Lecture 1: Value Iteration

Scribe: *David Millard*

There is a recitation on smoothing on Tuesday 2018-09-04.

1 Review of MDPs

Recall that a Markov decision process (MDP) is defined as (S, A, T, R) where

- S is a discrete state space.
- A is a discrete action space.
- $T : S \times A \to \Pi(S)$ is a transition function giving $p(s'|a, s)$, the probability of transition to state s' given action a and previous state s.
- $R : S \to \mathbb{R}$ is function mapping states to numerical rewards.

Given a policy π, we can compute the value of being in a given state s for a finite time horizon

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{T} R(s_t) \middle| \pi_t\right] \tag{1}$$

or for an infinite time horizon

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \middle| \pi_t\right] \tag{2}$$

2 Example: Grid world

We consider the simple grid world given in Figure 1.

-0.04	-0.04	-0.04	$+1$
-0.04		-0.04	-1
-0.04	-0.04	-0.04	-0.04

Figure 1: Grid world

Our goal is to produce an optimal policy

$$\pi^*(s) = \underset{\pi}{\operatorname{argmax}}\, V^{\pi}(s) \tag{3}$$

Note

Why have a policy at all? Why not generate a plan and then execute it?

A policy gives you the option for closed loop control. If we end up in a state that deviates from our plan's expectations, we can still recover and act optimally.

We describe the value function for an optimal policy as follows

$$V^*(s_t) = \max_a \left[R(s_t) + \gamma \sum_{s_{t+1}} P(S_{t+1}|s_t, a_t) \underbrace{V^*(s_{t+1})}_{\substack{\text{expected} \\ \text{future} \\ \text{rewards}}} \right] \tag{4}$$

Intuitively, an optimal policy should optimize the immediate reward and the discounted expected future rewards.

Note

We have n equations and n unknowns, so can't we solve the system of equations?

No, because $\max_{a_t}$ is nonlinear, we must solve some other way.

3 Value iteration algorithm

Algorithm 1: Value iteration algorithm

Result: $\pi*$ an optimal policy (within ε)
$V_0(s) \leftarrow 0$;
repeat
  for $s \in S$ **do**
    $V_{t+1}(s) \leftarrow \max_a \left[R(s) + \sum_{s'} P(s'|s,a)V_t(s')\right]$;
  end for
until $\max_s |V_{t+1}(s) - V_t(s)| < \varepsilon$;

Applying the value iteration algorithm to our grid world example, we get

−0.04	−0.04	−0.04	+1
−0.04		−0.04	−1
−0.04	−0.04	−0.04	−0.04

(a) $t = 1$

−0.08	−0.08	−0.68	+1
−0.08		−0.08	−1
−0.08	−0.08	−0.08	−0.08

(b) $t = 2$

...

0.81	0.89	0.92	+1
0.76		0.66	−1
0.71	0.66	0.61	0.39

(c) $t = T$

Figure 2: Value iteration on grid world

Note

What if -0.04 approaches 0? The robot will be extremely conservative, and run into the wall instead of attempting to move near -1.

What if -0.04 approaches -5? The robot will go straight towards either -1 or $+1$, since either are better than staying still.

4 Example: Robot cleaning a table

Consider a robot trying to clean a table by itself, and a human who can intervene. The human can trust or mistrust the robot, which impacts their willingness to intervene. Notably, this trust can change if the human observes the robot succeed at cleaning.

We define the following states (note this is not exactly an MDP, yet)

- $S^H = \{Trust, \neg Trust\}$ defines the state of the human.

- $S^W = Bottle \times Glass = \{TT, TR, TH, RT, RR, RH, HT, HR, HH\}$ defines the state of the world. TH means the bottle is on the Table and the glass is taken by the Human.
- $A^R = \{B, G\}$ are the robot actions. The robot can either pick up the Bottle or the Glass.
- $A^H = \{Intervene, \neg Intervene\}$ are the human actions. The human can either intervene or not.

The human will intervene based on their trust in the robot and based on whether the robot attempts to grasp the bottle or the glass. The intervention probability is given by the following conditional probability table.

s_t^H	a_t^R	$P(a_t^H = Intervene \mid s_t^H, a_t^H)$
$Trust$	B	0.1
$Trust$	G	0.2
$\neg Trust$	B	0.3
$\neg Trust$	G	0.8

The world state transitions deterministically as though the robot or human always succeeds at their actions. The following table gives two examples of the deterministic transition.

s_t^W	a_t^R	a_t^H	s_{t+1}^W
TT	B	$Intervene$	HT
TT	B	$\neg Intervene$	RT
		$\vdots$	$\vdots$

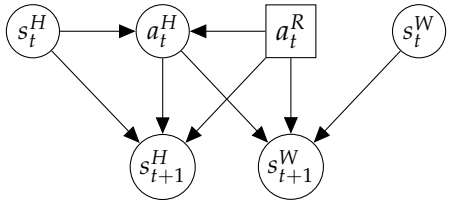
The human's trust in the robot can also change according to the following conditional probability table.

s_t^H	a_t^R	a_t^H	$P(s_{t+1}^H = Trust \mid s_t^H, a_t^H, a_t^R)$
$Trust$	B	$Intervene$	1.0
$Trust$	B	$\neg Intervene$	1.0
$\neg Trust$	B	$\neg Intervene$	0.8
$\neg Trust$	B	$\neg Intervene$	0.9

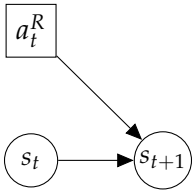
We specify a sparse reward function where the robot is rewarded only for clearing the glass or clearing both items.

$$R(RR) = 10$$
$$R(TR) = 5$$
$$R(*) = 0$$

These specifications give us the following causal model of the world



but we want something more similar to the traditional MDP graph



Recall

We do not include human actions in the graph because the robot cannot control them.

so we define our state space $S = S^H \times S^W$ ($|S| = 18$), we can derive our transition table $T : S \times A^R \to \Pi(S)$.

$$T(s_{t+1} | s_t, a_t^R) = P(s_{t+1}^W, s_{t+1}^H | a_t^R, s_t^W, s_t^H)$$

by marginalizing, we get

$$= \sum_{a_t^H} P(s_{t+1}^W, s_{t+1}^H, a_t^H | a_t^R, s_t^W, s_t^H)$$
$$= \sum_{a_t^H} P(s_{t+1}^W, s_{t+1}^H, | a_t^R, s_t^W, s_t^H, a_t^H) P(a_t^H | s_t^W, s_t^H, a_t^R)$$

since we have specified all causal ancestors, we see by conditional independence that

$$= \sum_{a_t^H} P(s_{t+1}^W | a_t^R, s_t^W, s_t^H, a_t^H) P(s_{t+1}^H | a_t^R, s_t^W, s_t^H, a_t^H) P(a_t^H | s_t^W, s_t^H, a_t^R)$$
$$= \sum_{a_t^H} P(s_{t+1}^W | a_t^R, s_t^W, a_t^H) P(s_{t+1}^H | a_t^R, s_t^H, a_t^H) P(a_t^H | s_t^H, a_t^R)$$

The size of our matrix is $|T| = |S \times A^R \to \Pi(s)| = 18 \times 2 \times 18$.

Applying value iteration to the MDP specification, we see

s^W	s^H	$V_1(s)$	$V_2(s)$	$V_3(s)$
TT	$\neg Trust$	0	1	4.76
TR	$\neg Trust$	5	12	12
TH	$\neg Trust$	0	0	0
RT	$\neg Trust$	0	2	2
RR	$\neg Trust$	10	10	10
RH	$\neg Trust$	0	0	0
HT	$\neg Trust$	0	0	0
HR	$\neg Trust$	0	0	0
HH	$\neg Trust$	0	0	0
TT	$Trust$	0	4	11.2
TR	$Trust$	5	14	14
TH	$Trust$	0	0	0
RT	$Trust$	0	8	8
RR	$Trust$	10	10	10
RH	$Trust$	0	0	0
HT	$Trust$	0	0	0
HR	$Trust$	0	0	0
HH	$Trust$	0	0	0

Notice that the robot develops and interesting policy. If the human doesn't trust the robot, the robot will pick up the bottle to increase trust before attempting to pick up the glass.

As an example, we present the computation of $V_2(TT, \neg Trust)$.

$$V_2(TT, \neg Trust) = \max_{a_1^R} \left\{ \begin{array}{l} 0 + P(RT, \neg Trust | TT, \neg Trust, B) V_1(RT, Trust) + \ldots, \\ 0 + P(TR, Trust | TT, \neg Trust, G) V_1(TR, Trust) + \\ \quad + P(TH, Trust | TT, \neg Trust, G) V_1(TH, Trust) \\ \quad + P(TH, \neg Trust | TT, \neg Trust, G) V_1(TR, \neg Trust) \\ \quad + P(TH, \neg Trust | TT, \neg Trust, G) V_1(TH, \neg Trust) \end{array} \right\}$$

So we are left with

$$\begin{aligned} P(TR, Trust | TT, \neg Trust, G) &= P(TR | TT, G, Intervene) P(Trust, | \neg Trust, G, Intervene) P(Intervene | \neg Trust, G) \\ &\quad + P(TR | TT, G, \neg Intervene) P(Trust, | \neg Trust, G, \neg Intervene) P(\neg Intervene | \neg Trus \\ &= 1 \times 0.9 \times 0.2 = 0.18 \end{aligned}$$

and

$$\begin{aligned} P(TR, \neg Trust | TT, \neg Trust, G) &= P(TR | TT, G, Intervene) P(\neg Trust, | \neg Trust, G, Intervene) P(Intervene | \neg Trust, \\ &\quad + P(TR | TT, G, \neg Intervene) P(\neg Trust, | \neg Trust, G, \neg Intervene) P(\neg Intervene | \neg T \\ &= 0.02 \end{aligned}$$

So $V_2(TT, \neg Trust) = 0.18 \times 5 + 0.02 \times 5 = 0.2 \times 5 = 1$.

Note

The robot could try to pick up the glass twice, which is impossible. How do we prevent this?

This is not modeled here, but could be handled by assigning a high negative value to these type of impossible configurations.