CSE6243: Advanced Machine Learning **Fall 2024**

Lecture 21: DP: Value and Policy Iteration

Lecturer: Bo Dai *Scribes: Mohammad Taher, Huizhong Xue*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

21.1 Recap

In reinforcement learning, the agent actively interacts with the environment to gather data and learn which actions to take. This process is called exploration, where the model discovers new states or actions instead of observing static data. The model takes a sequential set of actions to plan for the maximum future rewards.

Reinforcement learning problems can be mathematically modeled by as Markov Decision Process:

$$M =< S, A, R, P, H/\gamma >$$

where S is the state space, A is the action space, R is the rewards, P is the transition probabilities across states, and H is the horizon with discount factor γ. An agent can take some action a from state s, and with some probability p, go to a new state s' and gain reward r. It'll keep exploring till the horizon H with discount factor γ which ensures that the model weights actions in the future exponentially less than actions in the near term.

We use the value function $V(s)$ to keep track of the value at some state s. Similarly, the Q-function $Q(s, a)$ keeps track of the value of taking some action a at state s. To compute the functions, we use the Bellman equations. For example, here's the Bellman equation for the value function:

$$V^\pi(s) = \sum_a R(s,a)\pi(a|s) + \gamma \mathbb{E}_{p^\pi}[V^\pi(s')] \tag{21.1}$$

In this case, π represents the policy, which can be thought of as the decision-making process of the agent. For example, $\pi(a|s)$ dictates whether we choose some action a at state s. Similarly, we can use the Bellman optimality equation to get the optimal values V^*, Q^*.

21.2 New Content

21.2.1 Policy Evaluation

In Policy Evaluation, we use the policy π to calculate the value function V^π and objective function $J(\pi)$:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]$$

To create the best RL agent, we want to maximize our objective:

$$\pi^* = argmax_\pi J(\pi)$$

21.2.2 Computing the value function

Now let's see how to define the value function V^π using a finite, tabular setting and linear algebra. First, we can define S as a vector with n states and A as a vector with m states. Now R will take a state s and action a to output some real number, which gives us a vector of size $n * m$. Lastly, we need to define P, which takes a state s and action a to output a distribution of probabilities for S, which yields a matrix of size $(n * m) \times n$.

We can rewrite equation 21.1 as a linear equation by redefining:

$$R^\pi = \sum_a R(s,a)\pi(a|s)$$

and:

$$P^\pi \cdot V^\pi = \mathbb{E}_{p^\pi}[V^\pi(s')] = \sum_s p^\pi(s'|s)V^\pi(s') = \sum_s \sum_a \pi(a|s)P(s'|s,a)V^\pi(s')$$

to get:

$$\begin{aligned} V^\pi &= R^\pi + \gamma P^\pi \cdot V^\pi \\ \Rightarrow R^\pi &= V^\pi - \gamma P^\pi V^\pi = V^\pi(I - \gamma P^\pi) \\ \Rightarrow V^\pi &= (I - \gamma P^\pi)^{-1} R^\pi \end{aligned}$$

We've now successfully defined the value function as a linear function and solved for it. To show the intuition behind this formula, let's use the Taylor Expansion ($(1-x)^{-1}$ @ $x < 1 \Rightarrow 1 + x + x^2 + ...$):

$$(I - \gamma P^\pi)^{-1} = I + \gamma P^\pi + (\gamma P^\pi)^2 + (\gamma P^\pi)^3 + ...$$

adding R^π back in yields:

$$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi = R^\pi + \gamma P^\pi R^\pi + (\gamma P^\pi)^2 R^\pi + (\gamma P^\pi)^3 R^\pi + ...$$

which is the original formula for V^π!!

21.2.3 Policy Optimization

To get the optimal policy π^*, we solve for the following equation

$$\pi^*(a|x) = \arg\max_{\pi} \mathbb{E}_{\pi}\left[R(s,a) + \gamma \mathbb{E}_{p^{\pi}}\left[V^*\right]\right]$$

where V^* above is defined as follows

$$V^*(s') = \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s')$$

Then we can define the function

$$\Phi(V) := \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s')$$

so that $V^* = \Phi(V^*)$.

21.2.3.1 Value Iteration

With function Φ defined as above, the value iteration algorithm can be written as follows

```
Init $V_0$
For $t = 1, \ldots$
  $V_{t+1} = \Phi(V_t)$
  if $\|V_{t+1} - V_t\| \leq \epsilon$ :
    Break
```

We now want to show that the value iteration algorithm converges to the true solution V^*. First, We show that value iteration converges by proving that Φ is a contraction, i.e., we want to show that $\|\Phi(V) - \Phi(U)\|_{\infty} \leq \gamma \|V - U\|_{\infty}$ where $\gamma \in (0,1)$ is the discount factor. By definition of Φ, we have

$$\begin{aligned}\Phi(V) - \Phi(U) &\leq \Phi(V) - \left(R(S,a_V) + \gamma \sum_{s'} P(s'|s,a_V) U(s')\right) \\ &= \left(R(S,a_V) + \gamma \sum_{s'} P(s'|s,a_V) U(s')\right) - \left\|\gamma \sum_{s'} P(s'|s,a_V)(V(s') - U(s'))\right\|_{\infty} \\ &\leq \gamma \|V - U\|_{\infty}\end{aligned}$$

Therefore, Φ is a contraction. Letting $V_0 = 0$, we have

$$\begin{aligned}\|V_{n+1} - V_n\|_\infty &= \|\Phi(V_n) - \Phi(V_{n-1})\|_\infty \\ &\le \gamma \|V_n - V_{n-1}\|_\infty \\ &\le \gamma^n \|V_1 - V_0\|_\infty \\ &= \gamma^n \|V_1\|_\infty \\ &\le \gamma^n R_{\max}\end{aligned}$$

where the last inequality came from the general inequality $\left(1 - \frac{1}{x}\right)^x \le \frac{1}{e}$. Letting $1 - \gamma = \frac{1}{x}$, we then have

$$\begin{aligned}\gamma^n R_{\max} &= \left[(1-(1-\gamma))^{\frac{1}{1-\gamma}}\right]^{n(1-\gamma)} R_{\max} \\ &\le \left(\frac{1}{e}\right)^{n(1-\gamma)} R_{\max} \\ &= \epsilon\end{aligned}$$

The value iteration algorithm converges in $O\left(\log \frac{1}{\epsilon}\right)$ time steps. Now we show that value iteration converges to the true solution V^*. Again using the definition of Φ and V^*, and the fact that value iteration converges, we have

$$\begin{aligned}\|V^* - V_{n+1}\|_\infty &= \|V^* - \Phi(V_{n+1}) + \Phi(V_{n+1}) - V_{n+1}\|_\infty \\ &\le \|\Phi(V_*) - \Phi(V_{n+1})\|_\infty + \|\Phi(V_{n+1}) - V_{n+1}\|_\infty \\ &\le \gamma \|V^* - V_{n+1}\|_\infty + \gamma^n \|V_1\|_\infty \\ &\le \frac{\gamma^n}{1-\gamma} \|V_1\|_\infty\end{aligned}$$

Therefore, value iteration converges to the true solution V^*.

21.2.3.2 Policy Iteration

The convergence result for policy iteration algorithm is similar to that of value iteration. The algorithm can be expressed as follows

```
Init $\pi_0$
For $t = 1, \dots$
  $V^{\pi_t} = R^{\pi_t} + \gamma P^{\pi_t} V^{\pi_t}$
  $\pi_{t+1} = \arg\max_\pi \mathbb{E}_\pi \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, \pi(a,s)\, V^{\pi_t}(s')\right]$
```

where the first step in the for loop is called policy evaluation and runs in $O\left(|S|^3\right)$. The second step in the for loop is called policy update.