facebook
Artificial Intelligence Research

How to solve an MDP: Dynamic Programming

Alessandro Lazaric
Facebook AI Research

Outline

The Optimization Problem

$$\max_{\pi} \; V^{\pi}(s_0) =$$

$$= \max_{\pi} \; \mathbb{E}\big[r(s_0, \pi(s_0)) + \gamma r(s_1, \pi(s_1)) + \gamma^2 r(s_2, \pi(s_2)) + \dots\big]$$

$$= \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | a_t = \pi(s_t)\right]$$

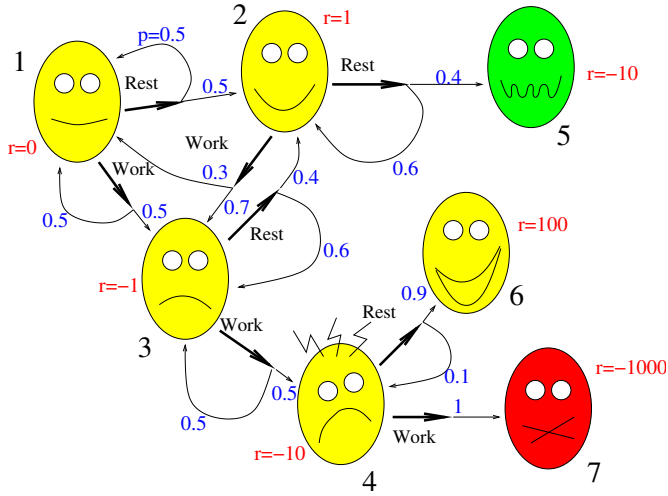
The Bellman Equation

Theorem

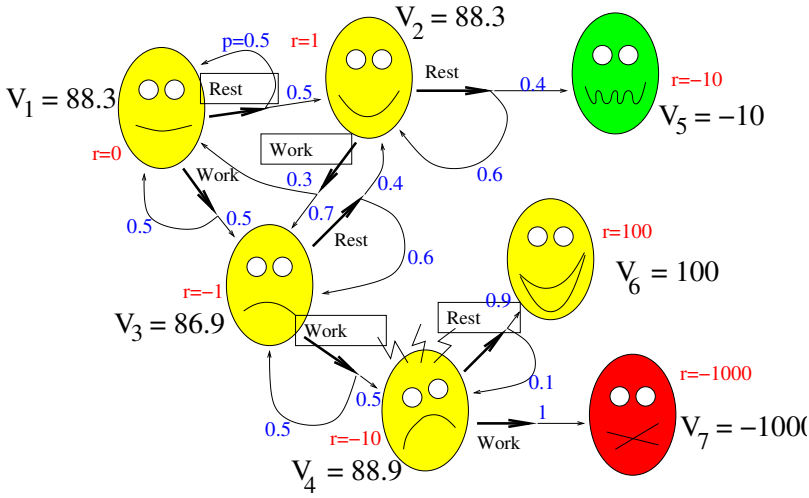
For any stationary deterministic policy $\pi = (d, d, \dots)$, at any state $s \in S$, the state value function satisfies the Bellman equation:

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{y} p(y|s, \pi(s)) V^{\pi}(y).$$

The student dilemma



The student dilemma

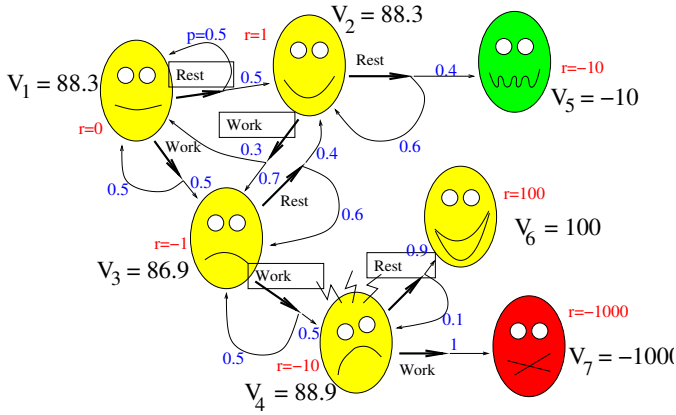


The student dilemma

Computing V_4:

$$V_6 = 100$$
$$V_4 = -10 + (0.9V_6 + 0.1V_4)$$

$$\Rightarrow V_4 = \frac{-10 + 0.9V_6}{0.9} = 88.8$$

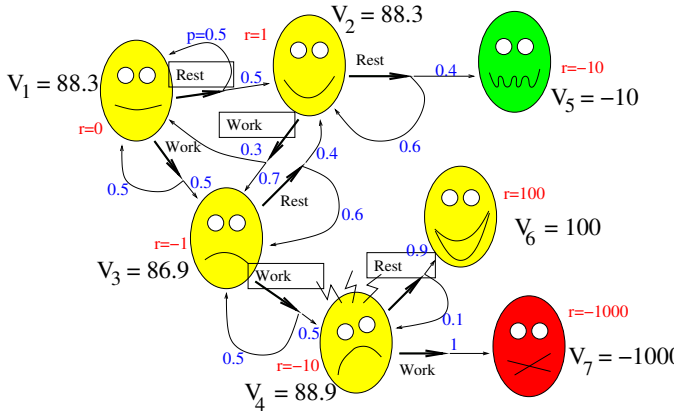


The student dilemma

Computing V_3: *no need* to consider all possible trajectories

$$V_4 = 88.8$$
$$V_3 = -1 + (0.5V_4 + 0.5V_3)$$

$$\Rightarrow V_3 = \frac{-1 + 0.5V_4}{0.5} = 86.8$$



Bellman Equation: a System of Equations

The Bellman equation

$$V^\pi(s) = r(s, \pi(s)) + \gamma \sum_y p(y|s, \pi(s)) V^\pi(y).$$

is a **linear** system of equations with $S = |\mathcal{S}|$ unknowns and S linear constraints.

Matrix notation

$$V^\pi \in \mathbb{R}^S, \quad r^\pi \in \mathbb{R}^S, \quad P^\pi \in \mathbb{R}^{S \times S}$$

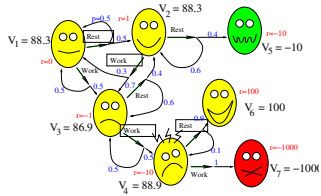
then

$$\begin{aligned} & V^\pi = r^\pi + \gamma P^\pi V^\pi \\ \implies \quad & V^\pi = (I - \gamma P^\pi)^{-1} r^\pi \end{aligned}$$

☞ V^π can be compute inverting a $S \times S$ matrix ($O(S^3)$ time)

The student dilemma

$$V^\pi(x) = r(x, \pi(x)) + \gamma \sum_y p(y|x, \pi(x)) V^\pi(y)$$



System of equations

$$\begin{cases} V_1 &= 0 + 0.5V_1 + 0.5V_2 \\ V_2 &= 1 + 0.3V_1 + 0.7V_3 \\ V_3 &= -1 + 0.5V_4 + 0.5V_3 \\ V_4 &= -10 + 0.9V_6 + 0.1V_4 \\ V_5 &= -10 \\ V_6 &= 100 \\ V_7 &= -1000 \end{cases} \Rightarrow$$

$$(V, R \in \mathbb{R}^7,\ P \in \mathbb{R}^{7\times 7})$$

$$V = R + PV$$

$$\Downarrow$$

$$V = (I - P)^{-1} R$$

The Optimal Bellman Equation

Theorem

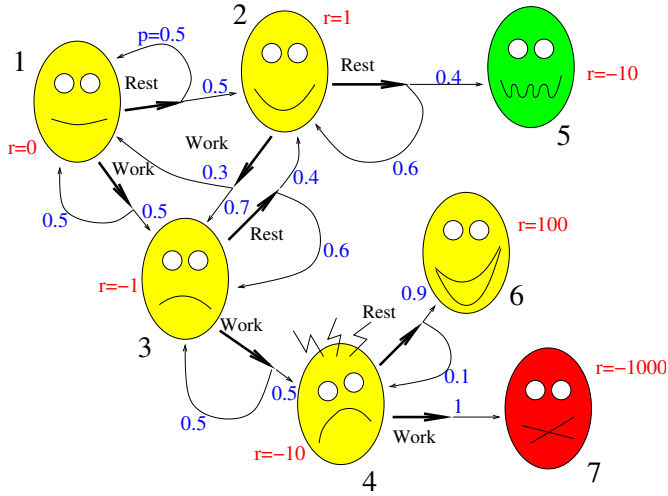
The optimal value function $V^\star$ (i.e., $V^\star = \max_\pi V^\pi$) is the solution to the optimal Bellman equation:

$$V^\star(s) = \max_{a \in A} \big[r(s,a) + \gamma \sum_{s'} p(s'|s,a) V^\star(s') \big].$$

and any optimal policy is such that

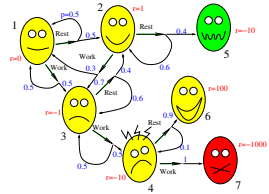
$$\pi^\star(s) \in \arg\max_{a' \in A} \big[r(s,a') + \gamma \sum_{s'} p(s'|s,a) V^\star(s') \big].$$

The Student Dilemma



The Student Dilemma

$$V^\star(x) = \max_{a \in A} \big[r(x,a) + \gamma \sum_y p(y|x,a) V^\star(y) \big]$$



System of equations

$$\begin{cases} V_1 &= \max\{0 + 0.5V_1 + 0.5V_2;\ \ 0 + 0.5V_1 + 0.5V_3\} \\ V_2 &= \max\{1 + 0.4V_5 + 0.6V_2;\ \ 1 + 0.3V_1 + 0.7V_3\} \\ V_3 &= \max\{-1 + 0.4V_2 + 0.6V_3;\ \ -1 + 0.5V_4 + 0.5V_3\} \\ V_4 &= \max\{-10 + 0.9V_6 + 0.1V_4;\ \ -10 + V_7\} \\ V_5 &= -10 \\ V_6 &= 100 \\ V_7 &= -1000 \end{cases}$$

$\Rightarrow$ too complicated, we need to find an alternative solution.

State-Action Value Function

Definition

In discounted infinite horizon problems, for any policy π, the *state-action value function* (or Q-function) $Q^\pi : S \times A \mapsto \mathbb{R}$ is

$$Q^\pi(s,a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) | s_0 = s, a_0 = a, a_t = \pi(s_t), \forall t \geq 1\Big],$$

The optimal Q-function is

$$Q^\star(s,a) = \max_\pi Q^\pi(s,a),$$

Greedy Policy

The *greedy* policy with respect to a value $V \in \mathbb{R}^S$, is defined as

$$\pi(s) \in \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') \right]$$

The *greedy* policy with respect to a value $Q \in \mathbb{R}^{S \times A}$, is defined as

$$\pi(s) \in \arg\max_{a \in \mathcal{A}} Q(s,a)$$

👉 from Bellman optimality equations

$$\pi^\star = \text{greedy}(V^\star) \quad \text{or} \quad \pi^\star = \text{greedy}(Q^\star)$$

State-Action and State Value Function

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) V^{\pi}(s')$$

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

$$Q^{\star}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) V^{\star}(s')$$

$$V^{\star}(s) = Q^{\star}(s, \pi^{\star}(s)) = \max_{a \in A} Q^{\star}(s,a)$$

Outline

Value Iteration

Input: $\mathcal{S}$, $\mathcal{A}$, r, p, ϵ
Set $Q_0(s,a) = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $k = 0$

repeat
  for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
    Compute

$$Q_{k+1}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} Q_k(s',a')$$

  end
  $k = k + 1$
until $\|Q_{k+1} - Q_k\|_\infty < \epsilon$
return greedy policy $\pi_\epsilon(s) = \arg\max_{a \in \mathcal{A}} Q_k(s,a)$

Value Iteration: the Guarantees

Theorem

Let $Q_0 \in \mathbb{R}^N$ be an arbitrary function, then the sequence of functions $\{Q_k\}_k$ generated by value iteration ***converges*** *to the optimal value function $Q^\star$.*

Furthermore, let $\varepsilon > 0$ and $\max_{s,a} |r(s,a)| \leq r_{\max} < \infty$, then after ***at most***

$$K = \left\lceil \frac{\log((1-\gamma)\varepsilon / r_{\max})}{\log(\gamma)} \right\rceil$$

iterations $\|Q_K - Q^\star\|_\infty \leq \varepsilon$.

Value Iteration: the Guarantees

Corollary

Let V_K the function computed after K iterations by value iteration, then the greedy policy

$$\pi_K(s) \in \arg\max_{a \in A} Q_K(s,a)$$

is such that

$$\underbrace{\|V^\star - V^{\pi_K}\|_\infty}_{\text{performance loss}} \leq \frac{2\gamma}{1-\gamma} \underbrace{\|Q^\star - Q_K\|_\infty}_{\text{approx. error}}.$$

Furthermore, there exists $\epsilon > 0$ such that if $\|Q_K - Q^\star\|_\infty \leq \epsilon$, then π_K is optimal.

Proof: Performance Loss

Definition

For any $W \in \mathbb{R}^N$, the *Bellman operator* $\mathcal{T}^\pi : \mathbb{R}^N \to \mathbb{R}^N$ is

$$\mathcal{T}^\pi W(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s'|s, \pi(s)) W(s'),$$

and the *optimal Bellman operator* (or dynamic programming operator) is

$$\mathcal{T} W(s) = \max_{a \in A} \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) W(s) \right].$$

The (policy/optimal) value functions are fixed point of the their operators

$$V^\pi = \mathcal{T}^\pi V^\pi,$$
$$V^* = \mathcal{T} V^\star.$$

Proof: Performance Loss

Theorem

The Bellman operators are γ-contractions in ell_∞-norm

$$\|\mathcal{T}W_1 - \mathcal{T}W_2\|_\infty \leq \gamma \|W_1 - W_2\|_\infty$$
$$\|\mathcal{T}^\pi W_1 - \mathcal{T}^\pi W_2\|_\infty \leq \gamma \|W_1 - W_2\|_\infty$$

For any $s \in S$

$$\begin{aligned}
&|\mathcal{T}W_1(s) - \mathcal{T}W_2(s)| \\
&= \left| \max_a \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) W_1(s') \right] - \max_{a'} \left[r(s,a') + \gamma \sum_{s'} p(s'|s,a') W_2(s') \right] \right| \\
&\leq \max_a \left| \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) W_1(s') \right] - \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) W_2(s') \right] \right| \\
&= \gamma \max_a \sum_{s'} p(s'|s,a) |W_1(s') - W_2(s')| \\
&\leq \gamma \|W_1 - W_2\|_\infty \max_a \sum_{s'} p(s'|s,a) = \gamma \|W_1 - W_2\|_\infty,
\end{aligned}$$

Proof: Performance Loss

$$
\begin{aligned}
\|V^* - V^\pi\|_\infty &\leq \|\mathcal{T}V^* - \mathcal{T}^\pi V\|_\infty + \|\mathcal{T}^\pi V - \mathcal{T}^\pi V^\pi\|_\infty \\
&\leq \|\mathcal{T}V^* - \mathcal{T}V\|_\infty + \gamma\|V - V^\pi\|_\infty \\
&\leq \gamma\|V^* - V\|_\infty + \gamma(\|V - V^*\|_\infty + \|V^* - V^\pi\|_\infty) \\
&\leq \frac{2\gamma}{1-\gamma}\|V^* - V\|_\infty.
\end{aligned}
$$

■

Value Iteration: the Complexity

Time complexity

- Each iteration takes $O(S^2A)$ operations

$$Q_{k+1}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} Q_k(s',a')$$

- The computation of the greedy policy takes $O(SA)$ operations

$$\pi_K(s) \in \arg\max_{a \in A} Q_K(s,a)$$

- Total time complexity $O(KS^2A)$

Space complexity

- Storing the MDP: dynamics $O(S^2A)$ and reward $O(SA)$.
- Storing the value function $O(SA)$ and the optimal policy $O(S)$.

Outline

Policy Iteration

Let π_0 be an arbitrary stationary policy

while $k = 1, \ldots, K$ **do**

Policy Evaluation: given π_k compute V^{π_k}

Policy Improvement: find π_{k+1} that is better than π_k

- e.g., compute the *greedy* policy

$$\pi_{k+1}(s) \in \arg\max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_y p(y|s,a) V^{\pi_k}(y) \right\}$$

return the last policy π_K

end

Policy Iteration: the Guarantees

Proposition

The policy iteration algorithm generates a sequences of policies with non-decreasing performance

$$V^{\pi_{k+1}} \geq V^{\pi_k},$$

and it converges to $\pi^\star$ in a finite number of iterations.

Policy Iteration: Complexity

Policy Evaluation Step

- Solution of the Bellman equations $O(S^3)$
- Iterative policy evaluation $O\Big(S^2 \frac{\log(1/\epsilon)}{\log(1/\gamma)}\Big)$

Policy Improvement Step

- If the policy is *evaluated with* V, then complexity $O(SA)$

Policy Iteration: Complexity

Policy Evaluation Step

- Solution of the Bellman equations $O(S^3)$
- Iterative policy evaluation $O\Big(S^2 \frac{\log(1/\epsilon)}{\log(1/\gamma)}\Big)$

Policy Improvement Step

- If the policy is *evaluated with* V, then complexity $O(SA)$
- If the policy is *evaluated with* Q, then complexity $O(A)$

$$\pi_{k+1}(s) \in \arg\max_{a \in A} Q^{\pi_k}(s, a),$$

Number of Iterations

- At most $O\left(\frac{SA}{1-\gamma} \log(\frac{1}{1-\gamma})\right)$
- Other results exist that do not depend on γ

Comparison between Value and Policy Iteration

Value Iteration

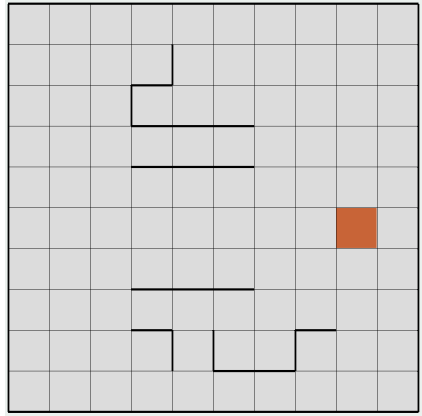
- *Pros:* each iteration is *computationally efficient*.
- *Cons:* convergence is *asymptotic*.

Policy Iteration

- *Pros:* converge in a *finite* number of iterations (often small in practice).
- *Cons:* each iteration requires a full *policy evaluation* and it might be expensive.

The Grid-World Problem

Hands-on session [link] – [5min]



Bibliography

Thank you!

facebook
Artificial Intelligence Research