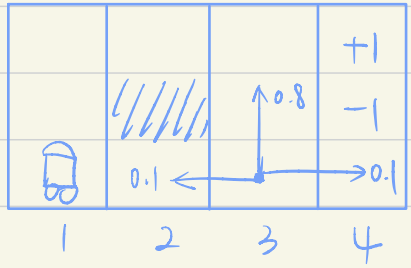
Lecture 17 RL: MDPs & Value / Policy Iteration

1. Markov Decision Process (MDP)

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

- S - Set of states
- A - Set of actions
- P_{sa} - State transition probabilities ($\sum_{s'} P_{sa}(s') = 1$)
- γ - discount factor, $\gamma \in [0, 1)$
- R - reward function

Example:



11 states, ~~Action~~: $\{N, S, E, W\}$

$P_{(3,1)N}((3,2)) = 0.8$

$P_{(3,1)N}((4,1)) = 0.1$

$P_{(3,1)N}((2,1)) = 0.1$

$P_{(3,1)N}((3,3)) = 0$

$R((4,3)) = +1$ reach destination asap.

$R((4,2)) = -1$

$R(s) = -0.02$ for all other states

Process: $S_0 \to$ choose action $a_0 \to$ get to $S_1 \sim P_{s_0,a_0} \to$ choose action $a_1 \to$ get to $S_2 \sim P_{s_1,a_1}$ .....

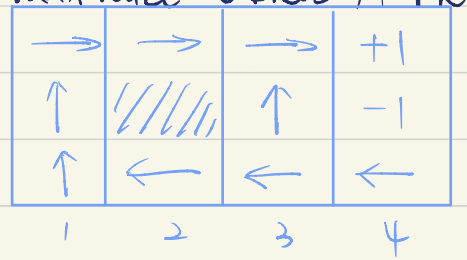
Total payoff: $R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots$

$\gamma = 0.99$ (close to 1) $\Rightarrow$ encourage to get reward faster

Goal: Choose actions over time to maximize $E[R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots]$

Result: find a policy $\pi: S \to A$

when in state s, take action $\pi(s)$



2. Some definition & Preparation

Define: V^π, V^*, π^*

- V^π: For a policy π, $V^\pi: S \to \mathbb{R}$ is s.t. $V^\pi(s)$ is the expected total payoff for starting in state s and executing π.

  is the value function for policy π.

$$V^{\pi}(s) = E[R(S_0) + \gamma R(S_1) + \cdots \mid \pi, S_0 = s]$$

Example:

→	→	→	+1
↓	////	→	−1
→	→	↑	↑

π

0.52	0.73	0.77	+1
−0.9	////	−0.82	−1
−0.88	−0.87	−0.85	−1.00

V^{π}

- Bellman's equations for V^{π}

$$V^{\pi}(s) = R(S_0) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^{\pi}(s')$$

<proof>:

$$V^{\pi}(s) = E[R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots \mid \pi, S_0 = s]$$

$$= E[R(S_0) + \gamma (R(S_1) + \gamma R(S_2) + \cdots) \mid \pi, S_0 = s]$$

$R(S_0)$: immediate reward

$R(S_1) + \gamma R(S_2) + \cdots = V^{\pi}(S_1)$: Expected future rewards

Let $s = S_0$, $s' = S_1$, $V^{\pi}(s) = E[R(s) + \gamma V^{\pi}(s')]$

$\because s' \sim P_{s\pi(s)}$ (In state s, take action $a = \pi(s)$)

$$\Rightarrow V^{\pi}(s) = R(S_0) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^{\pi}(s')$$

Given π, get a linear system of equations in terms of $V^{\pi}(s)$

e.g. $V^{\pi}((3,1)) = R((3,1)) + \gamma(0.8V^{\pi}((3,2)) + 0.1V^{\pi}((2,1)) + 0.1V^{\pi}((4,1)))$

- V^* is the optimal value function

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

  - Bellman's equations for V^*: expected future rewards

    $V^*(s) = R(s) + \max_a \gamma \sum_{s'} P_{sa}(s') V^*(s')$

    (P_{sa}: if take action a)

- π^* is the optimal policy

  $\pi^*(s) = \operatorname{argmax}_a \gamma \sum_{s'} P_{sa}(s') V^*(s')$ (γ can be omitted)

strategy: ① Find V^* ← Value Iteration

② Use argmax equation to find π^*

3. Value Iteration

Initialize $V(s) := 0$ for every s.

For every s, update: $V(s) := R(s) + \max_a \gamma \sum_{s'} P_{sa}(s') V(s')$ (use synchronous update)

old estimate of value

new estimate of value

(Bellman backup operator, $V := B(V)$)

Value iteration can make V converge to V^*

4. Policy Iteration

Initialize π randomly.

Repeat:

V^π – linear system of equations

Set $V := V^\pi$, (i.e. solve Bellman's equations to get V^π)

Set $\pi(s) := \operatorname{argmax}_a \sum_{s'} P_{sa}(s') V(s')$

will also converge.

5. Pros and Cons of value iteration and policy iteration.

policy iteration: solve V^{π}, a linear system of equations; when state space is small, then efficient; but if large, then slow; get the optimal value

value iteration: converge to V^{*}; more frequent use.

6. What if don't know P_{sa}?

$$P_{sa}(s') = \frac{\#\text{ times took action "a" in state s and get to } s'}{\#\text{ times took action "a" in state s.}}$$, or $\frac{1}{|S|}$ if above is $\frac{0}{0}$

Putting it together:

Repeat: {

Take actions wrt. π to get experience in MDP.

Update estimates of P_{sa}. (and possibly R)

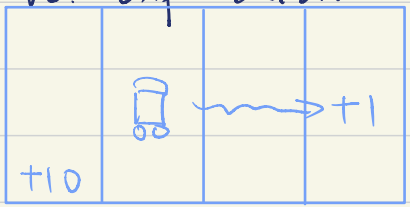
Solve Bellman's equation using value iteration to get V.

}

Update $\pi(s) := \arg\max_a \sum_{s'} P_{sa}(s') V(s')$

7. Exploration vs. Exploitation

example:



if it happens to reach +1 the first time, it may ignore +10, and thinks going to +1 is a good way. "greedy"

Modification:

Repeat: {

0.9 chance wrt. π

0.1 chance randomly $\Rightarrow$ ε-greedy

Take actions wrt. π to get experience in MDP.

Update estimates of P_{sa}. (and possibly R)

Solve Bellman's equation using value iteration to get V.

}

Update $\pi(s) := \arg\max_a \sum_{s'} P_{sa}(s') V(s')$