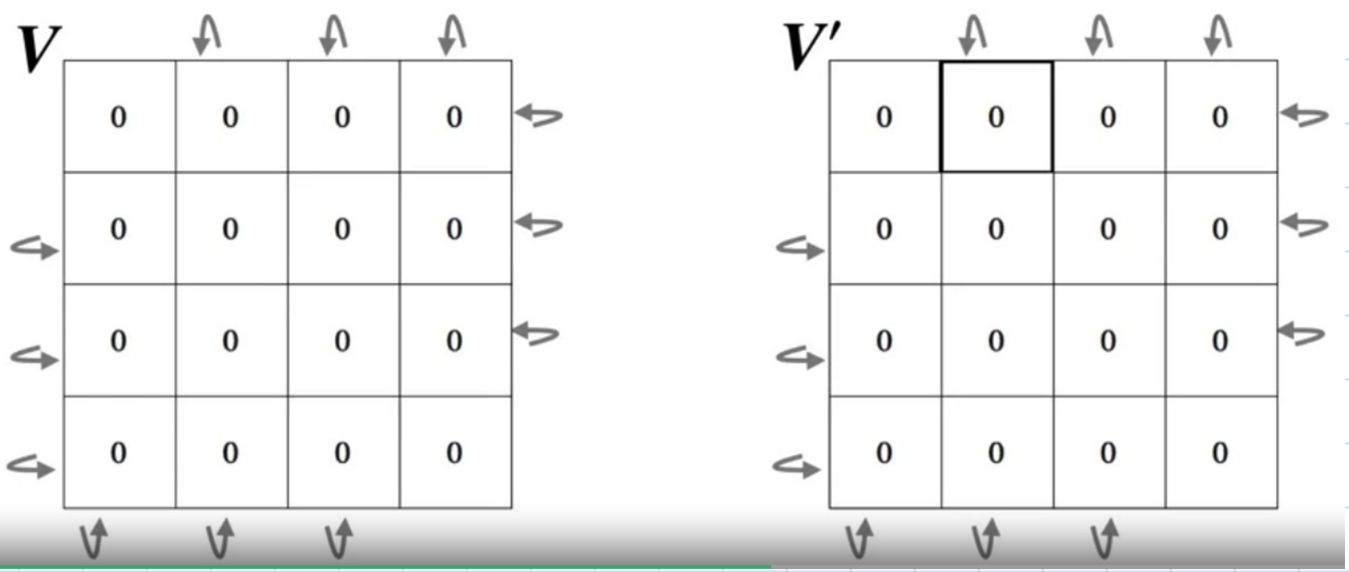
- The four-by-four grid world
- The terminal state located in the top left and bottom right corners.
- The reward will be minus one for every transition.
- gamma equals 1
- Four possible actions in each state up, down, left, and right.



Iterative Policy Evaluation

- Calculate:

$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

- Calculate:

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

$$0.25 * (-1+0) + 0.25 * (-1+0) + 0.25 * (-1+0) + 0.25 * (-1+0) = -1$$

V

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V'

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Iterative Policy Evaluation

- Calculate:

$$V'(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$

$$0.25 * (-1 + 0) + 0.25 * (-1 + 0) + 0.25 * (-1 + 0) + 0.25 * (-1 + 0) = -1$$

V

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

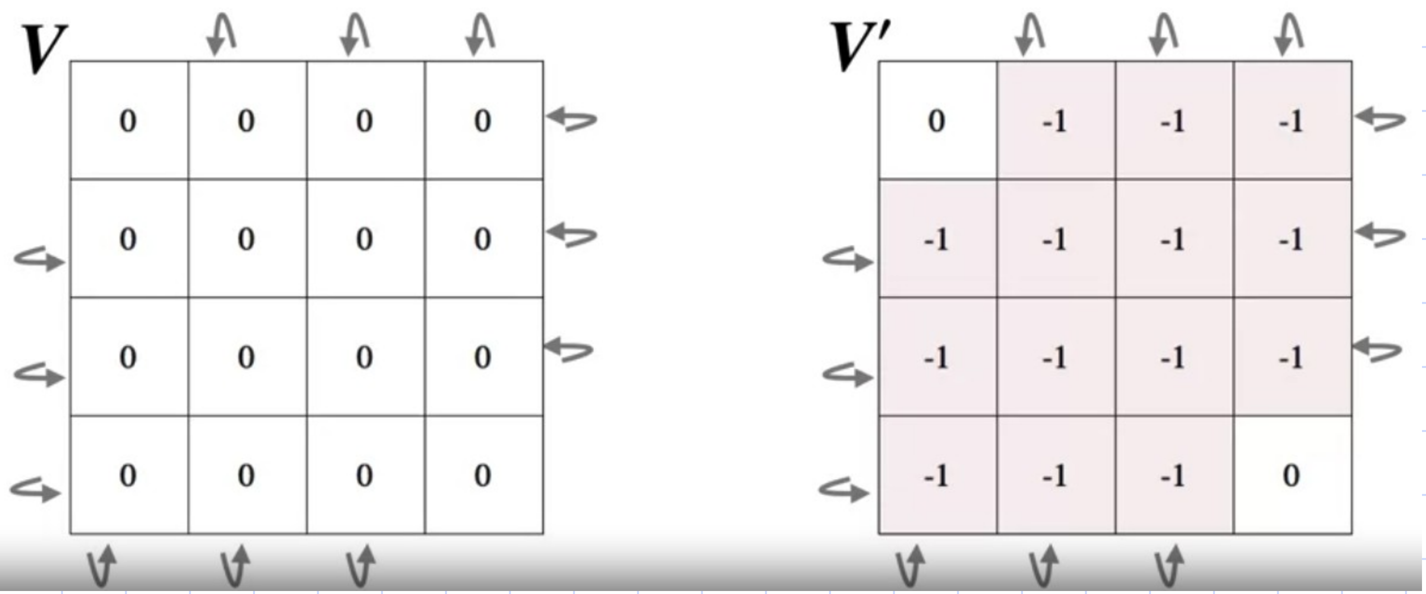
V'

0	-1	-1	0
0	0	0	0
0	0	0	0
0	0	0	0

Iterative Policy Evaluation

- Calculate:

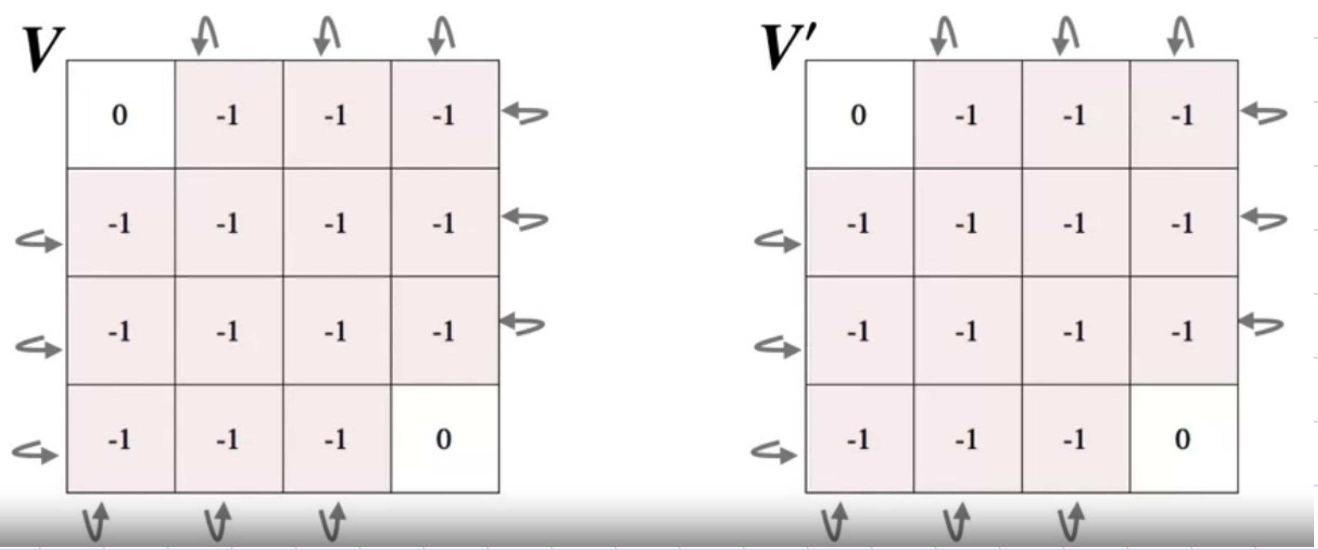
$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

- After completing this full sweep, we copy the updated values from V prime to V:

$$V'(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

- Iterative Policy Evaluation, for estimating V ~ Vpi:

Input π, the policy to be evaluated

$V \leftarrow \vec{0}, V' \leftarrow \vec{0}$

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

$$V'(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma V(s')]$$

$$\Delta \leftarrow \max(\Delta, |V'(s) - V(s)|)$$

$V \leftarrow V'$

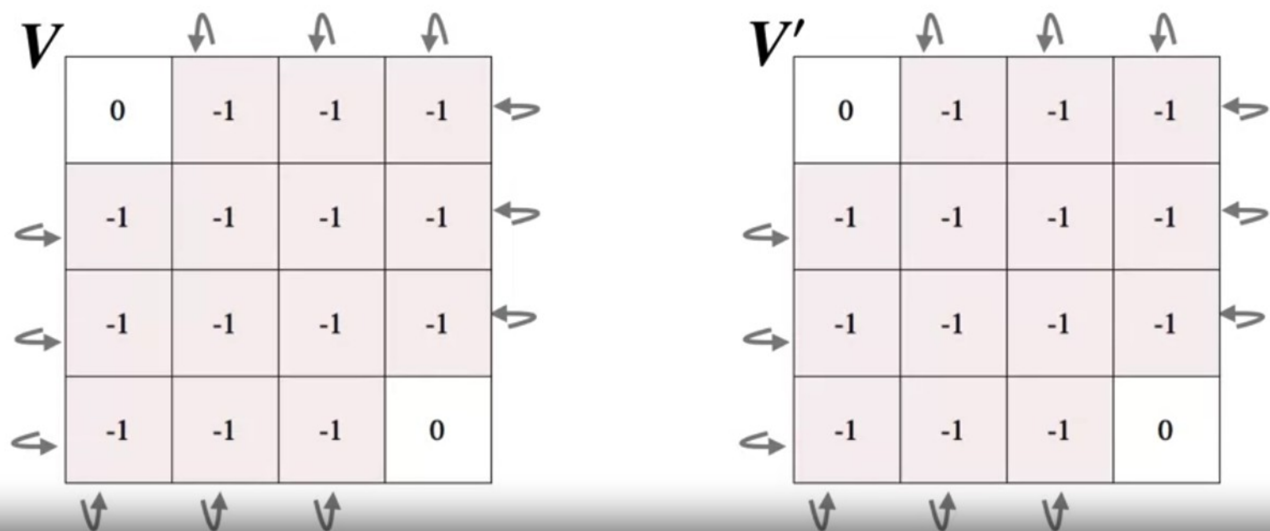
until $\Delta < \theta$ (a small positive number)

Output $V \approx v_\pi$

Iterative Policy Evaluation

- Our value of 0.001 for the stopping parameter theta
- $$V'(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s'} \sum_{r} p(s',r|s,a)\left[r + \gamma V(s')\right]$$

  $$\theta = 0.001 \quad \Delta = 1.0$$



Iterative Policy Evaluation

- The second sweep

$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$

V

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	0

V'

0	-1.7	-2	-2
-1.7	-2	-2	-2
-2	-2	-2	-1.7
-2	-2	-1.7	0

Iterative Policy Evaluation

- One more sweep

$$V'(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$

V

0	-1.7	-2	-2
-1.7	-2	-2	-2
-2	-2	-2	-1.7
-2	-2	-1.7	0

V'

0	-2.4	-2.9	-3
-2.4	-2.9	-3	-2.9
-2.9	-3	-2.9	-2.4
-3	-2.9	-2.4	0

Iterative Policy Evaluation

- More sweep

$$V'(s) \leftarrow \sum_{a} \pi(a \mid s) \sum_{s'} \sum_{r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$

V

0	-2.4	-2.9	-3
-2.4	-2.9	-3	-2.9
-2.9	-3	-2.9	-2.4
-3	-2.9	-2.4	0

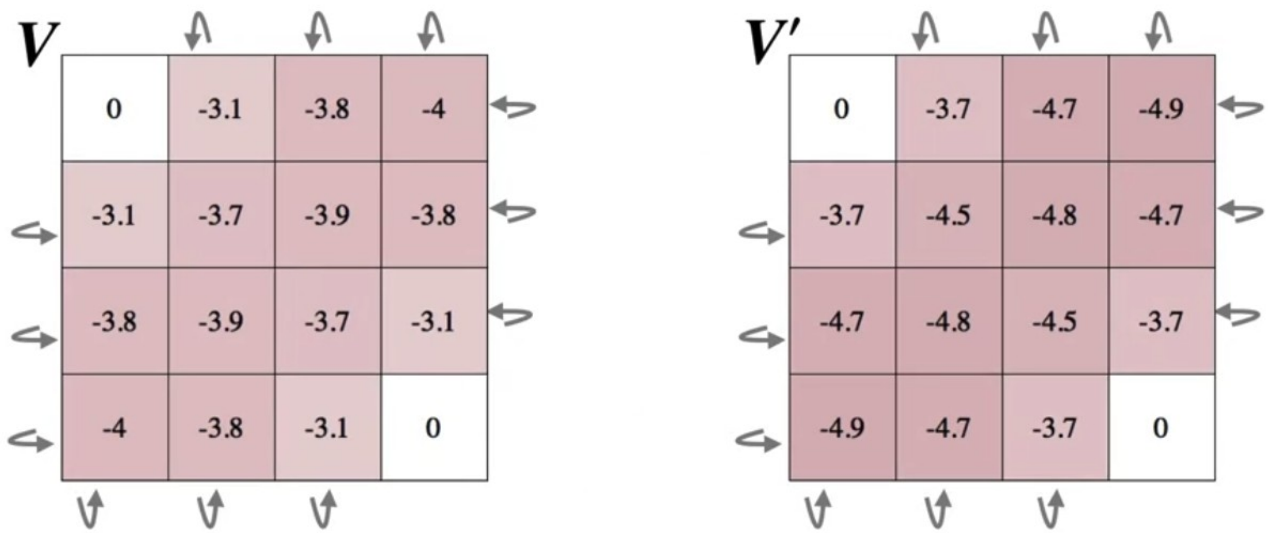
V'

0	-3.1	-3.8	-4
-3.1	-3.7	-3.9	-3.8
-3.8	-3.9	-3.7	-3.1
-4	-3.8	-3.1	0

Iterative Policy Evaluation

- More sweep

$$V'(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$



Iterative Policy Evaluation

- More sweep

$$V'(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma V(s')\right]$$

V

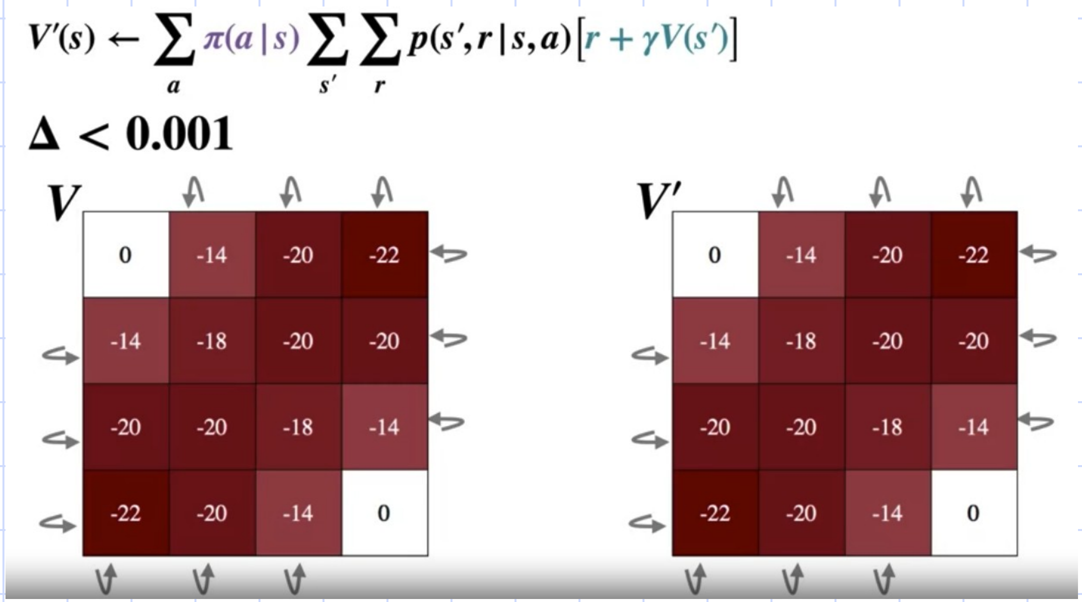
0	-3.7	-4.7	-4.9
-3.7	-4.5	-4.8	-4.7
-4.7	-4.8	-4.5	-3.7
-4.9	-4.7	-3.7	0

V'

0	-4.2	-5.5	-5.8
-4.2	-5.2	-5.6	-5.5
-5.5	-5.6	-5.2	-4.2
-5.8	-5.5	-4.2	0

Iterative Policy Evaluation

- Keep running until our maximum delta is less than theta.



- Our approximate value function has converged to the value function for the random policy

Summary

- Understand the distinction between policy evaluation and control
- Explain the setting in which dynamic programming can be applied, as well as its limitations
- Apply iterative policy evaluation to compute value functions

FPT Education
FPT UNIVERSITY

Q & A