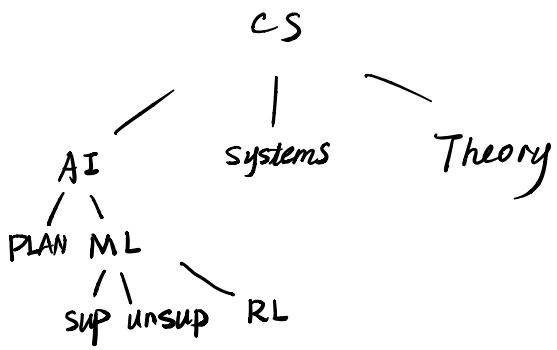
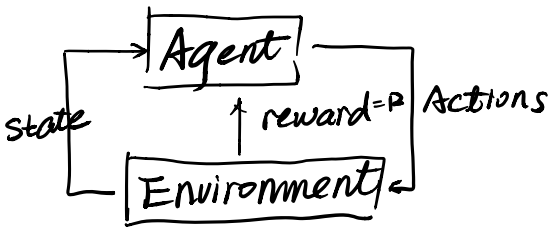
What's reinforcement Learning?

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

— Wiki



Agent Enviroment Diagram.



Key properties:
- Evaluate Feedback
- Sequential

Agent: Child, Dog, Robots, Program

Environment: World Lab, Software Enviroment

Neural Science: How do animals learn?

a specific agent of 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

≥50%

5?

49

Example Problem: Grid World

State: position of robot

Actions: Attempt ↑ ← → ↓ (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

"Staying" counts as re-entering state.

$\{1,2,3\}$ set $N>0$ $N\geq 0$

$\in$ elt $\times$ set X Random variables

$|X|$ 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" $N \geq 0$ iteration of the environment
- Let S_t 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

$S_t \xrightarrow{A_t} S_{t+1}$ R_t

MDP : $M = (S, A, P, d_R, d_0, \gamma)$

S = set of possible state of the environment

"State set". Finite for now. $\{1, 2, \cdots 23\}$

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

"Action set"

P = transition function, which describes how the state changes

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

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

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

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

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" $R: S \times A \to \mathbb{R}$

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

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

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

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

γ : 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

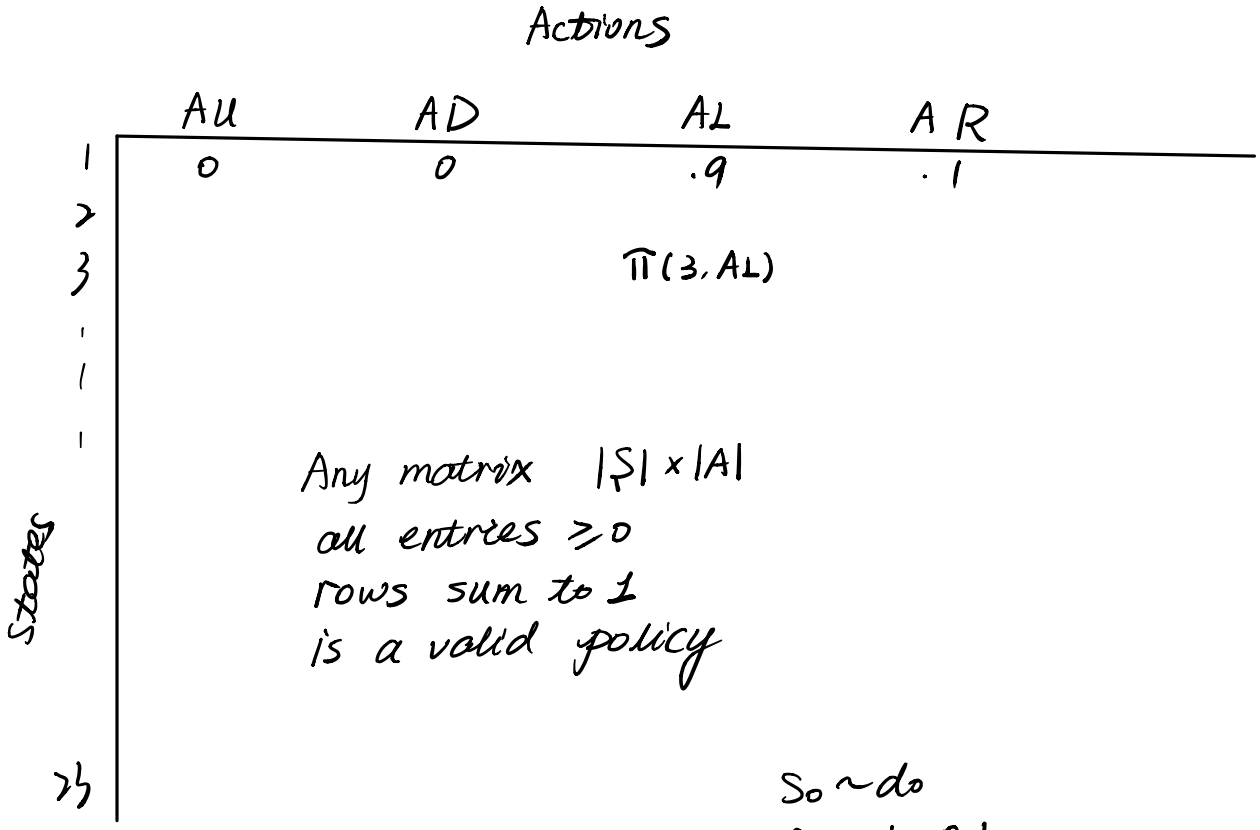
$\pi: S \times A \to [0, 1]$

$\pi(s,a) \triangleq P_r(A_t = a \mid S_t = s)$

Deterministic policy

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

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



$S_0 \sim d_0$

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

↓ sample from

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

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

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

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

$\vdots$

$S_0 \sim d_0$

for $t = 0, 1, \ldots$

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

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

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

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

Objective function:

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

↓ set of all policies

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

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

Optimal Policy:

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

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, t 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 γ ensures that $J(\pi)$ is finite

Terminal States:

A terminal state is a state that always transitions to a special state, S_∞, called the "terminal absorbing state"

- Once in S the agent can never leave

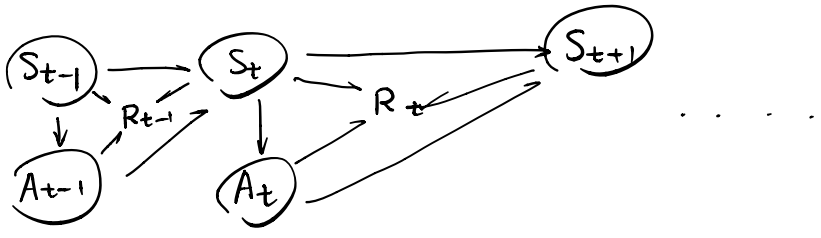
$$p(S_\infty, a, S_\infty) = 1$$

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

Episode:

When an agent reaches S_∞, the current trail called an "episode" ends, and we start a new one

- Set $t = 0$
- Sample $S_0 \sim d_0$
- Alert agent to this change



- d-seperation
- Markov blanket

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.

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

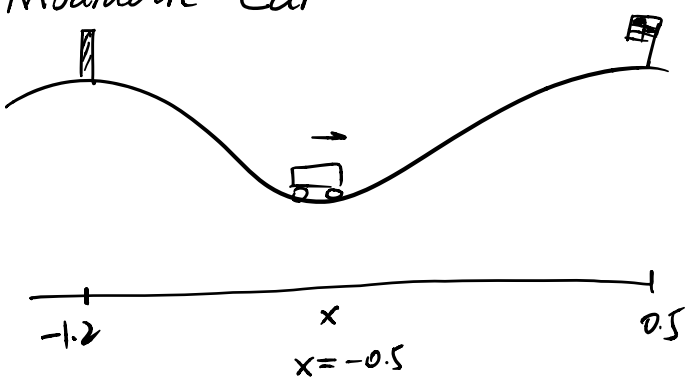
- A trajectory is the history of an entire episode H_∞
- The discounted return of a trajectory is the discounted sum of rewards.

$$G \triangleq \sum_{t=0}^{\infty} \gamma^t R_t \quad \xrightarrow{\text{simplify}} \quad J(\pi) = \mathbb{E}[G|\pi]$$

- The return from t is:

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

Mountain Car



State: $S = (X, v)$ ← velocity $\dot{x}$

Action: $a \in \{$forward, reverse, 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 → velocity = 0 if hit boundary

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

Rewards: $R_t = -1$ always

Discount: $\gamma = 1$

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

shaping rewards

Markov Property:

$$p(h, s, a, s') = Pr(S_{t+1} = s' \mid H_{t-1} = h, S_t = s, A_t = a)$$ 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, d_0, d_R are stationary

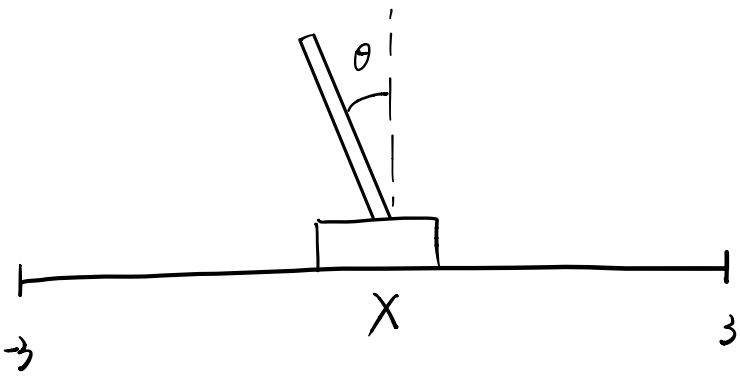
$$Pr(S_{t+1}=s' \mid S_t=s, A_t=a) = Pr(S_{i+1}=s \mid S_i=s, A_i=a) \quad \forall\, s,a,s',t,i$$

can be from different episodes.

- π is often non-stationary
  Stationary π is called a "fixed policy"

Finite-Horizon MDP

Cart-Pole:



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

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

$R_t = 1$ always

$\gamma = 1$

$S_0 = (0,0,0)$

Dynamics = physics of the system.

Episodes terminates after 20s or if the pole falls.

$\Delta t = 0.02$ sec

Walls at edges may end episodes

Finite-Horizon MDP

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

⇓

"horizon": Length of longest possible episode

- How implement the transition to S_∞ 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

1). Enumerate all policies and evaluate each for K episodes.

2). Phrase as a Black-Box optimization problem.

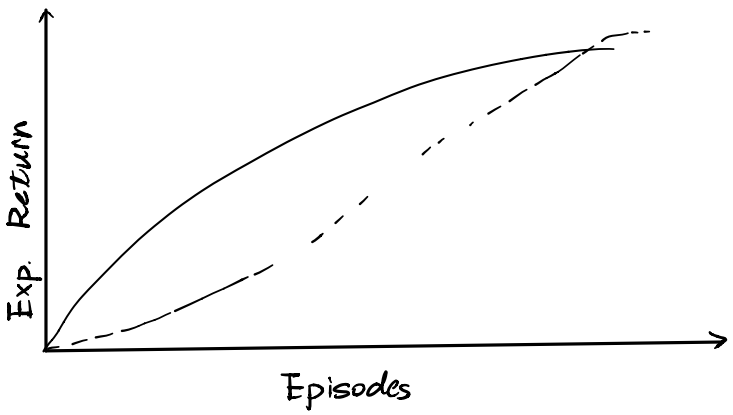
3). Model-based approaches.
↳ Estimate of transition Func
- Approximate P & R and solve for optimal policy for this model.

4) Value-Function based approaches

5) Gradient-Based methods (policy gradient)

Agent is implementation of:

```
for episode = 0, 1, 2, ...
    s ~ d0
    for t = 0, 1, 2, ...
        a = agent.getAction(s)
        s' ~ T(s, a)
        r ~ dR(s, a, s')
        agent.train(s, a, s', r)
        if s' = S∞
            break;
        s = s'
    agent.newEpisode();
```



Black-Box Optimization for policy search
- Simple agent
- Ignore MDP structure
- BBO algo solve problems of the form:

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

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

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

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

- For RL:

$$\arg\max_{\pi \in \Pi} J(\pi)$$

Estimate $J(\pi)$ by running π 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 π as a $|S| \times |A|$ matrix called P, with no constraints.

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

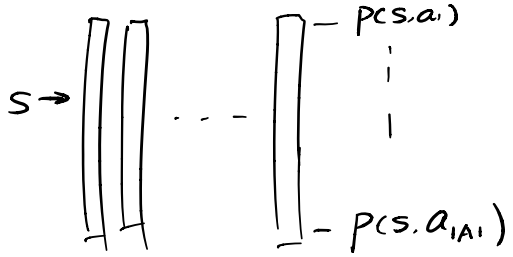
Tabular Softmax Policy

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

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

σ is constant scales "greediness"

- Drawback: can't represent deterministic policies.
- Policy parameters are typically denoted by θ, not P.

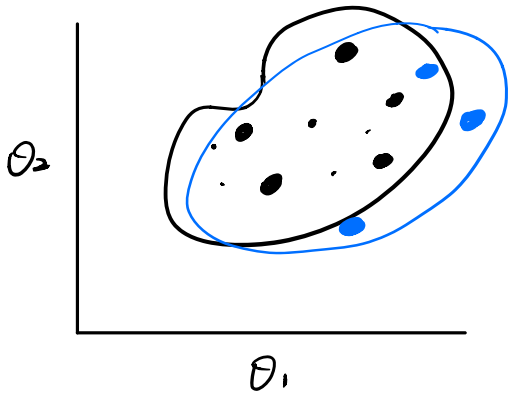


(rep. in NN)

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

Cross Entry Method.

- Stulp & Sigond



pseudo-code

Input θ : mean parameter vector

Σ : covariance matrix initially σ. const

K : number of elite

N : number of episodes to run for each

P : population

ε : small positive const.

1) For $i = 1$ to P

$\theta_i \sim N(\theta, \Sigma)$

vector of policy params

$\hat{J}_i$ = evaluate (θ_i, N)

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

keep paired values w/ policies

sort based on that produce

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

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

Repeat from 1

evaluate (θ, N)

1) Run N episodes with policy parameters θ

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

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

Parameterized Policy:

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

$\pi(s, a, \theta) = Pr(A_t = a \mid S_t = s, \theta)$

$\arg\max_{\pi \in \Pi} J(\pi)$

$\downarrow$

$\arg\max_{\theta \in \mathbb{R}^n} J(\theta)$

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

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

State-Value Function:

$v^\pi: \mathcal{S} \to \mathbb{R}$

$v^\pi(s) \triangleq \mathbb{E}[G \mid S_0 = s, \pi]$

$= \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s, \pi]$

$= \mathbb{E}[G_t \mid S_t = s, \pi]$

$= \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k} \mid S_t = s, \pi]$

: Expected discounted sum of rewards if the agent follows policy π from state s

- How "good" is it for the agent to be in state s.
- Value of state : s
- Does depend on policy π

1.) Describe env

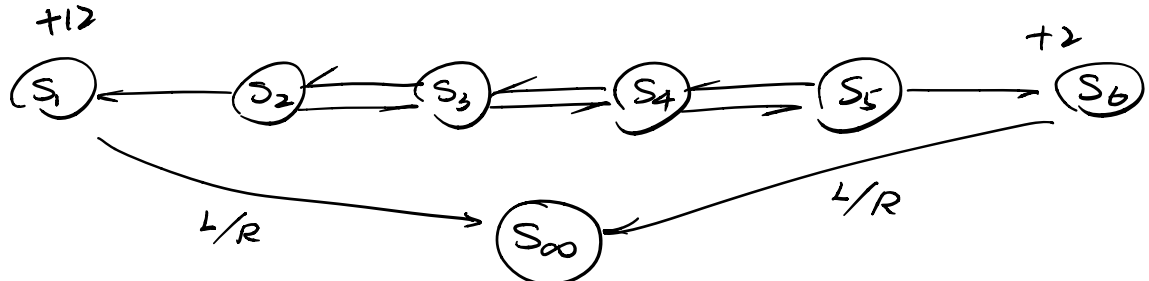
2). BBO – ignore structure

3) Value Functions
↳ Tool used

RL algo

R reward function

R_t reward at t.



$\gamma = 0.5$

π_1 = Left always

π_2 = Right always

value of terminal states is always zero.

$V^{\pi_1}(s_1) = 0$	$V^{\pi_2}(s_1) = 0$
$V^{\pi_1}(s_2) = 12$	$V^{\pi_2}(s_2) = 0.25$
$V^{\pi_1}(s_3) = 6$	$V^{\pi_2}(s_3) = 0.5$
$V^{\pi_1}(s_4) = 3$	$V^{\pi_2}(s_4) = 1$
$V^{\pi_1}(s_5) = 1.5$	$V^{\pi_2}(s_5) = 2$
$V^{\pi_1}(s_6) = 0$	$V^{\pi}(s_6) = 0$

Markov. P

$$w^{\pi}(s,h) = \mathbb{E}[G_t \mid S_t = s, H_{t-1} = h, \pi]$$

Action-Value Function / State-Action Value Function / Q-function.

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

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

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

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

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

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

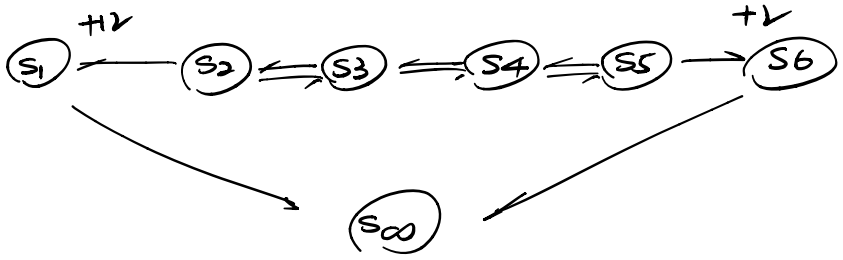
$$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k} \mid S_t = s, A_t = a, \pi\right] =$$

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

$$= \mathbb{E}\left[R_t + \sum_{k=1}^{\infty} \gamma^k \cdot R_{t+k} \mid \cdots\right]$$

$$= \sum_a \pi(s,a) \sum_{s'} P(s,a,s') + \gamma \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k+1} \mid \cdots\right]$$

- Expected-discounted return if the agent takes action a in state s, and follows policy π thereafter.
- How "good" is it to take action a in state s? (if using policy π otherwise)



π_1: Left

π_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. operator"

Bellman Equation for v^π

$$v^\pi(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k} \,\middle|\, S_t = s, \pi\right]$$

$$= \mathbb{E}\left[R_t + \sum_{k=1}^{\infty} \gamma^k \cdot R_{t+k} \,\middle|\, S_t = s, \pi\right]$$

$$= \sum_a \pi(s,a) \cdot R(s,a) + \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^{k+1} \cdot R_{t+1+k} \,\middle|\, S_t = s, \pi\right]$$

$\Downarrow$

$\gamma \cdot \gamma^k$

$\Downarrow$

$$\gamma\, \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \cdot R_{t+1+k} \,\middle|\, S_t = s, \pi\right]$$

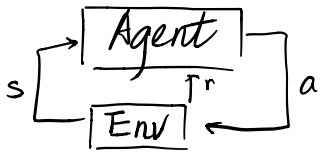
$$= \sum_a \pi(s,a)\, R(s,a) + \sum_a \pi(s,a) \sum_{s'} p(s,a,s') \cdot \gamma\, \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \,\middle|\, \cancel{S_t = s, A_t} = a, S_{t+1} = s'\right]$$

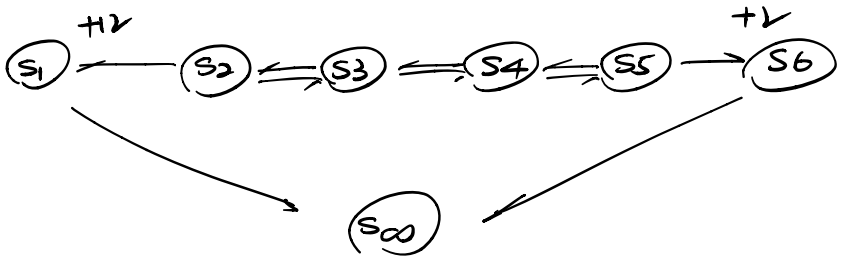
(π, γ)

Markov

$v^\pi(s')$

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





$\gamma = 0.5$.

π_1 = left

π_2 = right

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

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

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

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

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

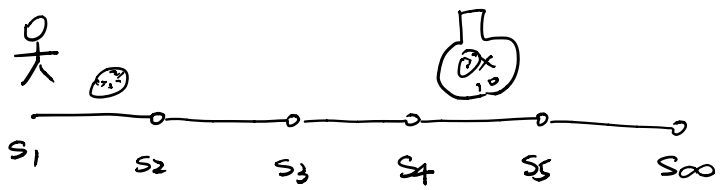
$V^{\pi_1}(s_4) = 0 + \gamma \cdot V^{\pi_1}(s_3) = 3$

$$V^{\pi}(s) = \mathbb{E}[R_1 + \gamma R_2 - - - \mid S_1 = s]$$

$$= \mathbb{E}[R_1 + \gamma (R_2 + \gamma R_3 \cdots) \mid S_1 = s]$$

$\Downarrow$

$V^{\pi}(s_2)$



$V^{\pi}(s_4) = 10$

$V^{\pi}(s_3) = 0 + \gamma \cdot V^{\pi}(s_4) = \gamma \cdot 10$

$V^{\pi}(s_2) = 0 + \gamma \cdot V^{\pi}(s_3) = \gamma^2 \cdot 10$

$V^{\pi}(s_1) = 1 + \gamma \cdot V^{\pi}(s_2) = 1 + \gamma^3 \cdot 10$

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

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 π, π' s.t. $\pi \not\geq \pi'$, $\pi' \not\geq \pi$

Optimal policy: any policy π^*

s.t. $\forall \pi, \; \pi^* \geq \pi$

Intuition: Given π, π'.

Construct π'' that applies π in states where it's better and applies in other states.

→ 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:

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

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

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

v^*, q^* are unique, unlike π.

— Given v^*, but don't know P & R, can you act optimally? No.

In state s, choose any action a that maximizes

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

$$= q^*(s,a)$$

What about q^*? Yes.

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

Bellman Optimality Equation

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

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

$$= \mathbb{E}\left[R_t + \underbrace{\sum_{k=1}^{\infty} \gamma^k \cdot R_{t+k}}_{\Downarrow} \;\middle|\; S_t=s, A_t=a, \pi^*\right]$$

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

$$= R(s,a) + \gamma \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a, \pi^*\right]$$

$$= R(s,a) + \sum_{s'} P(s,a,s') \underbrace{\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k+1} \mid \cancel{S_t = s}\ \cancel{A_t = a}, S_{t+1} = s', \pi^*\right]}$$

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

Bellman Optimality Eqn.

$$\textcircled{D} = \max_a \sum_{s'} P(s,a,s') \left(R(s,a) + \gamma V^*(s')\right)$$

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

Bellman optimality equations only hold for optimal policies.

Policy Iteration:

- policy evaluation.

  Give a policy π, find V^π

  - Assume P & R are known.

Method:

1) Solve Bellman equation
2) Dynamic programming. π^*

Dynamic Programming.

sequence of value function approximations. $\hat{V}_0^\pi$

$V_1, V_2, V_3 \cdots$ where $V_i: S \to \mathbb{R}$

arbitary guess (often zero vector)

$$V_{i+1}(s) = \sum \pi(s,a) \sum P(s,a,s') \left(R(s,a) + \gamma V_i(s)\right)$$

a s'

Properties:

$v_{i+1} = v_i$

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

2) $v_i \longrightarrow v^{\pi}$ as $i \longrightarrow \infty$ (not proven here)

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

$V_0 =$

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

S … G

$R_t = -1$, $\gamma = 1$ π: go down, unless bottom row, then right

$V_1 =$

-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1+0	-1	-1	-1

-1, S∞

$V_2 =$

-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

$V_4 =$ - -

$V_5 =$ - -

$V_7 =$

-7	-6	-5	-4
-6	-5	-4	-3
-5	-4	-3	-2
-4	-3	-2	-1

In-Place Implementation:

- When updates $v_i(s)$, 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: $q^{\pi}(s,a)$

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

- 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 Theorm:

Let π and π' be deterministic policies such that

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

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

$$\pi' > \pi$$

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

Proof:

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

(A_t comes from π')

$$= \mathbb{E}\left[R_t + \gamma V^{\pi}(S_{t+1}) \mid S_t = s, \pi'\right]$$

$$\leq \mathbb{E}\left[R_t + \gamma \cdot q^{\pi}(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s, \pi'\right]$$

$$= \mathbb{E}\left[R_t + \gamma\, \mathbb{E}\left[R_{t+1} + \gamma V^{\pi}(S_{t+2}) \mid S_t = s, \pi'\right] \mid S_t = s, \pi'\right]$$

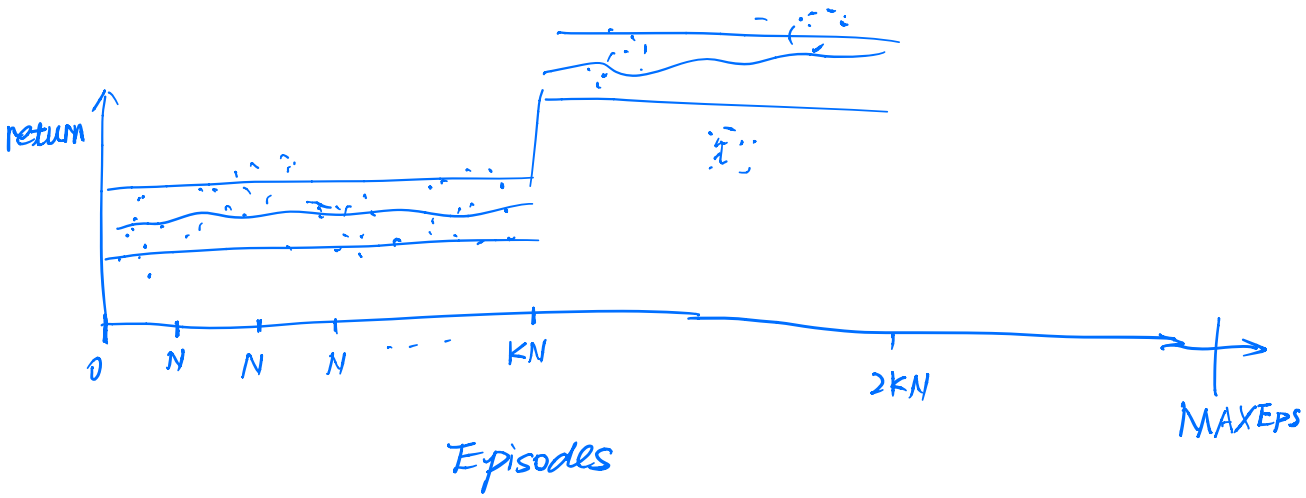
$$= \mathbb{E}[R_t + \gamma R_{t+1} + \gamma^2 V^{\pi}(S_{t+2}) \mid S_t = s, \pi']$$

$$\leq \text{"} \quad \text{"} \; q^{\pi}(S_{t+2}, \pi'(S_{t+2})) \text{"} \quad \text{"}$$

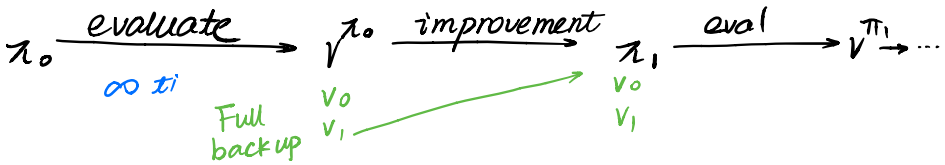
$$\leq \mathbb{E}[R_t + \gamma R_{t+1} \cdots \mid S_t = s, \pi']$$

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

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



Policy Iteration



value iteration.

What if $\pi' = \pi$

$$\pi'(s) \in \arg\max_a q^{\pi}(s,a)$$

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

$\forall s$

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

↑ Bellman optimality eqn.

Value Iteration.

V_0: arbitrary value function

π_0: arbitrary policy · deterministic

V_1: $\forall s \quad V_1(s) = \sum_{s'} P(s, \pi_0(s), s') \cdot (R(s, \pi_0(s)) + \gamma V_0(s'))$

π_1: $\forall s \quad \pi_1(s) \in \arg\max_a \sum_{s'} P(s,a,s') (R(s,a) + \gamma V_1(s'))$ $\quad \hat{q}(s,a)$

V_2: $\forall s \quad V_2(s) = \sum_{s'} P(s, \pi_1(s), s') (R(s, \pi_1(s)) + \gamma V_1(s'))$

π_2: $\forall s \quad \pi_2(s) \in \arg\max_a \sum_{s'} P(s,a,s') (R(s,a) + \gamma V_2(s'))$

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 $\mathbb{R}^{|S|}$.

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

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

$\mathcal{T}(V_k) \triangleq V_{k+1}$

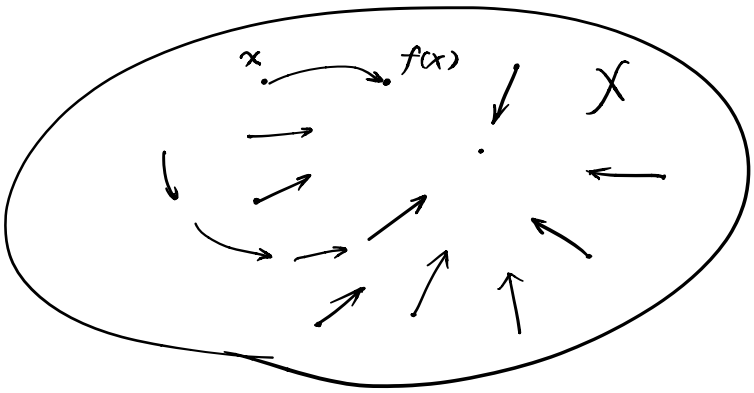
$\mathcal{T}_{V_k}(s) = V_{k+1}(s)$

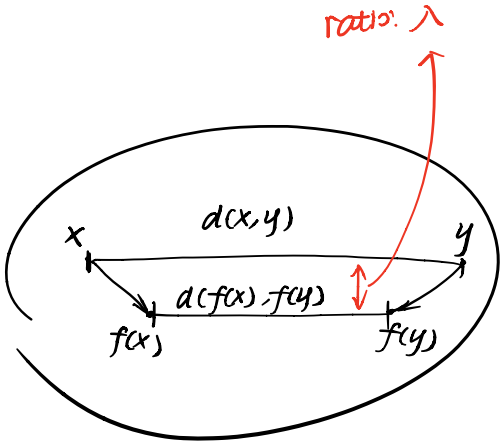
operator: function that takes elements of a space as input and produces elements of the same space as output

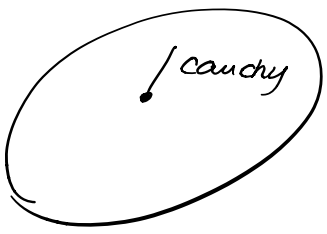
Contraction Mappings:

$f: S \to S$ 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: $f \leftarrow T$, $X = \mathbb{R}^{|S|}$

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

Show:

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

$$\|Tv - Tv'\| \leq \gamma \|v - v'\|$$

Oct 18. 7pm - 9pm. ISB 135

No class on Tuesday

Bellman operator is a contraction mapping:

$$T_V(s) = \max_{a \in A} \sum_{s'} P(s,a,s') \cdot (R(s,a) + \gamma V(s'))$$

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

$d(f(x), f(y)) \leq \lambda d(x,y)$ (from defi)

$$\Downarrow$$

$$\|T_V - T_{V'}\| \leq \gamma \|V - V'\| \qquad \forall\ V, V'$$

$$= \max_s |(T_V - T_{V'})(s)|$$

$$\|T_V - T_{V'}\| = \max_s |T_V(s) - T_{V'}(s)|$$

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

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

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

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

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

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

$$\Downarrow$$

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

Then:

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

$$\leq \max_s \max_a \left| \sum_{s'} P(s,a,s')\left(R(s,a)+\gamma V(s')\right) - \sum_{s'} P(s,a,s')\left(R(s,a)+\gamma V'(s')\right) \right|$$

$$= \gamma \max_s \max_a \left| \sum_{s'} P(s,a,s')\left(V(s')-V'(s')\right) \right|$$

↓ weighted average is $\leq$ the max value

$$\leq \gamma \max_s \max_a \max_{s'} \left| V(s')-V'(s') \right| = \gamma \max_{s'} \left| V(s')-V'(s') \right|$$

$$= \gamma \|V-V'\|$$

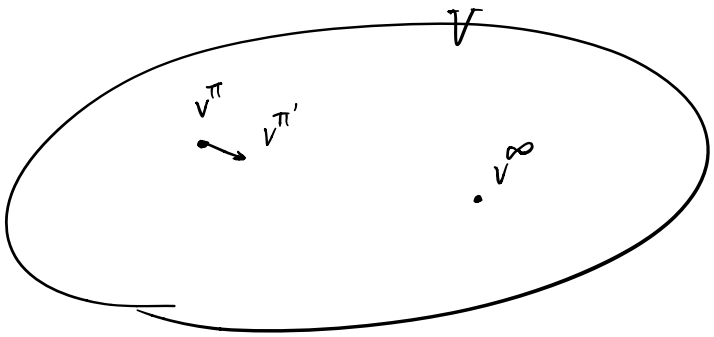
- Value iteration converges to a unique fixed point V^∞
- π^∞ be greedy w.r.t. V^∞

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

5%: there exists an optimal policy using V^π definition

$$V^{\pi^\infty}(s) = \sum_{s'} P(s,\overset{\uparrow}{\pi^\infty(s)},s')\left(R(s,\overset{\uparrow}{\pi^\infty(s)})+\gamma V^\infty(s')\right)$$

$$V^{\pi^\infty}(s) = \max_a \sum_{s'} P(s,a,s')\left(R(s,a)+\gamma V^\infty(s')\right)$$ Bellman Optimality Eqn



$V^{\pi^*} \not< V^{\pi'}$

$V^\pi \neq V^{\pi'}$

policy imp Thm

$\pi' > \pi$

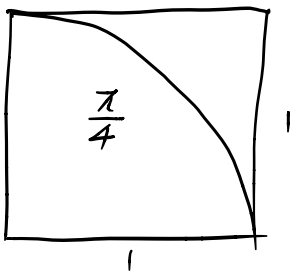
π not optimal

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: π



Policy Evaluation:

state Sapple — Sapple

History $H = (S_0, A_0, R_0, \ldots S_i)$

Generated by running π for one episode.

Monte Carlo estimate of $V^{\pi}(S_0) = \sum_{t=0}^{\infty} \gamma^t \cdot R_t = G$

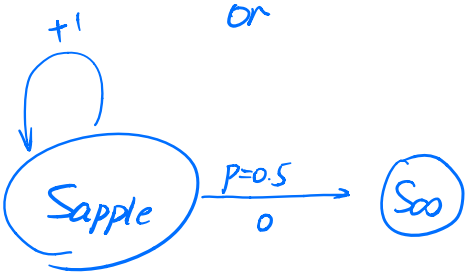
Sapple occurs many times in an episode

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

$V^{\pi}(\text{Sapple}) = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$

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

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



$V^{\pi}(S_a) = 0.5 \cdot 0 +$

Policy Evaluation: → start at s

History H, generated by π

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

First Visit Monte Carlo → estimating v^{π}

Intuition: Generate many episode of data

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

Pseudocode

Initialize:

$\pi \leftarrow$ policy to be evaluated

$V \leftarrow$ an arbitrary state value function.

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

Repeat Forever:

1) Generate an episode using π.

2) For each state S, appearing in the episode:

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

[if S_t is the first occurence of S, then $G = G_t = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k}$]

Append G to Returns(s)

V(s) $\leftarrow$ average(Returns(s))

Return V.

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

$$X_n \xrightarrow{as} X \quad \text{iff}$$

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

Khintchine Strong Law of Large Numbers

Let $X_1 X_2, \cdots$ be an infinite independent i.i.d. random variables, finite variance, same mean, 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 $\mathbb{E}[X_1]$ i.e.

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{a.s.} \mathbb{E}[X_1] \qquad Pr(\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} x_i = \mathbb{E}[X_1]) = 1$$

Properties of FVMC:

1. Converges (in probability and almost surely) to V^{π} if every state is visited infinitely often.

Proof:

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

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

G^i: 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 $X_1, X_2, \cdots$

⋮

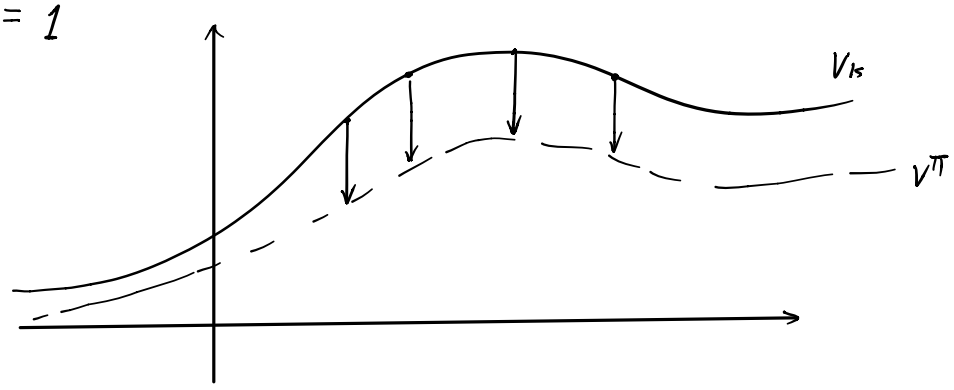
SEE NOTE!

$V_k(s) \xrightarrow{a.s} v^\pi(s) \quad \forall s$

$Pr(\lim_{n\to\infty} V_k(s) = v^\pi(s)) = 1, \quad \forall s$

$\Downarrow$ finite states (or at least countable)

$Pr(\forall s, \lim_{n\to\infty} V_k(s) = v^\pi(s)) = 1$



Every Visit Monte Carlo

- Use the return from every visit to s during an episode
  "and each time, t, that it occured"
- Also converges almost surely to v^π.

MC Estimation of Action Values

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

Problem: What if π never chooses action a in state s?

Solution: Exploring starts:

- Randomize s_0 and A_0 s.t. every (s,a) pair has non-zero probability.

Solution: Stochastic policies.

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

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

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