# 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 $\|.\|$ on the elements of $\mathcal{X}$.
- Norms are defined a mapping $\mathcal{X} \to \mathbb{R}$ 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/$L_\infty$ norm $\|.\|_\infty$
    - (weighted) $L_2$ norms $\|.\|_{2,\rho}$

# Contraction Mapping

**Definition**

Let $\mathcal{X}$ be a vector space, equipped with a norm $||.||$. An mapping $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ is a $\alpha$-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:

$$\text{If } x_n \to_{\|.\|} x \text{ then } \mathcal{T}x_n \to_{\|.\|} \mathcal{T}x$$

# 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

**Theorem (Banach Fixed Point Theorem)**

*Let $\mathcal{X}$ a complete normed vector space, equipped with a norm $||.||$ and $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ a $\gamma$-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* $\lim_{n\to\infty} \|x_n - x^*\| \leq \lim_{n\to\infty} (\gamma^n \|x_0 - x^*\|) = 0.$

# Markov Decision Processes and Dynamic Programming (Recap)

# (Recap) MDPs

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

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$$

- 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 $\pi$ :

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

- Action value function, for a policy $\pi$:

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

- 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 $\pi$, the value functions obey the following expectation equations:*

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

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

# (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 $T^*_{\mathcal{V}} : \mathcal{V} \to \mathcal{V}$ as:

$$(T^*_{\mathcal{V}} f)(s) = \max_a \left[ r(s,a) + \gamma \sum_{s'} p(s'|a,s) f(s') \right], \forall f \in \mathcal{V} \tag{5}$$

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 $\gamma$-contraction wrt. to $\|.\|_\infty$

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

3. $T^*$ is monotonic:

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

# Properties of the Bellman Operator $T^*$ (Proofs)

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

**Proof.**

$$
\begin{aligned}
|T^*v(s) - T^*u(s)| &= \left|\max_a \left[r(s,a) + \gamma \mathbb{E}_{s'|s,a} v(s')\right] - \max_b \left[r(s,b) + \gamma \mathbb{E}_{s''|s,b} u(s'')\right]\right| && (6)\\
&\leq \max_a \left|\left[r(s,a) + \gamma \mathbb{E}_{s'|s,a} v(s')\right] - \left[r(s,a) + \gamma \mathbb{E}_{s'|s,a} u(s')\right]\right| && (7)\\
&= \gamma \max_a \left|\mathbb{E}_{s'|s,a}\left[v(s') - u(s')\right]\right| && (8)\\
&\leq \gamma \max_{s'} \left|\left[v(s') - u(s')\right]\right| && (9)
\end{aligned}
$$

Thus we get:

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

$\square$

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)

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

**Proof.**

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

$$\begin{aligned} T^* v(s) - T^* u(s) &= \max_a \left[ r(s,a) + \gamma \mathbb{E}_{s'|s,a} v(s') \right] - \max_b \left[ r(s,b) + \gamma \mathbb{E}_{s''|s,b} u(s'') \right] && (10) \\ &\le \max_a \left( \left[ r(s,a) + \gamma \mathbb{E}_{s'|s,a} v(s') \right] - \left[ r(s,a) + \gamma \mathbb{E}_{s'|s,a} u(s') \right] \right) && (11) \\ &\le 0, \forall s. && (12) \end{aligned}$$

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

# Value Iteration through the lens of the Bellman Operator

## Value Iteration

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

As $k \to \infty$, $v_k \to_{\|.\|_\infty} v^*$.

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

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

# 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 $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$, we define, point-wise, the Bellman Expectation operator $T_{\mathcal{V}}^{\pi} : \mathcal{V} \to \mathcal{V}$ as:

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

# Properties of the Bellman Operator $T^\pi$

1. It has one unique fixed point $v_\pi$.

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

2. $T^\pi$ is a $\gamma$-contraction wrt. to $\|.\|_\infty$

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

3. $T^\pi$ is monotonic:

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

# Properties of the Bellman Operator $T^{\pi}$ (Proofs)

Prop. (2): $T^{\pi}$ is a $\gamma$-contraction wrt. to $\|.\|_{\infty}$

**Proof.**

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

Thus we get:

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

$\square$

Note: (14) gives us also Prop. (3), monotonicity of $T^{\pi}$.

# Policy Evaluation

## (Iterative) Policy Evaluation

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

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

*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 $\pi_0$.
- Iterate:
  - Policy Evaluation: $v_{\pi_i}$
    - ( E.g. For instance, by iterating $T^\pi$: $v_k = T^{\pi_i} v_{k-1} \Rightarrow v_k \rightarrow v^{\pi_i}$ as $k \rightarrow \infty$)
  - Greedy Improvement: $\pi_{i+1} = \arg\max_a q_{\pi_i}(s, a)$

## Similarly for $q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 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 $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, we define, point-wise, the Bellman Expectation operator $T^\pi_{\mathcal{Q}} : \mathcal{Q} \to \mathcal{Q}$ as:

$$(T^\pi_{\mathcal{Q}} f)(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|a, s) \sum_{a' \in \mathcal{A}} \pi(a'|s') f(s', a') \ , \forall f \in \mathcal{Q}$$

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

# Similarly for $q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 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 $T^*_{\mathcal{Q}} : \mathcal{Q} \to \mathcal{Q}$ 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 $T^*$: $\gamma$-contraction and monotonicity.

# Approximate Dynamic Programming

# Approximate DP

- So far, we have assume 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.
    $\Rightarrow$ sampling/estimation error, as we don't have access to the true operators $T^\pi$ ($T^*$)
  - We won't be able to represent the value function exactly after each update.
    $\Rightarrow$ 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 $\pi$ that is (close to) optimal.

# (Reminder) Value Iteration

## Value Iteration

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

As $k \to \infty$, $v_k \to_{\|.\|_\infty} v^*$.

# 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 $k \to \infty$, $v_k \to_{\|.\|_\infty} v^*$? **X**
Answer: In general, no.

# ADP: Approximating the value function

- 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 $v_{\theta_k}$

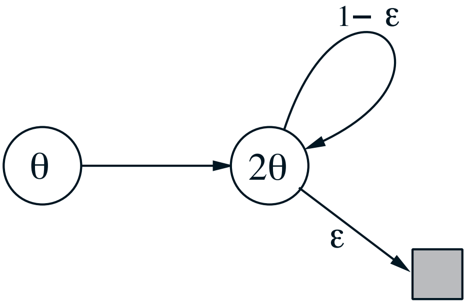
$$T^* v_k(s) = \max_a \mathbb{E}\left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s\right]$$

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

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

# 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.

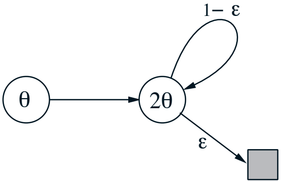


Blackboard

(Tsitsiklis and Van Roy's Example)

# 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.




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

- What is $\lim_{k\to\infty} \theta_k$ 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 $k \to \infty$, $v_k \to_{\|.\|_\infty} v^*$? **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

$k = 3$

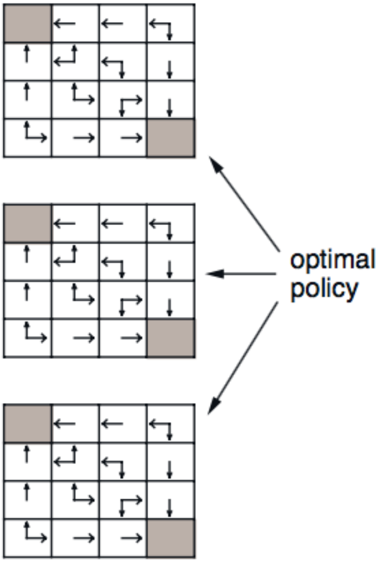
| | | | |
|---|---|---|---|
| 0.0 | -2.4 | -2.9 | -3.0 |
| -2.4 | -2.9 | -3.0 | -2.9 |
| -2.9 | -3.0 | -2.9 | -2.4 |
| -3.0 | -2.9 | -2.4 | 0.0 |

$k = 10$

| | | | |
|---|---|---|---|
| 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 |



# Performance of a Greedy Policy

**Theorem (Value of greedy policy)**

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

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

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

# Performance of a Greedy Policy (Proof)

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

**Proof.**

$$
\begin{aligned}
\|q^* - q^\pi\|_\infty &= \|q^* - T^\pi q + T^\pi q - q^\pi\|_\infty &(15)\\
&\le \|q^* - T^\pi q\|_\infty + \|T^\pi q - q^\pi\|_\infty &(16)\\
&= \|T^* q^* - T^* q\|_\infty + \|T^\pi q - T^\pi q^\pi\|_\infty &(17)\\
&\le \gamma\|q^* - q\|_\infty + \gamma \underbrace{\|q - q^\pi\|_\infty}_{\le \|q - q^*\|_\infty + \|q^* - q^\pi\|_\infty} &(18)\\
&\le 2\gamma\|q^* - q\|_\infty + \gamma\|q^* - q^\pi\|_\infty &(19)
\end{aligned}
$$

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

# 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 $\pi$ be the greedy policy associated with $q$, then:*

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

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

Observations:

- Small values of $\gamma$ 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?

# (Reminder) Policy Iteration

## Policy Iteration

- Start with $\pi_0$.
- Iterate:
  - Policy Evaluation: $q_i = q_{\pi_i}$
  - Greedy Improvement: $\pi_{i+1} = \arg\max_a q_{\pi_i}(s, a)$

As $i \to \infty$, $q_i \to_{\|.\|_\infty} q^*$. Thus $\pi_i \to \pi^*$.

# Approximate Policy Iteration

## Approximate Policy Iteration

- Start with $\pi_0$.
- Iterate:
  - Policy Evaluation: $q_i = \mathcal{A} q_{\pi_i}$ $\qquad (q_i \approx q_{\pi_i})$
  - Greedy Improvement: $\pi_{i+1} = \arg\max_a q_i(s, a)$

Question 1: As $i \to \infty$, does $q_i \to_{\|.\|_\infty} q^*$? **X**
Answer: In general, no.

Question 2: Or does $\pi_i$ 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 $\pi_0$.
- 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.