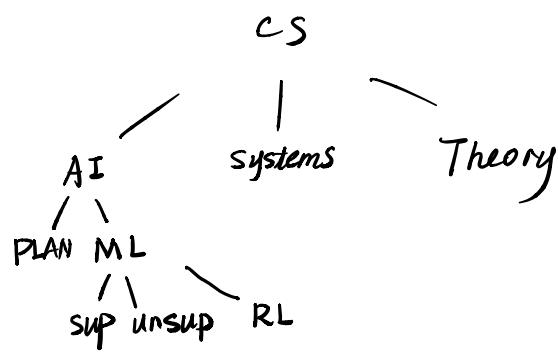


# What's reinforcement Learning?

"RL" is an area of ML inspired by behaviour psychology; concerned with how an agent can learn interactions with an environment.

- wiki



## Agent Environment Diagram.



Key properties:

- Evaluate Feedback
- Sequential

Agent: Child, Dog, Robots, Program

Environment: World Lab, Software Environment

Neural Science: How do animals learn?

a specific agent or set of agents

- study of some examples of learning & intelligence

RL: How can we make an agent that learns?

- study of learning & intelligence (in general)  
(machine or animal)

Donald Michie

MENACE

Controller

Plant

100

Pirates

10,000

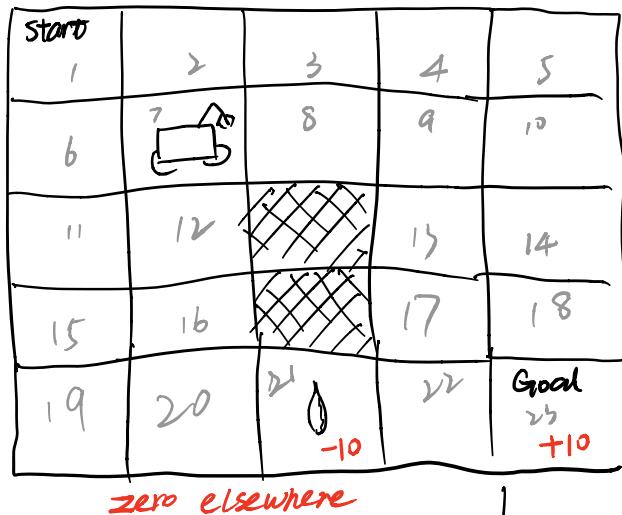
Gold



49

>50%

# Example Problem: Grid World



State: position of robot

Actions: Attempt  $\leftarrow \uparrow \rightarrow \downarrow$  UP Down  
L R  
AU/AD/AL/AR

Environment

Dynamics: Action succeeds  $P=0.8$

Veer right  $P=0.05$

Veer Left  $P=0.05$

Stays Put  $P=0.1$

① If the updates will cause the agent to leave the grid or hit an obstacle, it stays where it is

Rewards are for entering a state  
"straying" counts as re-entering state.

$\{1, 2, 3\}$	set	$\mathbb{N}^{>0}$	$\mathbb{N}^{>0}$
$\times$	set	$X$	Random variables
ett			
$ X $	abs		
$ X $	cardinality (size)		

No class Sep. 13

MDP

- A Mathematical specification of the environment and what we want the agent to learn.
- Let  $t \in \{0, 1, \dots\}$  be the "time step"
- Let  $s_t \in \mathbb{N}^{>0}$  be the state of the environment at time  $t$
- Let  $a_t$  be the action chosen by the agent at time  $t$
- Let  $r_t$  be the reward given to the agent at time  $t$

$$s_t \xrightarrow{a_t} s_{t+1} \quad r_t$$

$$\text{MDP} : M = (S, A, P, d_R, \gamma)$$

$S$  = set of possible states of the environment  
 "state set". Finite for now.  $\{1, 2, \dots, 23\}$

$A$  = set of possible actions the agent can take  $\{AU, AD, AL, AR\}$   
 "Action set"

$P$  = transition function, which describes how the state changes

$$P: S \times A \rightarrow [0, 1]$$

$$P(s, a, s') \triangleq \Pr(S_{t+1} = s' \mid S_t = s, A_t = a) \quad \begin{array}{l} s \in S \\ a \in A \\ s' \in S \\ t \in \mathbb{N} \geq 0 \end{array}$$

$$P(10, AD, 14) = 0.8$$

$$P(21, AR, 21) = 0.1 + 0.05 \times 2 = 0.2$$

Deterministic transition function

$$P(s, a, s') \in \{0, 1\}$$

$d_R$  = conditional distribution over  $R_t$  given  $S_t, A_t, S_{t+1}$

$R$  is the "reward function"  $R: S \times A \rightarrow \mathbb{R}$

$$R(s, a) \triangleq \mathbb{E}\{R_t \mid S_t = s, A_t = a\}$$

Assume  $|R_t| \leq R_{\max}$

$d_0$  is the "initial state distribution"  $d_0: S \rightarrow [0, 1]$

$$d_0(s) = P(S_0 = s)$$

$\gamma$ : discount factor  $\in [0, 1]$  is the "reward discount parameter"

Agent Formulation (for MDP)

Policy: the mechanism within the agent that determines which action to take

Learnings: corresponds to changing the policy

$$\pi: S \times A \rightarrow [0, 1]$$

$$\pi(s, a) \triangleq \Pr(A_t=a | S_t=s)$$

Deterministic policy

$$\pi(s, a) \in \{0, 1\} \quad \forall s, a, t$$

$$R(20, AR) = 0.8 \times (-10) + 0 = -8$$

Actions

	AU	AD	AL	AR
1	0	0	.9	.1
2				
3			$\pi(3, AL)$	
4				
5				

Any matrix  $|S| \times |A|$

all entries  $\geq 0$

rows sum to 1

is a valid policy

States

25

$S_0 \sim d_0$

for  $t=0, 1, \dots$

$A_t \sim \pi(S_t, \cdot)$

$S_{t+1} \sim P(S_t, A_t, \cdot)$

$R_t \sim d_R(S_t, A_t, S_{t+1}, \cdot)$

$S_0 \sim d_0$

$A_0 \sim \pi(S_0, \cdot)$

$\downarrow$   
sample from

$S_1 \sim P(S_0, A_0, \cdot)$

$R_0 \sim d_R(S_0, A_0, S_1, \cdot)$  -ish

$A_1 \sim \pi(S_1, \cdot)$

$S_2 \sim P(S_1, A_1, \cdot)$

Agent's Goal: Find a policy that maximizes the expected total amount of reward it will get

Objective function:

$$J: \Pi \rightarrow \mathbb{R}$$

$\downarrow$   
set of all policies

$$J(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} R_t | \pi \right]$$

$\hookrightarrow A_0 \sim \pi(S_0, \cdot)$   
 $A_1 \sim \pi(S_1, \cdot)$

Optimal Policy:

$$\pi^* \in \operatorname{argmax}_{\pi \in \Pi} J(\pi)$$

Properties: If  $|S| < \infty$ ,  $|A| < \infty$ , rewards are bounded, then there exists an optimal policy.

Reward discount:  $\gamma \in [0, 1]$

- Utility of reward  $r$  at time steps from now  $\gamma^t \cdot r$

$$J(\pi) \triangleq \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \cdot R_t \mid \pi \right]$$

- Including  $\gamma$  ensures that  $J(\pi)$  is finite

Terminal States:

A terminal state is a state that always transitions to a special state,  $S_{\infty}$ , called the "terminal absorbing state"

- Once in  $S$  the agent can never leave

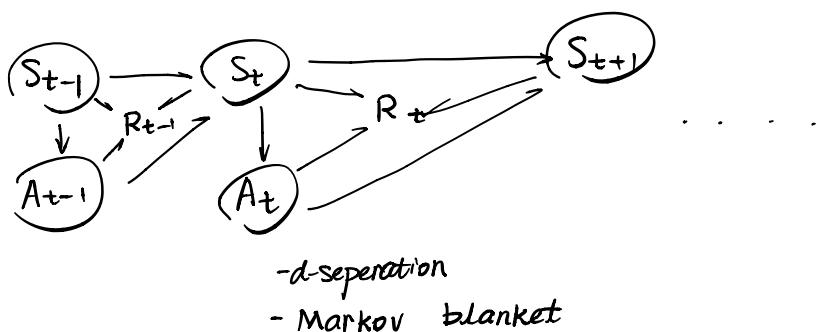
$$P(S_{\infty}, a, S_{\infty}) = 1$$

- If  $S_t = S_{\infty}$ , then  $R_t = 0$  always

Episode:

When an agent reaches  $S_{\infty}$ , the current trial called an "episode" ends, and we start a new one

- Set  $t = 0$
- Sample  $S_0 \sim \text{do}$
- Alert agent to this change



## Summary (Wrap-Up)

Def. of RL revisited:

"Learn from interactions with an environment"

Planning:  $P$  &  $R$  are known

RL: At least  $P$  is unknown.

More terminology:

- A history  $H_t$  is the trace of what has happened up to time  $t$  with one episode.

$$H_t \triangleq (S_0, A_0, R_0, S_1, A_1, R_1, \dots, S_t, A_t, R_t)$$

- A trajectory is the history of an entire episode  $H_\infty$

- The discounted Return of a trajectory is the discounted sum of rewards.

$$G \triangleq \sum_{t=0}^{\infty} \gamma^t R_t \quad \xrightarrow{\text{simplify}}$$

$$J(\pi) = \mathbb{E}[G | \pi]$$

- The return from  $t$  is:

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

velocity  $\dot{x}$

State:  $S = (X, V)$

Action:  $a \in \{\text{forward}, \text{reverse}, \text{neutral}\}$   
 $(+1) \quad (-1) \quad (0)$

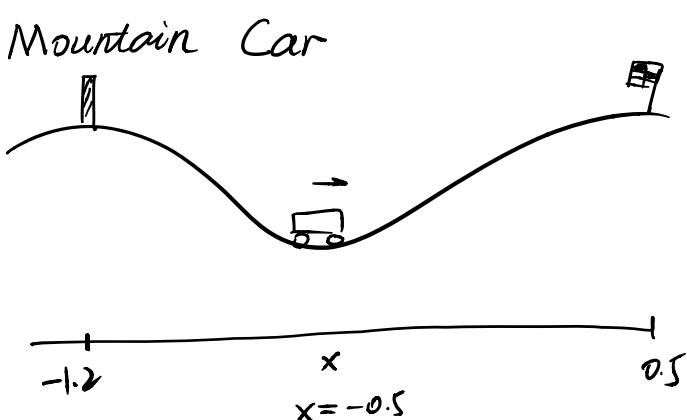
Dynamics:  $V_{t+1} = V_t + 0.001 A_t - 0.0025 \cdot \cos(3X_t)$

Capped  $[ -0.7, 0.7 ]$

$$X_{t+1} = X_t + V_{t+1}$$

$X$  capped to  $(-1.2, 0.5)$

with inelastic collisions  $\rightarrow$  Velocity = 0 if hit boundary



Terminal State: If  $X_t = 0.5$ , terminate.

shaping rewards

Rewards:  $R_t = -1$  always

Discount:  $\gamma = 1$

Initial State:  $S_0 = (-0.5, 0)$  always

Markov Property:

$$\downarrow P(h, s, a, s') = \Pr(S_{t+1} = s' \mid H_{t-1} = h, S_t = s, A_t = a) \quad \text{Not used because of MP}$$

Assumption / Property:

$$\Pr(S_{t+1} = s' \mid H_{t-1} = h, S_t = s, A_t = a) = \Pr(S_{t+1} = s' \mid S_t = s, A_t = a) \quad \forall s, a, s', h, t$$

The future is independent of history.

'conditionally'

- We also assume rewards are markovian.

$R_t$  is conditionally indep of  $H_{t-1}$  given  $S_t$

- Markov Policy

$$\Pr(A_t = a \mid S_t = s, H_{t-1} = h) = \Pr(A_t = a \mid S_t = s) \quad \forall s, a, h, t$$

Is mountain car markovian?

$s = (x, v)$  YES

$s = (x)$  NO

- Markov property is a property of a state representation, not a problem.

- One can always make a Markovian rep.

- Let  $s_t$  be non-Markovian, then  $(S_t, H_{t-1})$  is a Markovian State rep. ↗ "State Augmentation"

Stationary vs. Non-stationary.

Assume  $P$ ,  $\text{do}$ ,  $d_R$  are stationary

$$\Pr(S_{t+1} = s' \mid S_t = s, A_t = a) = \Pr(S_{i+1} = s \mid S_i = s, A_i = a) \quad \forall s, a, s', t, i$$

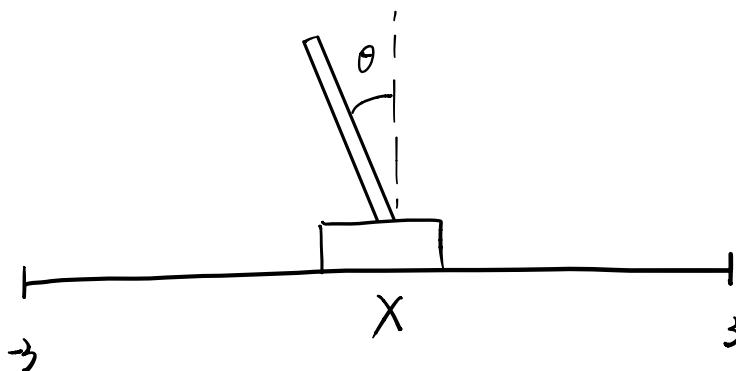
can be from  
different episodes.

-  $\pi$  is often non-stationary

Stationary  $\pi$  is called a "fixed policy"

Finite-Horizon MDP

Cart-Pole:



$$S = (X, V, \theta, \dot{\theta}, t)$$

$$a \in \{\text{LEFT}, \text{RIGHT}\}$$

$$R_t = 1 \quad \text{always}$$

$$\gamma = 1$$

$$S_0 = (0, 0, 0)$$

Dynamics = physics of the system.

Episodes terminates after 20s  
or if the pole falls.

Finite-Horizon MDP

$\Delta t = 0.02 \text{ sec}$   
walls at edges may end episodes

$$\exists L, \forall t \geq L, S_t = S_\infty \text{ always}$$



"horizon": Length of longest possible episode

- How implements the transition to  $S_\infty$  at time  $t=L$  within  $P$

- Augment state to include time:  $s = (X, V, \theta, \dot{\theta}, t)$

- Indefinite-horizon problem: will always terminate with  $pr=1$ .  
Within a finite num of steps, but  $L=\infty$

- Infinite horizon MDP : never terminate

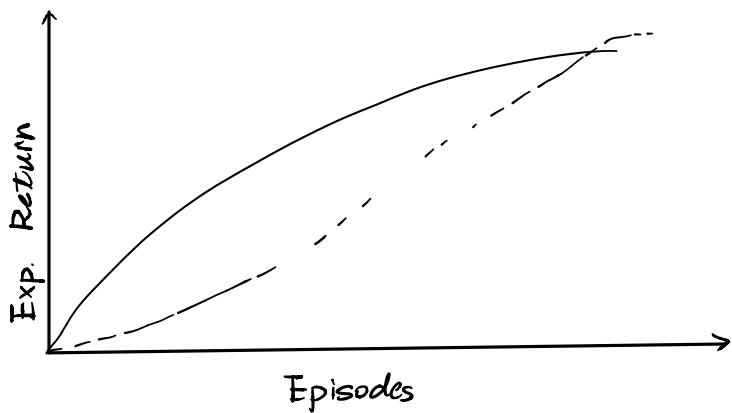
deterministic

Agent is implementation of:

for episode = 0, 1, 2, ...

```
s ~ do  
for t=0, 1, 2 ...  
    a = agent.getAction(s)  
    s' ~ π(s, a)  
    r ~ dr(s, a, s')  
    agent.train(s, a, s', r)  
    if s' = Soo  
        break;  
    s = s'  
agent.newEpisode();
```

- 1). Enumerate all policies and evaluate each for  $K$  episodes.
- 2). Phrase as a Black-Box optimization problem.
- 3). Model-based approaches.
  - ↳ Estimate of transition Func
  - Approximate  $P$  &  $R$  and solve for optimal policy for this model.
- 4). Value-Function based approaches
- 5). Gradient-Based methods (policy gradient)



Black-Box Optimization for policy search

- Simple agent
- Ignore MDP structure
- BBO algo solve problems of the form:

$$\underset{x \in \mathbb{R}^n}{\operatorname{argmax}} f(x)$$

$x \in \mathbb{R}^n$

assume access to an estimate  $\hat{f}(x)$  of  $f(x)$

→ some variants assume  $\nabla f(x)$  or estimate thereof are available

- BBO algos include hill-climbing, simulated annealing, and genetic algos

- For RL:

$$\underset{\pi \in \Pi}{\operatorname{argmax}} J(\pi)$$

Estimate  $J(\pi)$  by running  $\pi$  for  $N$  episodes:

$$\hat{J}(\pi) = \frac{1}{N} \sum_{i=1}^N G^i = \frac{1}{N} \sum_{i=1}^N \sum_{t=0}^{\infty} \gamma^t \cdot R_t^i$$

can represent policies as  $|S| \times |A|$  matrices, all entries  $> 0$ ,  
rows sum to 1.

Want to store  $\pi$  as a  $|S| \times |A|$  matrix called  $P$ , with no constraints.

- Increasing  $P(s, a)$  to increase  $\pi(s, a)$

### Tabular Softmax Policy

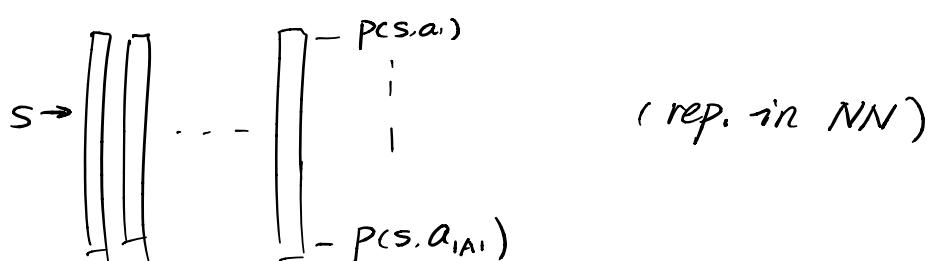
$$\pi(s, a) = \frac{e^{\theta P(s, a)}}{\sum_{a' \in A} e^{\theta P(s, a')}}$$

$$\pi(s, a) = \frac{e^{\theta_{sa}}}{\sum_{a' \in A} e^{\theta_{sa}}}$$

$\theta$  is constant scales "greediness"

- Drawback: can't represent deterministic policies.

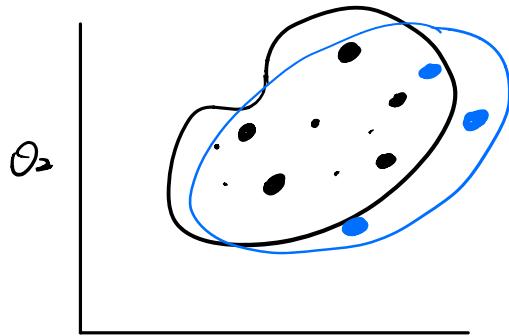
- Policy parameters are typically denoted by  $\Theta$ , not  $P$ .



$$\text{injection} = \frac{\text{current\_blood\_glucose} - \text{target\_blood\_glucose}}{\theta_1} + \frac{\text{med\_size}}{\theta_2}$$

## Cross Entry Method.

- Stulp & Sigond



pseudo-code

const

Input  $\theta$ : mean parameter vector

$\Sigma$ : covariance matrix initially  $0$ .

$K$ : number of elite

$N$ : number of episodes to run for

$P$ : population each

$\epsilon$ : small positive const.

1) For  $i = 1$  to  $P$

$$\underline{\theta}_i \sim N(\theta, \Sigma)$$

vector of policy params

$$\hat{J}_i = \text{evaluate}(\theta_i, N)$$

4) Sort  $(\theta_1, \theta_2, \dots, \theta_P, \hat{J}, \text{desc})$

$$5) \quad \theta = \frac{1}{K} \sum_{k=1}^K \theta_k$$

↓ keep paired values  
 ↑ sort based on policies that produce

$$6) \quad \Sigma = \frac{1}{K+\epsilon} \left[ \epsilon I + \sum_{k=1}^K (\theta_k - \theta)(\theta - \theta_k)^T \right]$$

Repeat from 1

$\text{evaluate}(\theta, N)$

1) Run  $N$  episodes with policy parameters  $\theta$

2) Compute returns  $G_1, G_2, \dots, G_N$

3) Return  $\frac{1}{N} \sum_{i=1}^N G^i$

Parametrized Policy:

$$\pi: S \times A \times \mathbb{R}^n \rightarrow [0, 1]$$

$$\pi(s, a, \theta) = \Pr(A_t=a | s_t=s, \theta)$$

$$\underset{\pi \in \Pi}{\operatorname{argmax}} J(\pi)$$



$$\underset{\theta \in \mathbb{R}^n}{\operatorname{argmax}} J(\theta)$$

$$J(\theta) = \mathbb{E}[G | \theta]$$



$$A_t \sim \pi(s_t, \cdot; \theta)$$

State-Value Function:

$$v^\pi: S \rightarrow \mathbb{R}$$

$$\begin{aligned} v^\pi(s) &\triangleq \mathbb{E}[G | s_0=s, \pi] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid s_0=s, \pi\right] \\ &= \mathbb{E}[G_t | s_t=s, \pi] \\ &= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid s_t=s, \pi\right] \end{aligned}$$

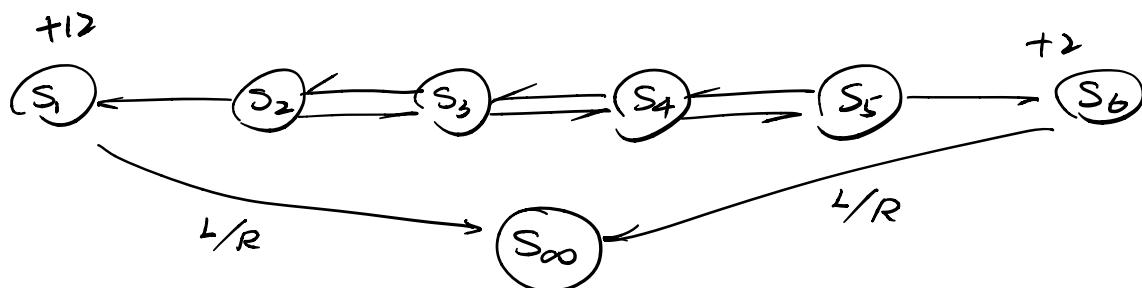
- Expected discounted sum of rewards if the agent follows policy  $\pi$  from state  $s$
- How "good" is it for the agent to be in state  $s$ .
- Value of state  $s$
- Does depend on policy  $\pi$

- 1.) Describe env
- 2.) BBD - ignore structure
- 3.) Value Functions
  - ↳ Tool used

RL algo

$R$  reward function

$R_t$  reward at  $t$ .



$$\gamma = 0.5$$

$\pi_1$  = Left always

$\pi_2$  = Right always

$$V^{\pi_1}(s_1) = 0$$

$$V^{\pi_2}(s_1) = 0$$

$$V^{\pi_1}(s_2) = 12$$

$$V^{\pi_2}(s_2) = 0.25$$

$$V^{\pi_1}(s_3) = 6$$

$$V^{\pi_2}(s_3) = 0.5$$

$$V^{\pi_1}(s_4) = 3$$

$$V^{\pi_2}(s_4) = 1$$

$$V^{\pi_1}(s_5) = 1.5$$

$$V^{\pi_2}(s_5) = 2$$

$$V^{\pi_1}(s_6) = 0$$

$$V^{\pi}(s_6) = 0$$

value of terminal states  
is always zero.

Markov P

$$w^{\pi}(s, h) = \mathbb{E}[G_t | S_t = s, A_{t+h} = h, \pi]$$

Action-Value Function

State-Action Value Function / Q-function.

$$q^{\pi}: S \times A \rightarrow \mathbb{R}$$

$\pi' = \pi$  but  $s \rightarrow a$

$$q^{\pi}(s, a) = V^{\pi'}(s)$$

$$q^{\pi}(s, a) = \mathbb{E}[G | S_0 = s, A_0 = a, \pi]$$

$$q^{\pi}(s, a)$$

$$= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, A_0 = a, \pi\right]$$

$$= \mathbb{E}[R_0 + \sum_{k=1}^{\infty} \gamma^k R_{t+k} | \dots]$$

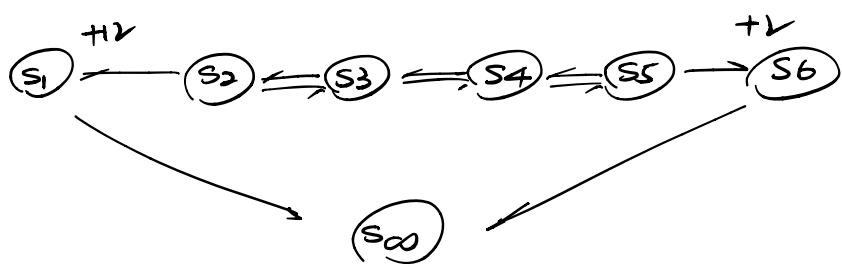
$$= \mathbb{E}[G_t | S_t = s, A_t = a, \pi]$$

$$= \sum_a \pi(s, a) \sum_s p(s, a, s') + \gamma \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | \dots\right]$$

$$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} | S_t = s, A_t = a, \pi\right] =$$

- Expected-discounted return if the agent takes action  $a$  in state  $s$ , and follows policy  $\pi$  thereafter.
- How "good" is it to take action  $a$  in state  $s$ ?

(if using policy  $\pi$   
otherwise)



$\pi_1$ : Left

$\pi_2$ : Right

$$q^{\pi_1}(s_1, L) = 0$$

$$q^{\pi_1}(s_1, R) = 0$$

$$q^{\pi_1}(s_2, L) = 12$$

$$q^{\pi_1}(s_2, R) = \gamma^0 \cdot 0 + \gamma^1 \cdot 0 + \gamma^2 \cdot 12 = 3$$

$$q^{\pi_1}(s_3, L) = 0.5 \cdot 12 = 6$$

$$q^{\pi_1}(s_3, R) = \gamma^0 \cdot 0 + \gamma^1 \cdot 0 + \gamma^2 \cdot 0 + \gamma^3 \cdot 12 = 1.5$$

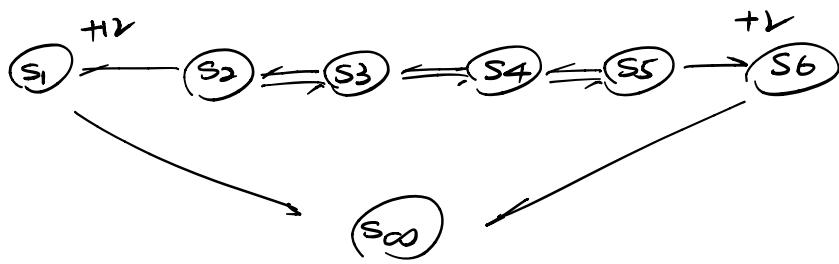
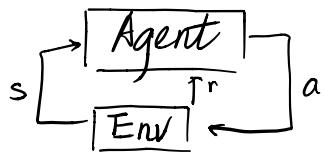
"consistant Bellman operation"

## Bellman Equation for $v^\pi$

$$\begin{aligned}
 v^\pi(s) &= \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k} \mid s_t = s, \pi \right] \\
 &= \mathbb{E} \left[ R_t + \sum_{k=1}^{\infty} \gamma^k \cdot R_{t+k} \mid s_t = s, \pi \right] \\
 &= \sum_a \pi(s, a) \cdot R(s, a) + \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^{k+1} \cdot R_{t+1+k} \mid s_t = s, \pi \right] \\
 &\quad \downarrow \\
 &\quad \underbrace{\gamma \cdot \gamma^k}_{\gamma^k} \\
 &\quad \downarrow \\
 &= \gamma \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k} \mid s_t = s, \pi \right]
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_a \pi(s, a) R(s, a) + \sum_a \pi(s, a) \sum_{s'} p(s, a, s') \cdot \gamma \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \mid s_t = s, a, s_{t+1} = s' \right] \\
 &\quad \downarrow \\
 &\quad \text{Markov} \\
 &= v^\pi(s)
 \end{aligned}$$

$$v^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P(s, a, s') (R(s, a) + \gamma v^\pi(s'))$$



$$\gamma = 0.5$$

$\pi_1 = \text{left}$

$\pi_2 = \text{right}$

$$v^{\pi_1}(s_2) = 12$$

$$v^{\pi_1}(s_3) = 6$$

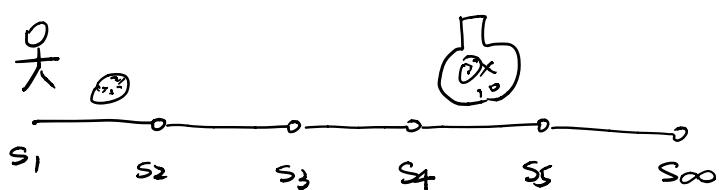
$$v^{\pi_1}(s_4) = 3$$

$$v^{\pi_1}(s_2) = 12 + \gamma 0$$

$$v^{\pi_1}(s_3) = 0 + \gamma \cdot v^{\pi_1}(s_2) = 6$$

$$v^{\pi_1}(s_4) = 0 + \gamma \cdot v^{\pi_1}(s_3) = 3$$

$$\begin{aligned} v^\pi(s) &= \mathbb{E}[R_1 + \gamma R_2 + \dots | S_1 = s] \\ &= \mathbb{E}[R_1 + \underbrace{\gamma(R_2 + \gamma R_3 + \dots)}_{\downarrow} | S_1 = s] \\ &= v^\pi(s_2) \end{aligned}$$



$$v^\pi(s_4) = 10$$

$$v^\pi(s_3) = 0 + \gamma \cdot v^\pi(s_4) = \gamma \cdot 10$$

$$v^\pi(s_2) = 0 + \gamma \cdot v^\pi(s_3) = \gamma^2 \cdot 10$$

$$v^\pi(s_1) = 1 + \gamma \cdot v^\pi(s_2) = 1 + \gamma^3 \cdot 10$$

$$q^\pi(s, a) = \sum_{s'} p(s, a, s') \sum_{a'} \pi(s', a') (R(s, a) + \gamma q^\pi(s', a'))$$

Optimal Value Function:

$$v^*: S \rightarrow \mathbb{R}$$

$$\forall s \in S \quad v^*(s) \triangleq \max_{\pi \in \Pi} v^\pi(s) = v^{\pi^*}(s)$$

Define:  $\pi \geq \pi'$  iff  $\forall s \in S, v^\pi(s) \geq v^{\pi'}(s)$

- partial ordering on policies

Can be  $\pi, \pi'$  s.t.  $\pi \not\geq \pi', \pi' \not\geq \pi$



Optimal policy: Any policy  $\pi^*$

$$\text{s.t. } \forall \pi, \pi^* \geq \pi$$

Intuition: Given  $\pi, \pi'$ .

Construct  $\pi''$  that applies  $\pi$  in states where it's better and applies  $\pi'$  in other states.

$\rightarrow$  This  $\pi'' \geq \pi$

$$\pi'' \geq \pi'$$

Formally: There exists at least one optimal policy for all MDP where  $|S| < \infty$ ,  $|A| < \infty$ ,  $R_{\max} < \infty$ ,  $\gamma < 1$

Optimal Q Function:

$$q^*: S \times A \rightarrow \mathbb{R}$$

$$q^*(s, a) = \max_{\pi} q^{\pi}(s, a)$$

$$q^* = q^{\pi^*}$$

$v^*$ ,  $q^*$  are unique, unlike  $\pi$ .

— Given  $v^*$ , but don't know  $P$  &  $R$ , can you act optimally? No.  
In state  $s$ , choose any action  $a$  that maximizes

$$\sum_{s'} p(s, a, s') \cdot (R(s, a) + \gamma v^*(s'))$$

$\underbrace{\phantom{p(s,a,s')}}_{=}$        $\underbrace{\phantom{v^*(s')}}_{=}$

↓  
 $q^*(s, a)$

What about  $q^*$ ? Yes.

$$a \in \arg \max_{a \in A} q^*(s, a)$$

Bellman Optimality Equation

$$v^*(s) = \sum_a \pi^*(s, a) \sum_{s'} p(s, a, s') (R(s, a) + \gamma v^*(s')) \quad \textcircled{1}$$

$\underbrace{\phantom{\sum_a \pi^*(s, a) \sum_{s'} p(s, a, s')}}_{q^*(s, a)}$

$$\begin{aligned} q^*(s, a) &= \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid S_t = s, A_t = a, \pi \right] \\ &= \mathbb{E} \left[ R_t + \underbrace{\sum_{k=1}^{\infty} \gamma^k R_{t+k}}_{\downarrow} \mid S_t = s, A_t = a, \pi^* \right] \\ &\quad \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \end{aligned}$$

$$\begin{aligned}
 &= R(s, a) + \gamma \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a, \pi^* \right] \\
 &= R(s, a) + \sum_{s'} p(s, a, s') \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a, S_{t+1} = s', \pi^* \right] \\
 &= R(s, a) + \sum_{s'} p(s, a, s') \gamma \cdot V^*(s')
 \end{aligned}$$

$\text{D} = \max_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma V^*(s'))$  ← Bellman Optimality Eqn.

$$q^*(s, a) = \sum_{s'} p(s, a, s') (R(s, a) + \gamma \max_{a'} q^*(s', a'))$$

Bellman optimality equations only hold for optimal policies.

Policy Iteration:

- policy evaluation.

Give a policy  $\pi$ , find  $V^\pi$

- Assume  $P$  &  $R$  are known.

Method:

- 1) Solve Bellman equation
- 2) Dynamic programming.  $\pi^*$

Dynamic Programming.

Sequence of value function approximations.

$$\hat{V}_0^\pi$$

$v_1, v_2, v_3, \dots$  where  $v_i: S \rightarrow \mathbb{R}$

arbitrary guess (often zero vector)

$$v_{i+1}(s) = \sum \pi(s, a) \sum p(s, a, s') (R(s, a) + \gamma v_i(s'))$$

Properties:

1)  $v_i = v^\pi$  is a fixed point

2)  $v_i \rightarrow v^\pi$  as  $i \rightarrow \infty$  (not proven here)

3) One pass over the set of states is called a "full backup"  
A single state update is called a "backup"

$$v_{i+1} = v_i$$

S	0	0	0	0
$v_0$	0	0	0	0
G	0	0	0	0

$$R_t = -1, \gamma = 1$$

$\pi$ : go down, unless bottom row, then right

$v_1 =$	-1	-1	-1	-1
	-1	-1	-1	-1
	-1	-1	-1	-1
	-1	-1	-1	-1
	-1	-1	-1	-1

(So)

$v_2 =$	-2	-2	-2	-2
	-2	-2	-2	-2
	-2	-2	-2	-2
	-2	-2	-2	-1

$v_3 =$	-3	-3	-3	-3
	-3	-3	-3	-3
	-3	-3	-3	-2
	-3	-3	-2	-1

$$v_4 = \dots \quad v_5 = \dots$$

$v_7 =$	-7	-6	-5	-4
	-6	-5	-4	-3
	-5	-4	-3	-2
	-4	-3	-2	-1

## In-Place Implementation:

- When updates  $v_t(s)$ , store back in same state table rather than new table (also converges)
- Update states in any order
- Can even update some states more often than others (as long as every state update infinitely often)

Know:  $q^{\pi}(s, a)$

$$q^{\pi}(s, a) = \sum_{s'} p(s, a, s') (R(s, a) + \gamma v^{\pi}(s'))$$

- For all states select the action that maximizes  $q^{\pi}(s, a)$

Deterministic policy:

$$\pi'(s) \in \operatorname{argmax}_a q^{\pi}(s, a)$$

Policy Improvement Theorem:

Let  $\pi$  and  $\pi'$  be deterministic policies such that

$$\forall s \in S \quad q^{\pi}(s, \pi'(s)) \geq v^{\pi}(s)$$

Then:

$$\pi' > \pi$$

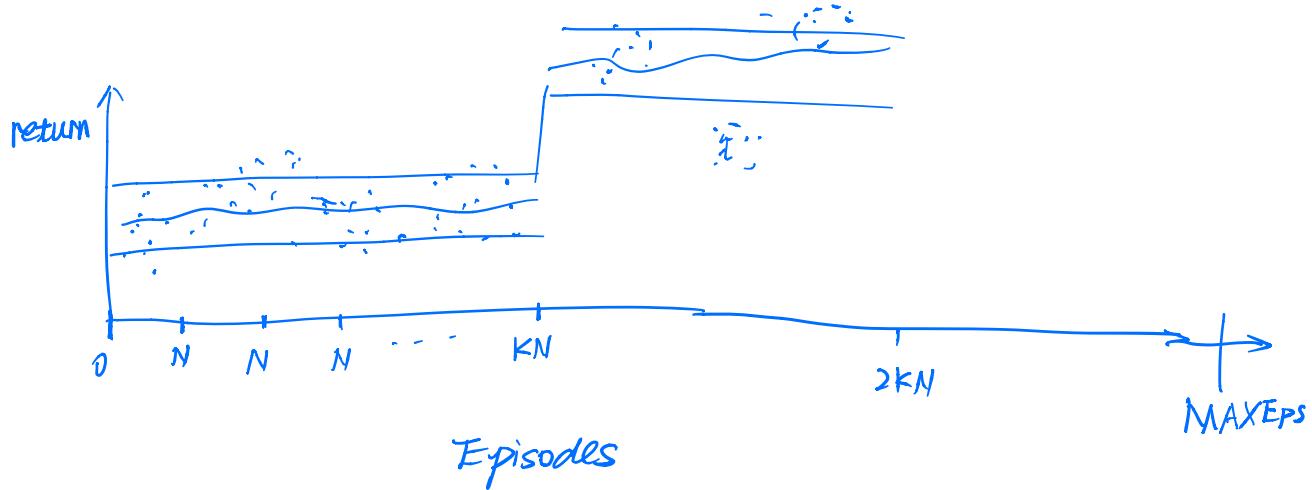
$$\text{I.e.: } \forall s \quad v^{\pi'}(s) \geq v^{\pi}(s), \quad \forall s \quad \pi'(s) > \pi(s)$$

Proof:

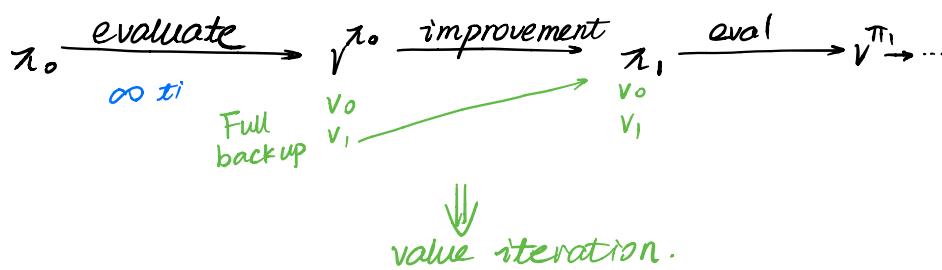
$$\begin{aligned} v^{\pi}(s) &\leq q^{\pi}(s, \pi'(s)) && \nearrow \text{At comes from } \pi' \\ &= \mathbb{E}[R_t + \gamma v^{\pi}(s_{t+1}) \mid s_t = s, \pi'] \\ &\leq \mathbb{E}[R_t + \gamma \cdot q^{\pi}(s_{t+1}, \pi'(s_{t+1})) \mid s_t = s, \pi'] \\ &= \mathbb{E}[R_t + \gamma \mathbb{E}[R_{t+1} + \gamma v^{\pi}(s_{t+2}) \mid s_t = s, \pi'] \mid s_t = s, \pi'] \end{aligned}$$

$$\begin{aligned}
 &= \mathbb{E}[R_t + \gamma R_{t+1} + \gamma^2 V^\pi(s_{t+2}) \mid S_t = s, \pi'] \\
 &\leq "B^\pi(s_{t+2}, \pi'(s_{t+2}))" \\
 &\leq \mathbb{E}[R_t + \gamma R_{t+1} \dots \mid S_t = s, \pi'] \\
 &= V^{\pi'}(s)
 \end{aligned}$$

If  $\forall s \in S$ ,  $\sum_a \pi'(s, a) q^\pi(s, a) \geq V^\pi(s)$ , then  $\pi' \geq \pi$



## Policy Iteration



What if  $\pi' = \pi$

$$\begin{aligned}
 \pi'(s) &= \arg\max_a q^\pi(s, a) \\
 &= \arg\max_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma V^\pi(s')) \\
 \text{As} \\
 V^\pi(s) &= \max_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma V^\pi(s'))
 \end{aligned}$$

↑  
Bellman optimality eqn.

## Value Iteration.

$v_0$ : arbitrary value function

$\pi_0$ : arbitrary policy · deterministic

$$v_1: \forall s \quad v_1(s) = \sum_{s'} p(s, \pi_0(s), s') \cdot (R(s, \pi_0(s)) + \gamma v_0(s'))$$

$$\pi_1: \forall s \quad \pi_1(s) \in \operatorname{argmax}_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma v_1(s'))$$

$\hat{\pi}(s, a)$

$$v_2: \forall s \quad v_2(s) = \sum_{s'} p(s, \pi_1(s), s') (R(s, \pi_1(s)) + \gamma v_1(s'))$$

$$\pi_2: \forall s \quad \pi_2(s) \in \operatorname{argmax}_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma v_2(s'))$$

$$v_3: \forall s \quad v_3(s) = \sum_{s'} p(s, \pi_2(s), s') (R(s, \pi_2(s)) + \gamma v_2(s'))$$

## Value Iteration

$$\forall s, \quad v_{k+1}(s) = \max_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma v_k(s'))$$

Bellman Optimality Eqn.

$$v^*(s) = \max_a \sum_{s'} p(s, a, s') (R(s, a) + \gamma v^*(s'))$$

Bellman Operator: Encodes one step of value iteration

- view value function as vectors in  $\mathbb{R}^{|S|}$

$$v_k = \begin{bmatrix} v_k(s_0) \\ v_k(s_1) \\ \vdots \\ v_k(s_{|S|-1}) \end{bmatrix}$$

$$\mathcal{T}: \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$$

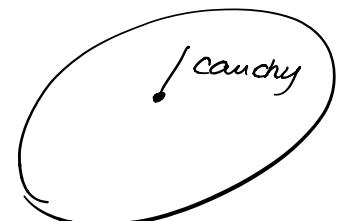
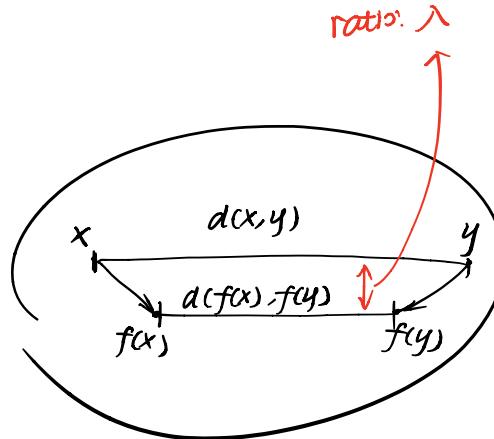
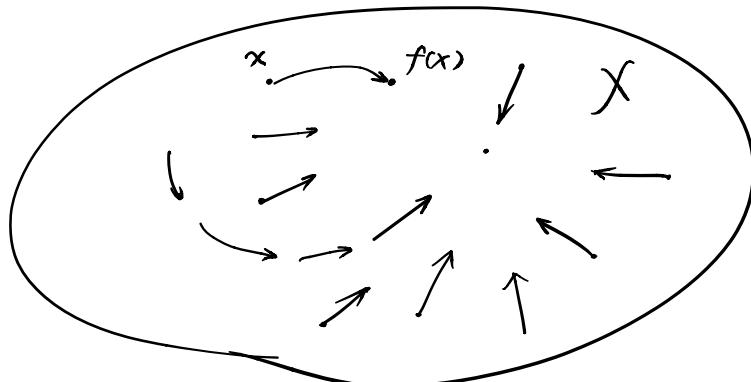
$$\mathcal{T}(v_k) \triangleq v_{k+1}$$

operator: function that takes elements of a space as input and produces elements of the same space as output

$$\mathcal{T}_{v_k}(s) = v_{k+1}(s)$$

## Contraction Mappings:

$f: S \rightarrow X$  is a contraction mapping if there exists a  $\lambda \in [0, 1)$  s.t.  $\forall x, y. d(f(x), f(y)) \leq \lambda d(x, y)$  where  $d$  is a distance function.



## Banach Fixed Point Thm.

$\mathbb{R}^n$

If  $f$  is a contraction mapping on a non-empty complete normed vector space, then  $f$  has a unique fixed point,  $x^*$ , and the sequence defined by  $x_{i+1} = f(x_i)$ , with  $x_0$  chosen arbitrarily, converges to  $x^*$ .

Bellman Operator is a contraction mapping:  $f \leftarrow T, X = \mathbb{R}^{IS}$

$$d(v, v') = \max_s |v(s) - v'(s)| = \underbrace{\|v - v'\|}_{\text{max norm}}$$

Show:

$$\|\hat{T}(v) - \hat{T}(v')\| \leq \gamma \|v - v'\| \quad \text{for all } v, v'$$

$$\|\hat{T}_v - \hat{T}_{v'}\| \leq \gamma \|v - v'\|$$

Oct 18. 7pm - 9pm. ISB 135

No class on Tuesday

Bellman operator is a contraction mapping:

$$T_v(s) = \max_{a \in A} \sum_{s'} p(s,a,s') \cdot (R(s,a) + \gamma v(s'))$$

$$d(v, v') = \max_s |v(s) - v'(s)| = \|v - v'\|$$

$$d(f(x), f(y)) \leq \lambda d(x, y) \text{ (from defi.)}$$

↓

$$\|T_v - T_{v'}\| \leq \gamma \|v - v'\| \quad \forall v, v'$$

$$= \max_s |(T_v - T_{v'})(s)|$$

$$\|T_v - T_{v'}\| = \max_s |T_v(s) - T_{v'}(s)|$$

$$= \max_s \left| \max_a \sum_{s'} p(s,a,s') (R(s,a) + \gamma v(s')) - \max_a \sum_{s'} p(s,a,s') (R(s,a) + \gamma v'(s')) \right|$$

Claim.  $\left| \max_x f(x) - \max_x g(x) \right| \leq \max_x |f(x) - g(x)|$

Proof:  $\forall x \quad f(x) \leq |f(x) - g(x)| + g(x)$

$$\max_x f(x) \leq \max_x |f(x) - g(x)| + g(x)$$

$$\max_x f(x) \leq \max_x |f(x) - g(x)| + \max_x g(x)$$

$$\max_x f(x) - \max_x g(x) \leq \max_x |f(x) - g(x)|$$

↓

$$\left| \max_x f(x) - \max_x g(x) \right| \leq \max_x |f(x) - g(x)|$$

Then:

$$\begin{aligned}
 & \max_s \left| \max_a \sum_{s'} P(s,a,s') (R(s,a) + \gamma v(s')) - \max_a \sum_{s'} P(s,a,s') (R(s,a) + \gamma v'(s')) \right| \\
 & \leq \max_s \max_a \left| \underbrace{\sum_{s'} P(s,a,s') (R(s,a) + \gamma v(s'))}_{\text{weighted average}} - \underbrace{\sum_{s'} P(s,a,s') (R(s,a) + \gamma v'(s'))}_{\text{the max value}} \right| \\
 & = \gamma \max_s \max_a \left| \sum_{s'} P(s,a,s') (v(s') - v'(s')) \right| \\
 & \leq \gamma \max_s \max_a \max_{s'} |v(s') - v'(s')| = \gamma \max_{s'} |v(s') - v'(s')| \\
 & = \gamma \|v - v'\|
 \end{aligned}$$

- Value iteration converges to a unique fixed point  $v^\infty$

-  $\pi^\infty$  be greedy w.r.t.  $v^\infty$

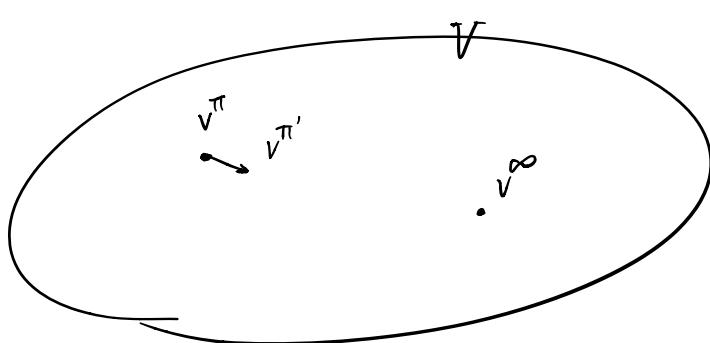
$$\pi^\infty(s) \in \arg\max_a \sum_{s'} P(s,a,s') (R(s,a) + \gamma v(s'))$$

5% : there exists

$$v^{\pi^\infty}(s) = \sum_{s'} P(s,\overset{\pi^\infty(s)}{\underset{\pi(s)}{\uparrow}},s') (R(s,\overset{\pi^\infty(s)}{\underset{\pi(s)}{\uparrow}},a) + \gamma v^\infty(s'))$$

an optimal policy  
using  $v^\pi$  definition

$$v^{\pi^\infty}(s) = \max_a \sum_{s'} P(s,a,s') (R(s,a) + \gamma v^\infty(s')) \quad \text{Bellman Optimality Ezn}$$



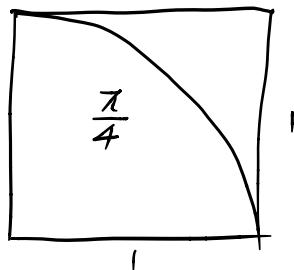
$v^\dagger \neq v^\pi$   
 $v^\pi \neq v^{\pi'}$   
 policy imp Thm  
 $\pi' > \pi$   
 $\pi$  not optimal

Prove that for all MDPs. with finite state, actions , bounded rewards and  $\gamma \in [0,1)$ , there exists an optimal deterministic policy. ???

Monte Carlo

Idea: use random samples to solve problems that are deterministic in principle

Estimate:  $\pi$



Policy Evaluation:

*state Sample*      *sapple*  
 History  $H = (s_0, a_0, r_0, -s_1)$

Generated by running  $\pi$  for one episode.

Monte Carlo estimate of  $v^\pi(s_0) = \sum_{t=0}^{\infty} \gamma^t \cdot R_t = G$

*Sapple* occurs many times in an episode

$$S_t \dots S_{t'} \dots S_{t''}$$

$$v^\pi(S_{\text{sapple}}) = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$$

or

$$= \sum_{k=0}^{\infty} \gamma^k R_{t'+k}$$

$$= \frac{1}{3} \left[ \sum_{k=0}^{\infty} \gamma^k (R_{t+k} + R_{t'+k} + R_{t''+k}) \right]$$



Policy Evaluation. → start at  $s$

History  $H$ , generated by  $\pi$

$$v^\pi(s) \approx G$$

First Visit Monte Carlo → estimating  $v^\pi$

Intuition: Generate many episode of data

For each state, average the discounted returns  
after it was first visited in each episode

Pseudocode

Initialize:

$\pi$  ← policy to be evaluated

$V$  ← an arbitrary state value function.

Returns( $s$ ) ← an empty list for all  $s \in S$

Repeat Forever:

1) Generate an episode using  $\pi$ .

2) For each state  $s$ , appearing in the episode:

$G \leftarrow$  Return following the first occurrence of  $s$

[ if  $s_t$  is the first occurrence of  $s$ , then ]

$$G = G_t = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k}$$

Append  $G$  to Returns( $s$ )

$v(s) \leftarrow \text{average}(\text{Returns}(s))$

Return  $v$ .

Definition.  $X_1, X_2, \dots, X_n$  real valued R.V.

$$X_n \xrightarrow{\text{a.s.}} X \text{ iff}$$

$$\Pr(\lim_{n \rightarrow \infty} X_n = X) = 1$$

### Khintchine Strong Law of Large Numbers

Let  $X_1, X_2, \dots$  be <sup>an infinite</sup> independent finite variance. same mean. i.i.d. random variables, then

$\left( \frac{1}{n} \sum_{i=1}^n X_i \right)_{n=1}^{\infty}$  is a sequence of random variables that

converges almost surely to  $\mathbb{E}[X_i]$  i.e.

$$\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{\text{a.s.}} \mathbb{E}[X_i]$$

$$\Pr(\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n X_i = \mathbb{E}[X_i]) = 1$$

### Properties of FVMC:

1. Converges (in probability and almost surely) to  $\hat{\pi}$  if every state is visited infinitely often.

Proof:

Consider a sequence of estimates  $v_k(s)$  for a particular state,

s.

$$v_k(s) = \frac{1}{k} \sum_{i=1}^k G^i$$

$\downarrow$   
ith element of Return(s)

$$\mathbb{E}[G^i] = v^{\pi}(s)$$

$$X_i \leftarrow G^i, \quad X_n \xrightarrow{a.s} V^\pi(s)$$

Kolmogorov's Strong Law of Large Numbers

Let  $X_1, X_2, \dots$

:

:

SEE NOTE !

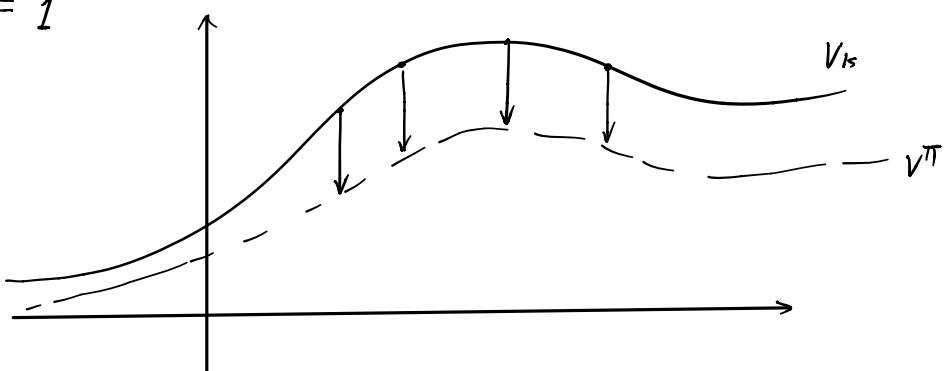
$$V_K(s) \xrightarrow{a.s} V^\pi(s) \quad \forall s$$

$$\Pr(\lim_{n \rightarrow \infty} V_K(s) = V^\pi(s)) = 1, \quad \forall s$$



finite states (or at least countable)

$$\Pr(\forall s, \lim_{n \rightarrow \infty} V_K(s) = V^\pi(s)) = 1$$



Every Visit Monte Carlo

- Use the return from every visit to  $s$  during an episode  
"and each time,  $t$ , that it occurred"
- Also converges almost surely to  $V^\pi$ .

## MC Estimation of Action Values

Same idea:  $\hat{q}^\pi(s, a) = \text{average return from first time action } a \text{ taken in state } s$

Problem: What if  $\pi$  never chooses action  $a$  in state  $s$ ?

Solution: Exploring starts:

- Randomize  $\pi_0$  and  $A_0$  st. every  $(s, a)$  pair has non-zero probability.

Solution: Stochastic policies.

Have a non-zero probability for every action in every state

$$\forall s, a, \pi(s, a) > 0.$$

$$\text{standard } (V_k(s)) \propto \frac{1}{\sqrt{k}}$$

Review:

Evaluate  $\hat{q}^\pi$

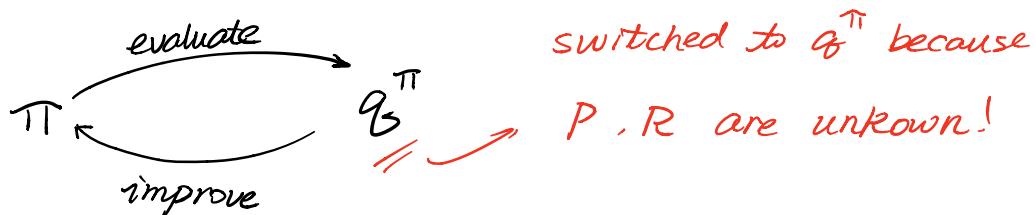
→ Run  $\pi$ . Use MC estimate

Some actions may never be taken (e.g. with policy iteration)

- Fix: Exploring Starts / Stochastic policies.

## Monte-Carlo Control

- Form of generalized policy iteration. (GPI)

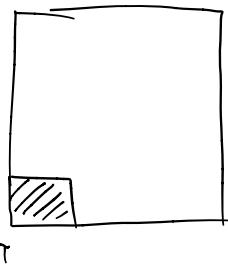


## Monte-Carlo Policy Iteration.

$$\pi_0 \xrightarrow{\text{eval}} q^{\pi_0} \xrightarrow{\text{imp}} \pi_1 \xrightarrow{\text{eval}} q^{\pi_1} \dots$$

Evaluate using FCMC

same prop as PI if use exploring starts.



Monte-Carlo - Exploring Starts (like VI for MC)

- Avoid infinite episodes in evaluation

step by terminating evaluation after  
one episode.

Init so randomly

$$\Pr(s) > 0 \quad \forall s$$

- Accumulate returns over all episodes. Don't only use returns generated by current policy.

PseudoCode :

Initialize:  $\forall s \in S, a \in A$

$$q(s, a) \leftarrow \text{arbitrary}$$

$$\pi(s) \leftarrow \text{arbitrary}$$

$$\text{Return}(s, a) \leftarrow \text{empty List}$$

Repeat Forever:

1). Generate one episode using exploring starts and  $\pi$

2). For each  $(s, a)$  appearing in the episode:

$G \leftarrow \text{Return following the first occurrence of } (s, a)$

Append  $G$  to  $\text{Returns}(s, a)$

$$q(s, a) \leftarrow \text{average}(\text{Returns}(s, a))$$

$$q^{\pi}$$

3). For each  $s$  in the episode

$$\pi(s) \leftarrow \underset{a}{\operatorname{argmax}} q(s, a)$$

? Can this converge to a sub-optimal policy?

1988 couldn't show converges to  $\pi^*$ , or  $q^*$

$\epsilon$ -Greedy MC Control Algorithm.

Idea: Avoid exploring starts by using a slightly stochastic policy.

replace

$$\begin{aligned} a^* &\leftarrow \operatorname{argmax}_a q^\pi(s, a) \\ \hat{\pi}(s, a) &= \begin{cases} (1-\epsilon) + \frac{\epsilon}{|A|} & \text{if } a = a^* \\ \frac{\epsilon}{|A|} & \text{if } a \neq a^* \end{cases} \end{aligned}$$

Properties: By variations on the policy improvement theorem, PI using  $\epsilon$ -Greedy converges to optimal  $\epsilon$ -greedy policy.

Another Monte Carlo Algorithm (TD1)

Given state  $s$  is visited at time  $t$ :

$$*\quad v(s_t) \leftarrow v(s_t) + \alpha(G_t - v(s_t)) \Rightarrow \text{WHY THIS?}$$

target

$$f(x) \leftarrow f(x) + \alpha(Y - f(x))$$

↓

$$L(v) = \mathbb{E} \left[ \frac{1}{2} (v^\pi(s) - v(s))^2 \right] \quad (\text{Loss Func})$$

TD       $\lambda$       MC

$\lambda = 1 \rightarrow MC$

$\lambda = 0 \rightarrow TD$

$$\frac{\partial L(v)}{\partial v} = \mathbb{E} \left[ (v^\pi(s) - v(s)) \frac{\partial v(s)}{\partial v} \right]$$

$$\frac{\partial L(v)}{\partial v(s)} = (V^\pi(s) - v(s))$$

$$\nabla \widehat{L}(v) = (G_t - V(s_t)) \frac{-\partial V(s_t)}{\partial v}$$

$$V \leftarrow V - \alpha \widehat{L}(v_t)$$

$$V(s_t) \leftarrow V(s_t) + \alpha (G_t - V(s_t))$$

$\downarrow$   
 $s_t \leftarrow \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

Properties:

- SGD on  $L(v)$
- Convergence of SGD "Gradient Convergence in Gradient Methods"
- Must wait until end of episode to update

Continuous:  $V_w(s)$

$$w \leftarrow w + \alpha (G_t - V_w(s)) \frac{\partial V_w(s)}{\partial w}$$

## MIDTERM REVIEW

Markov Decision Process.

$$M = (S, A, P, d_R, d_A, \gamma)$$

$\downarrow$  states       $\downarrow$  actions      dynamics of transition

$d_R$  distribution of rewards  $R_t$

$$\pi: S \times A \rightarrow [0, 1] \quad \pi(s, a)$$

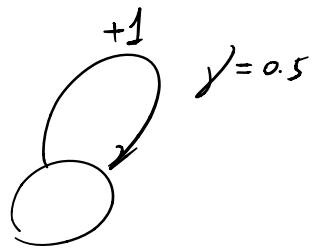
$$J(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_t \mid \pi \right]$$

$$\pi^* \in \arg \max_{\pi} J(\pi)$$

Soo

State  $s$  is terminal if  $P(s, a, s_\infty) = 1, \forall a \in A$

A sequence  $s_0, a_0, r_0, s_1, a_1, \dots$



$$s_\infty \in S \quad p(s_\infty, a, s_\infty) = 1 \quad \forall a \in A \quad \text{and} \quad r(s_\infty, a) = 0 \quad \forall a \in A$$

Planning:  $P$ , maybe  $R$  are known

RL: Don't know  $P$ , maybe  $R$ .

$$v^\pi(s) = \mathbb{E}[G_t | s_t = s, \pi]$$

$$q^\pi(s, a) = \mathbb{E}[G_t | s_t = s, A_t = a, \pi]$$

$$v^\pi(s) = \sum_a \pi(s, a) \sum_{s'} p(s, a, s') (R(s, a) + \gamma v^\pi(s'))$$

$$q^\pi(s, a) = \sum_{s'} p(s, a, s') \sum_{a'} \pi(s', a') (R(s, a) + \gamma q^\pi(s'))$$

$$V^*(s) = \max_{\pi} V^\pi(s) = V^{\pi^*}(s)$$

$$\pi' \geq \pi \text{ iff } \forall s \quad v^{\pi'}(s) \geq v^\pi(s)$$

$$\pi^* \text{ optimal } \forall \pi' \quad \pi^* \geq \pi'$$

$$V^*(s) = \max_a \sum_{s'} p(s, a, s') (R(s, a) + V^*(s'))$$

$$q^*(s, a) = \sum_{s'} p(s, a, s') (R(s, a) + \gamma \max_{a'} q^*(s', a'))$$

## Policy Evaluation (DP)

$$\pi(s) \in \arg\max_a \sum_{s'} P(s, a, s') (R(s, a) + \gamma V^\pi(s'))$$

$$q^\pi(s, \pi'(s)) \geq V^\pi(s)$$

stochastic

$$\sum_a \pi'(s, a) \cdot q^\pi(s, a) \geq V^\pi(s)$$

Review. 18/Oct/23

$$V(s) \leftarrow V(s) + \alpha \underbrace{\left( \sum_{k=0}^{\infty} \gamma^k R_{t+k} - V(s_t) \right)}_{\text{target "Monte Carlo Return"}}$$

$V_w$  estimate of  $V^\pi$  with weights  $w$ ,

$$w \leftarrow w - \alpha \nabla L(w) \quad L(w) = \mathbb{E} \left[ \frac{1}{2} (V^\pi(s) - V_w(s))^2 \right]$$

$$w \leftarrow w + \alpha \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k} - V_w(s_t) \right] \frac{\partial V_w(s_t)}{\partial w}$$

## Temporal Difference Learning (TD)

- How can we update before the end of an episode?

Bellman Eqn:

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P(s, a, s') (R(s, a) + \gamma V^\pi(s'))$$

$$= \mathbb{E}_{s,a} [R(s, a) + \gamma V^\pi(s')] \quad \text{estimate} \\ R_t + \gamma V(s_{t+1})$$

### TD UPDATE

Given a transition  $(s, a, r, s')$

$$V(s) \leftarrow V(s) + \alpha \underbrace{\left( \sum_{k=0}^{\infty} \gamma^k R_{t+k} - V(s) \right)}_{\text{breaks if rewards happens at the end only!}}$$

breaks if rewards happens at the end only!

Equivalently

$$V(S_t) \leftarrow V(S_t) + \gamma [R_t + \gamma V(S_{t+1}) - V(S_t)]$$

TD ERROR

Temporal Difference Error:

$$\begin{aligned} \delta_t &= R_t + \gamma V(S_{t+1}) - V(S_t) \\ &= r + \gamma v(s') - v(s) \end{aligned}$$

what we predicted

What actually happened

$$v(S_t) \leftarrow v(S_t) + \alpha \delta_t$$

- "Reward Prediction Error" (RPE)
- Not gradient algorithm.

Convergence?

$$L(w) = \mathbb{E} \left[ \frac{1}{2} (R_t + \gamma V_w(S_{t+1}) - V_w(S_t))^2 \right]$$

$$\begin{aligned} \frac{\partial L(w)}{\partial w} &= \mathbb{E} \left[ (R_t + \gamma V_w(S_{t+1}) - V_w(S_t)) \frac{\partial}{\partial w} (R_t + \gamma V_w(S_{t+1}) - V_w(S_t)) \right] \\ &= \mathbb{E} \left[ (R_t + \gamma V_w(S_{t+1}) - V_w(S_t)) \left( \frac{\gamma \partial V_w(S_{t+1})}{\partial w} - \frac{\partial V_w(S_t)}{\partial w} \right) \right] \end{aligned}$$

$$w \leftarrow w + \alpha (R_t + \gamma V_w(S_{t+1}) - V_w(S_t)) \left( \frac{\partial V_w(S_t)}{\partial w} - \gamma \frac{\partial V_w(S_{t+1})}{\partial w} \right)$$

$$v(S) \leftarrow v(S) + \alpha \delta$$

$$v(S') \leftarrow v(S') - \alpha \delta$$

implement in tabular case

"residual gradient"  
- much slower than TD

$$w \leftarrow w + \alpha \delta \left( \frac{\partial V_w(S_t)}{\partial w} - \frac{\gamma \partial V_w(S_{t+1})}{\partial w} \right)$$

$$V_w = \begin{bmatrix} v_w(s_1) \\ v_w(s_2) \\ \vdots \\ v_w(s_{1,1}) \end{bmatrix} \quad |S| \quad \frac{\partial V_w(S)}{\partial w} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow s$$

$$v(S) \leftarrow v(S) + \alpha [R_t + \gamma V(S') - V(S)]$$

$$w \leftarrow w + \alpha (R_t + \gamma V_w(S') - V_w(S)) \frac{\partial V_w(S)}{\partial w}$$

$$\mathbb{E}[S_t | S_t = s] = 0$$

$$\mathbb{E}[S_t | S_t = s, A_t = a] = ?$$

Positive TD Error

- 1).  $V(S)$  is too small
- 2). Stochastic environment + luck
- 3).  $V(S')$  is too big

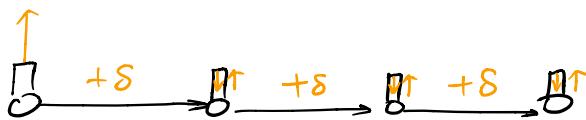
$$\begin{bmatrix} w \\ \vdots \\ w \end{bmatrix} \leftarrow \begin{bmatrix} w \\ \vdots \\ w \end{bmatrix} + \alpha \delta \left( \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} - \gamma \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} \right)$$

*s<sup>th</sup>*  
*s<sup>th</sup>*

$s, a, r, s'$



what TD will do



competing update.

	DP PF (PI, VI)	MC MC ES, TD(s)	TD TD
MUST KNOW P & R	YES	NO	NO
Must wait until end of episode	X	YES* *later we will do this without waiting	NO

### TD Properties

Tabular: Converges a.s. to  $v^\pi$  if  $\alpha_t$  decayed appropriately

Converges to  $v^\pi$  "in the mean" if  $\alpha$  sufficiently small constant  $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$

Linear : Converges <sup>(a.s)</sup> to  $w_\infty$

$$\text{MSVE}(w_\infty) \leq \frac{1}{1-\gamma} \text{MSVE}(w^*)$$

$\downarrow E[(v^\pi(s) - v_w(s))^2]$

where  $w^*$  are optimal weights.

