

Lecture 4: Theoretical Fundamentals of Dynamic Programming

Diana Borsa

21st January 2021, UCL



This Lecture

- ▶ Last lecture: MDP, DP, Value Iteration (VI), Policy Iteration (PI)
- ▶ This lecture:
 - ▶ Deepen the mathematical formalism behind the MDP framework.
 - ▶ Revisit the **Bellman equations** and introduce their corresponding **operators**.
 - ▶ Re-visit the paradigm of **dynamic programming**: VI and PI.
- ▶ Next lectures: approximate, sampled versions of these paradigms, mainly in the absence of perfect knowledge of the environment.



Preliminaries

(Quick Recap of Functional Analysis)



Normed Vector Spaces

- ▶ Normed Vector Spaces: vector space \mathcal{X} + a norm $\|\cdot\|$ on the elements of \mathcal{X} .
- ▶ Norms are defined a mapping $\mathcal{X} \rightarrow \mathbb{R}$ s.t:
 1. $\|x\| \geq 0, \forall x \in \mathcal{X}$ and if $\|x\| = 0$ then $x = \mathbf{0}$.
 2. $\|\alpha x\| = |\alpha| \|x\|$ (homogeneity)
 3. $\|x_1 + x_2\| \leq \|x_1\| + \|x_2\|$ (triangle inequality)
- ▶ For this lecture:
 - ▶ Vector spaces: $\mathcal{X} = \mathbb{R}^d$
 - ▶ Norms:
 - ▶ max-norm/ L_∞ norm $\|\cdot\|_\infty$
 - ▶ (weighted) L_2 norms $\|\cdot\|_{2,\rho}$



Contraction Mapping

Definition

Let \mathcal{X} be a vector space, equipped with a norm $\|\cdot\|$. A mapping $\mathcal{T} : \mathcal{X} \rightarrow \mathcal{X}$ is a **α -contraction** mapping if for any $x_1, x_2 \in \mathcal{X}$, $\exists \alpha \in [0, 1)$ s.t.

$$\|\mathcal{T}x_1 - \mathcal{T}x_2\| \leq \alpha \|x_1 - x_2\|$$

- ▶ If $\alpha \in [0, 1]$, then we call \mathcal{T} **non-expanding**
- ▶ Every contraction is also (by definition) **Lipschitz**, thus it is also **continuous**. In particular this means:

$$\text{If } x_n \rightarrow_{\|\cdot\|} x \text{ then } \mathcal{T}x_n \rightarrow_{\|\cdot\|} \mathcal{T}x$$

→ full contraction if $\alpha < 1$



Fixed point

Definition

A point/vector $x \in \mathcal{X}$ is a **fixed point** of an operator \mathcal{T} if $\mathcal{T}x = x$.



Banach Fixed Point Theorem

every convergent sequence limit's contains the space.

Theorem (Banach Fixed Point Theorem)

Let \mathcal{X} a *complete normed* vector space, equipped with a norm $\|\cdot\|$ and $\mathcal{T} : \mathcal{X} \rightarrow \mathcal{X}$ a γ -contraction mapping, then:

1. \mathcal{T} has a *unique fixed point* $x \in \mathcal{X}$: $\exists! x^* \in \mathcal{X}$ s.t. $\mathcal{T}x^* = x^*$
2. $\forall x_0 \in \mathcal{X}$, the sequence $x_{n+1} = \mathcal{T}x_n$ converges to x^* in a geometric fashion:

$$\|x_n - x^*\| \leq \gamma^n \|x_0 - x^*\|$$

Thus $\lim_{n \rightarrow \infty} \|x_n - x^*\| \leq \lim_{n \rightarrow \infty} (\gamma^n \|x_0 - x^*\|) = 0$.



Markov Decision Processes and Dynamic Programming (Recap)



(Recap) MDPs

- ▶ Markov Decision Processes (MDPs) formally describe an environment:

transition dynamics ↗ *reward under transition*.
 $M = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$

- ▶ Almost all RL problems can be formalised as MDPs, e.g.
 - ▶ Optimal control primarily deals with continuous MDPs
 - ▶ Partially observable problems can be converted into MDPs
 - ▶ Bandits are MDPs with one state



(Recap) Value functions

- ▶ State value function, for a policy π :

$$v_\pi(s) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | s_0 = s; \pi \right]$$

- ▶ Action value function, for a policy π :

$$q_\pi(s, a) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | s_0 = s, a_0 = a, \pi \right]$$

we take action
a (maybe not
accordingly to
 π)

- ▶ Optimal value functions: $q^* = \max_\pi q_\pi$ ($v^* = \max_\pi v_\pi$)



(Recap) Bellman Equations

Theorem (Bellman Expectation Equations)

Given an MDP, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, for *any policy π , the value functions obey the following expectation equations:*

$$v_{\pi}(s) = \sum_a \pi(s, a) \left[r(s, a) + \gamma \sum_{s'} p(s'|a, s) v_{\pi}(s') \right] \quad (1)$$

$$q_{\pi}(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|a, s) \sum_{a' \in \mathcal{A}} \pi(a'|s') q_{\pi}(s', a') \quad (2)$$



(Recap) The Bellman Optimality Equation

Theorem (Bellman Optimality Equations)

Given an MDP, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, the *optimal value functions* obey the following expectation equations:

$$v^*(s) = \max_a \left[r(s, a) + \gamma \sum_{s'} p(s'|a, s) v^*(s') \right] \quad (3)$$

$$q^*(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|a, s) \max_{a' \in \mathcal{A}} q^*(s', a') \quad (4)$$



Bellman Operators



The Bellman Optimality Operator

Definition (Bellman Optimality Operator $T_{\mathcal{V}}^*$)

Given an MDP, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, let $\mathcal{V} \equiv \mathcal{V}_{\mathcal{S}}$ be the space of bounded real-valued functions over \mathcal{S} . We define, point-wise, the **Bellman Optimality operator** $T_{\mathcal{V}}^* : \mathcal{V} \rightarrow \mathcal{V}$ as:

↳ The argument s returns number

↳ V^ instead of F is optimal.*

$$(T_{\mathcal{V}}^* f)(s) = \max_a \left[r(s, a) + \gamma \sum_{s'} p(s'|a, s) f(s') \right], \forall f \in \mathcal{V} \quad V^* \text{ is a fixed point.} \quad (5)$$

↳ returns function that takes s .

As a common convention we drop the index \mathcal{V} and simply use $T^* = T_{\mathcal{V}}^*$



Properties of the Bellman Operator T^*

1. It has one **unique fixed point** v^* .

$$T^*v^* = v^*$$

2. T^* is a **γ -contraction** wrt. to $\|.\|_\infty$

$$\|T^*v - T^*u\|_\infty \leq \gamma \|v - u\|_\infty, \forall u, v \in \mathcal{V}$$

3. T^* is **monotonic**: *these are functions.*

$\forall u, v \in \mathcal{V}$ s.t. $u \leq v$, component-wise, then $T^*u \leq T^*v$

To regardless of state.



Properties of the Bellman Operator T^* (Proofs)

Prop. (2): T^* is a γ -contraction wrt. to $\|\cdot\|_\infty$

$$\|T^*v - T^*u\|_\infty \leq \gamma \|v - u\|_\infty$$

Proof.

$$|T^*v(s) - T^*u(s)| = |\max_a [r(s, a) + \gamma \mathbb{E}_{s'|s,a} v(s')] - \max_b [r(s, b) + \gamma \mathbb{E}_{s''|s,b} u(s'')]| \quad (6)$$

$$\leq \max_a |[r(s, a) + \gamma \mathbb{E}_{s'|s,a} v(s')] - [r(s, a) + \gamma \mathbb{E}_{s'|s,a} u(s')]| \quad (7)$$

$$= \gamma \max_a |\mathbb{E}_{s'|s,a} [v(s') - u(s')]| \quad (8)$$

$$\leq \gamma \max_{s'} |v(s') - u(s')| \quad \Rightarrow \|v - u\|_\infty \quad (9)$$

$$\begin{aligned} & |\max_a f(a) - \max_b g(b)| \\ & \leq \max_a |f(a) - g(a)| \end{aligned}$$

Thus we get:

$$\|T^*v - T^*u\|_\infty \leq \gamma \|v - u\|_\infty, \forall u, v \in \mathcal{V}$$



Note: Step (6)-(7) uses: $|\max_a f(a) - \max_b g(b)| \leq \max_a |f(a) - g(a)|$



Properties of the Bellman Operator T^* (Proofs)

$$v(s) \leq u(s) \quad \forall s, \Rightarrow T^*v(s) \leq T^*u(s)$$

Prop. (3): T^* is **monotonic**

Proof.

Given $v(s) \leq u(s), \forall s \Rightarrow r(s, a) + \mathbb{E}_{s'|s,a} v(s') \leq r(s, a) + \mathbb{E}_{s'|s,a} u(s'). \quad \textcircled{1}$

$$T^*v(s) - T^*u(s) = \max_a [r(s, a) + \gamma \mathbb{E}_{s'|s,a} v(s')] - \max_b [r(s, b) + \gamma \mathbb{E}_{s''|s,b} u(s'')] \quad (10)$$

$$\leq \max_a ([r(s, a) + \gamma \mathbb{E}_{s'|s,a} v(s')] - [r(s, a) + \gamma \mathbb{E}_{s'|s,a} u(s')]) \quad (11)$$

$$\leq 0, \forall s. \quad \text{using } \textcircled{1} \Rightarrow v(s) \leq u(s) \quad \forall s \quad (12)$$

Thus $T^*v(s) \leq T^*u(s), \forall s \in \mathcal{S}$.

□



Value Iteration through the lens of the Bellman Operator

Value Iteration

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = T^*v_k$.

before: $v_{K+1}^{(s)} \leftarrow \max_a [r(s, a) + \gamma E(v_k(s))]$

As $k \rightarrow \infty$, $v_k \rightarrow \|.\|_\infty v^*$.

Proof: Direct application of the *Banach Fixed Point Theorem*.

$$\begin{aligned}\|v_k - v^*\|_\infty &= \|T^*v_{k-1} - v^*\|_\infty \\ &= \|T^*v_{k-1} - T^*v^*\|_\infty \quad (\text{fixed point prop.}) \\ &\leq \gamma \|v_{k-1} - v^*\|_\infty \quad (\text{contraction prop.}) \\ &\leq \underbrace{\gamma^k \|v_0 - v^*\|_\infty}_{< 0 \text{ if } \gamma < 1} \quad (\text{iterative application})\end{aligned}$$

$T^*v_k(s)$

$\gamma < 1$ so converge if



The Bellman Expectation Operator

Definition (Bellman Expectation Operator)

Given an MDP, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, let $\mathcal{V} \equiv \mathcal{V}_{\mathcal{S}}$ be the space of bounded real-valued functions over \mathcal{S} . For any policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, we define, point-wise, the **Bellman Expectation operator** $T_{\mathcal{V}}^{\pi} : \mathcal{V} \rightarrow \mathcal{V}$ as:

$$(T_{\mathcal{V}}^{\pi} f)(s) = \sum_a \pi(s, a) \left[r(s, a) + \gamma \sum_{s'} p(s'|a, s) f(s') \right], \forall f \in \mathcal{V} \quad (13)$$



Properties of the Bellman Operator T^π

1. It has one **unique fixed point** v_π .

$$T^\pi v_\pi = v_\pi$$

2. T^π is a **γ -contraction** wrt. to $\|.\|_\infty$

$$\|T^\pi v - T^\pi u\|_\infty \leq \gamma \|v - u\|_\infty, \forall u, v \in \mathcal{V}$$

3. T^π is **monotonic**:

$\forall u, v \in \mathcal{V}$ s.t. $u \leq v$, component-wise, then $T^\pi u \leq T^\pi v$



Properties of the Bellman Operator T^π (Proofs)

Prop. (2): T^π is a **γ -contraction** wrt. to $\|.\|_\infty$

Proof.

$$\begin{aligned} T^\pi v(s) - T^\pi u(s) &= \sum_a \pi(a|s) [r(s, a) + \gamma \mathbb{E}_{s'|s,a} v(s') - r(s, a) - \gamma \mathbb{E}_{s'|s,a} u(s')] \\ &= \gamma \sum_a \pi(a|s) \mathbb{E}_{s'|s,a} [v(s') - u(s')] \tag{14} \\ \Rightarrow |T^\pi v(s) - T^\pi u(s)| &\leq \gamma \max_{s'} |[v(s') - u(s')]| \end{aligned}$$

Thus we get:

$$\|T^\pi v - T^\pi u\|_\infty \leq \gamma \|v - u\|_\infty, \forall u, v \in \mathcal{V}$$

□

Note: (14) gives us also Prop. (3), monotonicity of T^π .



Policy Evaluation

(Iterative) Policy Evaluation

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = T^\pi v_k$.

As $k \rightarrow \infty$, $v_k \rightarrow_{\|\cdot\|_\infty} v_\pi$.

Proof: Direct application of the *Banach Fixed Point Theorem*.



(Summary) Dynamic Programming with Bellman Operators

Value Iteration

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = T^*v_k$.

Policy Iteration

- ▶ Start with π_0 .
- ▶ Iterate:
 - ▶ Policy Evaluation: v_{π_i}
 - ▶ (E.g. For instance, by iterating T^π : $v_k = T^{\pi_i}v_{k-1} \Rightarrow v_k \rightarrow v^{\pi_i}$ as $k \rightarrow \infty$)
 - ▶ Greedy Improvement: $\pi_{i+1} = \arg \max_a q_{\pi_i}(s, a)$



Similarly for $q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ functions

Definition (Bellman Expectation Operator)

Given an MDP, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, let $\mathcal{Q} \equiv \mathcal{Q}_{\mathcal{S}, \mathcal{A}}$ be the space of bounded real-valued functions over $\mathcal{S} \times \mathcal{A}$. For any policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, we define, point-wise, the **Bellman Expectation operator** $T_Q^\pi : \mathcal{Q} \rightarrow \mathcal{Q}$ as:

$$(T_Q^\pi f)(s, a) = r(s, a) + \gamma \sum_{s'} p(s' | a, s) \sum_{a' \in \mathcal{A}} \pi(a' | s') f(s', a') , \forall f \in \mathcal{Q}$$

- ▶ This operator has **unique fixed point** which corresponds to the *action-value function* q_π in our MDP \mathcal{M} .
- ▶ Same properties as T^π : γ -contraction and **monotonicity**.



Similarly for $q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ functions

Definition (Bellman Optimality Operator)

Given an MDP, $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, let $\mathcal{Q} \equiv \mathcal{Q}_{\mathcal{S}, \mathcal{A}}$ be the space of bounded real-valued functions over $\mathcal{S} \times \mathcal{A}$. We define the **Bellman Optimality operator** $T_{\mathcal{Q}}^* : \mathcal{Q} \rightarrow \mathcal{Q}$ as:

$$(T_{\mathcal{Q}}^* f)(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|a, s) \max_{a' \in \mathcal{A}} f(s', a') , \forall f \in \mathcal{Q}$$

- ▶ This operator has **unique fixed point** which corresponds to the *action-value function* q^* in our MDP \mathcal{M} .
- ▶ Same properties as T^* : **γ -contraction** and **monotonicity**.



Approximate Dynamic Programming



Approximate DP

- ▶ So far, we have assumed **perfect knowledge** of the MDP and **perfect/exact representation** of the value functions.
- ▶ Realistically, more often than not:
 - ▶ We **won't know the underlying MDP** (like in the next two lectures)
 - ▶ We **won't be able to represent the value function exactly** after each update (lectures to come)



Approximate DP

- ▶ Realistically, more often than not:
 - ▶ We won't know the underlying MDP.
⇒ sampling/estimation error, as we don't have access to the true operators T^π (T^*)
 - ▶ We won't be able to represent the value function exactly after each update.
⇒ approximation error, as we approximate the true value functions within a (parametric) class (e.g. linear functions, neural nets, etc).
- ▶ Objective: Under the above conditions, come up with a policy π that is (close to) optimal.



(Reminder) Value Iteration

Value Iteration

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = T^*v_k$.

As $k \rightarrow \infty$, $v_k \rightarrow_{\|\cdot\|_\infty} v^*$.



Approximate Value Iteration

Approximate Value Iteration

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = \underbrace{AT^*v_k}_{\text{sample or function approximator}} (v_{k+1} \approx T^*v_k)$

Question: As $k \rightarrow \infty$, $v_k \rightarrow_{\|\cdot\|_\infty} v^*$? **X**

Answer: In general, **no**.



ADP: Approximating the value function

→ Assumes we have
true access to MDP

- ▶ Using a **function approximator** $v_\theta(s)$, with a parameter vector $\theta \in \mathbb{R}^m$
- ▶ The estimated value function at iteration k is $v_k = v_{\theta_k}$
- ▶ Use dynamic programming to compute $v_{\theta_{k+1}}$ from v_{θ_k}

$$T^*v_k(s) = \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s]$$

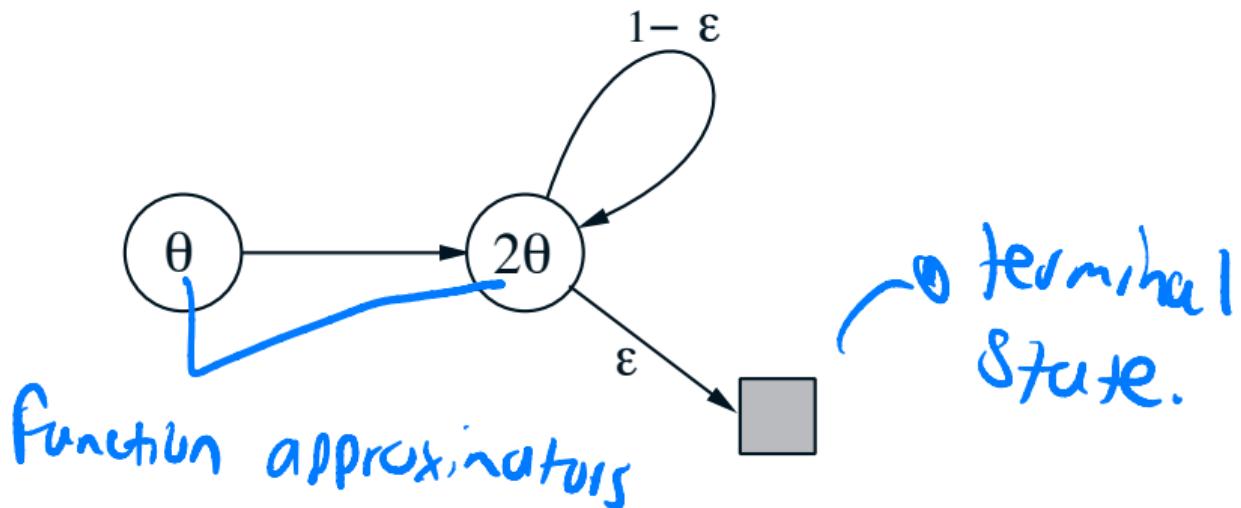
- ▶ Fit θ_{k+1} s.t. $v_{\theta_{k+1}} \approx T^*v_k(s)$
 - ▶ For instance, with respect to a squared loss over the state-space.

$$\theta_{k+1} = \arg \min_{\theta_{k+1}} \sum_s (v_{\theta_{k+1}}(s) - T^*v_k(s))^2$$



Example of divergence with dynamic programming

- ▶ Tsitsiklis and Van Roy made an example where dynamic programming with linear function approximation can diverge. Consider the two state example below, where the rewards are all zero, there are no decisions, and there is a single parameter for estimating the value.



Iteration $K+1$:

$$(D_y, p) \quad v_{K+1} \leftarrow T_{v_K}$$

$$v_K = v_{\theta_K} \quad \text{all rewards are 0}$$

Blackboard

$T_{v_K}(s) = r(s) + \gamma \mathbb{E}_{s'}[v_{K+1}(s')]$ (Tsitsiklis and Van Roy's Example)

$$(T v_K)(s) = \gamma \mathbb{E}_{s'}[v_K(s')] \quad \text{approx. setting: } j_{K+1} = \theta_{K+1} \Rightarrow v_{K+1} = T v_K$$

$$\theta_{K+1} = \arg \min_{\theta} \sum_{s \in S} (v_{\theta}(s) - \mathbb{E}_{s' \sim \pi}[v_{\theta_K}(s') | s' = s])^2$$



$$= \arg \min_{\theta} \left[[\theta - \gamma V_{\theta_k}(s_2)] + \frac{[2\theta - \gamma(\varepsilon \cdot 0 + (1-\varepsilon)V_{\theta_k}(s_1))]}{2\theta_k} \right]$$

Blackboard

(Tsitsiklis and Van Roy's Example)

$$\begin{aligned} &= \arg \min_{\theta} ((\theta - \gamma 2\theta_k)^2 + (2\theta - \gamma(1-\varepsilon)2\theta_k)^2) \\ \frac{\partial f(\theta)}{\partial \theta} &= 2(\theta - \gamma 2\theta_k) + 2 \cdot 2(2\theta - \gamma(1-\varepsilon)2\theta_k) \end{aligned}$$



$$\theta - 2\gamma \theta_K + u\theta - 4\gamma \theta_K(1-\epsilon) = 0$$

$$\rightarrow s\theta = 2\gamma \theta_K(1+2(1-\epsilon))$$

$$\Rightarrow \theta_{K+1} = \frac{2\gamma(3-2\epsilon)}{s}\theta_K$$

Blackboard

(Tsitsiklis and Van Roy's Example)

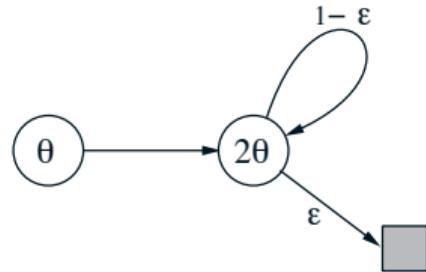
$$\square \leq 1 \Rightarrow \theta_{K+1} \rightarrow 0$$

if $\gamma \geq \theta_{K+1}$ diverges.



Example of divergence with dynamic programming

- ▶ Tsitsiklis and Van Roy made an example where dynamic programming with linear function approximation can diverge. Consider the two state example below, where the rewards are all zero, there are no decisions, and there is a single parameter for estimating the value.



$$\begin{aligned}\theta_{k+1} &= \operatorname{argmin}_{\theta} \sum_{s \in \mathcal{S}} (v_{\theta}(s) - \mathbb{E}[v_{\theta_k}(S_{t+1}) \mid S_t = s])^2 \\ &= \operatorname{argmin}_{\theta} (\theta - \gamma 2\theta_k)^2 + (2\theta - \gamma(1 - \epsilon)2\theta_k)^2 \\ &= \frac{2(3 - 2\epsilon)\gamma}{5} \theta_k\end{aligned}$$

- ▶ What is $\lim_{k \rightarrow \infty} \theta_k$ when $\theta_0 = 1$, $\epsilon = \frac{1}{8}$, and $\gamma = 1$?
- ▶ This is only a problem when we update the states, e.g., synchronously, without looking at the time an agent would spend in each state



Approximate Value Iteration

Approximate Value Iteration

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = \mathcal{A}T^*v_k$. $(v_{k+1} \approx T^*v_k)$

Question: As $k \rightarrow \infty$, $v_k \rightarrow_{\|\cdot\|_\infty} v^*$? **X**

Answer: In general, **no**.

Hopeless? Not quite!

- ▶ Sample versions of these algorithms converge under mild conditions
- ▶ Even for the function approximation case, the theoretical danger of divergence is rarely materialised in practice
- ▶ There may **many value functions** that can induce the **optimal policy**!



Example from last lecture: Many value functions \Rightarrow same optimal policy

$k = 3$

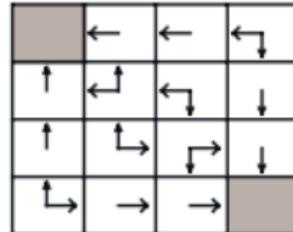
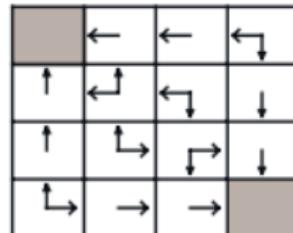
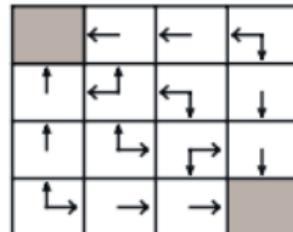
0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



optimal
policy



Performance of a Greedy Policy

$$q \rightarrow \pi = \text{greedy}(a)$$

Theorem (Value of greedy policy)

Consider a MDP. Let $q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be an arbitrary function and let π be the greedy policy associated with q , then:

→ approximation you used

$$\|q^* - q^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma} \|q^* - q\|_\infty$$

where q^* is the optimal value function associated with this MDP.



Performance of a Greedy Policy (Proof)

Statement: $\|q^* - q^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma} \|q^* - q\|_\infty$

Proof.

$$\|q^* - q^\pi\|_\infty = \|q^* - T^\pi q + T^\pi q - q^\pi\|_\infty \quad \text{triangle inequality} \quad (15)$$

$$\leq \|q^* - T^\pi q\|_\infty + \|T^\pi q - q^\pi\|_\infty \quad (16)$$

$$= \|T^* q^* - T^* q\|_\infty + \|T^\pi q - T^\pi q^\pi\|_\infty \quad (17)$$

$$\leq \gamma \|q^* - q\|_\infty + \gamma \underbrace{\|q - q^\pi\|_\infty}_{\leq \|q - q^*\|_\infty + \|q^* - q^\pi\|_\infty} \quad (18)$$

$$\leq 2\gamma \|q^* - q\|_\infty + \gamma \|q^* - q^\pi\|_\infty \quad (19)$$

Fix a point $T^\pi q = T^* q$

Re-arranging: $(1 - \gamma) \|q^* - q^\pi\|_\infty \leq 2\gamma \|q^* - q\|_\infty$.

→ divide by $(1 - \gamma)$



Performance of a Greedy Policy: Test your understanding!

Theorem (Value of greedy policy)

Consider a MDP. Let $q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be an arbitrary function and let π be the greedy policy associated with q , then:

$$\|q^* - q^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma} \|q^* - q\|_\infty$$

Smaller value of Gamma makes this term small.

where q^* is the optimal value function associated with this MDP.

Observations:

- ▶ Small values of γ obtain a better(smaller) upper bound on the potential loss of performance. How do you interpret that?
- ▶ In particular, what happens for $\gamma = 0$? How do you explain this?
- ▶ What if $q = q^*$? What does this bound imply in that case?

Low reward converge to optimal

only immediate reward.



(Reminder) Policy Iteration

Policy Iteration

- ▶ Start with π_0 .
- ▶ Iterate:
 - ▶ Policy Evaluation: $q_i = q_{\pi_i}$
 - ▶ Greedy Improvement: $\pi_{i+1} = \arg \max_a q_{\pi_i}(s, a)$

As $i \rightarrow \infty$, $q_i \rightarrow_{\|\cdot\|_\infty} q^*$. Thus $\pi_i \rightarrow \pi^*$.



Approximate Policy Iteration

Approximate Policy Iteration

- ▶ Start with π_0 .
- ▶ Iterate:
 - ▶ Policy Evaluation: $q_i = \mathcal{A}q_{\pi_i}$ $(q_i \approx q_{\pi_i})$
 - ▶ Greedy Improvement: $\pi_{i+1} = \arg \max_a q_i(s, a)$

Question 1: As $i \rightarrow \infty$, does $q_i \rightarrow_{\|\cdot\|_\infty} q^*$? **X**

Answer: In general, **no**.

Question 2: Or does π_i converge to the optimal policy? **X**

Answer: In general, **no**.

Hopeless? In some cases, **no**, depending on the nature of \mathcal{A} . (More: Next lectures)



(Summary) Approximate Dynamic Programming

Approximate Value Iteration

- ▶ Start with v_0 .
- ▶ Update values: $v_{k+1} = \mathcal{A}T^*v_k$. $(v_{k+1} \approx T^*v_k)$

Approximate Policy Iteration

- ▶ Start with π_0 .
- ▶ Iterate:
 - ▶ Policy Evaluation: $q_i = \mathcal{A}q_{\pi_i}$ $(q_i \approx q_{\pi_i})$
 - ▶ Greedy Improvement: $\pi_{i+1} = \arg \max_a q_i(s, a)$



Questions?

The only stupid question is the one you were afraid to ask but never did.
-Rich Sutton

For questions that may arise during this lecture please use Moodle and/or the next Q&A session.



