AML, 22 Oct 2019

Markov Decision Processes

State dependent rewards

$$p(s', r \mid s, a)$$

At s, action a leads to s' with reward r

Assume states, rewards are finite

Policy – "strategy to choose action"

Given a policy π, $V_\pi(s)$ - expected long term value of state s

Over a path, we discount rewards

$$G_t = R_{t+1} + \gamma G_{t+1}$$

$$v_\pi(s) = E_\pi [G_t | S_t = s]$$

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

$$q_\pi(s,a) = \dots$$

Optimal policy

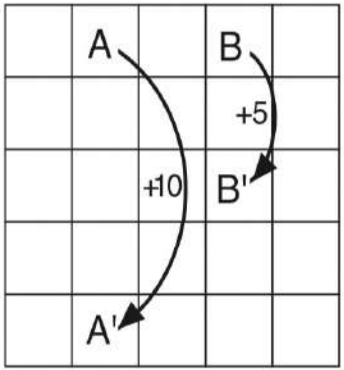
$\pi_1 \geq \pi_2$ if $\forall s.\ v_{\pi_1}(s) \geq v_{\pi_2}(s)$

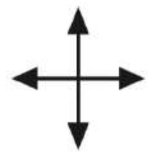
Optimal value function

$$v_*(s) = \max_a E\left[R_{t+1} + \gamma v_*(S_{t+1}) \,\middle|\, S_t = s, A_t = a\right]$$

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

$q_*(s,a) = \ldots$





N, S, E, W

v_π

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

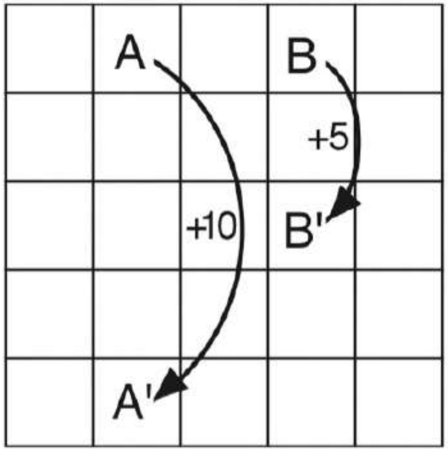
Uniform policy π ↑

At A, B all actions move as shown

All other rewards are 0

Except, -1 for hitting edge

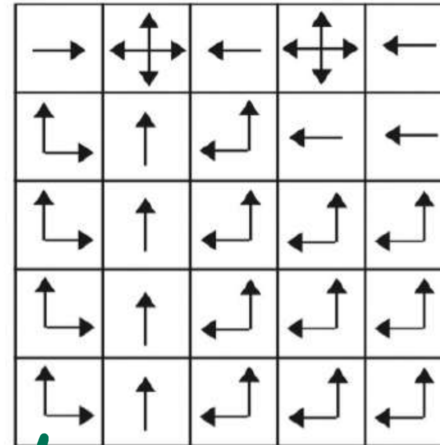
$\pi \rightarrow v_\pi$



Gridworld

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_*

Same Example

Choose ↑ →
with any probability

0 prob for all nonoptimal dir.

Computing v_π from π

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

Bellman Eqn

Provided $\gamma < 1$, this set of eqns has a unique solution

If we have episodic system with finite paths, assume a terminal state with value 0

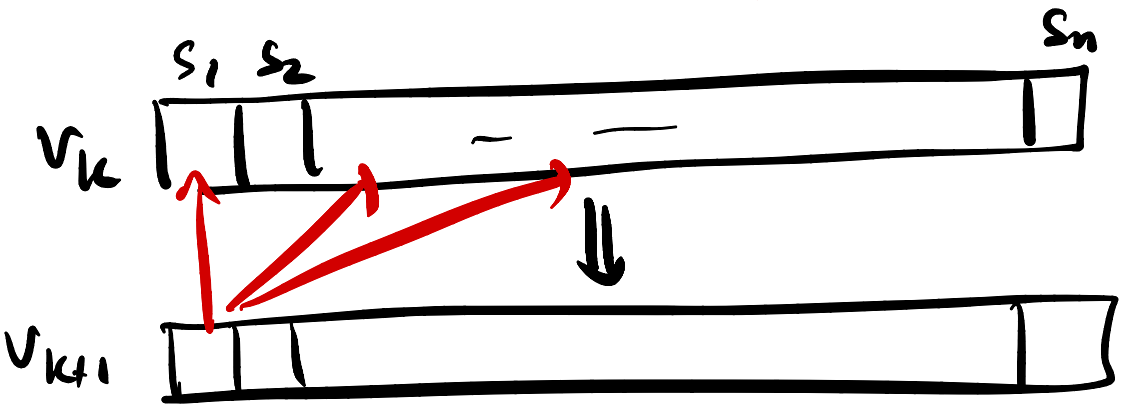
S original states, S^+ with added terminal state

Iterative soln

Treat equation as update rule

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

Update all $v_{k+1}(s)$ in parallel



Repeat until convergence ($\Delta \leq \varepsilon$)

Each update: $\Delta = \max_s \left(|V_{k+1}(s) - V_k(s)| \right)$

$\pi \rightarrow V_\pi$: Policy evaluation

Convergence guaranteed

In practice — need not update in parallel

Sequentially update $V_{k+1}(s_1)$, $V_{k+1}(s_2)$ ...

Goal is to reach v_*, π_*

Policy update

$$\pi \longrightarrow v_\pi$$

At s, choose action $\pi'(s)$ instead of $\pi(s)$ s.t

$$q(s, \pi'(s)) \geq v_\pi(s)$$

$$\Rightarrow \forall s \quad v_{\pi'}(s) \geq v_\pi(s)$$

Greedy policy update "works"


0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
random policy
0.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 0.0
greedy update.
0.0 -1.7 -2.0 -2.0
-1.7 -2.0 -2.0 -2.0
-2.0 -2.0 -2.0 -1.7
-2.0 -2.0 -1.7 0.0

$k = 3$

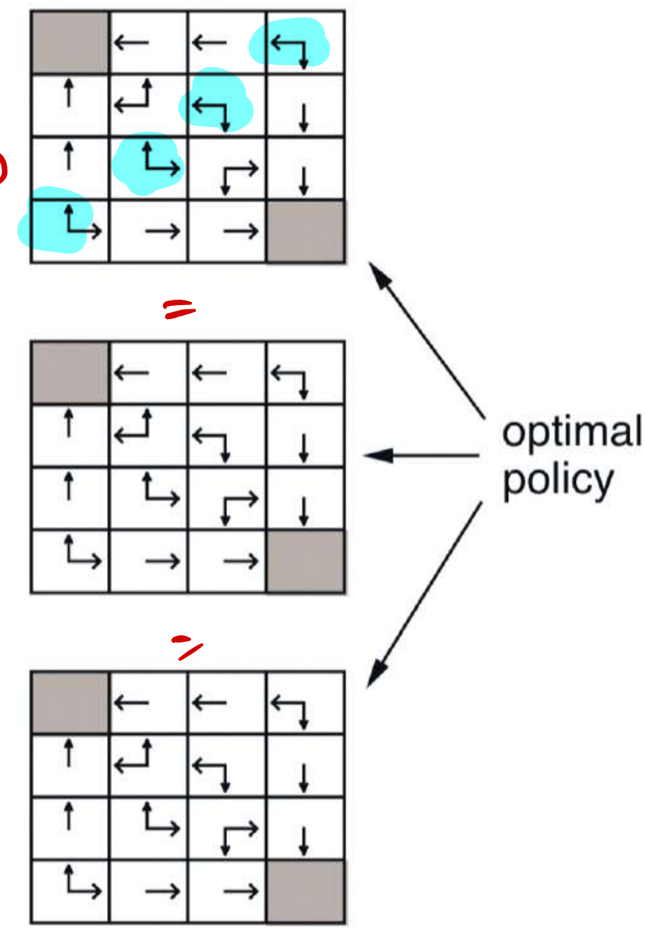
0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



$$
\begin{aligned}
v_\pi(s) &\leq q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s] \\
&= \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \mid S_t = s\big] \\
&\leq \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \mid S_t = s\big] \\
&\ \ \vdots \\
&\leq \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \mid S_t = s\big] \\
&= v_{\pi'}(s).
\end{aligned}
$$

Proof that greedy policy update is sound

Policy iteration

Initial policy

π_0 —policy evaluation→ V_{π_0} —greedy update→ π_1 → V_{π_1} → π_2 → V_{π_2} → π_3 ⋮

↑ iterative calculation

Expensive, but theoretically feasible

Treating Bellman equations as update rules

Dynamic Programming

Policy evaluation - repeatedly sweep across all states

Simplifying Policy Iteration

Don't need to fully evaluate V_π for each π in the iteration

Suffices to compute

$$V_\pi^0 \to V_\pi^1 \quad (1 \text{ sweep})$$

Collapse $\quad V_\pi \to \pi' \to V_{\pi'}$ [1 step]

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

Value Iteration

In practice

Asynchronous D.P.

Need not update $V(s)$ $\forall s$ in each pass

"Fairness"

In an infinite sequence of updates, every state is updated infinitely often