6.7950 Fall 2022: - Recitation 4 Handout

1 Potential-based reward shaping

In the previous recitation, we looked at a way modify the reward function of an MDP so it will fall between the range $[0, 1]$ without actually modifying the optimal policy how different states compare to each relatively. Now, let's look at another useful class of transformations

Using the notation we have been using so far, consider an infinite-horizon MDP with reward $r(s, a, s')$. We want to investigate another MDP with reward $\overline{r}(s, a, s') = r(s, a, s') + \gamma\phi(s') - \phi(s)$ for some function ϕ, called the potential function. Note that this is a specific case of the more general reward shaping framework $\overline{r}(s, a, s') = r(s, a, s') + F(s, a, s')$, which might be covered in future lectures. For this case, F is called a potential-based shaping funcion.

Answer the following question

1. Prove that the new MDP has a value function $\overline{V}^{\pi}(s) = V^{\pi}(s) - \phi(s)$ under a policy π
2. Prove that the new MDP has the same optimal policy π^*
3. What happens with the optimal policy when we consider an MDP with a reward only containing the potential function, that is $\tilde{r}(s, a, s') = \gamma\phi(s') - \phi(s)$
4. (Discussion) When would a reward shaping approach like that be desirable or not? What would be a good value for $\phi(s)$?

Solution:

1. Consider the expression for the value function $\overline{V}^{\pi}$ as

$$\overline{V}^{\pi}(s) = \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}\left[\overline{r}(s, \pi(s), s') + \gamma\overline{V}^{\pi}(s')\right] \tag{1}$$

$$\overline{V}^{\pi}(s) = \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}\left[r(s, \pi(s), s') + \gamma\phi(s') - \phi(s) + \gamma\overline{V}^{\pi}(s')\right] \tag{2}$$

$$\overline{V}^{\pi}(s) + \phi(s) = \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}\left[r(s, \pi(s), s') + \gamma\left(\overline{V}^{\pi}(s') + \phi(s)\right)\right] \tag{3}$$

   If we define a value function $\tilde{V}^{\pi}(s) = \overline{V}^{\pi}(s) + \phi(s)$, we have that

$$\tilde{V}^{\pi}(s) = \mathbb{E}_{s' \sim p(\cdot|s,\pi(s))}\left[r(s, \pi(s), s') + \gamma\left(\tilde{V}^{\pi}(s')\right)\right] \tag{4}$$

   which are also the same equations for the value function V^{π}, so $\tilde{V}^{\pi}(s) = V^{\pi}(s) = \overline{V}^{\pi}(s) + \phi(s)$ and so $\overline{V}^{\pi}(s) = V^{\pi}(s) - \phi(s)$, as desired.

2. If $\overline{\pi}^*$ is the optimal policy for $\overline{V}$ and π is a generic policy, then

$$V^{\overline{\pi}^*}(s) \geq \overline{V}^{\pi}(s) \tag{5}$$

$$\therefore V^{\overline{\pi}^*} = \overline{V}^{\overline{\pi}^*}(s) + \phi(s) \geq \overline{V}^{\pi}(s) + \phi(s) = V^{\pi} \tag{6}$$

$$\therefore V^{\overline{\pi}^*} \geq V^{\pi} \tag{7}$$

   Thus the optimal policy for the value $\overline{V}$ of the modified MDP is also optimal for the original one as it maximizes the value function. Proving that an optimal policy for the original MDP is also optimal for the modified one is analogous.

3. In this case, we have that the MDP with potential-based shaping has the same optimal policies with the MDP with $r(s, a, s') = 0$. Thus all policies are optimal in this case. This can provide an intuitive understanding as to why this shaping function does not change the optimal policies: it does not give preference to any particular action by assigning more or less weight to one action over the other. Another way to visualize the effects of this strategy is by considering that, even though an additional term $\gamma\phi(s')$ is added to the reward when moving to s', it's immediately removed at the next step by the term $-\phi(s')$ after being discounted by the factor γ.

4. When we are dealing with a hard RL or DP problem, we may try to apply this or other reward shaping approaches to make the optimization easier. For instance, consider a RL agent trying to move from point A to point B. If we set the rewards such that they are zero everywhere except when in proximity to point B, in which case they are equal to 1, then the optimal policy would follow the desired behavior. However, the agent would have difficulties learning this policy as most actions and trajectories would have zero reward and only eventually would it stumble upon the goal point and receive a positive reward. If we instead set a potential function that is proportional to the distance of the agent to point, then any action that moves it away from the B will have smaller reward than an action that moves it closer to B, thus guiding the training to the desired policy. Conversely, if the potentials ϕ were poorly chosen, they could also hinder training.

   Note that this also applies to DP and not just RL. In problem 3 of HW2, for example, you are asked to prove that both policy iteration and value iteration converge slowly for the given MDP and $V_0 = [0, \cdots, 0]$. If one were to suitably chose ϕ in order for $\overline{V}$ to be closer to V_0, then the problem would be simpler to solve. An observant reader might also notice that, for the cases we considered, we could instead pick $V_0 = \phi$ and have a better initialization for the value function and they would be right. In other cases, for example when approximating V, it might be more feasible to shape the rewards instead of modifying the initialization (see Potential-Based Shaping and Q-Value Initialization are Equivalent for proofs and more discussions).

   Thus, a good strategy is to try to incorporate some sort of domain knowledge about the problem into ϕ. Particularly, having $\phi \approx V^*$ can be a good choice if it can simplify the structure of $\overline{V}$ by, for example, making it less sparse. As in the example of the agent navigating to a goal, we don't need to know the exact value function, but only a reasonable guess that states closer to the goal point are more valuable.

   This particular reward shaping strategy is particularly interesting because it's the most general one that can be done without altering the underlying MDP and without having more assumptions

about the problem. A description of this result, as well as a discussion of other interesting examples where reward shaping can produce undesired effects can be found in Policy invariance under reward transformations: Theory and application to reward shaping.

2 Solving infinite-horizon MDPs with Linear Programming

Besides Value Iteration and Policy Iteration, Linear Programming (LP) is an alternative way of solving infinite horizon DPs. Recall that a linear program is an optimization problem with linear costs and linear (equality or inequality) constraints. Though most of the benefits this formulation appear when doing approximate dynamic programming, it's an interesting result nonetheless. Assuming the number of states and actions is finite, state how to use LP to compute the optimal value function V^*.

Solution: We'll start with our formulation with a set of positive weights c_i such that we want to solve the following problem

$$\begin{aligned} \min \ & \sum_i c_i |V(s_i) - V^*(s_i)| \\ \text{s.t. } & \mathcal{T}V \leq V \end{aligned}$$

Clearly, V^* both minimizes the problem and satisfies the constraints. Furthermore, by repeatedly applying the constraints and monotonicity of the Bellman operator, we have that

$$V^* \leq \cdots \leq \mathcal{T}^2 V \leq \mathcal{T}V \leq V$$

$$\therefore \sum_i c_i |V(s_i) - V^*(s_i)| = \sum_i c_i (V(s_i) - V^*(s_i)) = c^\top V - c^\top V^*$$

Thus, the presented problem has the same optimal as

$$\begin{aligned} \min \ & c^\top V \\ \text{s.t. } & \mathcal{T}V \leq V \end{aligned}$$

This is not yet a LP because $\mathcal{T}$ is nonlinear (it uses the "argmax" operator), so the constraints aren't linear. But we can instead expand the system of inequalities for the optimal Bellman operator to include inequalities for all possible states in the state space $\mathcal{S}$ and actions in the action space $\mathcal{A}$.

$$V(s) \geq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) V(s'), \ \forall s \in \mathcal{S}, a \in \mathcal{A}$$

The problem remains feasible with these additional constraints as no other action can increase V^* further, so V^* is still the unique minimizer.

Therefore, the LP formulation becomes

$$\begin{aligned} \min \ & c^\top V \\ \text{s.t. } & V(s) \geq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) V(s'), \ \forall s \in \mathcal{S}, a \in \mathcal{A} \end{aligned}$$

To learn more about this topic, a good reading option is The Linear Programming Approach to Approximate Dynamic Programming.