B9120-001 Dynamic Programming Fall 2023

Prof. Daniel Russo

Homework Assignment 8: Due Friday November 17

Approximate Fixed Point Operators.

Let $\mathcal{J} \subset \mathbb{R}^n$ and let $T : \mathcal{J} \to \mathcal{J}$ be a contraction operator with modulus α in some norm $\|\cdot\|$. That is, $\|TJ - TJ'\| \leq \alpha\|J - J'\|$ for all $J, J' \in \mathcal{J}$. Let J^* denote its unique fixed point. Let $\hat{T} : \mathcal{J} \to \mathcal{J}$ represent an approximation to T. Define the induced norm

$$\|\hat{T} - T\| = \sup_{J \in \mathcal{J}} \|\hat{T}J - TJ\|.$$

Part (a) Suppose $\hat{J} = \hat{T}\hat{J}$ is a fixed point of $\hat{T}$. Show

$$\|\hat{J} - J^*\| \leq \frac{\|\hat{T} - T\|}{1 - \alpha}$$

Part (b) No longer assume $\hat{T}$ has a fixed point. Consider the iteration $\hat{J}_{k+1} = \hat{T}\hat{J}_k$ for $k = 0, 1, 2, \cdots$. Show

$$\limsup_{k \to \infty} \|\hat{J}_k - J^*\| \leq \frac{\|\hat{T} - T\|}{1 - \alpha}$$

Remark: *For concreteness, one can have in mind for this problem that* $\mathcal{J} = \{J \in \mathbb{R}^n : \|J\|_\infty \leq \frac{M}{1-\alpha}\}$ *denotes a bounded subset of cost-to-go functions,* T *is the Bellman optimality operator and* $\hat{T}$ *is an approximation to* T *that is consistent with the apriori bounds on* J^*. *For example,* $\hat{T}J(s) = \min_u \hat{g}(s, a) + \alpha \sum_{x'} \hat{P}(x'|x, u)J(s')$ *could be a Bellman operator for an MDP whose transition probabilities were estimated from data.*

In addition, this result could be used to study approximate value iteration. Lemma 2 from lecture notes 9A follows from the results above.

Asynchronous value iteration

Consider an infinite horizon discounted problem as in Lecture 4.

Consider the iteration

$$J_{k+1}(x) \leftarrow \begin{cases} TJ_k(x) & \text{if } x = x_k \\ J_k(x) & \text{if } x \neq x_k \end{cases}$$

for some sequence of states $(x_i : i = 0, 1, 2, \cdots)$. Suppose the state space is finite and each state is updated infinitely often. Show that $\|J_k - J^*\|_\infty \to 0$.

Hint: We know that

$$J^* - \|J_0 - J^*\|_\infty e \preceq J_0 \preceq J^* - \|J_0 - J^*\|_\infty e,$$

where e denotes a vector of all ones. Show that if a state c has been updated *at least once* by time k, then

$$J^*(x) - \alpha\|J^* - J_0\|_\infty \leq J_k(x) \leq J^*(x) + \alpha\|J_0 - J^*\|_\infty.$$

Remark: *The index k is used for analysis only. On a computer this method could be implemented while storing only a single value function. There are a few ways to motivate a method like this.*

a) *The real-time DP algorithm we looked at in class is a special case of this iteration.*

b) *A variant of value iteration called Gauss-Seidel value iteration is more common in practice than the batch version we saw in class. It is a special case of iteration above, with cyclic order $x_1 = 1, x_2 = 2$,,..., $x_n = n$ and then $x_{n+1} = 1, x_{n+2} = 2$ and so on. This method has the benefits of (a) allowing one to update a single value function "in place" rather than store two copies and (b) often requiring fewer loops through the state space in practice. Intuitively, the latter property is due to using an updated value at states 1 & 2 when calculating the value of state* 3.

c) *(Very roughly...) Imagine a parallelized implementation of value iteration were $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_m$ is divided into m parts, with most (but not all) transitions from states in $\mathcal{X}_i$ going to other states in $\mathcal{X}_i$. Node i computes $TJ_k(x)$ for states $x \in \mathcal{X}_i$ and then communicates the results. A batch implementation of value iteration requires all nodes are idle while waiting for the slowest node to finish its computation. With appropriate asynchronous variants of value iteration, no processor sits idle. The version given in this problem above matches an extreme case where $\mathcal{X}_i$ is a singleton and update value estimates are communicated as soon as they are available. The sequence of states is determined by the timing of the processors.*