CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 21: DP Value and Policy Iteration

Lecturer: Bo Dai *Scribes: Ignat Georgiev, Chloe Saleh*

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

21.1 Recap

1. **Reinforced Learning VS Supervised Learning:** Reinforcement learning differs from supervised learning in a way that in supervised learning a mapping from X to Y is learned and the distribution remains fixed whereas in reinforcement learning, a policy is learned and the distribution changes according to your learning.
2. **Bellman Optimal Equation for Q^* and V^*:**
   (a) $Q^*(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[V^*, s']$:
   (b) $V^*(s) = \max_a \left(R(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[V^*, s']\right)$
3. **Policy Evaluation VS Policy Optimization:** In policy evaluation, we are given a policy π and we need to estimate $V(\pi)$ or $Q(\pi)$ whereas in policy optimisation, we are not given a policy π and we need to estimate an optimal π^*
4. **Planning VS Learning:** In planning, we have access to the transition probability and the reward function whereas in learning we don't, in short learning is going from experience to a policy, whereas planning is going from a model to a policy.

21.2 New Content

We have previously seen two algorithms designed to handle planning in the context of policy optimisation. We have assumed that the environment model is known. That is, the transition probability $p(a|s,a)$ and the expected reward $\mathbb{E}[r(s,a)]$ for all s, s' $\in$ S and a $\in$ A are assumed to be given.

21.2.1 Value Iteration

In value iteration, we compute the optimal state value function by iteratively updating the estimate $V(s)$: the new values of V(s) are determined using the Bellman equations and the current values. This process is repeated until a convergence condition is met.

We start with a random value of function V(s). At each step we update it:

$$V(s) = \max_a \sum_{s',r'} p(s', r|s, a)[r + \gamma V(s')]$$

Value Iteration converges to the optimal value function $V^*(s)$ as $k \to \infty$.

Theorem 21.1. The Value Iteration algorithm ran for k iterations with discount rate γ converges at a rate:

$$\|V^* - V_{k+1}\|_\infty \leq \frac{\gamma^k}{1-\gamma} R_{max} \tag{21.1}$$

where $R(s,a) \leq R_{max} \quad \forall s, \forall a$.

Proof. To prove this property, we need the following proposition:

Proposition 21.2. $\phi(\cdot)$ is γ - contractive; a type of function or operator that contracts distances between points in a specific way. $\phi()$ is said to be γ-contrastive if $\forall x, y \in dom(\phi)$, the following holds:

$$d(\phi(x), \phi(y)) \leq \gamma d(x, y)$$

By definition: $\pi_V(a|s) = \arg\max_\pi \Phi(V)$. Using the property $|\max_a f(a) - \max_a g(a)| \leq \max_a |f(a) - g(a)|$ *(Proof in Mohri et al, 2018)*, we can show:

$$\begin{aligned}
\Phi(V) - \Phi(U) &\leq \Phi(V) - \sum_a \pi_V(a|s)(R(s,a) + \gamma PU) \\
&= \langle \pi_V(a|s), R(s,a) + \gamma PV \rangle - \langle \pi_V(a|s), R(s,a) + \gamma PU \rangle \\
&= \langle \pi_V(a|s), \gamma P(V - U) \rangle \\
&= \sum_{s'} \sum_a \gamma \pi_V(a|s) p(s'|s,a)(V(s') - U(s')) \\
&\leq \gamma \|V - U\|_\infty
\end{aligned}$$

where in the last step we use the Cauchy–Schwarz inequality. Next we want to quantify this results

$$\begin{aligned}
\|V^* - V_{k+1}\|_\infty &\leq \|V^* - \Phi(V_{k+1})\|_\infty + \|\phi(V_{k+1}) - V_{k+1}\|_\infty && \text{(triangle inequality)} \\
&= \|\Phi(V^*) - \Phi(V_{k+1})\|_\infty + \|\phi(V_{k+1}) - \Phi(V_{k+1})\|_\infty \\
&\vdots \\
&\leq \frac{\gamma^k}{1-\gamma} \|V_1 - V_0\|_\infty && \text{(telescoping sum)} \\
&= \frac{\gamma^k}{1-\gamma} R_{max}
\end{aligned}$$

□

21.2.2 Policy Iteration

In policy iteration, we start by choosing an arbitrary policy π. Then, we iteratively evaluate and improve the policy until convergence:

We evaluate a policy $\pi(s)$ by calculating the state value function V(s):

$$V(s) = \sum_{s', r'} p(s', r|s, \pi(s))[r + \gamma V(s')]$$

Then, we calculate the improved policy by using one-step look-ahead to replace the initial policy $\pi(s)$:

$$\pi(s) = arg\max_a \sum_{s',r'} p(s', r|s, a)[r + \gamma V(s')]$$

Theorem 21.3. Denote $U_k{}_{k=1}^K$ and $V_k{}_{k=1}^K$ as the value estimates of Value Iteration and Policy Iteration respectively. If $U_0 = V_0$ then

$$U_k \leq V_k \quad \forall k \in 0, 1, .., K \tag{21.2}$$

In other words, Policy Iteration converges at at least the same rate as Value iteration (but can be better!). In order to prove this, we first need:

Proposition 21.4. $V^{\pi_k} \to V^*$ as $k \to \infty$

Proof. Using the fact that if $V' \geq V$, then $\Phi(V') \geq \Phi(V)$:

$$\begin{aligned} V^{\pi_k} &\leq \Phi^{\pi_{k+1}}(V^{\pi_k}) && (21.3)\\ &\leq (\Phi^{\pi_{k+1}})^2(V^{\pi_k}) && (21.4)\\ &\vdots && (21.5)\\ \lim_{n\to\infty} &\leq (\Phi^{\pi_{k+1}})^n(V^{\pi_k}) && (21.6)\\ &= V^{\pi_{k+1}} && (21.7) \end{aligned}$$

□

Where the last line comes from the fact that $\gamma^n \to 0$ as $n \to \infty$. See proof of Lemma 4.1 in Mathematical Foundations of Reinforcement Learning

Now we can return to prove Theorem 21.3:

Proof. First we need to show that Φ is monotonic. Let U and V be such that $U \leq V$ and let π be such that $\phi(U) = R(s, a) + \gamma PU$, then

$$|\Phi(U) \leq R_\pi + \gamma PV \leq \max_{\pi'} (R(s, a) + \gamma PV) = \Phi(V)$$

Since Φ is monotonic, assuming $U_k \leq V_k$, gives us

$$U_{n+1} = \Phi(U_k) \leq \Phi(V_k) = \max_{\pi} (R(s, a) + \gamma PV)$$

If π_{k+1} is the maximising policy, then

$$\Phi(V_k) = R(s_{k+1}, a_{k+1}) + \gamma PV_k \leq R(s_{k+1}, a_{k+1}) + \gamma PV = V_{k+1}$$

Therefore, $U_{k+1} \leq V_{k+1}$ which by induction holds for $k \in 0, 1, .., K$. □