CSE 250A. Principles of AI

Probabilistic Reasoning and Decision-Making

Lecture 18 – Policy and value iteration

Lawrence Saul
Department of Computer Science and Engineering
University of California, San Diego

Fall 2020

Outline

Markov decision processes

- **MDP** $= \{\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s)\}$
- **A policy** $\pi : \mathcal{S} \to \mathcal{A}$ **maps states to actions**
- **State and action value functions**

$$\begin{aligned} V^{\pi}(s) &= \mathrm{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t) \,\middle|\, s_0 = s\right] \\ Q^{\pi}(s,a) &= \mathrm{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t) \,\middle|\, s_0 = s, a_0 = a\right] \end{aligned}$$

- **Optimality**

  There exists at least one policy π^* such that $V^{\pi^*}(s) \geq V^{\pi}(s)$ for all policies π and states s.

Policy evaluation

- **How to compute the state value function?**

$$V^\pi(s) \;=\; \mathrm{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s\right]$$

- **Bellman equation:**

$$V^\pi(s) \;=\; R(s) + \gamma \sum_{s'} P(s'|s,\pi(s))\, V^\pi(s')$$

- **Solve linear system:**

$$\underbrace{\begin{bmatrix} R \end{bmatrix}}_{\substack{n \times 1 \\ \text{vector}}} \;=\; \underbrace{\begin{bmatrix} I - \gamma P^\pi \end{bmatrix}}_{\substack{n \times n \\ \text{matrix}}} \underbrace{\begin{bmatrix} V^\pi \end{bmatrix}}_{\substack{n \times 1 \\ \text{vector}}}$$

Policy improvement

- **Greedy policy:**

$$\pi'(s) \quad = \quad \operatorname*{argmax}_a Q^\pi(s, a)$$

- **Theorem:**

$$V^{\pi'}(s) \ \geq V^\pi(s) \quad \text{for all states } s \in \mathcal{S}$$

- **Intuition:**

  If it's better to choose action a in state s before following π, then it's always better to make this choice.

- **Proof:** later today.

Policy iteration

- **How to compute π^*?**

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s,a) \end{matrix} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} \cdots$$

  This process is guaranteed to terminate.
  But does it converge to an optimal policy?

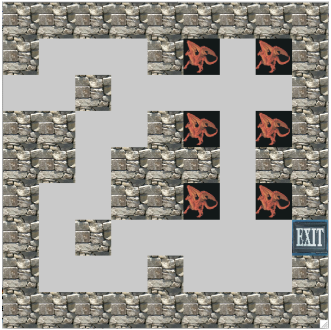
- **Theorem:**

  If $\pi'(s) = \arg\max_a Q^\pi(s,a)$ and $V^{\pi'}(s) = V^\pi(s)$ for all $s \in \mathcal{S}$,
  then $V^\pi(s) = V^*(s)$ for all $s \in \mathcal{S}$.

- **Proof:** later today.

Demo — policy iteration

How to exit the maze with high probability?



$\mathcal{S}$	location in maze
$\mathcal{A}$	$\{\uparrow, \leftarrow, \downarrow, \rightarrow\}$
$P(s' \| s, a)$	move *with some probability* in direction of arrow
$R(s)$	+1 (exit), -1 (dragon), 0 (otherwise)
γ	0.99 (close to one)

Demo — no uncertainty

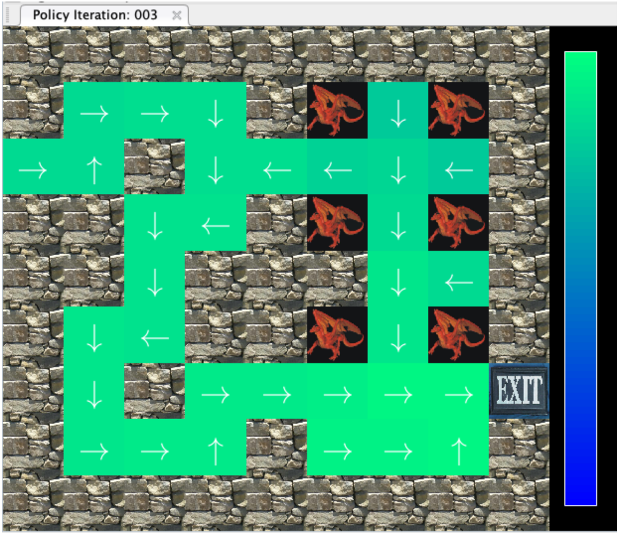
Agent moves **deterministically** in the direction of the action.

Converges in 14 iterations.

Demo — low uncertainty

Agent moves with **low probability** in unintended direction.

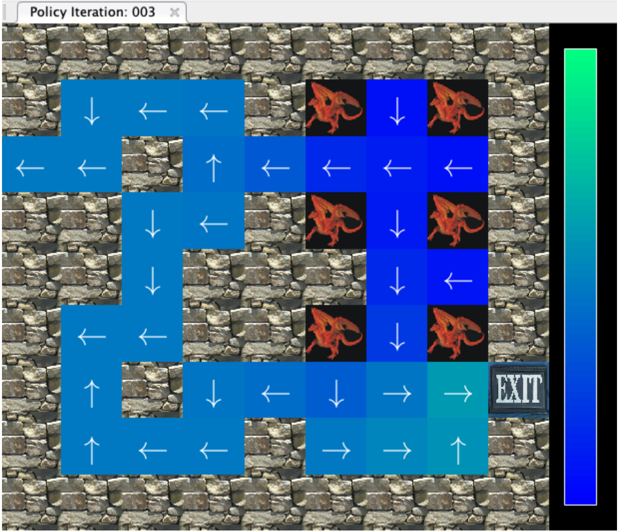
Converges in 3 iterations.



Demo — high uncertainty

Agent moves with **high probability** in unintended direction.

Converges in 3 iterations.



Outline

1. **Review**
2. **Policy improvement**
3. **Policy iteration**
4. **Value iteration**

Policy improvement theorem

- **Theorem**

  The greedy policy $\pi'(s) = \arg\max_a Q^\pi(s, a)$ improves everywhere on the policy π from which it was derived:

  $$V^{\pi'}(s) \;\geq\; V^\pi(s) \quad \text{for all states} \quad s \in \mathcal{S}$$

- **Intuition**

  If it's better to choose action a in state s before following π, then it's always better to make this choice.

- **Proof idea**

  We'll prove a key inequality for *one-step deviations* from π, then we'll extend this inequality by an iterative argument.

Proof — 1. Deriving the inequality

- **Comparing value functions:**

$$\begin{aligned} V^\pi(s) &= Q^\pi(s, \pi(s)) \\ &\leq \max_a Q^\pi(s, a) \\ &= Q^\pi(s, \pi'(s)) \\ &= R(s) + \gamma \sum_{s'} P(s'|s, \pi'(s)) V^\pi(s') \end{aligned}$$

- **Combining these steps:**

$$V^\pi(s) \leq R(s) + \gamma \sum_{s'} P(s'|s, \pi'(s)) V^\pi(s')$$

- **Intuition:**

  It is better to take one step under π', then revert to π, than to always follow π.

Proof — 2. Leveraging the inequality

- **One-step inequality:**

$$V^\pi(s) \;\leq\; R(s) + \gamma \sum_{s'} P(s'|s,\pi'(s))\, V^\pi(s')$$

What happens if we plug this inequality into itself?
Then we obtain ...

- **Two-step inequality:**

$$V^\pi(s) \;\leq\; R(s) + \gamma \sum_{s'} P(s'|s,\pi'(s)) \left[R(s') + \gamma \sum_{s''} P(s''|s',\pi'(s'))\, V^\pi(s'') \right]$$

- **Intuition:**

It is better to take **two** steps under π', then revert to π,
than to always follow π.

Proof — 3. Taking the limit

- **Two-step inequality:**

$$V^{\pi}(s) \;\leq\; R(s) + \gamma \sum_{s'} P(s'|s,\pi'(s)) \left[R(s') \;+\; \gamma \sum_{s''} P(s''|s',\pi'(s')) V^{\pi}(s'') \right]$$

- **Apply the inequality t times:**

  It is better to take t steps under π', then revert to π, than to always follow π. Last term is of order $O(\gamma^t)$.

- **Take the limit $t \rightarrow \infty$:**

  It is better to follow π' (always) than to follow π (always). Conclude that $V^{\pi}(s) \leq V^{\pi'}(s)$ for all states $s \in \mathcal{S}$.

Outline

1. **Review**
2. **Policy improvement**
3. **Policy iteration**
4. **Value iteration**

Policy iteration

- **How to compute π^*?**

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s,a) \end{matrix} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} \cdots$$

This process is guaranteed to terminate.
But does it converge to an optimal policy?

- **Theorem**

If $\pi'(s) = \arg\max_a Q^\pi(s,a)$ and $V^{\pi'}(s) = V^\pi(s)$ for all $s \in \mathcal{S}$, then $V^\pi(s) = V^*(s)$ for all $s \in \mathcal{S}$.

- **Proof idea**

Prove a key equality/inequality for terminal/non-terminal policies; iterate t times, then compare the limits as $t \to \infty$.

Proof — 1. Bellman optimality equation

- **Suppose policy iteration converges to π'.**

$$V^{\pi'}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi'(s)) V^{\pi'}(s') \qquad \boxed{\textbf{Bellman equation}}$$

$$V^{\pi}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi'(s)) V^{\pi}(s') \qquad \boxed{\textbf{at convergence}}$$

Now exploit that π' is greedy with respect to π ...

- **Bellman optimality equation**

$$\boxed{V^{\pi}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) V^{\pi}(s')}$$

These equations are nonlinear due to the max operation.
There are n equations for n unknowns (where $s = 1, 2, \ldots, n$).

Proof — 2. Inequality

- **Let $\tilde{\pi}$ be any policy of the MDP:**

$$
\begin{aligned}
V^{\tilde{\pi}}(s) &= R(s) + \gamma \sum_{s'} P(s'|s,\tilde{\pi}(s))V^{\tilde{\pi}}(s') && \boxed{\textbf{Bellman equation}} \\
V^{\tilde{\pi}}(s) &\leq R(s) + \gamma \max_a \sum_{s'} P(s'|s,a)V^{\tilde{\pi}}(s') && \boxed{\textbf{greedy}}
\end{aligned}
$$

- **Compare to Bellman optimality equation (BOE):**

$$
V^{\pi}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a))V^{\pi}(s')
$$

- **Understanding the difference:**

  The inequality holds for any policy $\tilde{\pi}$ of the MDP.
  The **BOE** only holds for a solution π from policy iteration.

Proof — 3. Taking the limit

- **Iterating the inequality:**

$$\begin{aligned} V^{\tilde{\pi}}(s) &\leq R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^{\tilde{\pi}}(s') \\ &\leq R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) \left[R(s') + \gamma \max_{a'} \sum_{s''} P(s''|s',a') V^{\tilde{\pi}}(s'') \right] \end{aligned}$$

- **Iterating the BOE:**

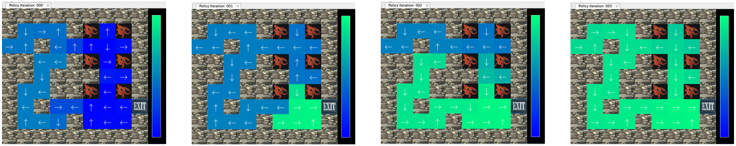
$$\begin{aligned} V^{\pi}(s) &= R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^{\pi}(s') \\ &= R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) \left[R(s') + \gamma \max_{a'} \sum_{s''} P(s''|s',a') V^{\pi}(s'') \right] \end{aligned}$$

- **Iterating t times:**

Both right sides agree up to term of order γ^t.
Taking the limit $t \to \infty$, we find $V^{\tilde{\pi}}(s) \leq V^{\pi}(s)$ for all $s \in \mathcal{S}$.

$\boxed{\textbf{Since } \tilde{\pi} \textbf{ is arbitrary, we conclude that } \pi \textbf{ is optimal}}$.

Policy iteration in practice



- **Pros and cons**

  [+] Policy iteration converges quickly in practice.
  [−] But each iteration costs $O(n^2)$ for policy evaluation.

- **Speedups for policy evaluation**

  HW 9.5 describes a fast iterative approximation.
  Sparse matrices $P(s'|s,\pi(s))$ may simplify policy evaluation.

Outline

1. Review
2. Policy improvement
3. Policy iteration
4. $\boxed{\text{Value iteration}}$

Motivation

- **How policy iteration works:**

  It searches directly (and quite efficiently) through the combinatorially large space of policies in the MDP.

- **Is there another way?**

  Given an MDP $= \{\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s), \gamma\}$, recall how its optimal policies and value functions are connected:

$$
\begin{aligned}
\pi^*(s) &= \operatorname*{argmax}_a \Big[Q^*(s,a) \Big] \\
&= \operatorname*{argmax}_a \Big[R(s) + \gamma \sum_{s'} P(s'|s,a)\, V^*(s') \Big]
\end{aligned}
$$

  So if we can directly compute the optimal value function $V^*(s)$, then we can use it to derive an optimal policy π^*.

Bellman optimality equation

- **Derivation:**

$$\begin{aligned} V^*(s) &= \max_a \Big[Q^*(s,a) \Big] \\ &= \max_a \Big[R(s) + \gamma \sum_{s'} P(s'|s,a)\, V^*(s') \Big] \end{aligned}$$

- **Solution?**

  Suppose we know the parameters $\{R(s), P(s'|s,a), \gamma\}$. Then the above gives us n equations for n unknowns:

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

  But how to solve these **nonlinear** equations for $V^*(s)$?

Value iteration

- **Idea in a nutshell**

  Replace the equality sign in the Bellman optimality equation by an assignment operation:

$$V^*(s) \quad = \quad \max_a \left[R(s) + \gamma \sum_{s'} P(s'|s,a)\, V^*(s') \right] \quad \boxed{\textbf{BOE}}$$

$$V_{\text{new}}(s) \quad \longleftarrow \quad \max_a \left[R(s) + \gamma \sum_{s'} P(s'|s,a)\, V_{\text{old}}(s') \right] \quad \boxed{\textbf{algorithm}}$$

- **Why this might work**

  The value function $V^*(s)$ is a *fixed point* of this iteration. But does this iteration always converge to a valid solution?

Algorithm for value iteration

1. **Initialize:** $V_o(s) = 0$ for all $s \in \mathcal{S}$.

2. **Iterate until convergence:**

$$V_{k+1}(s) \;=\; \max_a \left[R(s) + \gamma \sum_{s'} P(s'|s,a)\, V_k(s') \right] \text{ for all } s \in \mathcal{S}.$$

3. **Solve for optimal policy**:

$$\begin{aligned} Q_k(s,a) &= R(s) + \gamma \sum_{s'} P(s'|s,a)\, V_k(s'), \\ \pi^*(s) &= \lim_{k\to\infty} \operatorname{argmax}_a Q_k(s,a). \end{aligned}$$

Value iteration (VI) versus policy iteration (PI)

- **Compare and contrast:**

  PI searches through the **combinatorial** space of policies.
  VI searches through the **continuous** space of value functions.

- **Convergence:**

  PI converges in a finite number of steps.
  VI converges asymptotically (in the limit).

- **Next lecture:**

  Proof of convergence for value iteration.
  More demos, effect of discount factor, etc.

Final exam update

Basic information:

- It will be a take-home, open-book exam.
- It is designed to take about one afternoon.
- Okay to check (but not do) work in Python, R, Matlab, Mathematica, etc.
- No collaboration is allowed.
- Questions will be broken down into simpler parts (like HW).
- Roughly speaking: $\frac{2}{3}$ straightforward, $\frac{1}{3}$ novel (but familiar).
- Solutions will be collected via Gradescope.
- For MS students in CSE: the final is the comprehensive exam.

Tentative plans (to be confirmed):

- Sun Dec 5 @ noon to Mon Dec 6 @ noon (PST)
- 10-12 questions with parts of varying difficulty
- 100 points total