L3: Bellman’s Equations & Dynamic Programming

EECE 571N | Sequential Decision Making | Fall 2025

Cyrus Neary | cyrus.neary@ubc.ca

Last lecture…

$\mathcal{M} = (S, A, T, R, \gamma, \mu)$

R- Reward function

$R(s, a) \in \mathbb{R}$ — A function defining the reward the agent receives for transitioning from state $s \in S$ to $s' \in S$ under action $a \in A$.

A - Actions

Set of all possible actions the agent could take.

μ - Initial state distribution

$\mu(s)$ denotes the probability that the initial state is s.

S - States

Set of all possible states from which the agent will make decisions.

γ - Discount factor

$\gamma \in [0,1]$ — a modeling parameter that reduces the impact of future rewards on decisions made now.

T - Transition function

$s' \sim T(\cdot \mid s, a)$ a stochastic transition function defining the probability of the agent transitioning to next state $s' \in S$, given that it executed action $a \in A$ from state $s \in S$.

Last lecture…

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

Intuition: How “valuable” is it for the agent to be in state s, if it follows policy π from that point onwards?

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

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

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a\right]$$

Intuition: How “valuable” is it for the agent to take action a from state s, if it follows policy π from that point onwards?

Last lecture…

A policy $\pi^*(\cdot \mid s)$ is considered optimal whenever

$$V^{\pi^*}(s) \geq V^{\pi}(s), \quad \forall s \in S, \forall \pi \in \Pi^{HR}$$

The optimal value of an MDP is defined as:

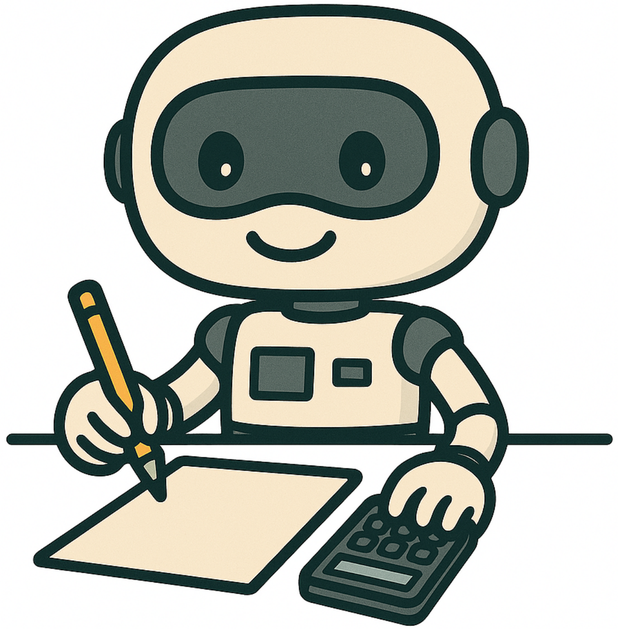
$$V^*(s) := \sup_{\pi \in \Pi^{HR}} V^{\pi}(s)$$

An optimal policy $\pi^* \in \Pi^{\{HR,MR,HD,MD\}}$ exists when:

$$V^{\pi^*}(s) \geq V^*(s), \quad \forall s \in S$$

This lecture…

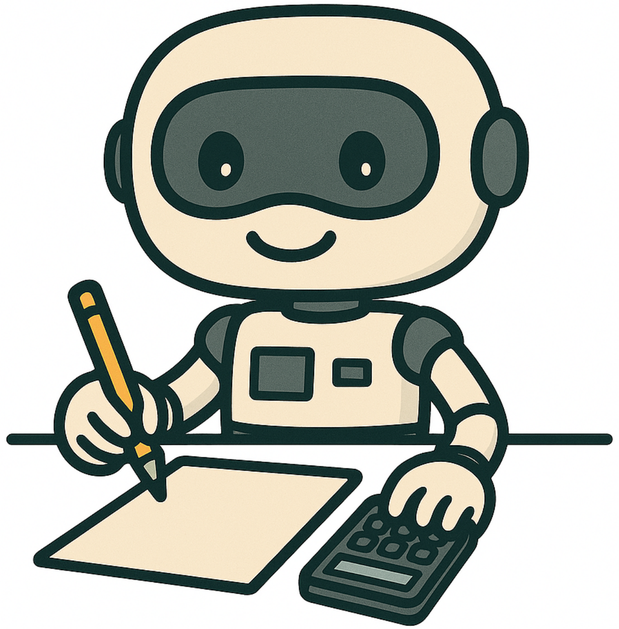
How can we solve for optimal policies?



This lecture…

How can we solve for optimal policies?

One solution technique: Dynamic programming!



Dynamic programming

41 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

Not to be confused with Dynamic programming language or Dynamic problem.

Dynamic programming is both a mathematical optimization method and an algorithmic paradigm. The method was developed by Richard Bellman in the 1950s and has found applications in numerous fields, from aerospace engineering to economics.

In both contexts it refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. While some decision problems cannot be taken apart this way, decisions that span several points in time do often break apart recursively. Likewise, in computer science, if a problem can be solved optimally by breaking it into sub-problems and then recursively finding the optimal solutions to the sub-problems, then it is said to have *optimal substructure*.

If sub-problems can be nested recursively inside larger problems, so that dynamic programming methods are applicable, then there is a relation between the value of the larger problem and the values of the sub-problems.[1] In the optimization literature this relationship is called the Bellman equation.

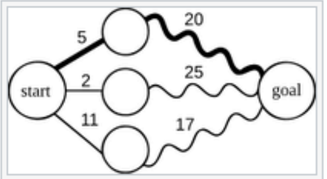
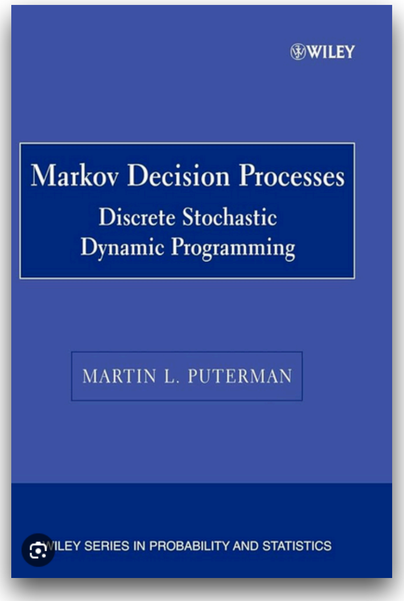


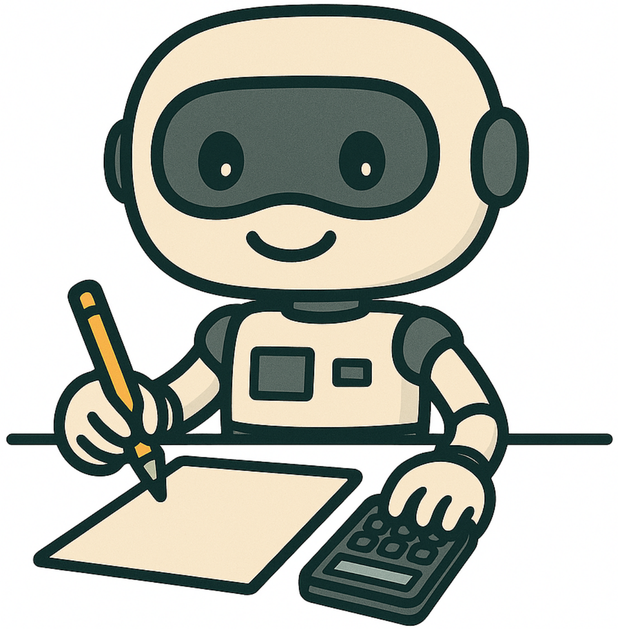
Figure 1. Finding the shortest path in a graph using optimal substructure; a straight line indicates a single edge; a wavy line indicates a shortest path between the two vertices it connects (among other paths, not shown, sharing the same two vertices); the bold line is the overall shortest path from start to goal.

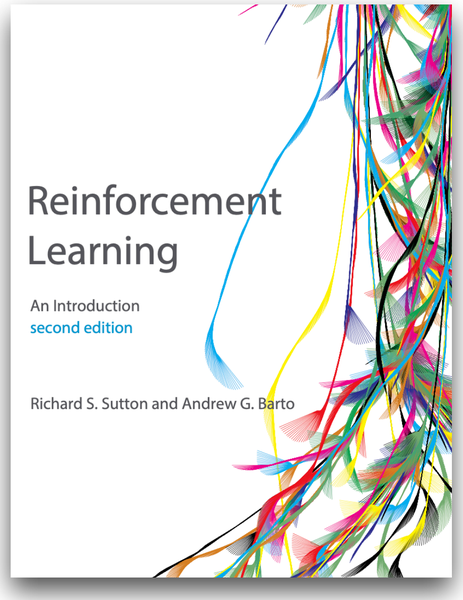
This lecture…

How can we solve for optimal policies?



Chapters 6.1-6.4





Chapter 4

Why use dynamic programming?

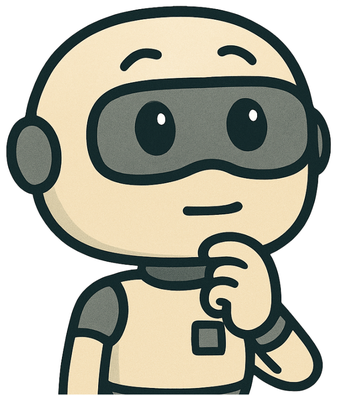
What would an alternate, naieve approach look like? What are the drawbacks?

- enumerate all possible trajectories and compute expected returns.
- MDP with branching factor b and time horizon T, this requires $O(b^T)$ computational effort.

How do we naturally solve long-horizon decision problems?

- Naturally break problems into stages. Think about what a "good" next state is, and try to get there first.
- Make decisions now that put you in a position that aligns with your long-term goals.

What is the core intuition behind the use of dynamic programming?



Solve small problems once and reuse their solns when appear in larger problems.

Bellman's equation

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^{\pi}(s') \right]$$

Bellman's equation

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^{\pi}(s') \right]$$

Value at state s

Bellman's equation

Immediate reward from state s under policy π.

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{\pi}(s') \right]$$

Value at state s

Bellman's equation

Immediate reward from state s under policy π.

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{\pi}(s') \right]$$

Value at state s

Probability of transitioning from s to new state s' under policy π.

Bellman's equation

Immediate reward from state s under policy π.

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{\pi}(s') \right]$$

Value at state s

Probability of transitioning from s to new state s' under policy π.

Value of the next state s'

Deriving Bellman's equation

Tower property

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right] = \mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

$$= \mathbb{E}\left[\mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a, s_1 = s'\right] \mid \pi, s_0 = s\right]$$

$$\mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a, s_1 = s'\right] = R(s,a) + \gamma \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s'\right]$$

$$= R(s,a) + \gamma V^\pi(s') \qquad \gamma V^\pi(s')$$

Take outer expectation

$$\mathbb{E}\left[R(s,a) + \gamma V^\pi(s') \mid \pi, s_0 = s\right] = \sum_{a,s'} \text{Prob}(a,s') \cdot \left[R(s,a) + \gamma V^\pi(s')\right]$$

$$= \sum_{a \in A} \pi(a|s) \sum_{s' \in S} T(s'|s,a) \cdot \left[R(s,a) + \gamma V^\pi(s')\right]$$

Bellman's equation

$$\boxed{V^\pi(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^\pi(s')\right]}$$

$$= \sum_{a \in A} \pi(a|s) \sum_{s' \in S} T(s'|s,a) \cdot [R(s,a) + \gamma V^{\pi}(s')]$$

$$\sum_{a \in A} \pi(a|s) \underbrace{\sum_{s' \in S} T(s'|s,a)}_{=1} R(s,a) + \sum_{a \in A} \pi(\cdot|\cdot) \sum_{s' \in S} T(s'|s,a) \gamma V^{\pi}(s')$$

Tower property

Variables X, Y, Z

$$\mathbb{E}[X|Z] = \mathbb{E}[\mathbb{E}[X|Y,Z]|Z]$$

Average of something given info on Z

= first averaging over more detailed info Y & Z

Then averaging results of that avg. over Z.

Why is this recursive structure helpful?

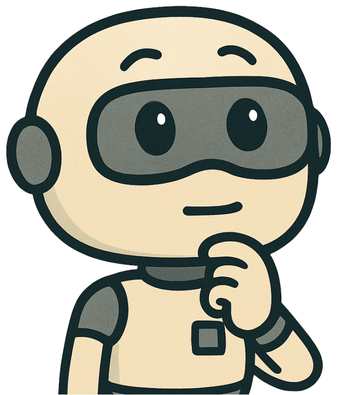
$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{\pi}(s') \right]$$

- Reveals the optimal substructure for value function computation.
- Reduces a quantity over infinite sums of paths into a quantity in terms of a single step, and the value at the next step.

$$\sum_{a \in A} \pi(a|s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) \, \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s' \right] \right]$$

$$\sum_{a \in A} \pi(a|s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) \, \mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s' \right] \right]$$

gross!



Q-functions have an identical recursive structure

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a\right] = \mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a\right]$$

apply tower property:

$$\mathbb{E}\left[\mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a, s_1 = s', a_1 = a'\right] \mid \pi, s_0 = s, a_0 = a\right]$$

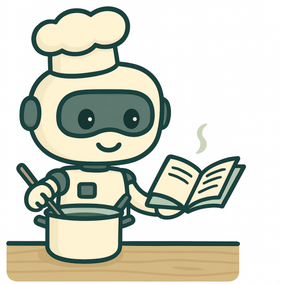
$$R(s,a) + \gamma\, \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s', a_0 = a'\right]$$

Plugging into outer expectation we get our result.

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

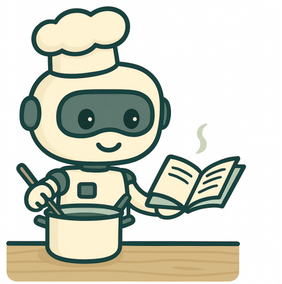
Policy evaluation, policy improvement, policy iteration

The overall recipe:



Policy evaluation, policy improvement, policy iteration

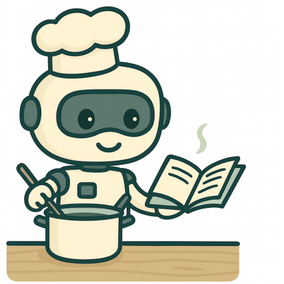
The overall recipe:



Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy evaluation, policy improvement, policy iteration

The overall recipe:



Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy evaluation, policy improvement, policy iteration

The overall recipe:

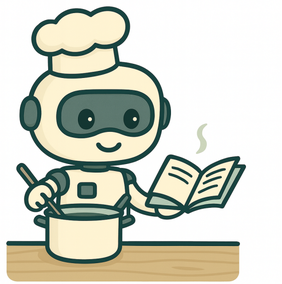
Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy iteration: Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

Policy evaluation, policy improvement, policy iteration

The overall recipe:



Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy iteration: Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

Policy evaluation, linear algebra solution

Suppose we are given a policy π and an MDP model $\mathscr{M}$.
How can we compute the policy's value?

$$V^\pi(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^\pi(s') \right]$$

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

Define $R^\pi(s) = \sum_{a \in A} \pi(a|s) R(s,a)$, $P^\pi(s,s') = \sum_{a \in A} \pi(a|s) T(s'|s,a)$

Then $V^\pi(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s,s') V^\pi(s')$, $\forall s \in S$

Define:

- $V_\pi \in \mathbb{R}^{|S|}$ vector with entries $V^\pi(s)$
- $R_\pi \in \mathbb{R}^{|S|}$ vector with entries $R^\pi(s)$
- $P_\pi \in \mathbb{R}^{|S| \times |S|}$ matrix with entries $P^\pi(s,s')$

$\Rightarrow V_\pi = R_\pi + \gamma P_\pi V_\pi$

$\Rightarrow \boxed{V_\pi = (I - \gamma P_\pi)^{-1} R_\pi}$

Policy evaluation, iterative solution

Suppose we are given a policy π and an MDP model $\mathcal{M}$.
How can we compute the policy's value?

$$V^\pi(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^\pi(s') \right]$$

Define Bellman's operator functions to value functions. $T_\pi : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ as a function mapping from value

$$(T_\pi V)(s) = \sum_{a \in A} \pi(a|s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) V(s') \right]$$

Compute $V^\pi(s)$ using the following iterative procedure:

- Pick some initial guess for $V^\pi(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_\pi V_k$

The value function V^π in question is the fixed point $V^\pi = T_\pi V^\pi$.

Policy evaluation, iterative solution

Compute $V^{\pi}(s)$ using the following iterative procedure:

- Pick some initial guess for $V^{\pi}(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_{\pi} V_k$

Policy evaluation, iterative solution

Compute $V^{\pi}(s)$ using the following iterative procedure:

- Pick some initial guess for $V^{\pi}(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_{\pi} V_k$

The value function V^{π} in question is the fixed point $V^{\pi} = T_{\pi} V^{\pi}$.

Policy evaluation, iterative solution

Compute $V^{\pi}(s)$ using the following iterative procedure:

- Pick some initial guess for $V^{\pi}(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_{\pi} V_k$

The value function V^{π} in question is the fixed point $V^{\pi} = T_{\pi} V^{\pi}$.

Proof sketch

Equip $\mathbb{R}^{|S|}$ with the norm $||V||_{\infty} = \max_{s \in S} |V(s)|$.

Lemma: for any V, V', $||T_{\pi} V - T_{\pi} V'||_{\infty} \leq \gamma ||V - V'||_{\infty}$.

- Proof of lemma: Homework!!

Consequence of lemma:

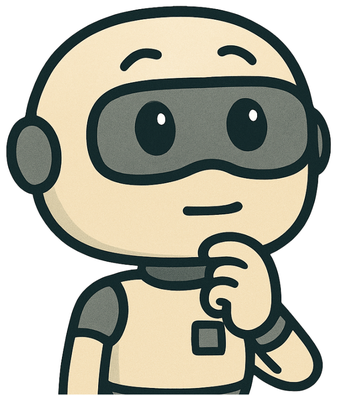
- By the Banach fixed-point theorem, T_{π} has a **unique fixed point.**
- By definition of T_{π} and V_{π}, this unique fixed point must be V_{π}
- The iterates converge from any starting point V_0.
- The error in the iterates shrinks geometrically with γ:
  - $||T_{\pi} V_0 - T_{\pi} V^{\pi}||_{\infty} \leq \gamma ||V_0 - V^{\pi}||_{\infty}$

When might we use a direct solution to the system of equations vs. using an iterative approach to policy evaluation?

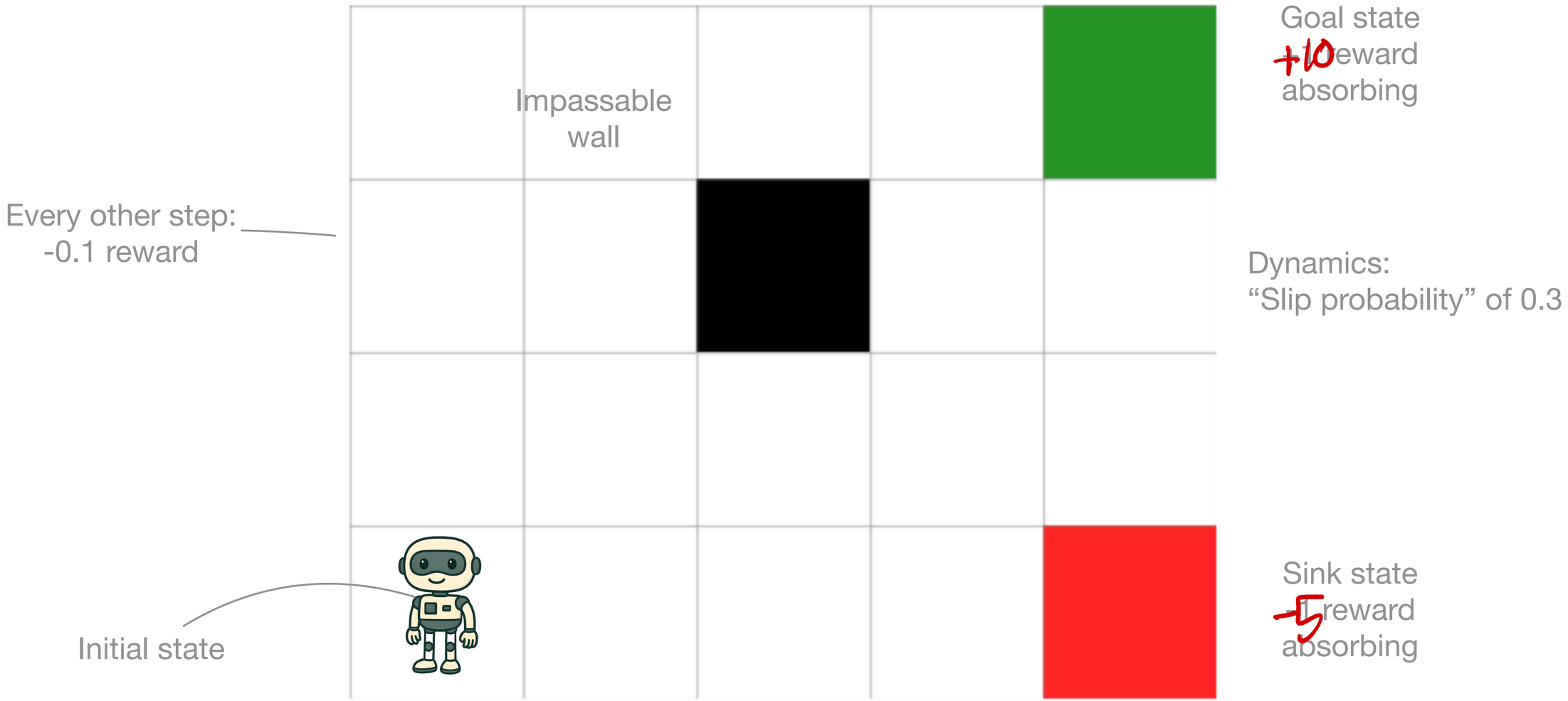
$$V_\pi = (I - \gamma P_\pi)^{-1} R_\pi$$

For k = 0, 1, 2, …
$$V_{k+1} \leftarrow T_\pi V_k$$

- For large state spaces, linear algebra soln requires $O(|S|^3)$ for matrix inversion.
- For iterative approach $O(|S|^2)$ computation per iterate.



Introducing a running gridworld example



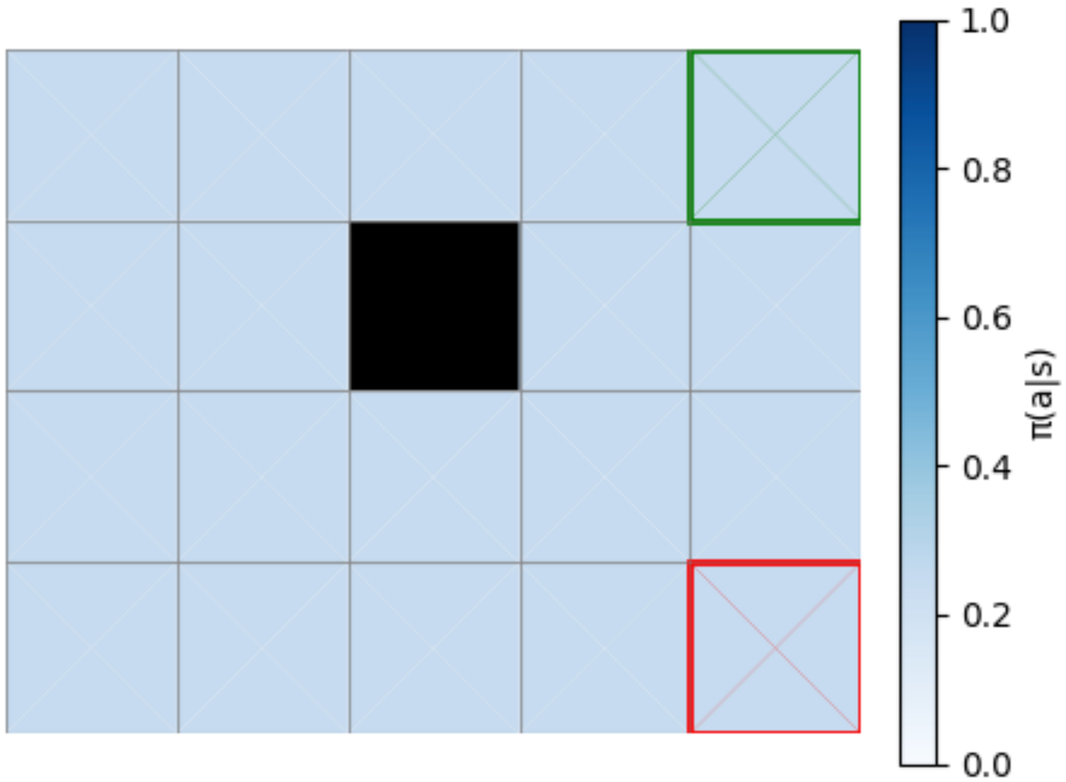
Visualizing two example policies

Uniform distribution over actions

Always try moving right

Visualizing two example policies

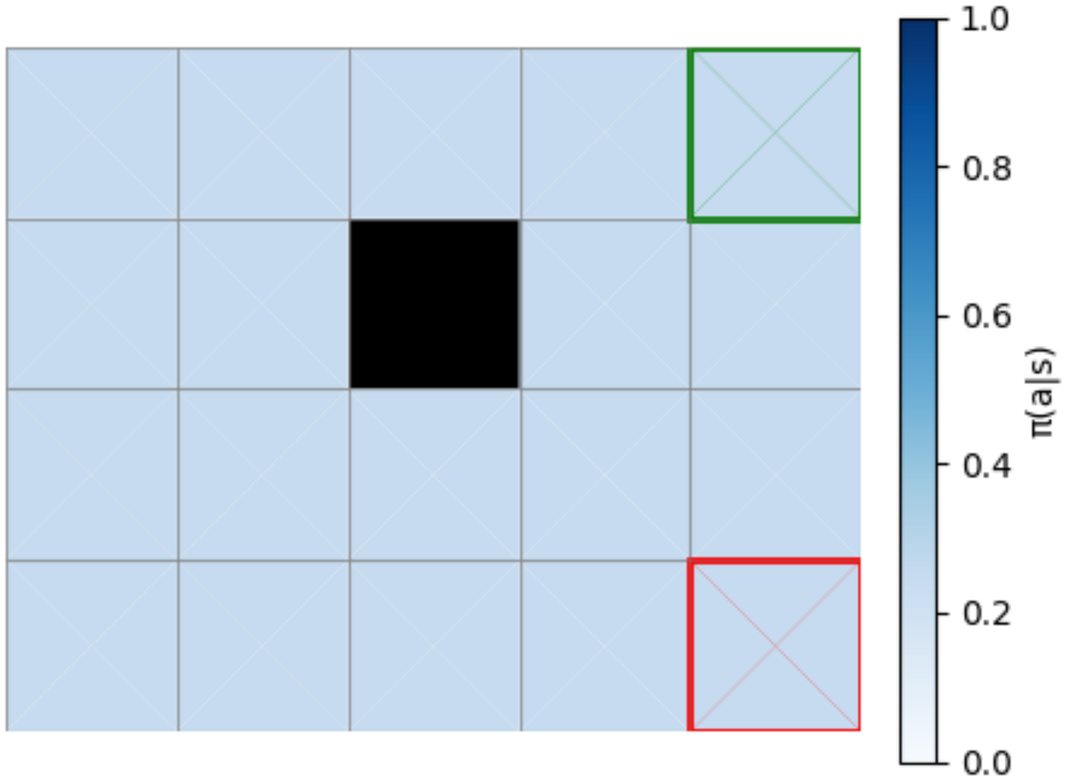
Uniform distribution over actions



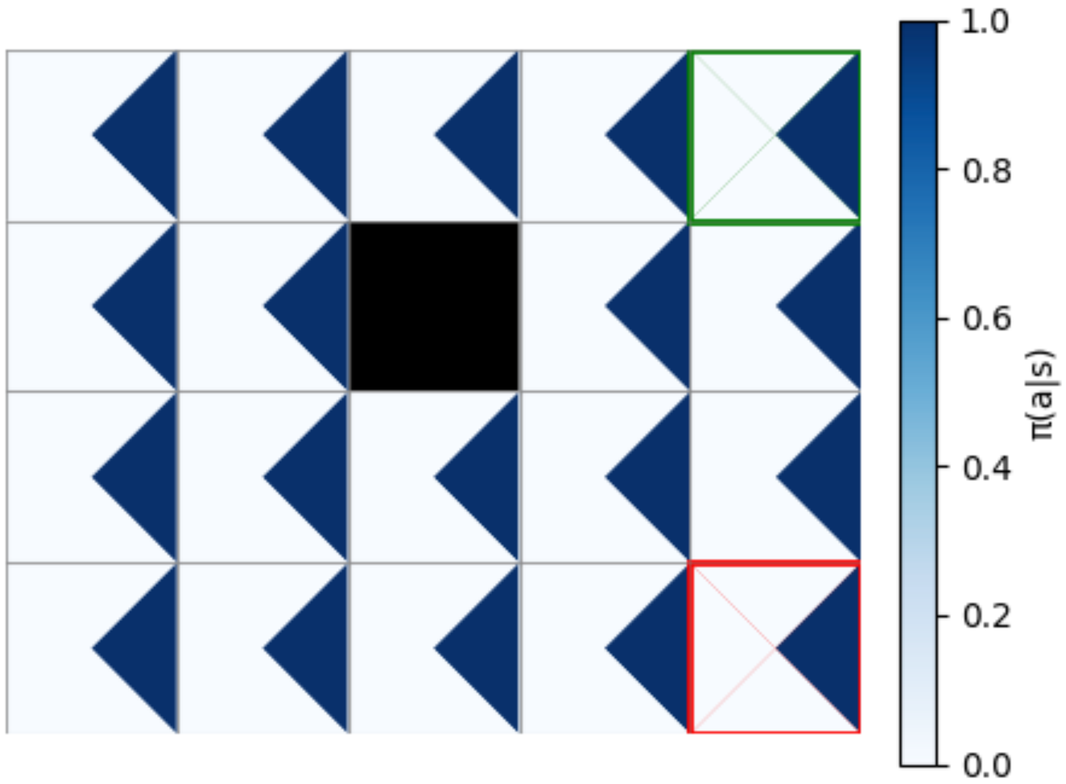
Always try moving right

Visualizing two example policies

Uniform distribution over actions



Always try moving right



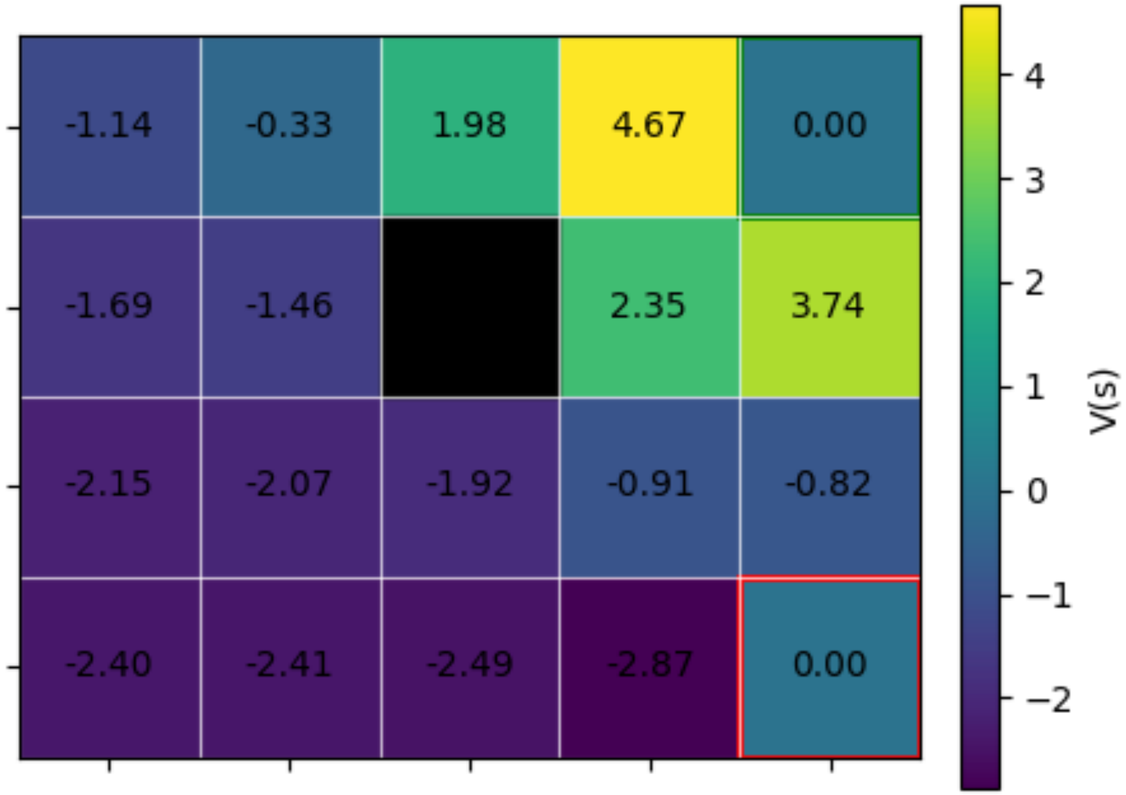
Policy evaluation: visualizing state value function $V^{\pi}(s)$

Uniform distribution over actions

Always try moving right

Policy evaluation: visualizing state value function $V^{\pi}(s)$

Uniform distribution over actions

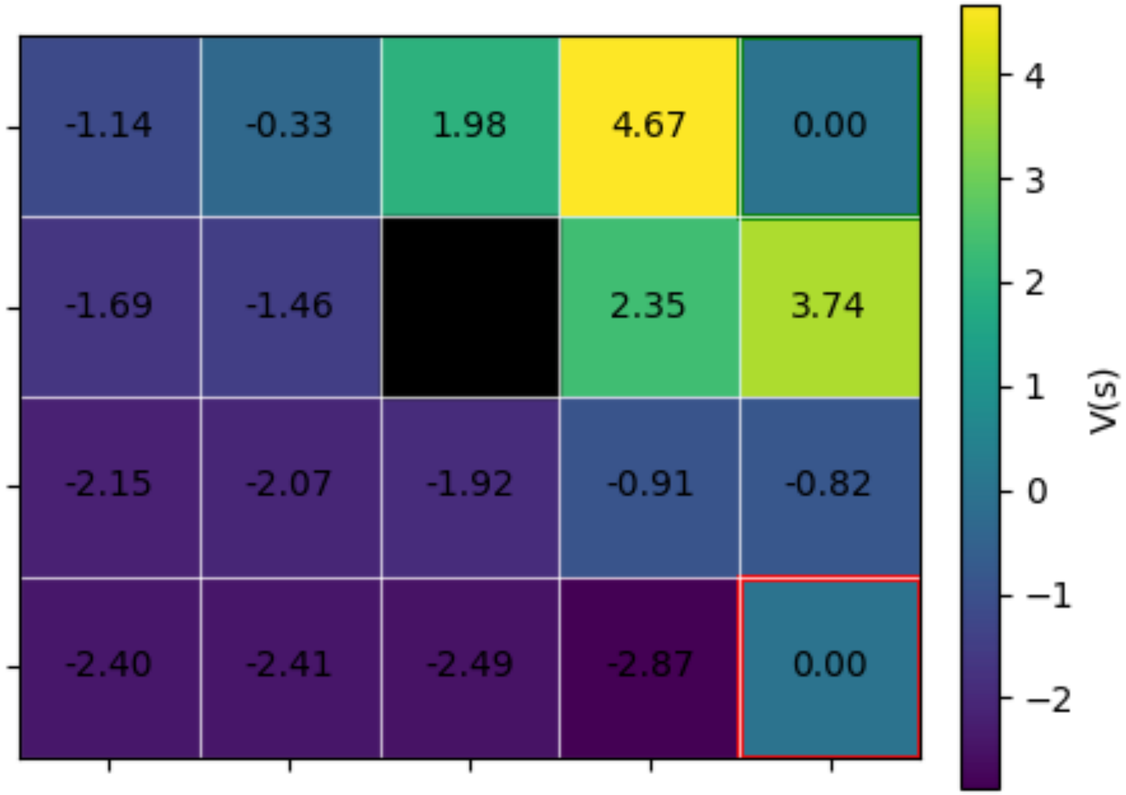


Always try moving right

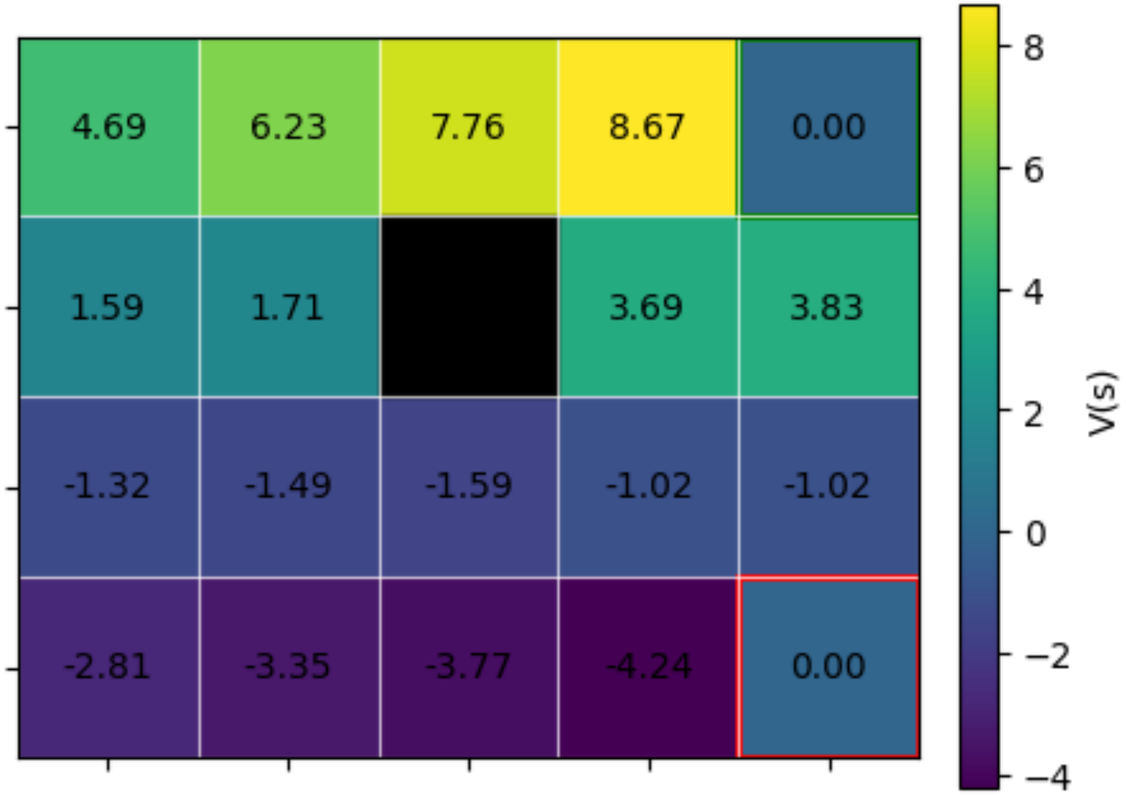
Policy evaluation: visualizing state value function $V^{\pi}(s)$

$R(s,a) \rightarrow R(s,a,s')$

Uniform distribution over actions



Always try moving right



Policy evaluation, policy improvement, policy iteration

The overall recipe:

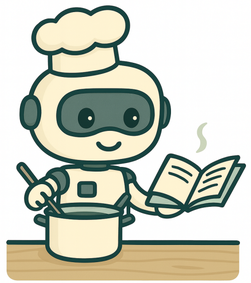
Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy iteration: Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

Policy evaluation, policy improvement, policy iteration

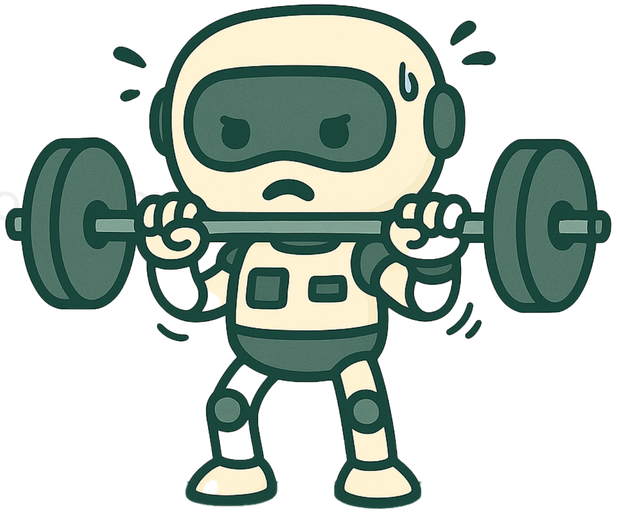
The overall recipe:



Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy iteration: Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.



Policy Improvement

Consider a given policy π, and its corresponding value $V^{\pi}(s)$ and state-action value $Q^{\pi}(s, a)$ functions.

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

Policy improvement theorem:

Let π, π' be two deterministic policies such that, for all $s \in S$, $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$.

Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all states $s \in S$.

Policy Improvement

Consider a given policy π, and its corresponding value $V^{\pi}(s)$ and state-action value $Q^{\pi}(s, a)$ functions.

Policy improvement theorem:

Let π, π' be two deterministic policies such that, for all $s \in S$, $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$.

Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all states $s \in S$.

Policy improvement theorem (in words):

If, for every state, the value of counterfactually replacing the action output by $\pi(s)$ with the action output by $\pi'(s)$ before following π thereafter increases the value…

Then policy π' is a strict improvement (in terms of value) over policy π.

Policy Improvement

Policy improvement theorem:

Let π, π' be two deterministic policies such that, for all $s \in S$, $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$.

Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all states $s \in S$.

Policy Improvement

Policy improvement theorem:

Let π, π' be two deterministic policies such that, for all $s \in S$, $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$.

Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all states $s \in S$.

Proof:

$$
\begin{aligned}
v_\pi(s) &\leq q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] && \text{(by (4.6))} \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] && \text{(by (4.7))} \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s] \\
&= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \mid S_t = s\right] \\
&\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \mid S_t = s\right] \\
&\vdots \\
&\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \mid S_t = s\right] \\
&= v_{\pi'}(s).
\end{aligned}
$$

Policy Improvement

Policy improvement theorem:

Let π, π' be two deterministic policies such that, for all $s \in S$, $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$.

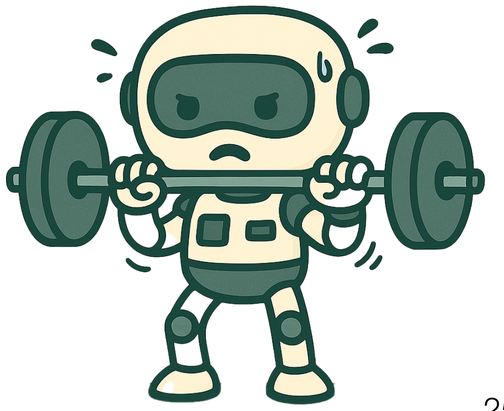
Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all states $s \in S$.

How can we use the policy improvement theorem?

Given the initial policy π and its value functions V^{π}, Q^{π}, construct π' as:

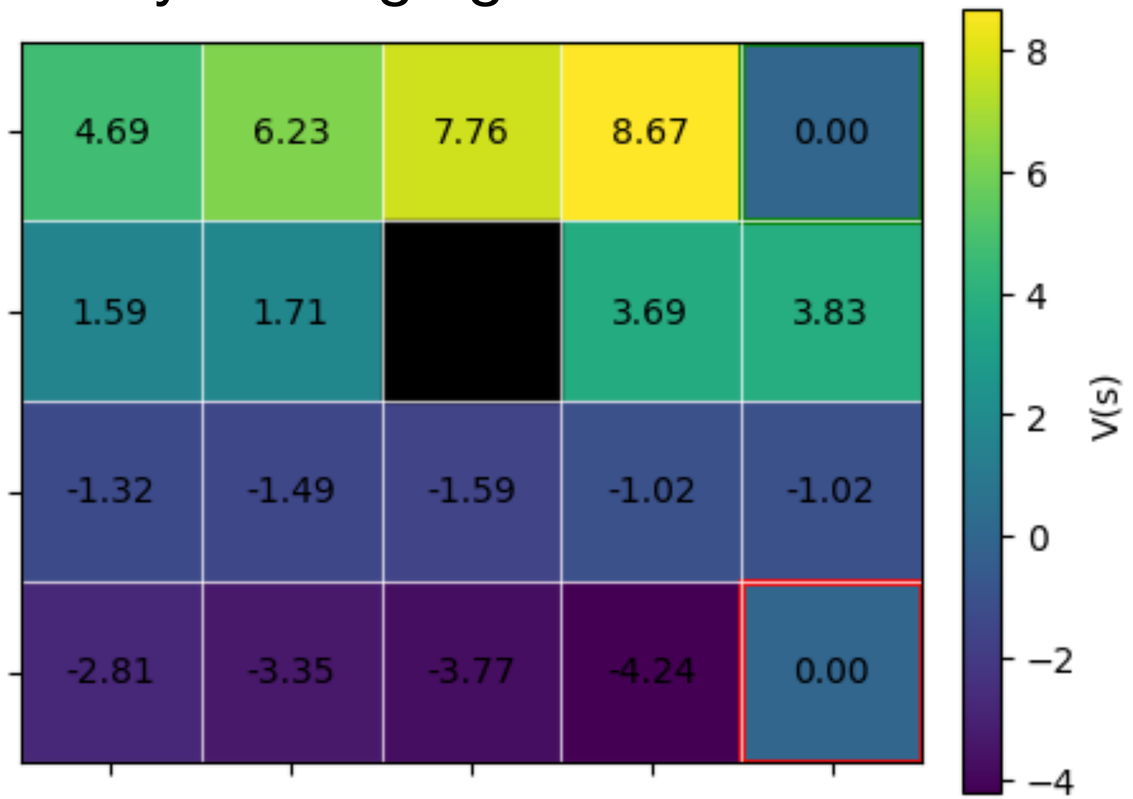
$$\pi'(s) \in \text{argmax}_{a \in A} Q^{\pi}(s, a) = \text{argmax}_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^{\pi}(s') \right]$$

Then, trivially, we have $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$ and so by the policy improvement theorem $V^{\pi'}(s) \geq V^{\pi}(s)$.

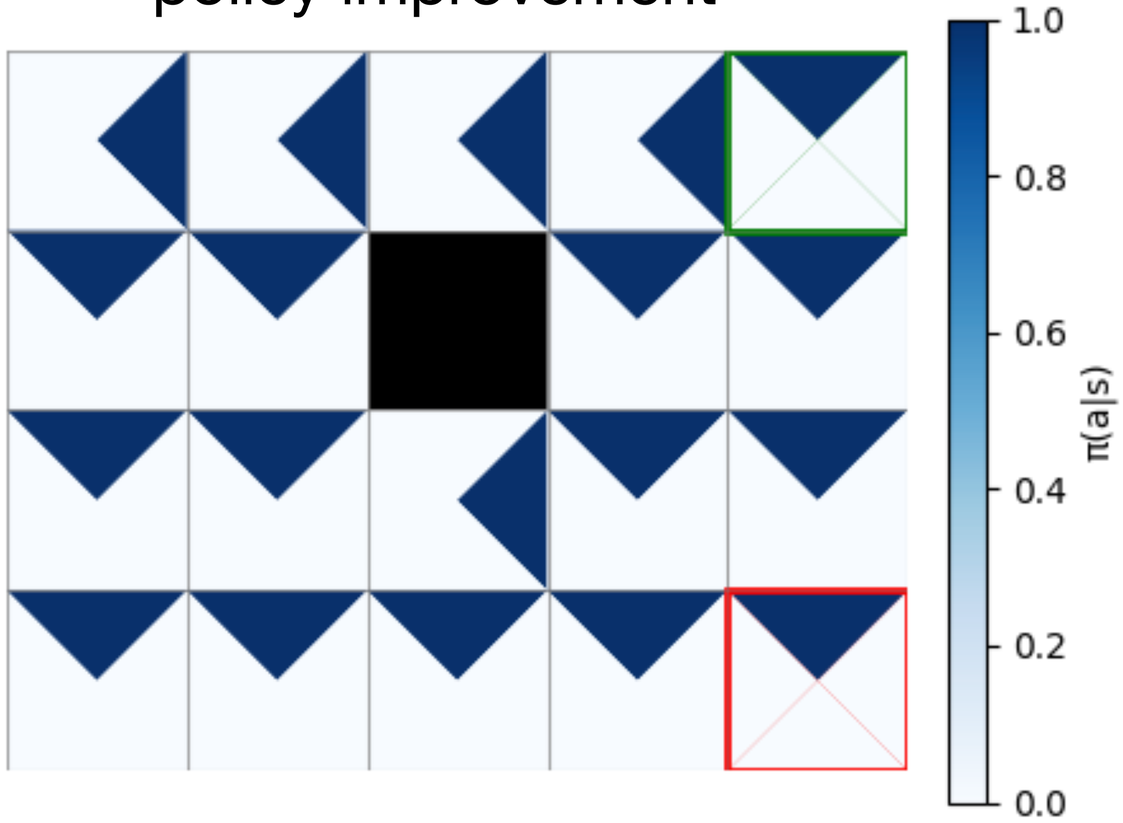


Return to example: Example policy improvement step

Value for policy: always try moving right



After one iteration of policy improvement



Policy evaluation, policy improvement, policy iteration

The overall recipe:

Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy iteration: Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

Policy evaluation, policy improvement, policy iteration

The overall recipe:

Policy evaluation: Given a policy π, compute the value function $V^{\pi}(s)$ of that policy.

Policy improvement: Use the computed value $V^{\pi}(s)$ function to compute an *improved policy* π' such that $V^{\pi'}(s) \geq V^{\pi}(s), \quad \forall s \in S.$

Policy iteration: Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

Policy iteration

Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \left[r + \gamma V(s') \right]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma V(s') \right]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Figure courtesy of Sutton & Barto.

Policy iteration convergence

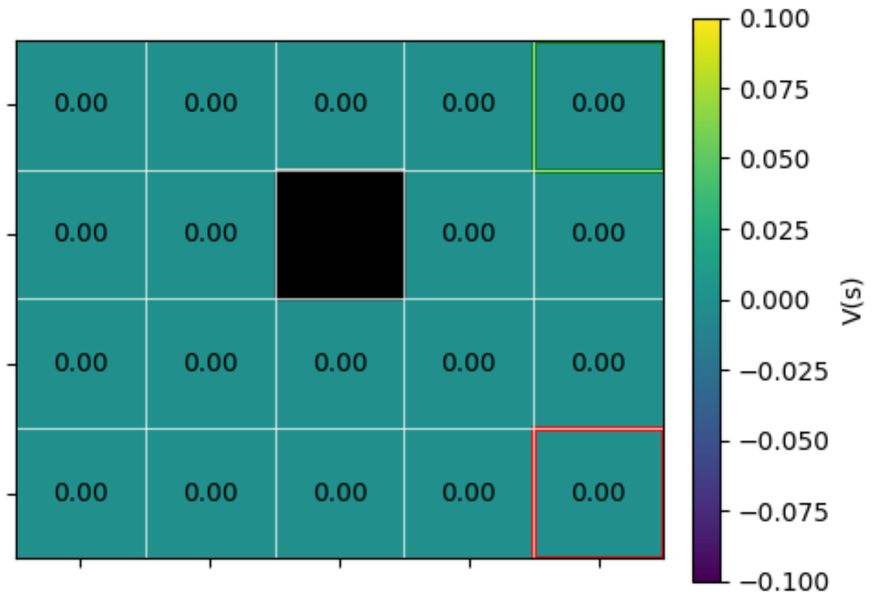
Iterate between policy evaluation and policy improvement to solve for an optimal policy π^* and optimal value function $V^*(s)$.

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*,$$

Policy improvement converges in a finite number of steps to an optimal policy π^* and its value function $V^*(s)$, because each improvement step yields a strictly better policy across all states, and the number of distinct deterministic markovian policies is finite.

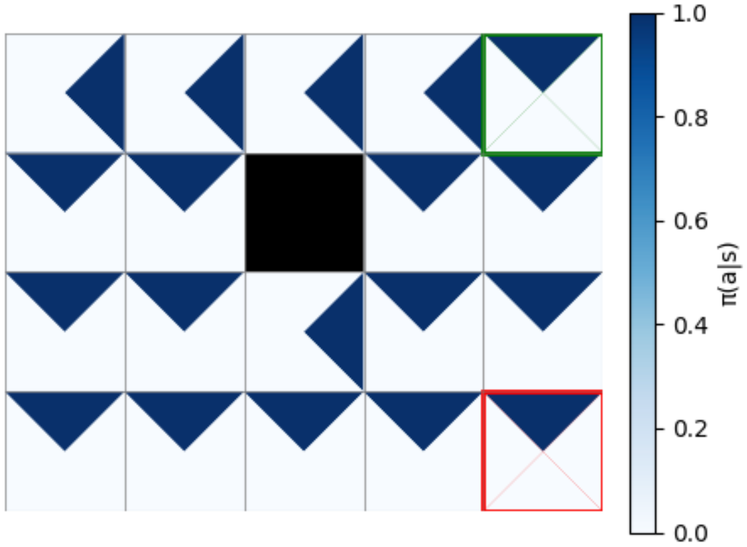
Policy iteration gridworld example

$V_0(s)$

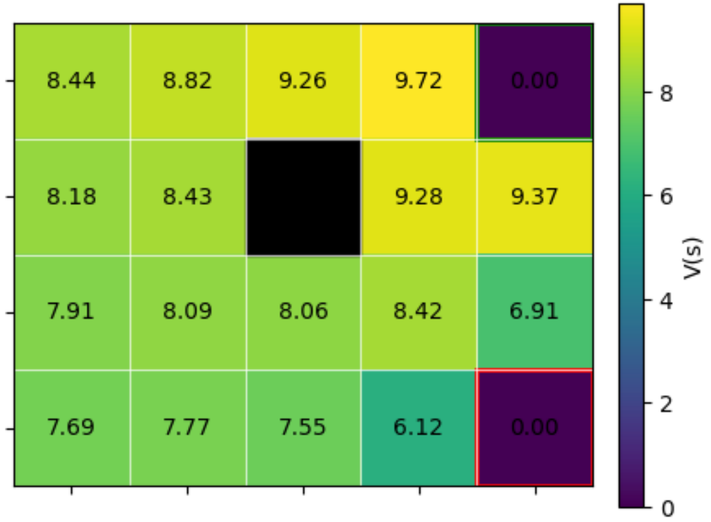


Policy iteration

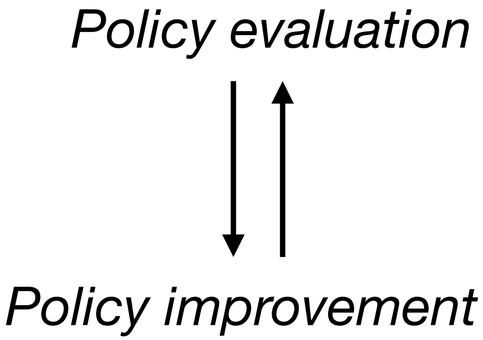
$\pi^*(s)$



$V^*(s)$



Policy iteration:



Wait for complete convergence of value function before proceeding with policy improvement.

Value iteration:

Interleave policy evaluation and improvement into a single iterative process.

More specifically, alternate between policy evaluation and improvement after only a single sweep of the entire state space.

Bellman's optimality equation

Last time $V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) V^{\pi}(s') \right)$

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

Optimal value at state s

One-step maximization problem.

Optimal value at state s'

Derivation of Bellman's optimality equation

Recall Bellman's equation:

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

Derivation of Bellman's optimality equation

Recall Bellman's equation:

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

And the definition of the optimal value function:

$$V^*(s) = \sup_{\pi} V^{\pi}(s)$$

Derivation of Bellman's optimality equation

Recall Bellman's equation:

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

And the definition of the optimal value function:

$$V^*(s) = \sup_\pi V^\pi(s)$$

because $V^\pi(s') \leq V^*(s') \quad \forall s'$

Then, we have

$$V^\pi(s) \leq \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

$$\Longrightarrow \sup_\pi V^\pi(s) \leq \sup_\pi \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

$$\Longrightarrow V^*(s) \leq \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

Derivation of Bellman's optimality equation continued

We established that $$V^*(s) \leq \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

And, by definition of $V^*(s)$ we have that for every $a \in A$

$$V^*(s) \geq R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s')$$

Which clearly implies $$V^*(s) \geq \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

Combining these inequalities, we get arrive at the desired result.

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^*(s') \right]$$

Value iteration

Define *Bellman's optimality operator* $T_* : \mathbb{R}^{|S|} \to \mathbb{R}^{|s|}$ as:

$$(T_* V)(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V(s) \right] = V'(s)$$

Value iteration operates by turning Bellman's optimality operator T_* into an iterative procedure.

- Pick some initial guess for $V^*(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_* V_k$

Value iteration

Define *Bellman's optimality operator* $T_* : \mathbb{R}^{|S|} \to \mathbb{R}^{|s|}$ as:

$$(T_* V)(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V(s) \right]$$

Value iteration operates by turning Bellman's optimality operator T_* into an iterative procedure.

- Pick some initial guess for $V^*(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_* V_k$

The optimal value function V^* is the unique fixed point $V^* = T_* V^*$.

Value iteration

Define *Bellman's optimality operator* $T_* : \mathbb{R}^{|S|} \to \mathbb{R}^{|s|}$ as:

$$(T_* V)(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V(s) \right]$$

Value iteration operates by turning Bellman's optimality operator T_* into an iterative procedure.

- Pick some initial guess for $V^*(s)$. e.g., $V_0(s) = 0, \forall s \in S$.
- For k = 0, 1, 2, …
  - $V_{k+1} \leftarrow T_* V_k$

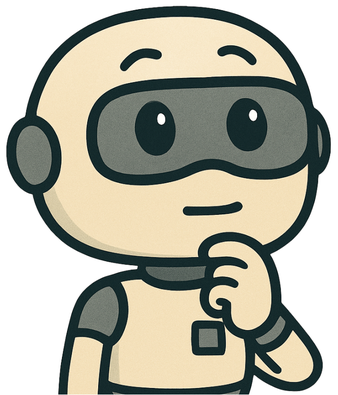
The optimal value function V^π is the unique fixed point $V^* = T_* V^*$.

Given $V^*(s)$, the optimal policy $\pi^*(s)$ may be optained by solving one-step optimization problems at every state:

$$\pi^*(s) \in \text{argmax}_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^*(s) \right]$$

How could we analyze the convergence of value iteration?

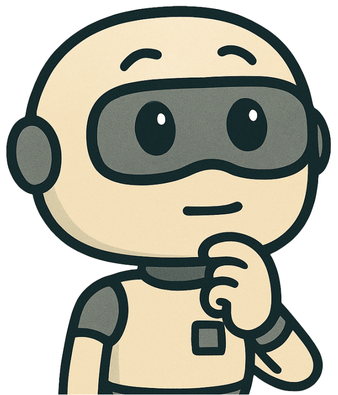
- Same as with policy iteration.
- First step: show that Bellman optimality operator is a contraction mapping.
- Then, by Banach fixed point theorem, T_* must have a unique fixed point.
- Since V^* is a fixed point of T_* by definition, the iterative procedure must converge to the unique fixed point V^*.



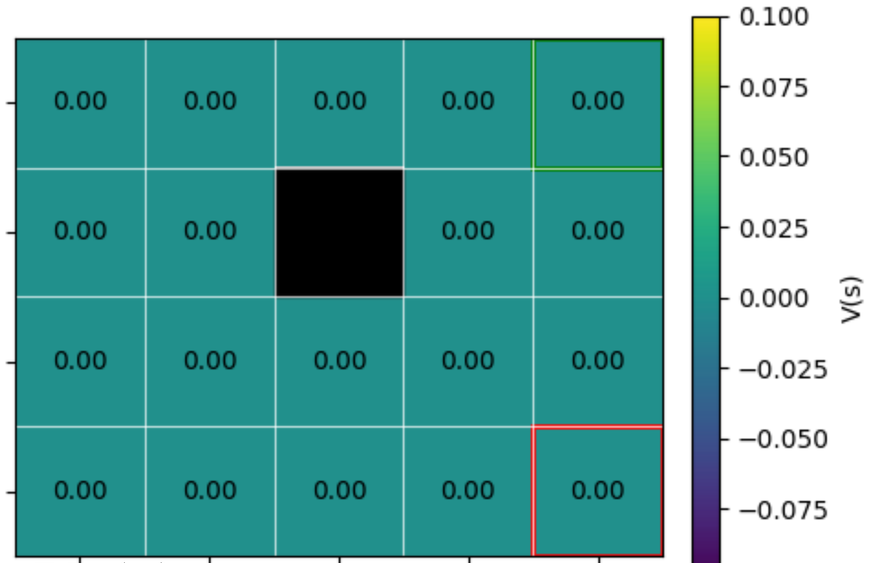
What's the relationship between optimal value functions and policies?

- optimal policy $\Rightarrow$ we can do policy evaluation to find the optimal value function
- optimal value function $\Rightarrow$ we can extract the optimal policy by solving a one-step maximization problem.

$$\pi^*(s) = \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) V^*(s') \right]$$

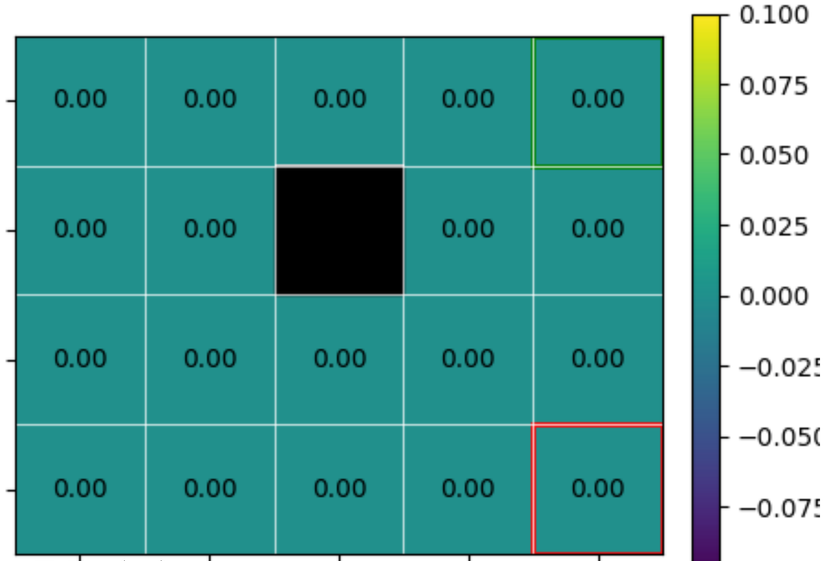


Gridworld example: value iteration



$V_0(s)$

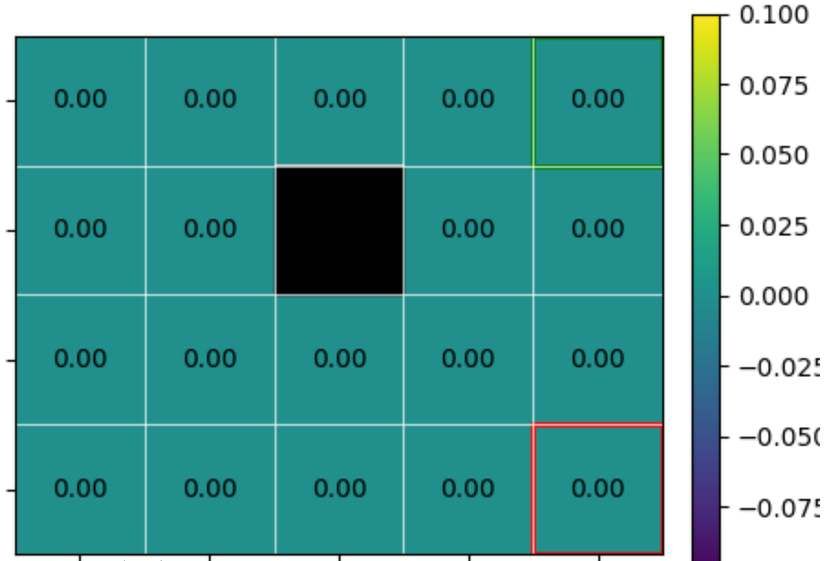
Gridworld example: value iteration



$V_0(s)$

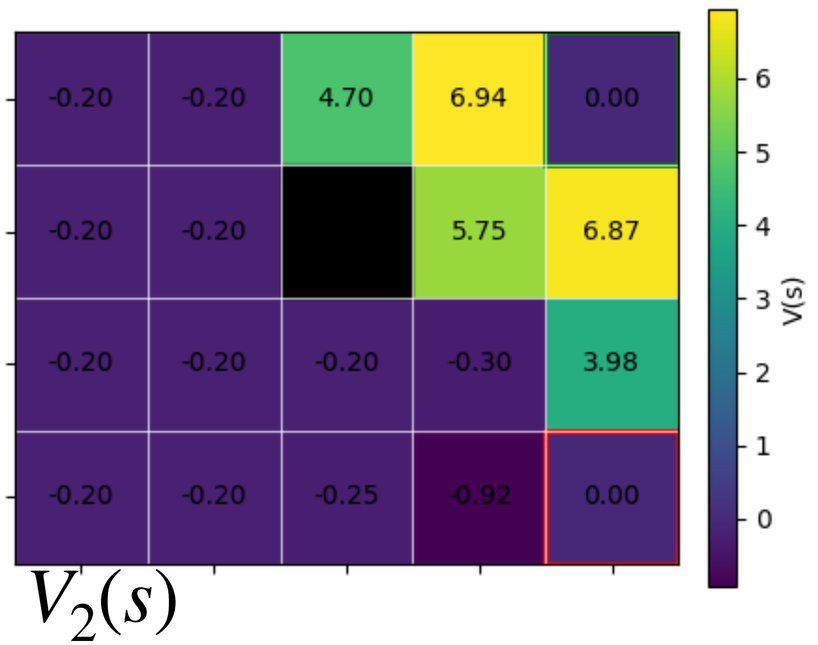
$V_1(s)$

Gridworld example: value iteration



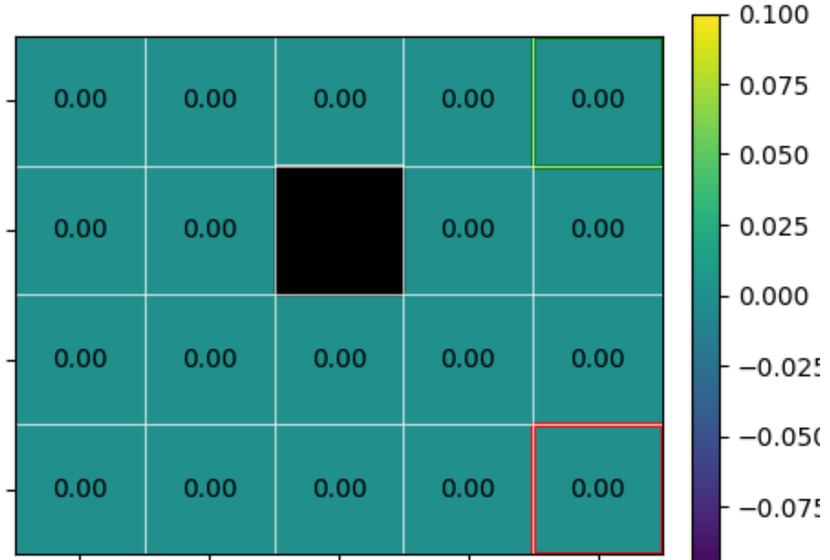
$V_0(s)$

$V_1(s)$



$V_2(s)$

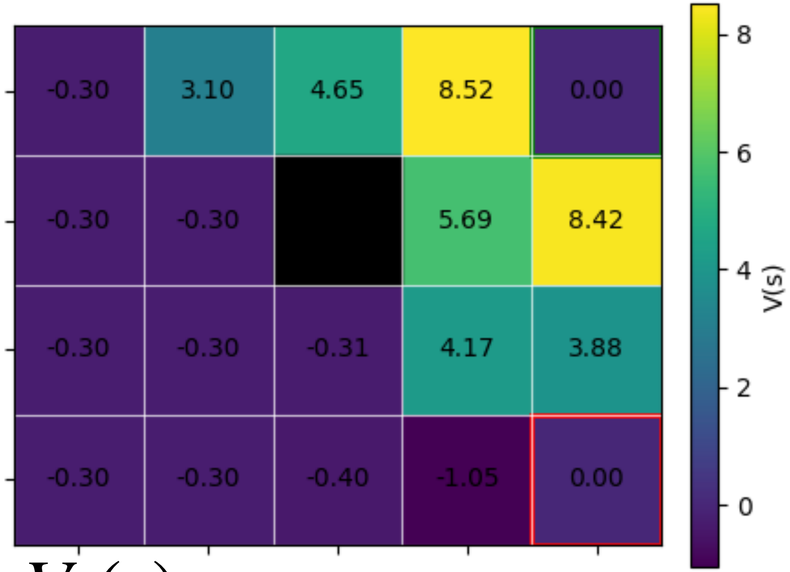
Gridworld example: value iteration



$V_0(s)$

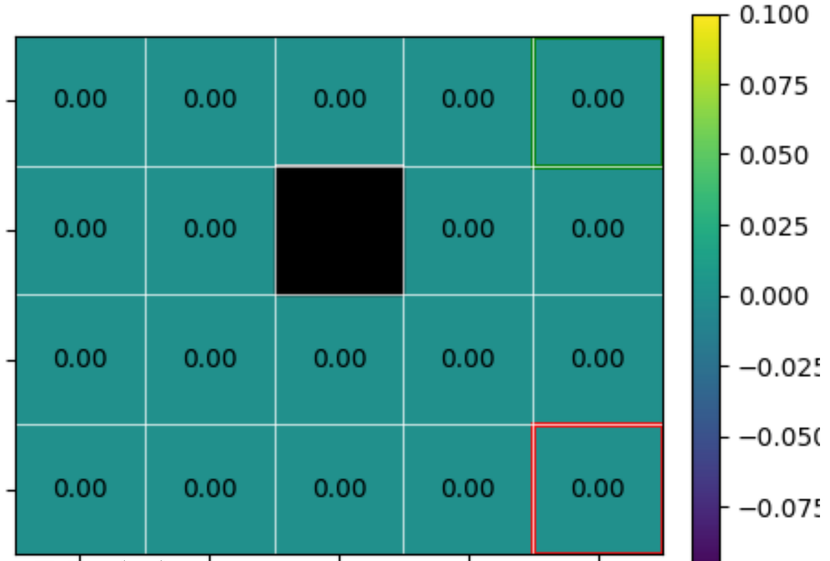
$V_1(s)$

$V_2(s)$



$V_3(s)$

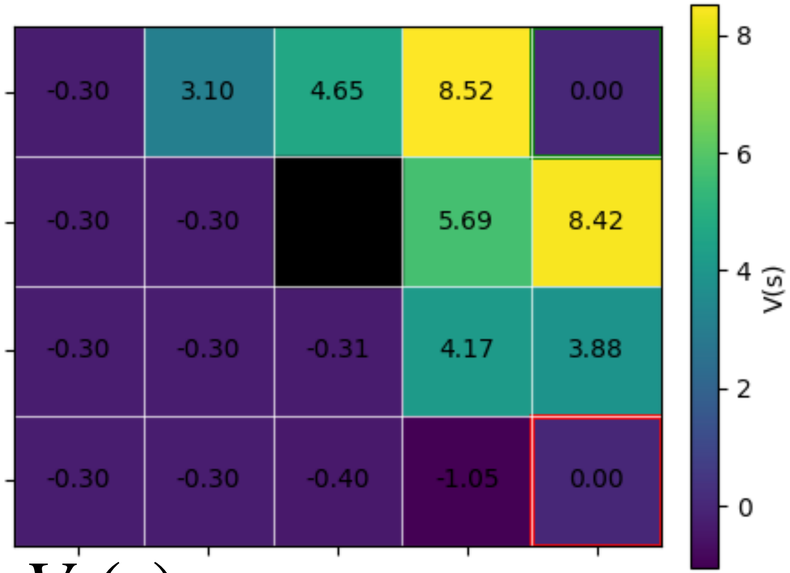
Gridworld example: value iteration



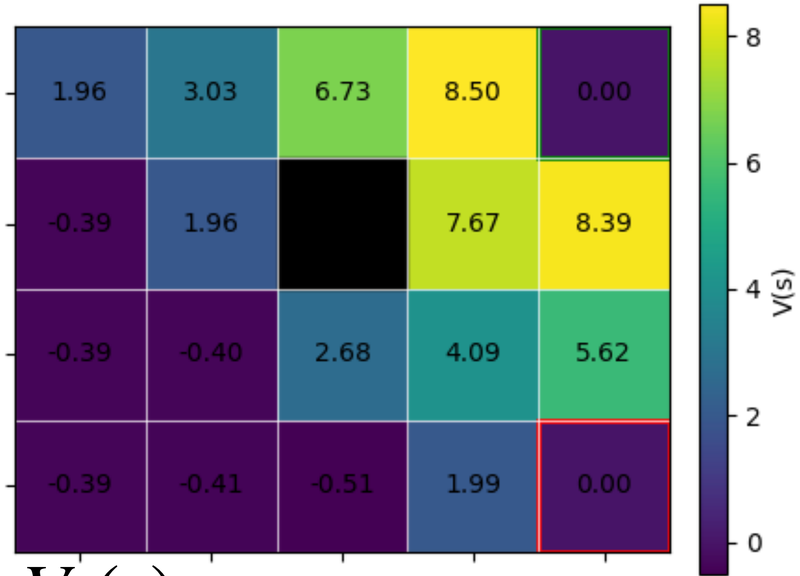
$V_0(s)$

$V_1(s)$

$V_2(s)$

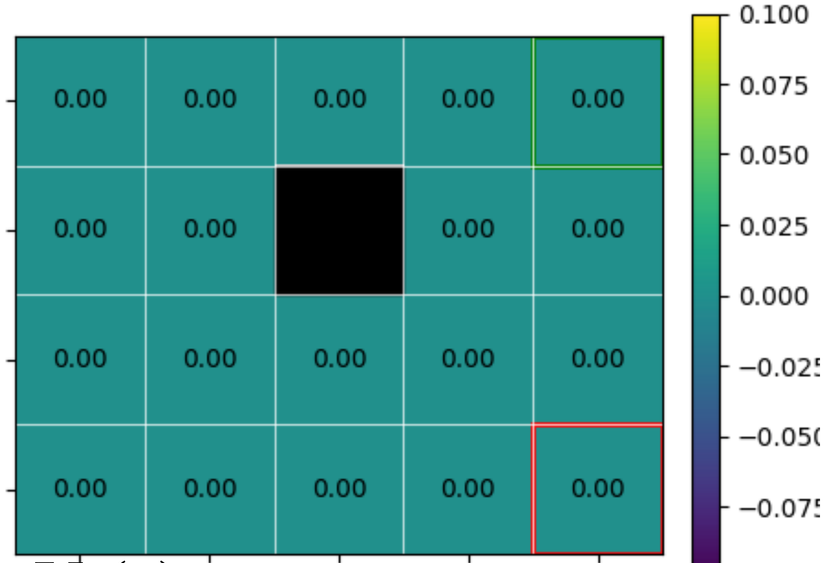


$V_3(s)$



$V_4(s)$

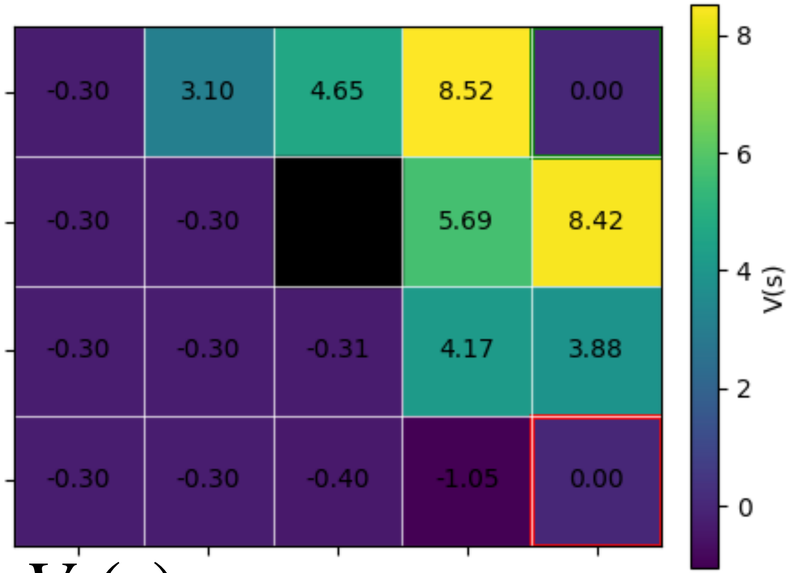
Gridworld example: value iteration



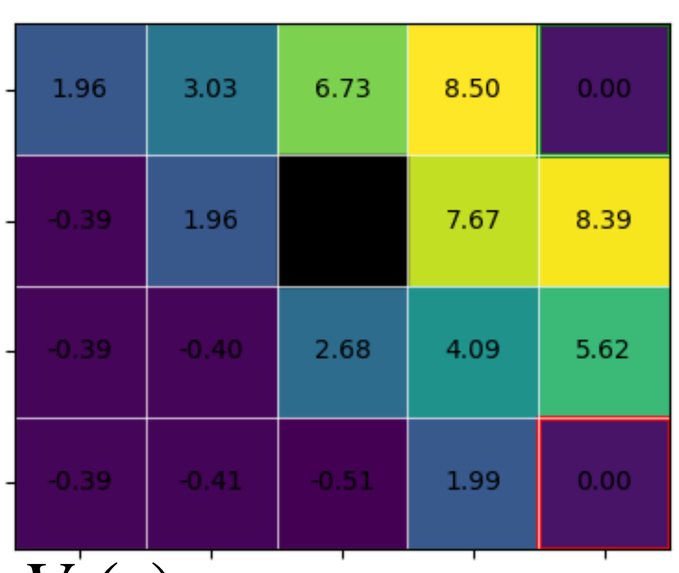
$V_0(s)$

$V_1(s)$

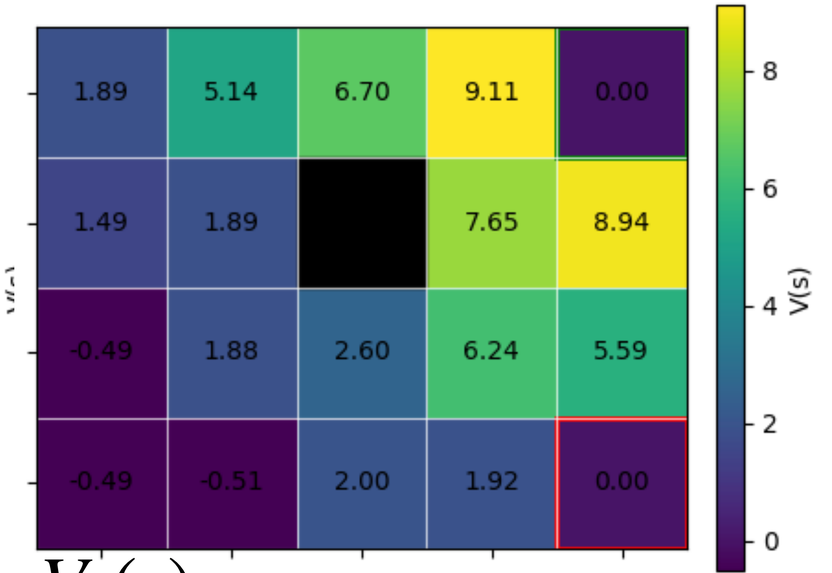
$V_2(s)$



$V_3(s)$



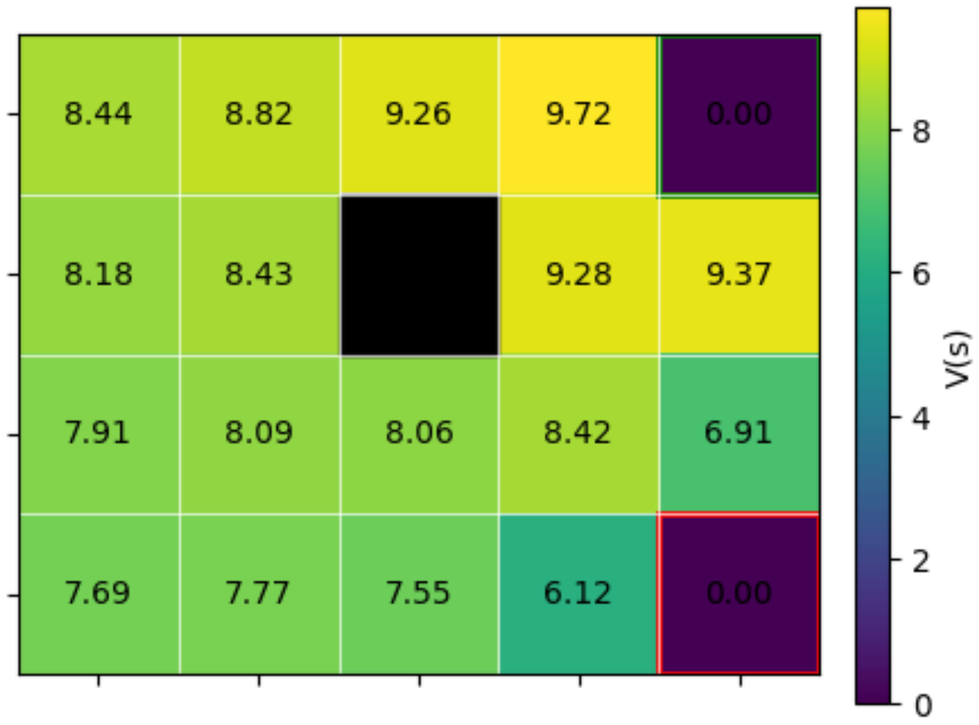
$V_4(s)$



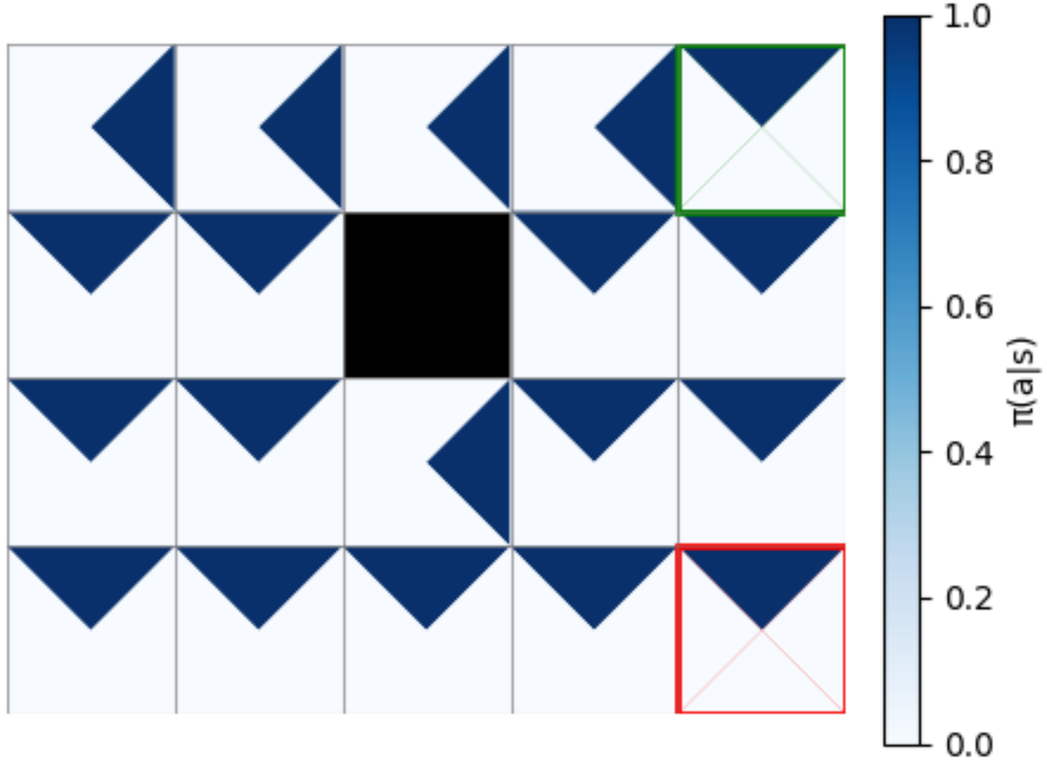
$V_5(s)$

Gridworld example: value iteration

$V_{82}(s)$

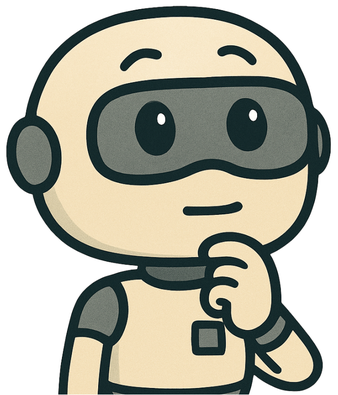


$\pi^*(s)$



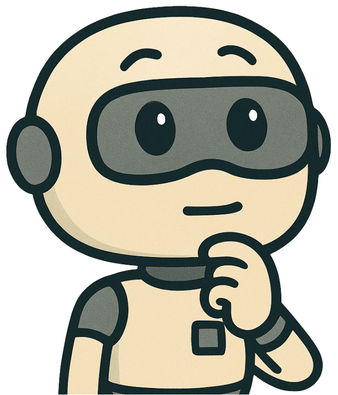
What assumptions are required for Bellman's equations to hold?

- Time invariance: $T(s'|s,a)$, $R(s,a)$, are <u>not</u> changing in time.
  - ↳ Hard to apply these algorithms in dynamically changing environments.
- Markov property $\mathbb{P}(s'|s,a,\text{history}) = T(s'|s,a)$
- Infinite time horizon
  - ↳ Bellman's equations still hold for finite time horizons, but policy/value need to depend on time.
- Scalar rewards + values.
  - ↳ Eg. LLM finetuning. Is the reward/value really scalar??



What are some practical examples in which the assumptions don't hold?

- Markovian ⇒ Partially observable environments.
  - ↳ dynamics + Rewards are <u>not</u> markovian w.r.t. the information available to the agent.
- Rewards/dynamics that depend on time
  - ↳ e.g. wheels on a tire heating up and changing dynamics.
- No proper scalar reward function.



Summary

Bellman's equation and operator

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{\pi}(s') \right] \qquad (T_{\pi}V)(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V(s') \right]$$

Bellman's optimality equation and operator

$$V^{*}(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{*}(s') \right] \qquad (T_{*}V)(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V(s') \right]$$

Policy iteration

Policy evaluation

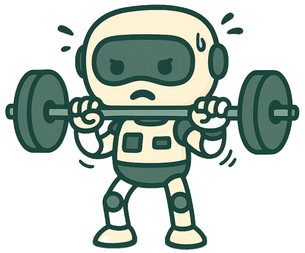
$$V_{\pi} = (I - \gamma P_{\pi})^{-1} R_{\pi}$$

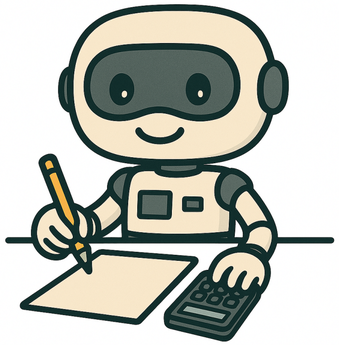
For k = 0, 1, 2, …

$$V_{k+1} \leftarrow T_{\pi} V_k$$

Policy improvement

$$\pi'(s) \in \text{argmax}_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{\pi}(s') \right]$$





Value iteration

For k = 0, 1, 2, …

$$V_{k+1} \leftarrow T_{*} V_k$$

$$\pi^{*}(s) \in \text{argmax}_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s,a) V^{*}(s') \right]$$

Next class

Linear programming solutions to MDPs

&

Connections with optimal control