HW3: Markov Decision Processes

CS6300: Artificial Intelligence, Spring 2018 **Jake Pitkin**
University of Utah

---

1 Life as a Professor

1. Using value iteration we can calculate $V_t^*(state)$ for $t = 1 \dots 5$ with $\gamma = 0.5$. We will use the following rewards for each state and the recursive value iteration formula:

$$r_{asst} = 20, \quad r_{assc} = 60, \quad r_{full} = 400, \quad r_{hl} = 10, \quad r_{dead} = 0$$

$$V_s^1 = r_s, \quad V_s^t = r_s + \gamma \sum_{j=1}^{n} p_{sj} V_j^{t-1}$$

t	V_t^*(asst)	V_t^*(assc)	V_t^*(full)	V_t^*(hl)	V_t^*(dead)
0	0	0	0	0	0
1	20	60	400	10	0
2	33	119	540	13.5	0
3	43.15	151.05	589	14.725	0
4	49.5225	165.6875	606.15	15.15375	0
5	52.940875	171.836625	612.1525	15.3038125	0

Table 1: Five iterations of value iteration on the professor MDP.

2. To compute the max-norm of V between steps 4 and 5, we will consider the change for each state and take the maximum:

$$||U|| = max(V_5(asst) - V_4(asst), V_5(assc) - V_4(assc), V_5(full) - V_4(full), V_5(hl) - V_4(hl), V_5(dead) - V_4(dead))$$

$$||U|| = max(3.418375, 6.151325, 6.0025, 0.1500625, 0)$$

$$||U|| = 6.151325$$

So the final values are each at most 6.151325 away from the true values (as the Bellman update is a contraction so the max-norm will be strictly decreasing with additional iterations of value iteration).

From the text (RN pages 654-655), we know that the Bellman update is a contraction by a factor of γ on the space of utility vectors. Let B be the Bellman update, U_i be the utility vector and U_i' be the utility vector at the next step. We can say:

$$||BU_i - B_i'|| \leq \gamma ||U_i - U_i'||$$

Simply put the Bellman update when applied to U_i and U_i' will produce values that are closer together by a factor of γ. Since $\gamma = 0.5$, the next update will produce a value at most roughly 3 away from the true value (half of 6.151 the max-norm).

So if we are ok with a small error, say being within 1 of the true value, we can expect to be at most three more steps away from convergence.

2 Clarence the Evil Professor

1. Alice's policy is to always work. This means we can remove the *max* from the value iteration formula with actions added and just consider the action *work*:

$$r_{DF} = 0, \quad r_{SF} = 0, \quad r_{DP} = 10, \quad r_{SP} = 10$$

$$V_s^1 = r_s, \quad V_s^t = r_s + \gamma \sum_{j=1}^{n} p_{sj}^{work} V_j^{t-1}$$

We will consider this for $t = 0 \dots 4$ and $\gamma = 0.5$:

t	D+F	S+F	D+P	S+P
0	0	0	0	0
1	0	0	10	10
2	0	0	10	10
3	0	0	10	10
4	0	0	10	10

Table 2: Policy: always work.

This outcome makes sense. Professor Clarence doesn't value hard work and *work* only leads to *Dumb & Fail* or *Smart & Fail*. As such, the most value Alice can obtain is 10 by beginning in the state *Dumb & Pass* or *Smart & Pass*.

2. Alice now updates her policy to work when she's dumb and sweet talk when she's smart. Using the same formula as part 1 with this updated policy:

t	D+F	S+F	D+P	S+P
0	0	0	0	0
1	0	0	10	10
2	0	2.5	10	15
3	0.625	3.75	10.625	16.25
4	1.09375	4.21875	11.09375	16.71875

Table 3: Policy: work when dumb and sweet talk when smart.

This policy is better which makes sense as Alice incorporates sweet talking into her policy. Professor Clarence only passes students that sweet talk him. This allows students in any state to have a chance to move into a passing state.

3. Now we will run value iteration without a policy. Calculating the value of V for $t = 1, \ldots, 5$ and $\gamma = 0.5$ using the value iteration formula for multiple actions:

$$r_{DF} = 0, \quad r_{SF} = 0, \quad r_{DP} = 10, \quad r_{SP} = 10$$

$$V_s^1 = r_s, \quad V_s^t = \max_l \left[r_s + \gamma \sum_{j=1}^{n} p_{sj}^l V_j^{t-1} \right]$$

t	D+F	S+F	D+P	S+P
0	0	0	0	0
1	0	0	10	10
2	0	2.5	12.5	15
3	0.625	3.75	13.125	16.875
4	1.09375	4.375	13.4375	17.5

Table 4: Value iteration without an initial policy.

4. For a fixed policy π, which was to always work, values were found for each state:

$$V^{\pi_i}(D+F) = 0, \quad V^{\pi_i}(S+F) = 0, \quad V^{\pi_i}(D+P) = 10, \quad V^{\pi_i}(S+P) = 10$$

Using these values and the MDP graph, we will use policy extraction to estimate a new policy for each state:

$$\pi_{i+1}(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} T(s'|s,a) V^{\pi_i}(s') \right]$$

	D+F	S+F	D+P	S+P
sweet talk	0	2.5	12.5	15
work	0	0	10	10
$\pi_{i+1}(s)$	*work*	*sweet talk*	*sweet talk*	*sweet talk*

Table 5: Estimated policy using the values from part 1.

The new policy for each state is to *work* if you are *Dumb & Failing* and *sweet talk* otherwise (with an arbitrary choice made for the tie for D + F). We will perform value iteration on this policy with $t = 1, \ldots, 5$ and $\gamma = 0.5$:

t	D+F	S+F	D+P	S+P
1	0	0	10	10
2	0	2.5	12.5	15
3	0.625	3.75	13.125	16.875
4	1.09375	4.375	13.4375	17.5
5	1.3671875	4.6484375	13.6328125	17.734375

Table 6: Value iteration with a fixed policy of sweet talking.

Using these values, we will perform another round of policy extraction for a final policy:

	D+F	S+F	D+P	S+P
sweet talk	0	4.375	13.3203125	17.6953125
work	1.0546875	2.109375	11.0546875	12.109375
$\pi_{i+1}(s)$	*work*	*sweet talk*	*sweet talk*	*sweet talk*

Table 7: Third policy.

The policy has converged as it hasn't changed since the last step of policy extraction.

This policy seems intuitively optimal to me. When in the state D+F, the only way to leave is to *work*. In the other states, *sweet talking* has a better chance of putting you in a passing state and *working* has a better of putting you in a failing state.