REINFORCEMENT LEARNING

Replace Y with reward (Y is the supervision)

Maximize reward over time

In supervised learning i.i.d., goal was to do well in that example, however in R.L. the goal is to maximize reward over time

The concept of time makes R.L. special

Markov Decision Process (MDP)

Tuple $(S, A, \{P_{sa}\}, \gamma, R)$

S - set of states

A - set of actions

$P_{sa} \rightarrow$ Transition probabilities $S_{next} \sim P_{S_{current} \cdot a_{current}}$

$\gamma \in (0,1)$ - Discount factor

$$\left. \begin{array}{c} R: S \times A \rightarrow \mathbb{R} \\ \text{or} \\ R: S \rightarrow \mathbb{R} \end{array} \right\} \rightarrow \text{Reward Function}$$

3	-0.02	-0.02	-0.02	+1
2			-0.02	-1
1				
	1	2	3	4

$S = \{(1,1), (1,2) \dots\}$

$|S| = 11$

$A = \{N, S, W, E\}$

$$P_{sa} = \begin{array}{cl} \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0.1 \\ 0.1 \\ 0.8 \\ 0 \\ 0 \\ 0 \end{array}\right] & \begin{array}{l} \\ \\ \\ \rightarrow (1,4) \\ \rightarrow (1,2) \\ \rightarrow (2,3) \\ \\ \\ \\ \end{array} \end{array}$$

for $s = (1,3)$
$a = N$

(Assumption that 80% of times when $a = N$, it goes N, 10% west, 10% east)

$R(S) \rightarrow \mathbb{R}$ or $R(s,a) \rightarrow \mathbb{R}$

$R(S)$: reward depends on state

$R(s,a)$: sometimes even if state is the same, action taken to reach may have different cost, so to account for it, we use (s,a)

$R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \gamma^3 R(S_3)$ → maximize this

$S_0 \xrightarrow{a_0} S_1 \xrightarrow{a_1} S_2 \xrightarrow{a_2} S_3 \xrightarrow{a_3} \ldots$ Trial / Episode / Trajectory

$S_1 \sim P_{S_0 a_0}$ $\quad$ $S_2 \sim P_{S_1 a_1}$

[Why is P_{sa} a probability vector?

→ Suppose robot is placed in an environment, you have told it to go 100cm, it may go 95/105 cm. Hence, there is stochasticity. Hence to account for it, we have probability]

γ → Discount factor. It incentivizes model to earn large positive rewards sooner, as it discounts reward. Another way to think, we push negative reward at the end so that it is highly discounted.

[In Finance, γ is the interest rate, we want profits sooner & loss later]

Policy $\pi : S \to A$

We want to learn a policy that maximizes value

Value: $V^{\pi} : S \to \mathbb{R}$

$$V^{\pi}(s) = \mathbb{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) \ldots \mid s_0 = s, \pi\right]$$

starting state

Rewards can be random because states are random due to stochastic nature of transition probabilities

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

($s = s_0$ is not random)

$$V^{\pi}(s) = R(s) + \gamma \mathbb{E}(R(s_1) + \gamma R(s_2) \ldots)$$

$$V^{\pi}(S) = R(S) + \gamma \mathbb{E}(V^{\pi}(S_1))$$

$$= \underbrace{R(S) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')}_{\text{Bellman Equation}}$$

future state

If our goal was to maximize immediate reward, back to supervised learning

R.L. → focus on value. We look at long term reward

Policy and value duals of each other

Policy	Value
$S \to A$	$S \to R$
π	V^{π}
π	V

the action is optimized to reach next state with highest value.

$\pi \to V^\pi$

$$V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^\pi(s)$$

$$V^\pi = \begin{array}{c} \left[\begin{array}{c} - \\ - \\ - \\ \vdots \\ \vdots \end{array}\right] \end{array} \begin{array}{ll} s_1 & V^\pi(s_1) \\ s_2 & V^\pi(s_2) \\ s_3 & \vdots \\ & \vdots \\ & \end{array}$$

$$V^\pi(s_1) = R(s_1) + \gamma \sum_{s' \in S} P_{s\pi(s_1)}(s') V^\pi(s')$$

$$V^\pi(s_2) = R(s_2) + \gamma \sum_{s' \in S} P_{s\pi(s_2)}(s') V^\pi(s')$$

$$\vdots$$

$$V^\pi(s_{|S|}) = R(s_{|S|}) + \gamma \sum_{s' \in S} P_{s\pi(s_{|S|})}(s') V^\pi(s')$$

$$P^{\pi} = \left.\underbrace{\begin{bmatrix} \text{———} \; P_{s_1,\pi(s_1)} \; \text{———} \\ \vdots \\ \text{———} \; P_{s_i \pi(s_i)} \; \text{———} \\ \\ \end{bmatrix}}_{|S|}\right\} |S|$$

$$\underbrace{V^{\pi}}_{|S|} = \underbrace{R}_{|S|} + \gamma \underbrace{P^{\pi}}_{|S| \times |S|} \underbrace{V^{\pi}}_{|S|}$$

$$V^{\pi} = [I - \gamma P^{\pi}]^{-1} R$$

Optimal Value Function

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

$$V^*(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s')$$

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V^*(s')$$

VALUE ITERATION

Algorithm

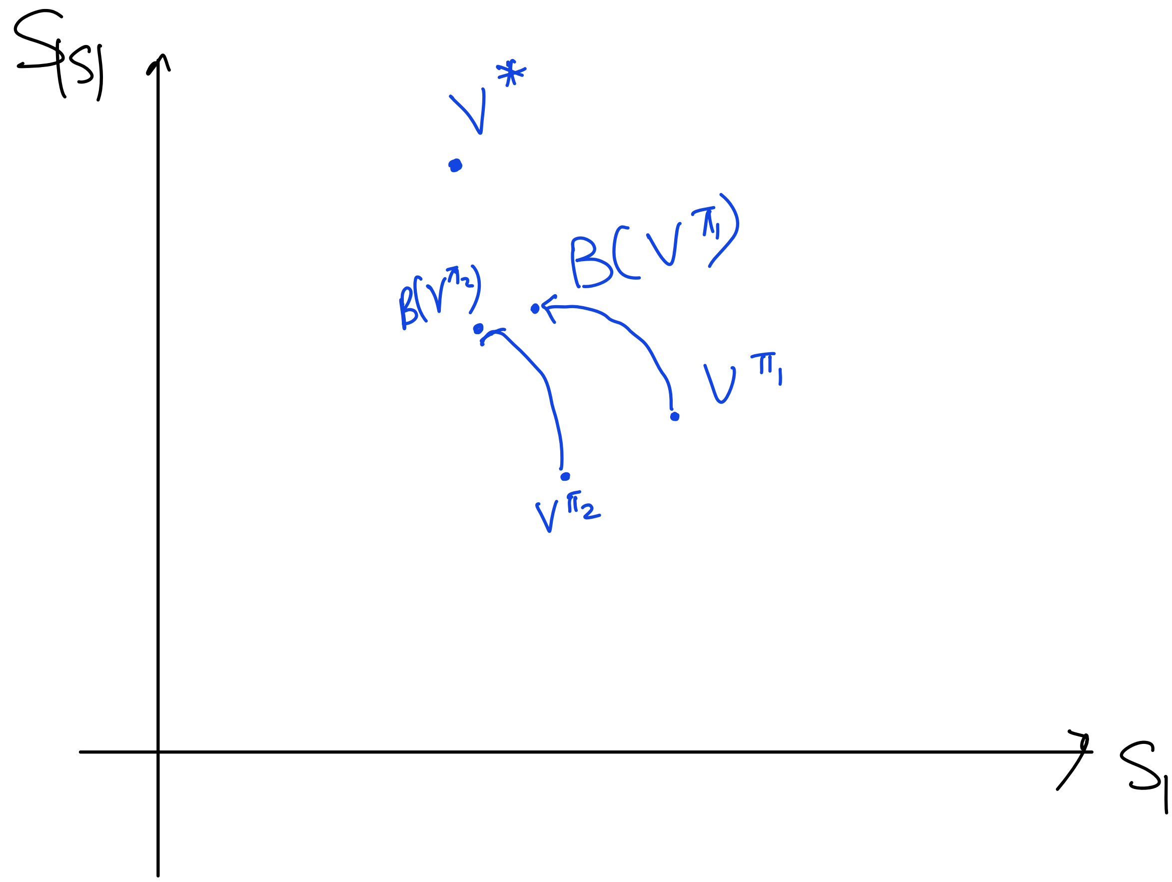
1. For each state s, initialize $V(s) := 0$
2. Repeat until convergence

   For every state, update

$$V(s) := R(s) + \max_{a \in A} \gamma \sum_{s'} P_{sa}(s') V(s')$$

Bellman Backup Operator

$O(|S|^2 |A|)$



$V^{\pi}(S_1)$ is the projection of S_1

Bellman operator is contraction mapping.

Suppose you have 2 points in space.

You apply Bellman. The result will be

closer than the input. They all converge to a single point → fixed point. V^*

POLICY ITERATION

1. Initialize π randomly
2. Repeat until convergence $\leftarrow O(|S|^3)$

   (a) Set $V := V^{\pi}$

   (b) For each state s, set

   $$\pi(s) := \arg\max_{a \in A} \sum_{s'} P_{sa}(s') V(s')$$

   $\hookleftarrow O(|A||S|)$

Policy: you will surely converge

Value: you get closer but may not reach final.

1. Policy $\pi(s) = a$

2. $V^{\pi}(s) = E[R(s_0) + \gamma R(s_1) \ldots \mid s_0 = s, \pi)$

$$= R(s) + \gamma \mathbb{E}_{s' \sim P_{sa}}[V^{\pi}(s')]$$

$\gamma < 1$ makes $V^{\pi}(s)$ bounded

Optimal Value

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

$$= R(s) + \max_{a} \underbrace{\mathbb{E}_{s' \sim P_{sa}}[V^{*}(s')]}_{\text{highest expected reward}}$$

Optimal Policy

$$\pi^*(s) = \arg\max_a E_{s' \sim P_{sa}}[V^*(s')]$$

$$V^\pi = (I - P^\pi)^{-1} R$$ (Policy evaluation)

1

$$\begin{tikzcd} \pi \arrow[r, bend left] & V \arrow[l, bend left] \end{tikzcd}$$

$$\pi(s) = \arg\max_a E_{s \sim P_{sa}}[V(s')]$$

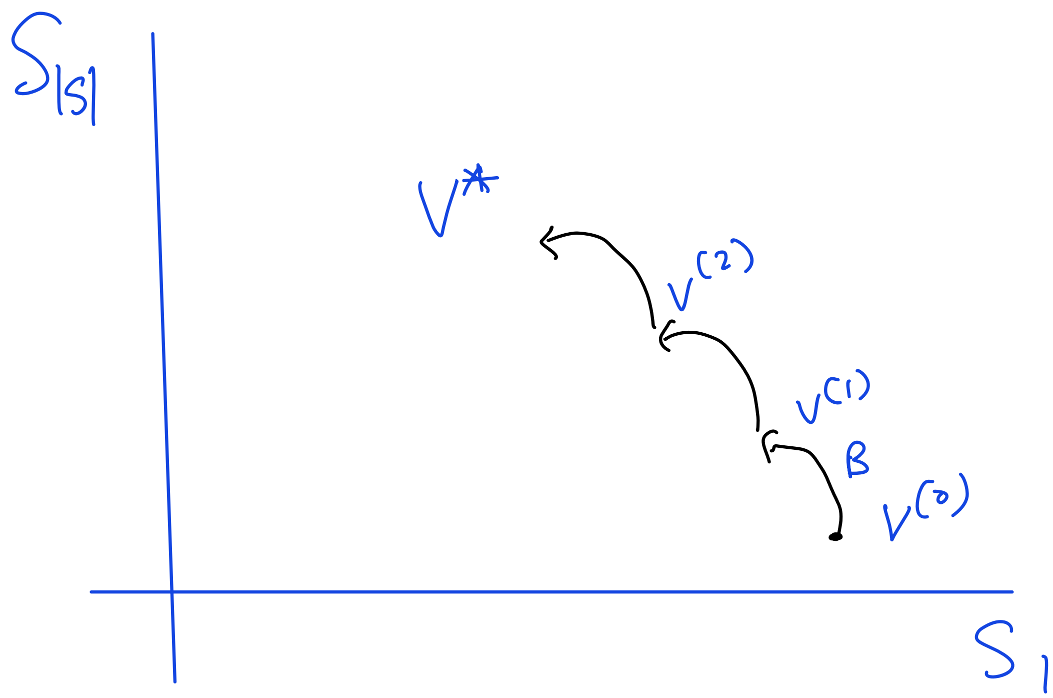
(Greedy Policy wrt A)

Value Iteration

Loop:

$$V^{(t+1)} = B(V^{(t)})$$

$$B(V(s)) = R(s) + \max_a \mathbb{E}_{s' \sim P_{sa}} [V(s')]$$



V^* is its fixed point

Policy Iteration

Loop

1. $V \leftarrow \pi$ using Policy evaluation

2. $\pi \leftarrow V$ greed Policy

Policy we get in step 2 is different from that in 1

P_{sa} is not given

Model based vs Model free

Model = P_{sa} Here model refers to environment

Learning Model

$$s_0^{(1)} \xrightarrow{a_0^{(1)}} s_1^{(1)} \; \cdots \; - -$$

$$s_0^{(2)} \xrightarrow{a_0^{(2)}} s_1^{(2)} \; - \cdot - \cdot$$

$$\vdots$$

$$\hat{P}_{sa}(s') = \frac{\text{\# of time we took action } a \text{ at state } s \text{ \& got to state } s'}{\text{\# of time we took action } a \text{ at state } s}$$

Sometimes $= \frac{0}{0} \Rightarrow$ Uniform