CS294-40 Learning for Robotics and Control Lecture 3 - 9/4/2008

Contractions, Asychronous Value Iteration

Lecturer: Pieter Abbeel *Scribe: Zhang Yan*

1 Lecture outline

- Review.
- Contractions.
- Asynchronous value iteration.

2 Review

We assume finite state space and finite action space.

2.1 Value of a policy

$$V_\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0 = s, \pi]$$

2.2 Value function

$$V^*(s) = \max_\pi \quad V_\pi(s)$$

2.3 Bellman/Dynamic programming operator

$$(TV)(s) = \max_{a \in A} [R(s) + \gamma \sum_{s'} P(s'|s,a) V(s')]$$

2.4 Theorem

$$\lim_{H \to \infty} \quad (T^H V) = V^*$$

In this lecture (amongst others) we will show that there is a *stationary* optimal policy $\pi^* = (\mu^*, \mu^*, \ldots)$, which achieves $V_{\pi^*} = V^*$, and it satisfies:

$$\mu^*(s) \in \arg\max_{a \in A} [R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s')]$$

2.5 Bellman/Dynamic programming operator for a fixed policy

To compute the value of a specific stationary policy $\pi = (\mu, \mu, \ldots)$, we can use the operator T_μ:

$$(T_\mu V)(s) = [R(s) + \gamma \sum_{s'} P(s'|s, \mu(s))V(s')].$$

Properties of the operator T can be directly translated in properties of the operator T_μ by realizing T_μ is the "T operator" for a special MDP, where there is only a single action $\mu(s)$ available from each state s.

For example, we have:

Theorem

$$\lim_{H\to\infty} (T_\mu^H V) = V_\pi$$

2.6 Theorem (Contractions)

T is a maximum norm γ-contraction, i.e.,

$$\|TV - T\overline{V}\|_\infty \leq \gamma \|V - \overline{V}\|_\infty$$

3 Contractions

3.1 Theorems

Let F be a α-contraction w.r.t. some norm $\|\cdot\|$, i.e.,

$$\|FX - F\overline{X}\|_\infty \leq \alpha \|X - \overline{X}\|_\infty$$

Theorem 1. *The sequence $X, FX, F^2X, \ldots$ converges for every X.*

Cauchy sequences: If for $x_0, x_1, x_2, \ldots$, we have that

$$\forall \epsilon, \exists K : \|x_M - x_N\| < \epsilon \quad for \quad M, N > K$$

then we call $x_0, x_1, x_2, \ldots$ a Cauchy sequence.

Property of Cauchy sequences: If $x_0, x_1, x_2, \ldots$ is a Cauchy sequence, and $x_i \in \Re^n$, then there exists $x^* \in \Re^n$ such that $\lim_{i\to\infty} x_i = x^*$.

Proof. Assume $N > M$.

$$\begin{aligned}
\left\|F^M X - F^N X\right\| &= \left\| \sum_{i=M}^{N-1} (F^i X - F^{i+1} X) \right\| \\
&\leq \sum_{i=M}^{N-1} \left\|F^i X - F^{i+1} X\right\| \\
&\leq \sum_{i=M}^{N-1} \alpha^i \left\|X - FX\right\| \\
&= \|X - FX\| \sum_{i=M}^{N-1} \alpha^i \\
&= \|X - FX\| \frac{\alpha^M}{1-\alpha}.
\end{aligned}$$

As $\|X - FX\| \frac{\alpha^M}{1-\alpha}$ goes to zero for M going to infnity, we have that for any $\epsilon > 0$ for $\left\|F^M X - F^N X\right\| \le \epsilon$ to hold for all $M, N > K$, it suffices to pick K large enough.

Hence $X, FX, \ldots$ is a Cauchy sequence and converges.

Theorem 2. *F has a unique fixed point.*

Proof. Suppose F has two fixed points. Let's say

$$FX_1 = X_1,$$
$$FX_2 = X_2,$$

this implies,

$$\|FX_1 - FX_2\| = \|X_1 - X_2\|.$$

At the same time we have from the contractive property of F

$$\|FX_1 - FX_2\| \le \alpha \|X_1 - X_2\|.$$

Combining both gives us

$$\|X_1 - X_2\| \le \alpha \|X_1 - X_2\|.$$

Hence,

$$X_1 = X_2.$$

Therefore, the fixed point of F is unique. □

Theorem 3. *A policy $\pi = (\mu, \mu, \mu, ...)$ is an optimal policy if and only if $TV^* = T_\mu V^*$.*

Proof. First suppose,

$$\begin{aligned} & TV^* = T_\mu V^* \\ \Rightarrow\ & T_\mu V^* = V^* \quad (\text{as } V^* = TV^*) \\ \Rightarrow\ & V^* \text{ is the fixed point for } T_\mu \\ \Rightarrow\ & V^* = V_{\pi=(\mu,\mu,\mu,\ldots)} \end{aligned}$$

Now suppose,

$$\begin{aligned} & \pi = (\mu, \mu, \mu, ...) \text{ is optimal} \\ \Rightarrow\ & V_{\pi=(\mu,\mu,\mu,\ldots)} = V^* \\ \Rightarrow\ & T_\mu V_\pi = TV^* (\text{as: } T_\mu V_\pi = V_\pi, TV^* = V^*) \\ \Rightarrow\ & T_\mu V^* = TV^* (\text{as: } V_\pi = V^*) \end{aligned}$$

□

Theorem 2 implies there is always a stationary optimal policy, namely the policy $\pi = (\mu, \mu, \ldots)$ such that $T_\mu V^* = TV^*$.

4 Various way of performing the value function updates in practice

4.1 The value function updates we have covered so far: $V \leftarrow TV$

Iterate

- $\forall s : \tilde{V}(s) \leftarrow \max_a [R(s) + \gamma \sum_{s'} P(s'|s,a)V(s')]$
- $V(s) \leftarrow \tilde{V}(s)$

From our theoretical results we have that no matter with which vector V we start, this procedure will converge to V^*.

4.2 Gauss-Seidel value iteration (problem set #1, prove this converges)

Iterate

- for $s = 1, 2, 3, \ldots$

$$V(s) \leftarrow \max_a [R(s) + \gamma \sum_{s'} P(s'|s,a)V(s')].$$

In most cases, Gauss-Seidel value iteration requires less computational time. It also requires less storage (only V, rather than both $\tilde{V}$ and V).

4.3 Asynchronous value iteration

Pick an infinite sequence of states,

$$s^{(0)}, s^{(1)}, s^{(2)}, \ldots$$

such that every state $s \in S$ occurs infinitely often. Define the operators $T_{s^{(k)}}$ as follows:

$$(T_{s^{(k)}}V)(s) = \begin{cases} (TV)(s), & \text{if } s^{(k)} = s \\ V(s), & \text{otherwise} \end{cases}$$

Asynchronous value iteration initializes V and then applies, in sequence, $T_{s^{(0)}}, T_{s^{(1)}}, \ldots$.

We now give a proof sketch of the convergence of asynchronous value iteration:
Let l_1 be a sequence such that all states have appeared at least once in: $s^{(0)}, s^{(1)}, s^{(2)}, \ldots s^{(l_1)}$
Let l_2 be a sequence such that all states have appeared at least once in: $s^{(l_1+1)}, s^{(l_1+2)}, \ldots s^{(l_2)}$
And so forth for $l_3, l_4, \ldots$.

To prove asynchronous value iteration converges to V^*, it suffices to show that for all i we have that the combined operator $T_{s^{(l_{i+1})}} \ldots T_{s^{(l_i)}}$ is a contraction, i.e., for any $V, \bar{V}$ we have that:

$$\left\| T_{s^{(l_{i+1})}} \ldots T_{s^{(l_i)}} V - T_{s^{(l_{i+1})}} \ldots T_{s^{(l_i)}} \bar{V} \right\|_\infty \leq \gamma \left\| V - \bar{V} \right\|_\infty .$$

Proving this contraction property is left as an exercise. (There is a very similar exercise in problem set #1, namely proving Gauss-Seidel value iteration converges.)

4.4 A back-up schedule that can work very fast in practice

Recall the Bellman back-up:

$$V(s) \leftarrow \max_a [R(s) + \gamma \sum_{s'} P(s'|s,a)V(s')]$$

This update is only useful when $V(s')$ has changed for some s', a, s.t. $P(s'|s,a) \neq 0$.

In practice, the transition matrix is often sparse. In these cases, the following scheduling can substantially speed up convergence:

Initialize the queue $q = (1, 2, 3, ..., |s|)$, while the queue q is not empty:

$$s = \text{pop first element from the queue } q$$

$$V(s) \leftarrow \max_a [R(s) + \gamma \sum_{s'} P(s'|s,a)V(s')]$$

$$\forall s'': P(s|s'', a) \neq 0, \text{for some } a, \text{add } s'' \text{ to the back of the queue } q, \text{when doing so, avoid duplication}$$