CS294-40 Learning for Robotics and Control Lecture 19 - 30/10/2008

Reward Shaping

Lecturer: Pieter Abbeel *Scribe: Pål From*

1 Algorithm Review

1.1 Q-learning

The Q-learning algorithm can be summarized as

$$\begin{array}{ll} \text{initialize} & s_0 \\ \text{for} & t = 1, 2, 3 \dots \\ & \text{choose an action } a_t \text{, let's say } \epsilon\text{-greedy w.r.t. } Q \\ & \text{execute } a_t \\ & Q(s_t, a_t) = (1 - \alpha_t) Q(s_t, a_t) + \alpha_t \left[R(s_t, a_t, s_{t+1}) + \gamma \max_a Q(s_{t+1}, a) \right] \end{array} \tag{1}$$

We can for example choose $\alpha_t = \frac{1}{t+1}$. Although we have not shown it, we know this algorithm converges as long as we visit all the state with probability $P(\cdot) > 0$.

1.2 Value iteration

Value iteration is summarized as

$$\begin{array}{lll} \text{iterate} & & \\ & \forall s & \bar{V}(s) = \max_a \sum_{s'} P(s'|a, s) \left[R(s, a, s') + \gamma V(s') \right] \\ & \forall s & V = \bar{V} \end{array} \tag{2}$$

which we normally write as $V = TV$.

2 Example

We want to find the optimal policy to move from the start to the goal state.

	S				G
Reward for entering state	0	0	0	0	1
State	1	2	3	4	5

(3)

$$\begin{array}{lll} \text{S} & - & \text{Start} \\ \text{G} & - & \text{Goal} \end{array} \tag{4}$$

We have two actions

$$\begin{array}{lll} \hat{L} & - & \text{move to the state on the left} \\ \hat{R} & - & \text{move to the state on the right} \end{array} \tag{5}$$

2.1 Q-learning

We apply Q-learning to the problem: We start by initializing $s_0 = 1$ and $a = \hat{R}$

$$Q(1,\hat{R}) = (1-\alpha)\underbrace{Q(1,\hat{R})}_{0} + \alpha\left[\underbrace{R(1,\hat{R},2)}_{0} + \gamma \max_a \underbrace{Q(2,a)}_{0}\right] = 0 \tag{6}$$

we how have $s_1 = 2$ and choose $a = \hat{L}$

$$Q(2,\hat{L}) = (1-\alpha)\underbrace{Q(2,\hat{L})}_{0} + \alpha\left[\underbrace{R(2,\hat{L},1)}_{0} + \gamma \max_a \underbrace{Q(1,a)}_{0}\right] = 0 \tag{7}$$

We see that until we coincidentally happen to hit state 5 Q will always be zero. Very little information is added to the Q-learning algorithm before this happens as there is nothing guiding us towards the right state.

We now propose the alternative reward function

	S				G
Reward for entering state	0	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	1
State	1	2	3	4	5

(8)

We now get

$$Q(1,\hat{R}) = \alpha \cdot \gamma \cdot \frac{1}{4} \tag{9}$$

$$Q(2,\hat{L}) = 0 \tag{10}$$

Then at time 3 we are in state 1 and have

$$Q(1,\hat{R}) = \alpha \cdot \gamma \cdot \frac{1}{4}, \qquad Q(1,\hat{L}) = 0 \tag{11}$$

We see that we can shape the reward to guide us in the right direction.

2.2 Value Iteration

$$\begin{array}{lccc} \text{State} & 0 & 1 & 2 & \dots \\ \text{Value Function} & \begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} & \begin{bmatrix}0\\0\\\gamma\cdot 1\\\gamma\cdot 1+1\\\gamma\cdot 1+1\end{bmatrix} & \dots \end{array} \tag{12}$$

We do this propagating from back to front. After four iterations we have found the optimal policy, although we have not yet converged to the value function.

3 Reward shaping without changing the optimal policy

So far the approach has been very heuristic. We will now look into how we can shape the reward function without changing the relative optimality of policies.

We start by looking at a bad example: let's say we want an agent to reach a goal state for which it has to climb over three mountains to get there. The original reward function has a zero reward everywhere, and a positive reward at the goal state (which is beyond the three mountains). We could imagine changing the

reward by adding a positive reward for making progress towards the goal by adding a positive reward when the agent reaches the top of each mountain. Indeed, this would favor some policies that move the agent towards the goal. However, it would also favor the following policy: climb to the top of the first mountain, take a few steps back, go back to the top, and keep repeating ... In fact, depending on the discounting and the exact reward function, the latter could even be the optimal policy.

Intuitively, the reward shaping in the example fails because the agent gets rewarded every time they reach the top of the mountain, independent of whether they already reached the top before. A natural solution is to, indeed reward the agent for reaching the top, but also *penalize* the agent for moving away from the goal/top such that cyclic behaviour results in a zero reward. This suggests shaping the reward function by using a potential function ϕ:

$$\bar{R}(s, a, s') = R(s, a, s') + F(s, a, s') \tag{13}$$

where

$$F(s, a, s') = \phi(s') - \phi(s). \tag{14}$$

For now, we assume:

1. $\exists$ "absorbing" state s_F
2. No discounting

The first assumption means that for all policies, we will end up in state $s_t = s_F$ with probability $P = 1$, i.e.

$$\forall \pi \quad \text{eventually } s_t = s_F. \tag{15}$$

We can remove the restriction on the discounting at a later stage.

Now consider the rewards accumulated during one episode:

$$\sum_{t=0}^{\tau-1} \left[R(s_t, a_t, s_{t+1}) + \phi(s_{t+1}) - \phi(s_t)\right] = \left[\sum_{t=0}^{\tau-1} R(s_t, a_t, s_{t+1})\right] + \phi(s_F) - \phi(s_0) \tag{16}$$

Let $\bar{M}$ denote the MDP identical to the original MDP M, except for the reward function: in $\bar{M}$ we have the shaped reward function. Then the above can be written as:

$$\forall \pi : V_{\bar{M}}^{\pi}(s) = V_M^{\pi}(s) + \phi(s_\tau) - \phi(s_0) \tag{17}$$

This implies that for all policies π_1, π_2 we have that the ordering of their value is preserved between M and $\bar{M}$:

$$V_M^{\pi_1}(s) > V_M^{\pi_2}(s) \Longrightarrow V_{\bar{M}}^{\pi_1}(s) > V_{\bar{M}}^{\pi_2}(s) \tag{18}$$

Hence reward shaping based upon differencing a potential function has the desired property of keeping the optimality ordering of policies invariant.

3.1 Infinite horizon

We re-write (16) with the discount factor

$$\sum_{t=0}^{\tau-1} \gamma^t \left[R(s_t, a_t, s_{t+1}) + \phi(s_{t+1}) - \phi(s_t)\right] \tag{19}$$

We write out

$$\sum_{t=0}^{\tau-1} \gamma^t \left[\phi(s_{t+1}) - \phi(s_t)\right] = (\phi(s_1) - \phi(s_0)) + (\gamma\phi(s_2) - \gamma\phi(s_1)) + (\gamma^2\phi(s_3) - \gamma^2\phi(s_2)) \dots \tag{20}$$

We see that if we multiply the first part with γ, we get

$$\sum_{t=0}^{\tau-1} \gamma^t \left[\gamma\phi(s_{t+1}) - \phi(s_t)\right] = \gamma\phi(s_1) - \phi(s_0) + \gamma^2\phi(s_2) - \gamma\phi(s_1) + \gamma^3\phi(s_3) - \gamma^2\phi(s_2)\dots$$
$$= -\phi(s_0) + \underbrace{(\gamma\phi(s_1) - \gamma\phi(s_1))}_{0} + \underbrace{(\gamma^2\phi(s_2) - \gamma^2\phi(s_2))}_{0} + \gamma^{\tau-1}\phi(s_\tau)\dots \tag{21}$$

So we choose

$$F(s, a, s') = \gamma\phi(s') - \phi(s). \tag{22}$$

Proposition 1 *Reward shaping with the function $F(s, a, s') = \gamma\phi(s') - \phi(s)$ leaves the optimality ordering of policies invariant.*

Proof *Let M be the original MDP, and let $\bar{M}$ be identical except for the reward function being shaped by $F(s, a, s') = \gamma\phi(s') - \phi(s)$.*

$$Q^*_M(s, a) = \sum_{s'} P(s' \mid s, a)\left[R(s, a, s') + \gamma \max_a Q^*_M(s, a)\right] \tag{23}$$

we add the shaping (by adding and subtracting the same terms)

$$\underbrace{Q^*_M(s, a) - \phi(s)}_{Q^*_{\bar{M}}(s,a)} = \sum_{s'} P(s' \mid s, a)\left[\underbrace{R(s, a, s') + \gamma\phi(s') - \phi(s)}_{Shaped\ Reward\ Function\ R_{\bar{M}}(s,a,s')} + \gamma \max_a \underbrace{(Q^*_M(s, a) - \phi(s'))}_{Q^*_{\bar{M}}(s',a)}\right] \tag{24}$$

For $\bar{M}$ we get

$$Q^*_{\bar{M}}(s, a) = \sum_{s'} P(s' \mid s, a)\left[R_{\bar{M}}(s, a, s') + \gamma \max_a Q^*_{\bar{M}}(s', a)\right] \tag{25}$$

We note that $Q^*_{\bar{M}}(s, a) = Q^*_M(s, a) - \phi(s)$ satisfies the Bellman equation for the reward $R_{\bar{M}}(s, a, s') = R(s, a, s') + \gamma\phi(s') - \phi(s)$. Thus we have

$$Q^*_{\bar{M}}(s, a) = Q^*_M(s, a) - \phi(s)$$
$$\arg\max_a Q^*_{\bar{M}}(s, a) = \arg\max_a Q^*_M(s, a) \tag{26}$$

as we wanted and the optimal policy is preserved.

Now consider a special MDP, in which only 1 action is available in each state, namely the action prescribed by a policy π. Then the same reasoning as above goes through, and we obtain:

$$Q^\pi_{\bar{M}}(s, a) = Q^\pi_M(s, a) - \phi(s). \tag{27}$$

This holds true for any policy π, hence for any pair of policies π_1, π_2 we have:

$$Q^{\pi_1}_M(s, \pi_1(s)) \geq Q^{\pi_2}_M(s, \pi_2(s)) \tag{28}$$

implies:

$$Q^{\pi_1}_M(s, \pi_1(s)) - \phi(s) \geq Q^{\pi_2}_M(s, \pi_2(s)) - \phi(s) \tag{29}$$

hence:

$$Q^{\pi_1}_{\bar{M}}(s, \pi_1(s)) \geq Q^{\pi_2}_{\bar{M}}(s, \pi_2(s)). \tag{30}$$

.667em

4 What is ideal shaping?

Consider value iteration, where we iteratively compute the value function as follows:

for $k = 0, 1, \ldots$

$\forall s \qquad V_{k+1}(s) = \max_a \sum_{s'} P(s'|a, s) \left[R(s, a, s') + \gamma V_k(s')\right]$

With reward shaping we have:

$$V_{k+1}(s) = \max_a \sum_{s'} P(s'|a, s) \left[R(s, a, s') + \gamma\phi(s') - \phi(s) + \gamma V_k(s')\right]$$

Assume we choose $V_0 = 0$, and $\phi = V^*$, then we obtain:

$$V_1(s) = \max_a \sum_{s'} P(s'|a, s) \left[R(s, a, s') + \gamma V^*(s') - V^*(s) + 0\right] = V^*(s) - V^*(s) = 0$$

And similarly, for all $k > 0$, we have $V_k(s) = 0$ for all states s. Note we also have:

$$\arg\max_a \sum_{s'} P(s'|a, s) \left[R(s, a, s') + \gamma V^*(s') - V^*(s) + V_k(s')\right] = \arg\max_a \sum_{s'} P(s'|a, s) \left[R(s, a, s') + \gamma V^*(s')\right] = \pi^*(s)$$

Hence, from the first iteration onwards, we have converged to the optimal value function and optimal policy.

This also suggests there is a close connection between initializing the value function and potential shaping.