CS 747, Autumn 2020: Week 6, Lecture 1

Shivaram Kalyanakrishnan

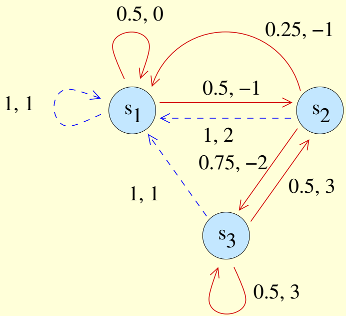
Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2020

MDPs: Weeks 4 and 5

1. MDP, policy, value function
2. MDP planning
3. Alternative formulations
4. Applications
5. Policy Evaluation

6. Banach's Fixed-point theorem
7. Bellman optimality operator
8. Value Iteration
9. Linear Programming



Markov Decision Problems

1. Policy Iteration
   - Policy evaluation (review)
   - Policy improvement
   - Algorithm and variants

2. Proof of Policy Improvement Theorem
   - Bellman operator
   - Proof

3. Computational complexity of MDP planning

4. Summary

Markov Decision Problems

1. Policy Iteration
    - Policy evaluation (review)
    - Policy improvement
    - Algorithm and variants
2. Proof of Policy Improvement Theorem
    - Bellman operator
    - Proof
3. Computational complexity of MDP planning
4. Summary

Policy Evaluation (Review)

- Recall that for MDP (S, A, T, R, γ) and policy $\pi : S \to A$, value function $V^\pi : S \to \mathbb{R}$ and action value function $Q^\pi : S \times A \to \mathbb{R}$ are obtained as below.

Policy Evaluation (Review)

- Recall that for MDP (S, A, T, R, γ) and policy $\pi : S \to A$, value function $V^{\pi} : S \to \mathbb{R}$ and action value function $Q^{\pi} : S \times A \to \mathbb{R}$ are obtained as below.
- For $s \in S$:

$$V^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s')\{R(s, \pi(s), s') + \gamma V^{\pi}(s')\}.$$

Policy Evaluation (Review)

- Recall that for MDP (S, A, T, R, γ) and policy $\pi : S \to A$,
  value function $V^{\pi} : S \to \mathbb{R}$ and
  action value function $Q^{\pi} : S \times A \to \mathbb{R}$
  are obtained as below.
- For $s \in S$:

$$V^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s')\{R(s, \pi(s), s') + \gamma V^{\pi}(s')\}.$$

- For $s \in S, a \in A$:

$$Q^{\pi}(s, a) = \sum_{s' \in S} T(s, a, s')\{R(s, a, s') + \gamma V^{\pi}(s')\}.$$

Policy Evaluation (Review)

- Recall that for MDP (S, A, T, R, γ) and policy $\pi : S \to A$,
  value function $V^\pi : S \to \mathbb{R}$ and
  action value function $Q^\pi : S \times A \to \mathbb{R}$
  are obtained as below.
- For $s \in S$:

$$V^\pi(s) = \sum_{s' \in S} T(s, \pi(s), s')\{R(s, \pi(s), s') + \gamma V^\pi(s')\}.$$

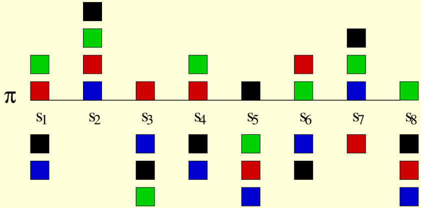
- For $s \in S, a \in A$:

$$Q^\pi(s, a) = \sum_{s' \in S} T(s, a, s')\{R(s, a, s') + \gamma V^\pi(s')\}.$$

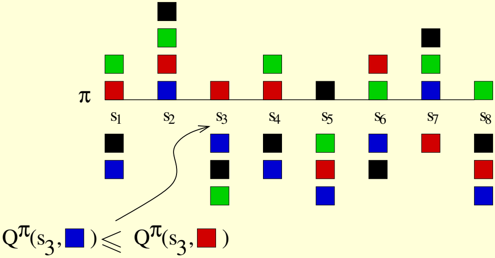
- V^π and Q^π computable in poly(n, k) arithmetic operations.

Policy Improvement

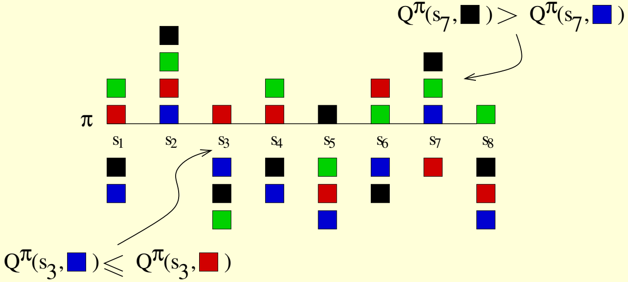
Policy Improvement



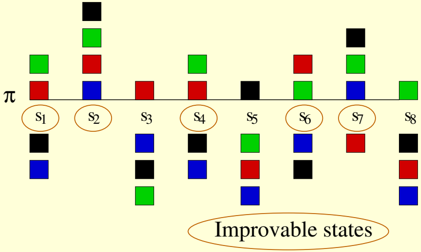
Policy Improvement



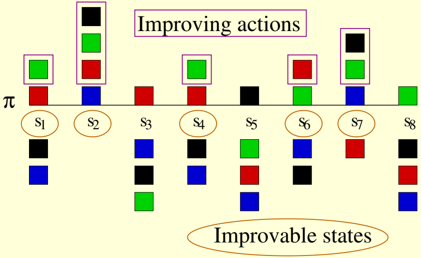
Policy Improvement



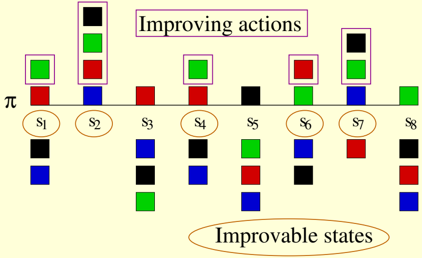
Policy Improvement



Policy Improvement



Policy Improvement



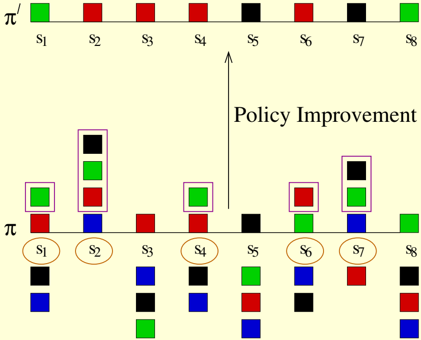
Given π,

Pick one or more improvable states, and in them,

Switch to an arbitrary improving action.

Let the resulting policy be π'.

Policy Improvement



Given π,

Pick one or more improvable states, and in them,

Switch to an arbitrary improving action.

Let the resulting policy be π'.

Policy Improvement Theorem

- For $\pi \in \Pi, s \in S$,

$$\mathbf{IA}(\pi, s) \stackrel{\text{def}}{=} \{a \in A : Q^{\pi}(s, a) > V^{\pi}(s)\}.$$

Policy Improvement Theorem

- For $\pi \in \Pi, s \in S$,

$$\mathbf{IA}(\pi, s) \stackrel{\text{def}}{=} \{a \in A : Q^{\pi}(s, a) > V^{\pi}(s)\}.$$

- For $\pi \in \Pi$,

$$\mathbf{IS}(\pi) \stackrel{\text{def}}{=} \{s \in S : |\mathbf{IA}(\pi, s)| \geq 1\}.$$

Policy Improvement Theorem

- For $\pi \in \Pi, s \in S$,

$$\mathbf{IA}(\pi, s) \stackrel{\text{def}}{=} \{a \in A : Q^{\pi}(s, a) > V^{\pi}(s)\}.$$

- For $\pi \in \Pi$,

$$\mathbf{IS}(\pi) \stackrel{\text{def}}{=} \{s \in S : |\mathbf{IA}(\pi, s)| \geq 1\}.$$

- Suppose $\mathbf{IS}(\pi) \neq \emptyset$ and $\pi' \in \Pi$ is obtained by policy improvement on π. Thus, π' satisfies

$$\forall s \in S : \pi'(s) = \pi(s) \text{ or } \pi'(s) \in \mathbf{IA}(\pi, s), \text{ and}$$
$$\exists s \in S : \pi'(s) \in \mathbf{IA}(\pi, s).$$

Policy Improvement Theorem

- For $\pi \in \Pi, s \in S$,

$$\mathbf{IA}(\pi, s) \stackrel{\text{def}}{=} \{a \in A : Q^\pi(s, a) > V^\pi(s)\}.$$

- For $\pi \in \Pi$,

$$\mathbf{IS}(\pi) \stackrel{\text{def}}{=} \{s \in S : |\mathbf{IA}(\pi, s)| \geq 1\}.$$

- Suppose $\mathbf{IS}(\pi) \neq \emptyset$ and $\pi' \in \Pi$ is obtained by policy improvement on π. Thus, π' satisfies

$$\forall s \in S : \pi'(s) = \pi(s) \text{ or } \pi'(s) \in \mathbf{IA}(\pi, s), \text{ and}$$
$$\exists s \in S : \pi'(s) \in \mathbf{IA}(\pi, s).$$

Policy Improvement Theorem:
(1) If $\mathbf{IS}(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If **IS**$(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If $\mathbf{IS}(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

- If $\pi \in \Pi$ is such that $\mathbf{IS}(\pi) \neq \emptyset$, then there exists $\pi' \in \Pi$ such that $\pi' \succ \pi$.

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If $\textbf{IS}(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

- If $\pi \in \Pi$ is such that $\textbf{IS}(\pi) \neq \emptyset$, then there exists $\pi' \in \Pi$ such that $\pi' \succ \pi$.
- But Π has a finite number of policies (k^n).

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If $\mathbf{IS}(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

- If $\pi \in \Pi$ is such that $\mathbf{IS}(\pi) \neq \emptyset$, then there exists $\pi' \in \Pi$ such that $\pi' \succ \pi$.
- But Π has a finite number of policies (k^n).
- Hence, there must exist a policy $\pi^\star \in \Pi$ such that $\mathbf{IS}(\pi^\star) = \emptyset$.

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If **IS**$(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

- If $\pi \in \Pi$ is such that **IS**$(\pi) \neq \emptyset$, then there exists $\pi' \in \Pi$ such that $\pi' \succ \pi$.
- But Π has a finite number of policies (k^n).
- Hence, there must exist a policy $\pi^\star \in \Pi$ such that **IS**$(\pi^\star) = \emptyset$.
- The theorem itself also tells us that $\pi^\star$ must be optimal.

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If $\mathbf{IS}(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

- If $\pi \in \Pi$ is such that $\mathbf{IS}(\pi) \neq \emptyset$, then there exists $\pi' \in \Pi$ such that $\pi' \succ \pi$.
- But Π has a finite number of policies (k^n).
- Hence, there must exist a policy $\pi^\star \in \Pi$ such that $\mathbf{IS}(\pi^\star) = \emptyset$.
- The theorem itself also tells us that $\pi^\star$ must be optimal.
- Observe that $\mathbf{IS}(\pi^\star) = \emptyset \iff B^\star(V^{\pi^\star}) = V^{\pi^\star}$.

Implication of Policy Improvement Theorem

Policy Improvement Theorem:
(1) If $\mathbf{IS}(\pi) = \emptyset$, then π is optimal, else
(2) if π' is obtained by policy improvement on π, then $\pi' \succ \pi$.

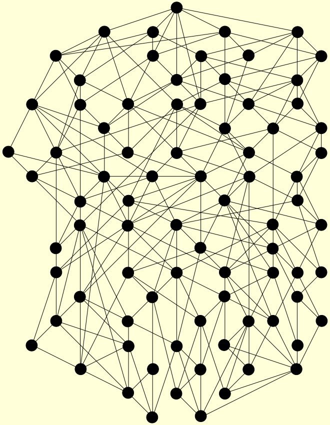
- If $\pi \in \Pi$ is such that $\mathbf{IS}(\pi) \neq \emptyset$, then there exists $\pi' \in \Pi$ such that $\pi' \succ \pi$.
- But Π has a finite number of policies (k^n).
- Hence, there must exist a policy $\pi^\star \in \Pi$ such that $\mathbf{IS}(\pi^\star) = \emptyset$.
- The theorem itself also tells us that $\pi^\star$ must be optimal.
- Observe that $\mathbf{IS}(\pi^\star) = \emptyset \iff B^\star(V^{\pi^\star}) = V^{\pi^\star}$.
- In other words, $V^{\pi^\star}$ satisfies the Bellman optimality equations—which we know has a unique solution. It is a convention to denote $V^{\pi^\star} = V^\star$.

Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
    $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.

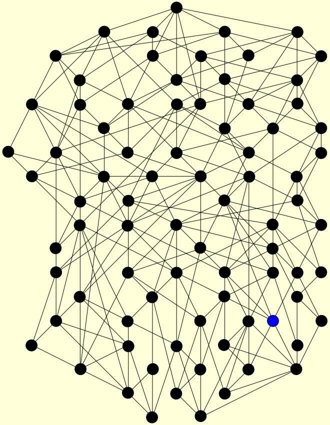
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
    $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



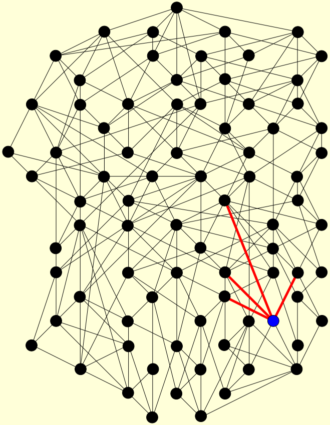
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



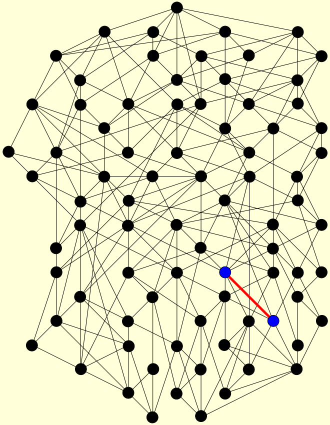
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
    $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



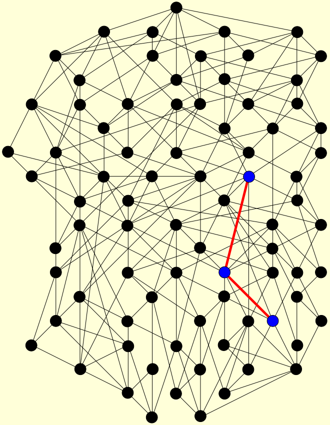
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



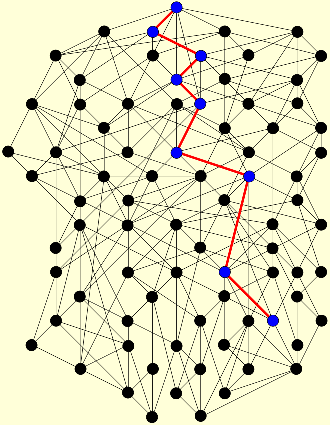
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
$\quad \pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



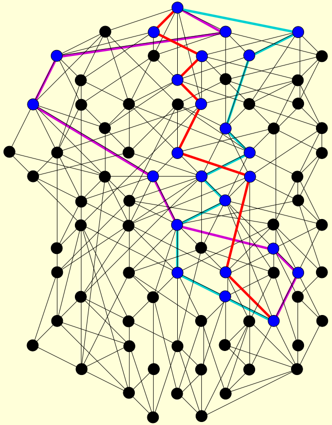
Policy Iteration Algorithm

$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
    $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



Policy Iteration Algorithm

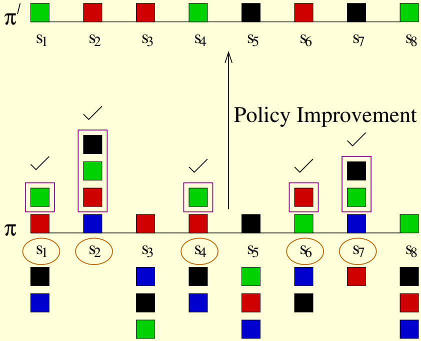
$\pi \leftarrow$ Arbitrary policy.
While π has improvable states:
  $\pi' \leftarrow$ PolicyImprovement(π); $\pi \leftarrow \pi'$.
Return π.



Path taken (and hence the number of iterations) in general depends on the switching strategy.

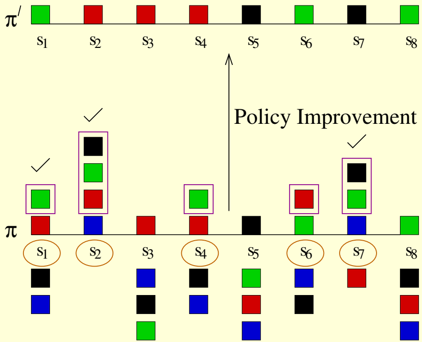
Howard's Policy Iteration

- Reference: Howard (1960).
- Greedy; switch all improvable states.



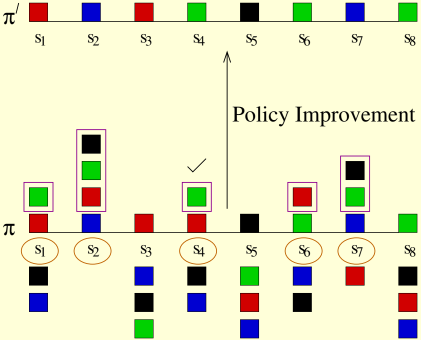
Random Policy Iteration

- Reference: Mansour and Singh (1999).
- Switch a non-empty subset of improvable states chosen uniformly at random.



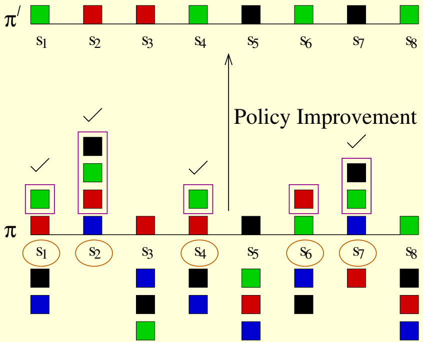
Random Policy Iteration

- Reference: Mansour and Singh (1999).
- Switch a non-empty subset of improvable states chosen uniformly at random.



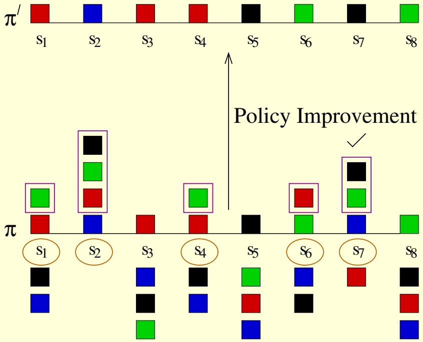
Random Policy Iteration

- Reference: Mansour and Singh (1999).
- Switch a non-empty subset of improvable states chosen uniformly at random.



Simple Policy Iteration

- Reference: Melekopoglou and Condon (1994).
- Assume a fixed indexing of states.
- Switch the improvable state with the highest index.



Markov Decision Problems

1. Policy Iteration
   - Policy evaluation (review)
   - Policy improvement
   - Algorithm and variants
2. Proof of Policy Improvement Theorem
   - Bellman operator
   - Proof
3. Computational complexity of MDP planning
4. Summary

Bellman Operator B^π

- For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows: For $X : S \to \mathbb{R}$ and for $s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s') \right).$$

Bellman Operator B^π

- For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows:
  For $X : S \to \mathbb{R}$ and for $s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s')\right).$$

- One Bellman operator for each $\pi \in \Pi$. No "max" like $B^\star$.

Bellman Operator B^π

- For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows: For $X : S \to \mathbb{R}$ and for $s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s') \right).$$

- One Bellman operator for each $\pi \in \Pi$. No "max" like $B^\star$.

- Some facts about B^π for all $\pi \in \Pi$.
  - B^π is a contraction mapping with contraction factor γ.
  - For $X : S \to \mathbb{R}$: $\lim_{l \to \infty} (B^\pi)^l(X) = V^\pi$.
  - For $X : S \to \mathbb{R}$, $Y : S \to \mathbb{R}$: $X \succeq Y \implies B^\pi(X) \succeq B^\pi(Y)$.

Bellman Operator B^π

- For $\pi \in \Pi$, we define $B^\pi : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ as follows:
  For $X : S \to \mathbb{R}$ and for $s \in S$,

$$(B^\pi(X))(s) \stackrel{\text{def}}{=} \sum_{s' \in S} T(s, \pi(s), s') \left(R(s, \pi(s), s') + \gamma X(s')\right).$$

- One Bellman operator for each $\pi \in \Pi$. No "max" like $B^\star$.

- Some facts about B^π for all $\pi \in \Pi$.
  - B^π is a contraction mapping with contraction factor γ.
  - For $X : S \to \mathbb{R}$: $\lim_{l \to \infty} (B^\pi)^l(X) = V^\pi$.
  - For $X : S \to \mathbb{R}$, $Y : S \to \mathbb{R}$: $X \succeq Y \implies B^\pi(X) \succeq B^\pi(Y)$.

- Observe that for $\pi, \pi' \in \Pi, \forall s \in S$:

$$B^{\pi'}(V^\pi)(s) = Q^\pi(s, \pi'(s)).$$

Proof of Policy Improvement Theorem

Proof of Policy Improvement Theorem

$\textbf{IS}(\pi) = \emptyset$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^{2}(V^{\pi})$

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$\implies \forall \pi' \in \Pi : V^\pi \succeq B^{\pi'}(V^\pi)$

$\implies \forall \pi' \in \Pi : V^\pi \succeq B^{\pi'}(V^\pi) \succeq (B^{\pi'})^2(V^\pi)$

$\implies \forall \pi' \in \Pi : V^\pi \succeq B^{\pi'}(V^\pi) \succeq (B^{\pi'})^2(V^\pi) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^\pi)$

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$

$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}$.

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

$\mathbf{IS}(\pi) \neq \emptyset$ and policy improvement on π yields π'

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty}(B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

$\mathbf{IS}(\pi) \neq \emptyset$ and policy improvement on π yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$
$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$
$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$
$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

$\mathbf{IS}(\pi) \neq \emptyset$ and policy improvement on π yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$
$$\implies (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

Proof of Policy Improvement Theorem

$\mathbf{IS}(\pi) = \emptyset$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi})$$

$$\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.$$

$\mathbf{IS}(\pi) \neq \emptyset$ and policy improvement on π yields π'

$$\implies B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

$$\implies \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi}) \succeq \cdots \succeq (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi}$$

Proof of Policy Improvement Theorem

$$
\begin{aligned}
&\mathbf{IS}(\pi) = \emptyset \\
&\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \\
&\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \\
&\implies \forall \pi' \in \Pi : V^{\pi} \succeq B^{\pi'}(V^{\pi}) \succeq (B^{\pi'})^2(V^{\pi}) \succeq \cdots \succeq \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi}) \\
&\implies \forall \pi' \in \Pi : V^{\pi} \succeq V^{\pi'}.
\end{aligned}
$$

$\mathbf{IS}(\pi) \neq \emptyset$ and policy improvement on π yields π'

$$
\begin{aligned}
&\implies B^{\pi'}(V^{\pi}) \succ V^{\pi} \\
&\implies (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi} \\
&\implies \lim_{l \to \infty} (B^{\pi'})^l(V^{\pi}) \succeq \cdots \succeq (B^{\pi'})^2(V^{\pi}) \succeq B^{\pi'}(V^{\pi}) \succ V^{\pi} \\
&\implies V^{\pi'} \succ V^{\pi}.
\end{aligned}
$$

A More General Class of Policies

- In principle, an agent can follow a policy λ that maps every possible history $s^0, a^0, r^0, s^1, a^1, r^1, \ldots, s^t$ for $t \geq 0$ to a probability distribution over A.
- Let Λ be the set of such policies λ (which are in general non-Markovian, non-stationary, and stochastic).

A More General Class of Policies

- In principle, an agent can follow a policy λ that maps every possible history $s^0, a^0, r^0, s^1, a^1, r^1, \dots, s^t$ for $t \geq 0$ to a probability distribution over A.
- Let Λ be the set of such policies λ (which are in general non-Markovian, non-stationary, and stochastic).

- Recall that we only considered Π, the set of all policies $\pi : S \to A$ (which are Markovian, stationary, and deterministic). Observe that $\Pi \subset \Lambda$.
- We have shown that there exists $\pi^\star \in \Pi$ such that for all $\pi \in \Pi$, $\pi^\star \succeq \pi$.

A More General Class of Policies

- In principle, an agent can follow a policy λ that maps every possible history $s^0, a^0, r^0, s^1, a^1, r^1, \dots, s^t$ for $t \geq 0$ to a probability distribution over A.
- Let Λ be the set of such policies λ (which are in general non-Markovian, non-stationary, and stochastic).

- Recall that we only considered Π, the set of all policies $\pi : S \to A$ (which are Markovian, stationary, and deterministic). Observe that $\Pi \subset \Lambda$.
- We have shown that there exists $\pi^\star \in \Pi$ such that for all $\pi \in \Pi$, $\pi^\star \succeq \pi$.

Could there exist $\lambda \in \Lambda \setminus \Pi$ such that $\neg(\pi^\star \succeq \lambda)$?

A More General Class of Policies

- In principle, an agent can follow a policy λ that maps every possible history $s^0, a^0, r^0, s^1, a^1, r^1, \dots, s^t$ for $t \geq 0$ to a probability distribution over A.
- Let Λ be the set of such policies λ (which are in general non-Markovian, non-stationary, and stochastic).
- Recall that we only considered Π, the set of all policies $\pi : S \to A$ (which are Markovian, stationary, and deterministic). Observe that $\Pi \subset \Lambda$.
- We have shown that there exists $\pi^\star \in \Pi$ such that for all $\pi \in \Pi$, $\pi^\star \succeq \pi$.

Could there exist $\lambda \in \Lambda \setminus \Pi$ such that $\neg(\pi^\star \succeq \lambda)$? No.

History and Stochasticity

- In MDPs, the agent can sense state, and the consequence of each action depends solely on state.

History and Stochasticity

- In MDPs, the agent can sense state, and the consequence of each action depends solely on state.
- Moreover, we are maximising an infinite sum of expected discounted rewards—the challenge at each time step is the same: to maximise the expected long-term reward starting from the current state!

History and Stochasticity

- In MDPs, the agent can sense state, and the consequence of each action depends solely on state.
- Moreover, we are maximising an infinite sum of expected discounted rewards—the challenge at each time step is the same: to maximise the expected long-term reward starting from the current state!
- History and stochasticity can help if the agent is unable to sense state perfectly. Such a situation arises in an abstraction called the Partially Observable MDP (POMDP).

History and Stochasticity

- In MDPs, the agent can sense state, and the consequence of each action depends solely on state.
- Moreover, we are maximising an infinite sum of expected discounted rewards—the challenge at each time step is the same: to maximise the expected long-term reward starting from the current state!

- History and stochasticity can help if the agent is unable to sense state perfectly. Such a situation arises in an abstraction called the Partially Observable MDP (POMDP).
- Optimal policies for the finite horizon reward setting are in general non-stationary (time-dependent).

History and Stochasticity

- In MDPs, the agent can sense state, and the consequence of each action depends solely on state.
- Moreover, we are maximising an infinite sum of expected discounted rewards—the challenge at each time step is the same: to maximise the expected long-term reward starting from the current state!

- History and stochasticity can help if the agent is unable to sense state perfectly. Such a situation arises in an abstraction called the Partially Observable MDP (POMDP).
- Optimal policies for the finite horizon reward setting are in general non-stationary (time-dependent).
- Optimal policies ("strategies") in many types of multi-player games are in general stochastic ("mixed").

Markov Decision Problems

1. Policy Iteration
   - Policy evaluation (review)
   - Policy improvement
   - Algorithm and variants
2. Proof of Policy Improvement Theorem
   - Bellman operator
   - Proof
3. Computational complexity of MDP planning
4. Summary

Running-time Bounds

(Full references: see Kalyanakrishnan, Misra, Gopalan, 2016.)

Running-time Bounds

(Full references: see Kalyanakrishnan, Misra, Gopalan, 2016.)

- **Value Iteration**: $V_0 \to V_1 \to \cdots \to V_\infty = V^{\pi^\star}$.
  Upper Bound: poly$(n, k, B, \frac{1}{1-\gamma})$. iterations [LKM95]; B is the number of bits used to represent the MDP.

Running-time Bounds

(Full references: see Kalyanakrishnan, Misra, Gopalan, 2016.)

- **Value Iteration**: $V_0 \to V_1 \to \cdots \to V_\infty = V^{\pi^\star}$.
  Upper Bound: $\text{poly}(n, k, B, \frac{1}{1-\gamma})$. iterations [LKM95]; B is the number of bits used to represent the MDP.
- **Linear Programming**: With (n variables, nk constraints) or (nk variables, n constraints).
  $\text{poly}(n, k, B)$ [K80, K84].
  $\text{poly}(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].

Running-time Bounds

(Full references: see Kalyanakrishnan, Misra, Gopalan, 2016.)

- **Value Iteration**: $V_0 \to V_1 \to \cdots \to V_\infty = V^{\pi^\star}$.
  Upper Bound: $\text{poly}(n, k, B, \frac{1}{1-\gamma})$. iterations [LKM95]; B is the number of bits used to represent the MDP.

- **Linear Programming**: With (n variables, nk constraints) or (nk variables, n constraints).
  $\text{poly}(n, k, B)$ [K80, K84].
  $\text{poly}(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].

- Strong bounds are those that depend solely on n and k.

Running-time Bounds

(Full references: see Kalyanakrishnan, Misra, Gopalan, 2016.)

- **Value Iteration**: $V_0 \to V_1 \to \cdots \to V_\infty = V^{\pi^\star}$.
  Upper Bound: $\text{poly}(n, k, B, \frac{1}{1-\gamma})$. iterations [LKM95]; B is the number of bits used to represent the MDP.
- **Linear Programming**: With (n variables, nk constraints) or (nk variables, n constraints).
  $\text{poly}(n, k, B)$ [K80, K84].
  $\text{poly}(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ (Expected) [MSW96].
- Strong bounds are those that depend solely on n and k.
- **Policy Iteration** naturally yields strong bounds (also enjoys good weak bounds [P94]). We review strong bounds.

PI: Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's (Greedy) PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Random PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$

PI: Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's (Greedy) PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Random PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$

Lower bounds on number of iterations

$\Omega(n)$ Howard's PI on n-state, 2-action MDPs [HZ10].

PI: Switching Strategies and Bounds

Upper bounds on number of iterations

PI Variant	**Type**	$k = 2$	**General** k
Howard's (Greedy) PI [H60, MS99]	Deterministic	$O\left(\frac{2^n}{n}\right)$	$O\left(\frac{k^n}{n}\right)$
Mansour and Singh's Random PI [MS99]	Randomised	1.7172^n	$\approx O\left(\frac{k}{2}\right)^n$

Lower bounds on number of iterations

$\Omega(n)$ Howard's PI on n-state, 2-action MDPs [HZ10].

$\Omega(2^n)$ Simple PI on n-state, 2-action MDPs [MC94].

PI: Some Recent Results ($k = 2$)

(Authors with names underlined once took CS 747!)

PI: Some Recent Results ($k = 2$)

(Authors with names underlined once took CS 747!)

- Kalyanakrishnan, Mall, and Goyal (2016) devise the Batch-switching PI algorithm (deterministic), and show an upper bound of 1.6479^n iterations.

- Taraviya and Kalyanakrishnan (2019) use a similar approach to provide an upper bound of 1.6001^n iterations (in expectation) for a randomised PI variant.

PI: Some Recent Results ($k \geq 2$)

- Gupta and Kalyanakrishnan (2017) give a deterministic PI variant with an upper bound of $k^{0.7207n}$ iterations, improved by Taraviya and Kalyanakrishnan to $k^{0.7019}$ iterations.

- Kalyanakrishnan, Misra, and Gopalan (2016) show an upper bound of $(2 + \ln(k-1))^n$ iterations for a randomised PI variant.

- Taraviya and Kalyanakrishnan (2019) show an upper bound of $(\sqrt{k \log(k)})^n$ iterations for a randomised variant of Howard's PI.

- Ashutosh, Consul, Dedhia, Khirwadkar, Shah, and Kalyanakrishnan (2020) show a lower bound of $\sqrt{k}^n$ iterations for a particular deterministic variant of PI.

Open Problems

- Is the complexity of Howard's PI on 2-action MDPs upper-bounded by the Fibonacci sequence ($\approx 1.6181^n$)?
- Is Howard's PI the most efficient among deterministic PI algorithms (worst case over all MDPs)?
- Is there a super-linear lower bound on the number of iterations taken by Howard's PI on 2-action MDPs?
- Is Howard's PI strongly polynomial on deterministic MDPs?
- Is there a variant of PI that can visit all k^n policies in some n-state, k-action MDP—implying an $\Omega(k^n)$ lower bound?
- Is there a strongly polynomial algorithm for MDP planning?

Markov Decision Problems

1. Policy Iteration
   - Policy evaluation (review)
   - Policy improvement
   - Algorithm and variants
2. Proof of Policy Improvement Theorem
   - Bellman operator
   - Proof
3. Computational complexity of MDP planning
4. Summary

Summary of MDP Planning

- MDPs are an abstraction of sequential decision making.
- Many applications; many different formulations.
- Essential solution concept: optimal policy (known to exist).

- Three main families of planning algorithms: Value Iteration, Linear Programming, Policy Iteration.
- Have strengths and weaknesses in theory and in practice.
- Can combine (especially Value Iteration, Policy Iteration).

- We showed correctness of all three methods.
- Used Banach's fixed-point theorem, Bellman optimality operator, Bellman operator.

- What if T, R were not given, but have to be *learned* from interaction? Can we still learn to act optimally?
- Yes: that's the Reinforcement Learning problem. Next week!