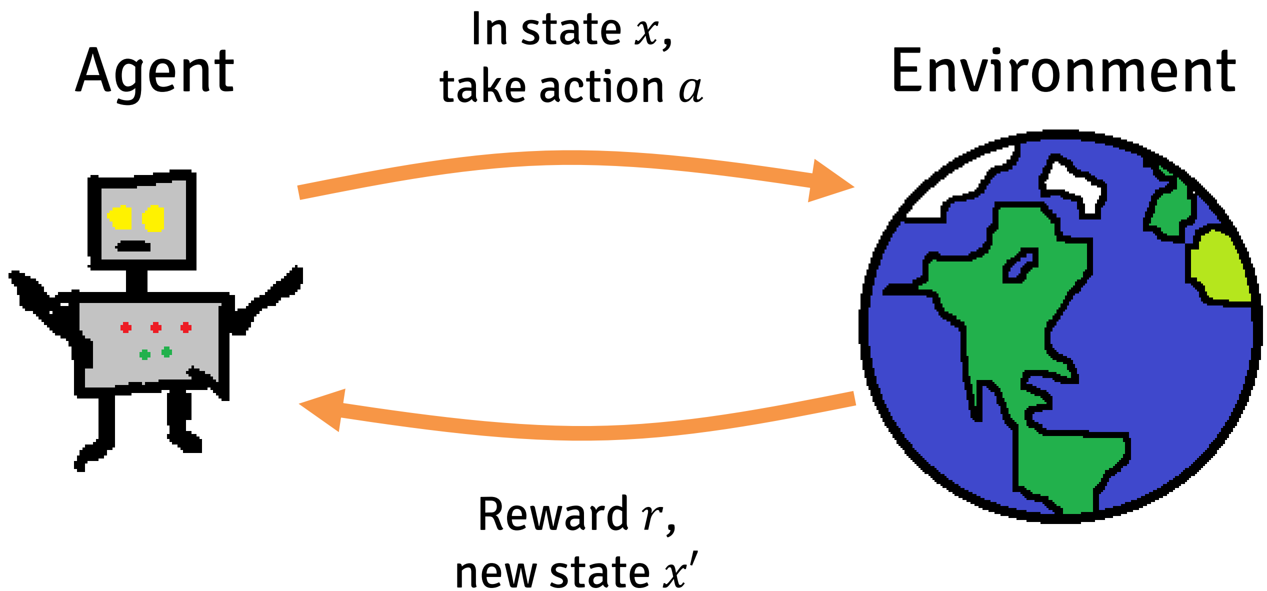
REINFORCEMENT LEARNING

Gergely Neu

upf. Universitat Pompeu Fabra Barcelona

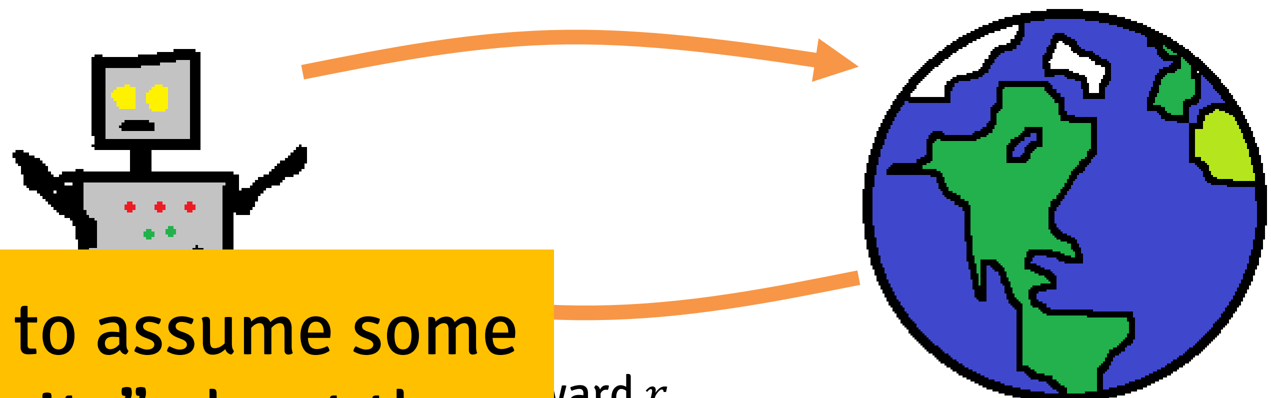
Lecture 2: Dynamic Programming

WHAT IS REINFORCEMENT LEARNING?



Learning to
- maximize reward
- in a reactive environment
- under partial feedback

WHAT IS REINFORCEMENT LEARNING?



We need to assume some "regularity" about the environment so that the agent can predict the evolution of the states

aximize reward
a reactive environment
der partial feedback

EVOLUTION OF THE STATES

The states generally evolve according to the probability distribution

$$x_t \sim P(\cdot \mid x_{t-1}, a_{t-1}, x_{t-2}, a_{t-2}, \dots, x_0, a_0)$$

Problems:

- Long-term planning is intractable: too many paths!
- Learning from experience is very hard: the agent will never get to sample from the same distribution twice!

EVOLUTION OF THE STATES

The states generally evolve according to the probability distribution

$$x_t \sim P(\cdot \mid x_{t-1}, a_{t-1}, x_{t-2}, a_{t-2}, \dots, x_0, a_0)$$

Problems:

- Long-term planning is intractable: too many paths!
- Learning from experience is very hard: the agent will never get to sample from the same distribution twice!

Solution: Markov assumption

THE MARKOV ASSUMPTION

Assumption:
For any t, the states evolve according to
$$x_t \sim P(\cdot\,|x_{t-1}, a_{t-1})$$

Andrey Markov
(1856-1922)

THE MARKOV ASSUMPTION

Assumption:
For any t, the states evolve according to
$$x_t \sim P(\cdot\,|x_{t-1}, a_{t-1})$$

- The history preceding x_{t-1} does not influence the state x_t
- In other words: x_t is a "sufficient statistic" for predicting the future states

Andrey Markov
(1856-1922)

THE MARKOV ASSUMPTION

Assumption:
For any t, the states evolve according to
$$x_t \sim P(\cdot\,|x_{t-1}, a_{t-1})$$

- The history preceding x_{t-1} does not influence the state x_t
- In other words: x_t is a "sufficient statistic" for predicting the future states
- **Advantages:**
  - Easier to plan long-term behavior
  - Easier to learn by experience

Andrey Markov
(1856-1922)

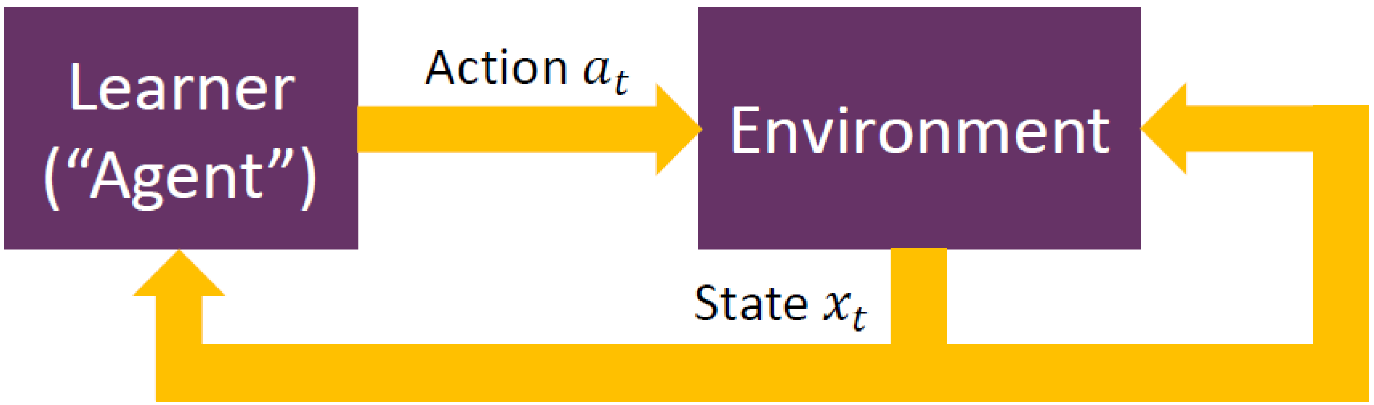
THE MARKOV ASSUMPTION

Assumption:
For any t, the states evolve according to
$$x_t \sim P(\cdot\,|x_{t-1}, a_{t-1})$$

- The history preceding x_{t-1} does not influence the state x_t
- In other words: x_t is a "sufficient statistic" for predicting the future states
- **Advantages:** Dynamic programming
  - Easier to plan long-term behavior
  - Easier to learn by experience

Andrey Markov
(1856-1922)

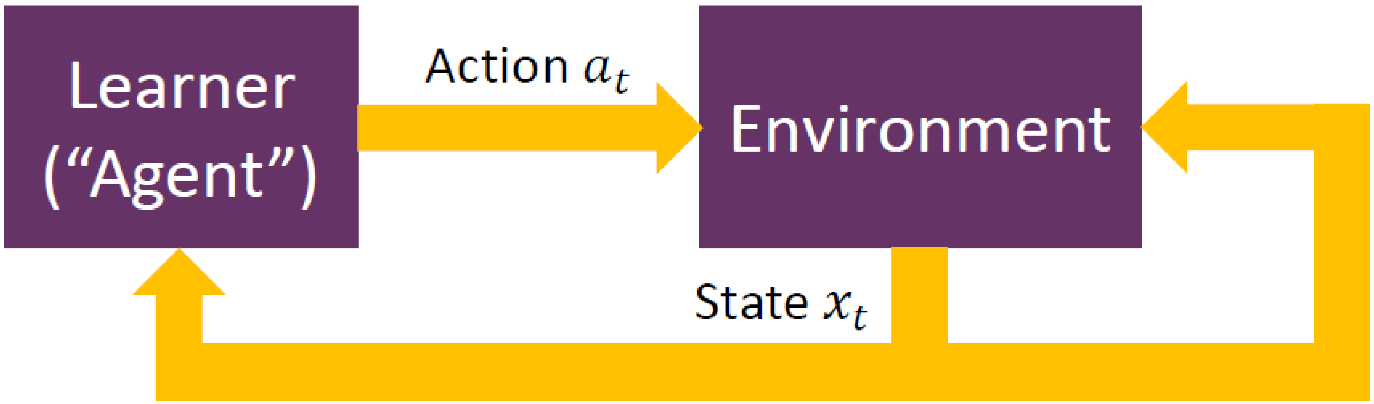
MARKOV DECISION PROCESSES (MDPs)



A Markov Decision Process (MDP) is characterized by

- X: a set of states
- A: a set of actions, possibly different in each state
- P: $X \times A \times X \rightarrow [0,1]$: a transition function with $P(\cdot\,|x,a)$ being the distribution of the next state given previous state x and action a:
$$\mathbf{P}[x_{t+1} = x' | x_t = x, a_t = a] = P(x'|x,a)$$
- $r: X \times A \rightarrow [0,1]$: a reward function

MARKOV DECISION PROCESSES (MDPs)



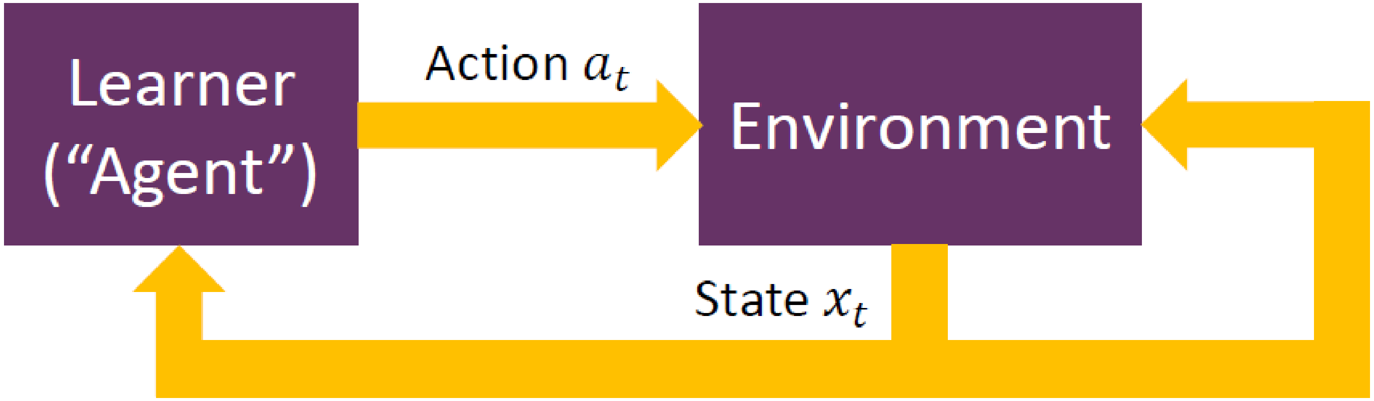
A Markov Decision Process (MDP) is characterized by (X, A, P, r)

- X: a set of states
- A: a set of actions, possibly different in each state
- P: $X \times A \times X \to [0,1]$: a transition function with $P(\cdot\,|x, a)$ being the distribution of the next state given previous state x and action a:

$$\mathbf{P}[x_{t+1} = x' | x_t = x, a_t = a] = P(x' | x, a)$$

- r: $X \times A \to [0,1]$: a reward function

MARKOV DECISION PROCESSES (MDPs)

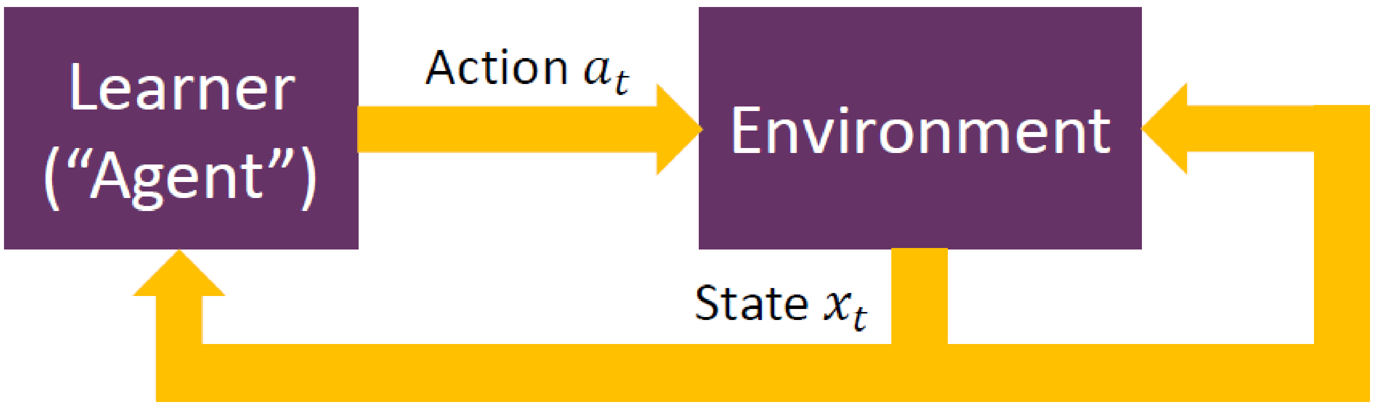


A Markov Decision Process (MDP) is characterized by (X, A, P, r)

Interaction in an MDP: in each round $t = 1,2, \ldots$

- Agent observes state x_t and selects action a_t
- Environment moves to state $x_{t+1} \sim P(\cdot \mid x_t, a_t)$
- Agent receives reward r_t such that $\mathbf{E}[r_t | x_t, a_t] = r(x_t, a_t)$

MARKOV DECISION PROCESSES (MDPs)



A Markov Decision Process (MDP) is characterized by (X, A, P, r)
Interaction in an MDP: in each round $t = 1,2, \dots$

- Agent observes state x_t and selects action a_t
- Environment moves to state $x_{t+1} \sim P(\cdot\,|x_t, a_t)$
- Agent receives reward r_t such that $\mathbf{E}[r_t|x_t, a_t] = r(x_t, a_t)$

GOAL:
maximize "total rewards"!

NOTIONS OF "TOTAL REWARD"

Episodic MDPs:

- There is a terminal state x^*
- GOAL: maximize total reward until final round T when x^* is reached:

$$R^* = \mathbf{E}[\sum_{t=0}^{T} r_t]$$

NOTIONS OF "TOTAL REWARD"

Episodic MDPs:

- There is a terminal state x^*
- GOAL: maximize total reward until final round T when x^* is reached:

$$R^* = \mathbf{E}[\sum_{t=0}^{T} r_t]$$

Discounted MDPs:

- No terminal state
- Discount factor $\gamma \in (0,1)$
- GOAL: maximize total discounted reward (a.k.a. "return")

$$R_\gamma = \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$$

NOTIONS OF “TOTAL REWARD”

+ other notions:
- long-term average reward
- total reward up to fixed horizon
- …

Episodic MDPs:

- There is a terminal state x^*
- GOAL: maximize total reward until final round T when x^* is reached:

$$R^* = \mathbf{E}[\sum_{t=0}^{T} r_t]$$

Discounted MDPs:

- No terminal state
- Discount factor $\gamma \in (0,1)$
- GOAL: maximize total discounted reward (a.k.a. “return”)

$$R_\gamma = \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$$

NOTIONS OF "TOTAL REWARD"

\+ other notions:
- long-term average reward
- total reward up to fixed horizon
- …

Episodic MDPs:
- There is a terminal state x^*
- GOAL: maximize total reward until final round T when x^* is reached:

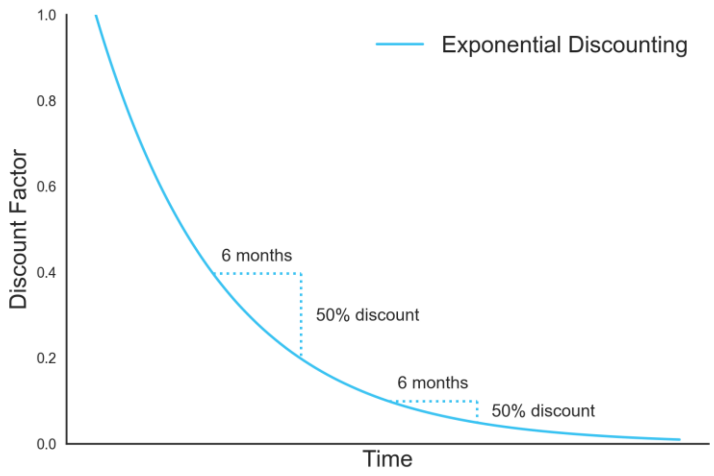
$$R^* = \mathbf{E}[\textstyle\sum_{t=0}^{T} r_t]$$

Discounted MDPs:
- No terminal state
- Discount factor $\gamma \in (0,1)$
- GOAL: maximize total discounted reward (a.k.a. "return")

$$R_\gamma = \mathbf{E}[\textstyle\sum_{t=0}^{\infty} \gamma^t r_t]$$

WHY DISCOUNT?

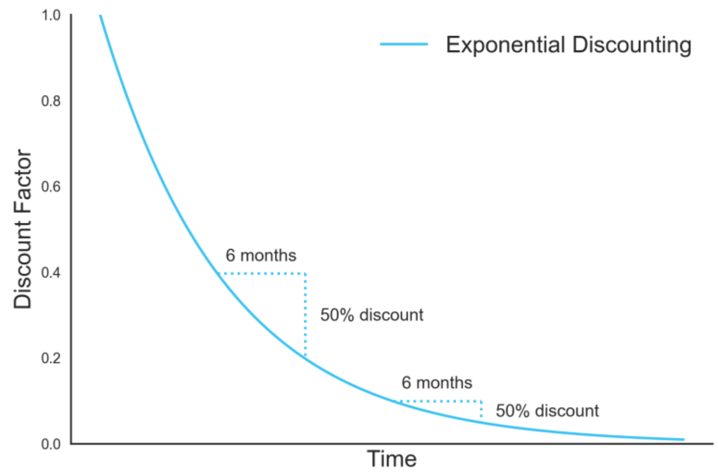
- “Earlier rewards matter more”
- Well-motivated in economics



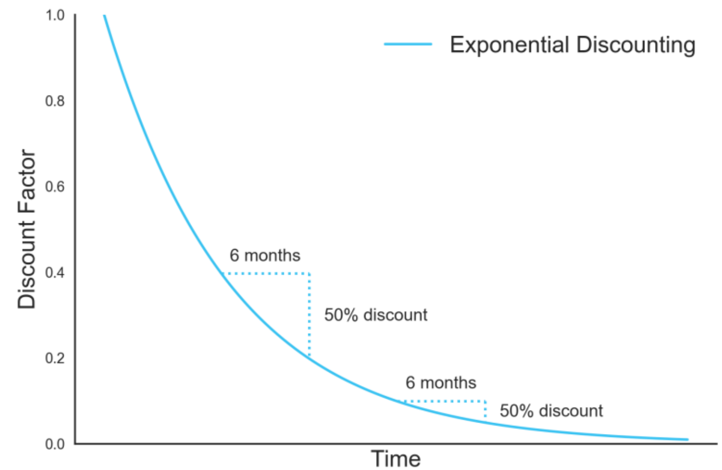
WHY DISCOUNT?

- "Earlier rewards matter more"
- Well-motivated in economics
- Mathematically convenient: if $r_t \in [0, R]$, then

$$\sum_{t=0}^{\infty} \gamma^t r_t \leq R \sum_{t=0}^{\infty} \gamma^t = \frac{R}{1-\gamma}$$



WHY DISCOUNT?



- "Earlier rewards matter more"
- Well-motivated in economics
- Mathematically convenient:
  if $r_t \in [0, R]$, then

$$\sum_{t=0}^{\infty} \gamma^t r_t \leq R \sum_{t=0}^{\infty} \gamma^t = \frac{R}{1-\gamma}$$

- Discounted return blows up as $\gamma \to 1$ and becomes harder to optimize
- The factor $\frac{1}{1-\gamma}$ is sometimes called an "effective time horizon" as rewards after these many steps "don't matter too much"

Dynamic Programming for discounted rewards

Dynamic Programming for discounted rewards

1. basic definitions
2. value functions and optimal policies
3. the Bellman equations
4. value iteration and policy iteration

Dynamic Programming for discounted rewards

1. basic definitions
2. value functions and optimal policies
3. the Bellman equations
4. value iteration and policy iteration

POLICIES AND TRAJECTORY DISTRIBUTIONS

Policy: mapping from histories to actions

$$\pi: x_1, a_1, x_2, a_2, \ldots, x_t \mapsto a_t$$

POLICIES AND TRAJECTORY DISTRIBUTIONS

Policy: mapping from histories to actions

$$\pi: x_1, a_1, x_2, a_2, \ldots, x_t \mapsto a_t$$

Stationary policy: mapping from states to actions
(no dependence on history or t)

$$\pi: x \mapsto a$$

POLICIES AND TRAJECTORY DISTRIBUTIONS

Policy: mapping from histories to actions

$$\pi: x_1, a_1, x_2, a_2, \ldots, x_t \mapsto a_t$$

Stationary policy: mapping from states to actions
(no dependence on history or t)

$$\pi: x \mapsto a$$

Let $\tau = (x_1, a_1, x_2, a_2, \ldots)$ be a trajectory generated by running π in the MDP $\tau \sim (\pi, P)$:

- $a_t = \pi(x_t, a_{t-1}, x_{t-1}, \ldots, x_1)$
- $x_{t+1} \sim P(\cdot \mid x_t, a_t)$

POLICIES AND TRAJECTORY DISTRIBUTIONS

Policy: mapping from histories to actions

$$\pi: x_1, a_1, x_2, a_2, \ldots, x_t \mapsto a_t$$

Stationary policy: mapping from states to actions
(no dependence on history or t)

$$\pi: x \mapsto a$$

Let $\tau = (x_1, a_1, x_2, a_2, \ldots)$ be a trajectory generated by running π in the MDP $\tau \sim (\pi, P)$:

- $a_t = \pi(x_t, a_{t-1}, x_{t-1}, \ldots, x_1)$
- $x_{t+1} \sim P(\cdot \mid x_t, a_t)$

Expectation under this distribution: $\mathbf{E}_\pi[\cdot]$

DEFINING OPTIMALITY

Optimal policy π^*: a policy that maximizes

$$\mathbf{E}_\pi[R_\gamma] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

DEFINING OPTIMALITY

Optimal policy π^*: a policy that maximizes

$$\mathbf{E}_\pi[R_\gamma] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \ldots, x_t) = \pi^*(x_t)$$

DEFINING OPTIMALITY

Optimal policy π^*: a policy that maximizes

$$\mathbf{E}_\pi[R_\gamma] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \dots, x_t) = \pi^*(x_t)$$

Consequence: it's enough to study stationary policies

$$\pi: x \mapsto a$$

DEFINING OPTIMALITY

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \dots, x_t) = \pi^*(x_t)$$

Consequence: it's enough to study stationary policies

$$\pi: x \mapsto a$$

Intuitive "proof": Future transitions $x_{t+1} \sim P(\cdot\,|x_t, a_t)$ do not depend on the previous states $x_1, x_2, \dots$

DEFINING OPTIMALITY

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \dots, x_t) = \pi^*(x_t)$$

Consequence: it's enough to study stationary policies

$$\pi: x \mapsto a$$

Intuitive "proof": Future transitions $x_{t+1} \sim P(\cdot \mid x_t, a_t)$ do not depend on the previous states $x_1, x_2, \dots$

"Markov property"

Dynamic Programming for discounted rewards

1. basic definitions
2. value functions and optimal policies
3. the Bellman equations
4. value iteration and policy iteration

VALUE FUNCTIONS

Value function: evaluates policy π starting from state x:

$$V^{\pi}(x) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^{t} r_{t} \,|x_{0} = x]$$

VALUE FUNCTIONS

Value function: evaluates policy π starting from state x:

$$V^{\pi}(x) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t \,|x_0 = x]$$

Action-value function: evaluates policy π starting from state x and action a:

$$Q^{\pi}(x, a) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t \,|x_0 = x, a_0 = a]$$

VALUE FUNCTIONS

Value function: evaluates policy π starting from state x:

$$V^{\pi}(x) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t \,|x_0 = x]$$

Action-value function: evaluates policy π starting from state x and action a:

$$Q^{\pi}(x, a) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t \,|x_0 = x, a_0 = a]$$

"Optimal policy $\pi^* = \arg\max_{\pi} V^{\pi}(x_0)$"

VALUE FUNCTIONS AND THE OPTIMAL POLICY

Theorem

There exists a policy π^* that satisfies

$$V^{\pi^*}(x) = \max_{\pi} V^{\pi}(x) \quad (\forall x)$$

VALUE FUNCTIONS AND THE OPTIMAL POLICY

Theorem

There exists a policy π^* that satisfies

$$V^{\pi^*}(x) = \max_{\pi} V^{\pi}(x) \quad (\forall x)$$

VALUE FUNCTIONS AND THE OPTIMAL POLICY

Theorem
There exists a policy π^* that satisfies
$$V^{\pi^*}(x) = \max_{\pi} V^{\pi}(x) \quad (\forall x)$$

Optimal policy: a policy π^* that satisfies the above

VALUE FUNCTIONS AND THE OPTIMAL POLICY

Theorem

There exists a policy π^* that satisfies

$$V^{\pi^*}(x) = \max_{\pi} V^{\pi}(x) \quad (\forall x)$$

Optimal policy: a policy π^* that satisfies the above

The optimal value function:

$$V^* = V^{\pi^*}$$

WHY IS THIS IMPORTANT?

- **Previous result** only establishes that for any initial state, there exists an optimal stationary policy maximizing $\mathbf{E}_\pi\left[R_\gamma\right]$

Optimal policy π^*: a policy that maximizes

$$\mathbf{E}_\pi\left[R_\gamma\right] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \dots, x_t) = \pi^*(x_t)$$

WHY IS THIS IMPORTANT?

- **Previous result** only establishes that for any initial state, there exists an optimal stationary policy maximizing $\mathbf{E}_\pi[R_\gamma]$

Optimal policy π^*: a policy that maximizes

$$\mathbf{E}_\pi[R_\gamma] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \dots, x_t) = \pi^*(x_t)$$

- **New result** states that there is a stationary policy π^* that is simultaneously optimal for all initial states x

WHY IS THIS IMPORTANT?

- **Previous result** only establishes that for any initial state, there exists an optimal stationary policy maximizing $\mathbf{E}_\pi\left[R_\gamma\right]$

Optimal policy π^*: a policy that maximizes

$$\mathbf{E}_\pi\left[R_\gamma\right] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Theorem

There exists a deterministic optimal policy π^* such that

$$\pi^*(x_1, a_1, \dots, x_t) = \pi^*(x_t)$$

- **New result** states that there is a stationary policy π^* that is simultaneously optimal for all initial states x

An optimal policy π^* does not "make compromises"

Dynamic Programming for discounted rewards

1. basic definitions
2. value functions and optimal policies
3. the Bellman equations
4. value iteration and policy iteration

THE BELLMAN EQUATIONS

Richard E. Bellman
(1920-1984)

THE BELLMAN EQUATIONS

Theorem

The value function of a stationary policy π satisfies the system of equations $(\forall x \in X)$

$$V^{\pi}(x) = r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x))\, V^{\pi}(y)$$

THE BELLMAN EQUATIONS

Theorem

The value function of a stationary policy π satisfies the system of equations ($\forall x \in X$)

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

Proof:

$$V^\pi(x) = \mathbf{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)\,|x_0 = x]$$

THE BELLMAN EQUATIONS

Theorem

The value function of a stationary policy π satisfies the system of equations ($\forall x \in X$)

$$V^{\pi}(x) = r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x))\, V^{\pi}(y)$$

Proof:

$$\begin{aligned} V^{\pi}(x) &= \mathbf{E}_{\pi}[\textstyle\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)\,|x_0 = x] \\ &= r\big(x, \pi(x)\big) + \mathbf{E}_{\pi}[\textstyle\sum_{t=1}^{\infty} \gamma^t r(x_t, a_t)\,|x_0 = x] \end{aligned}$$

THE BELLMAN EQUATIONS

Theorem

The value function of a stationary policy π satisfies the system of equations ($\forall x \in X$)

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

Proof:

$$V^\pi(x) = \mathbf{E}_\pi[\textstyle\sum_{t=0}^\infty \gamma^t r(x_t, a_t)\,|x_0 = x]$$

$$= r\big(x, \pi(x)\big) + \mathbf{E}_\pi[\textstyle\sum_{t=1}^\infty \gamma^t r(x_t, a_t)\,|x_0 = x]$$

$$= r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x))\, \mathbf{E}_\pi[\textstyle\sum_{t=1}^\infty \gamma^{t-1} r(x_t, a_t)\,|x_1 = y]$$

THE BELLMAN EQUATIONS

Theorem

The value function of a stationary policy π satisfies the system of equations ($\forall x \in X$)

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

Proof:

$$V^\pi(x) = \mathbf{E}_\pi[\textstyle\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)\,|x_0 = x]$$

$$= r\big(x, \pi(x)\big) + \mathbf{E}_\pi[\textstyle\sum_{t=1}^{\infty} \gamma^t r(x_t, a_t)\,|x_0 = x]$$

$$= r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x))\, \mathbf{E}_\pi[\textstyle\sum_{t=1}^{\infty} \gamma^{t-1} r(x_t, a_t)\,|x_1 = y]$$

$$= r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x))\, V^\pi(y) \qquad \blacksquare$$

THE BELLMAN OPTIMALITY EQUATIONS

Theorem

The optimal value function satisfies the system of equations

$$V^*(x) = \max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a)\, V^*(y) \right\}$$

THE BELLMAN OPTIMALITY EQUATIONS

Theorem

The optimal value function satisfies the system of equations

$$V^*(x) = \max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a)\, V^*(y) \right\}$$

Theorem

An optimal policy π^* satisfies

$$\pi^*(x) \in \arg\max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a)\, V^*(y) \right\}$$

OPTIMAL ACTION-VALUE FUNCTIONS

Theorem

The optimal action-value function satisfies

$$Q^*(x,a) = r(x,a) + \gamma \sum_y P(y|x,a) \max_b Q^*(y,b)$$

OPTIMAL ACTION-VALUE FUNCTIONS

Theorem

The optimal action-value function satisfies

$$Q^*(x,a) = r(x,a) + \gamma \sum_y P(y|x,a) \max_b Q^*(y,b)$$

Theorem

An optimal policy π^* satisfies

$$\pi^*(x) \in \arg\max_a Q^*(x,a)$$

OPTIMAL ACTION-VALUE FUNCTIONS

Theorem

The optimal action-value function satisfies

$$Q^*(x,a) = r(x,a) + \gamma \sum_y P(y|x,a) \max_b Q^*(y,b)$$

Theorem

An optimal policy π^* satisfies

$$\pi^*(x) \in \arg\max_a Q^*(x,a)$$

= greedy with respect to Q^*

SHORT SUMMARY SO FAR

So far, we have characterized

- The value functions of a given policy
- The optimal policy through value functions
- The optimal value functions
- The optimal policy through the optimal value functions

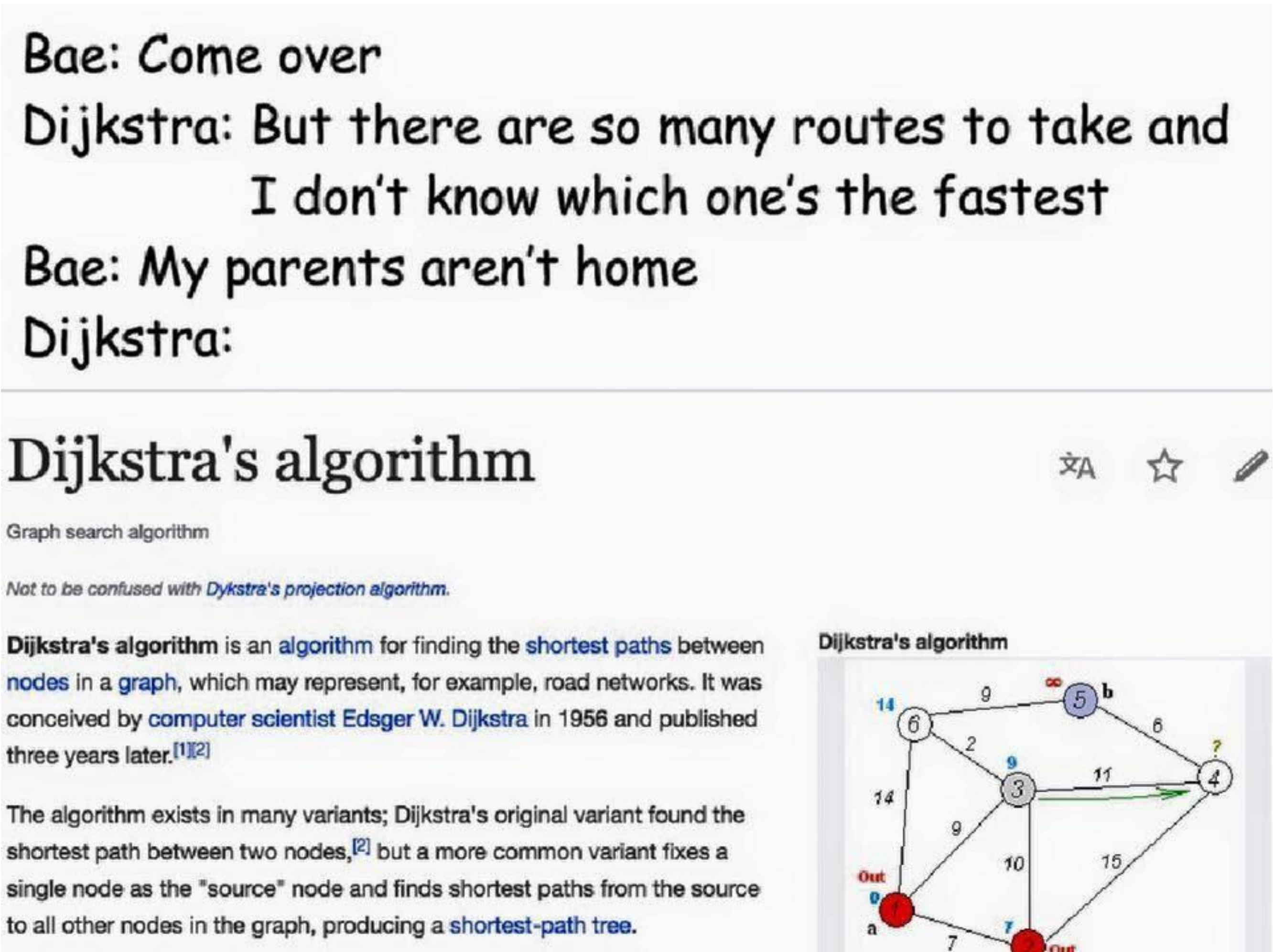
SHORT SUMMARY SO FAR

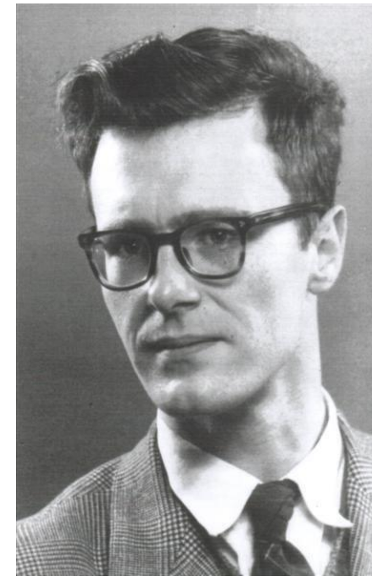
So far, we have characterized

- The value functions of a given policy
- The optimal policy through value functions
- The optimal value functions
- The optimal policy through the optimal value functions

BUT HOW DO WE FIND THE OPTIMAL VALUE FUNCTION??

EASY ANSWER FOR FINITE-HORIZON PROBLEMS





Edsger Dijkstra
(1920-2002)

BELLMAN AND DIJKSTRA

Theorem

The optimal value function satisfies the system of equations

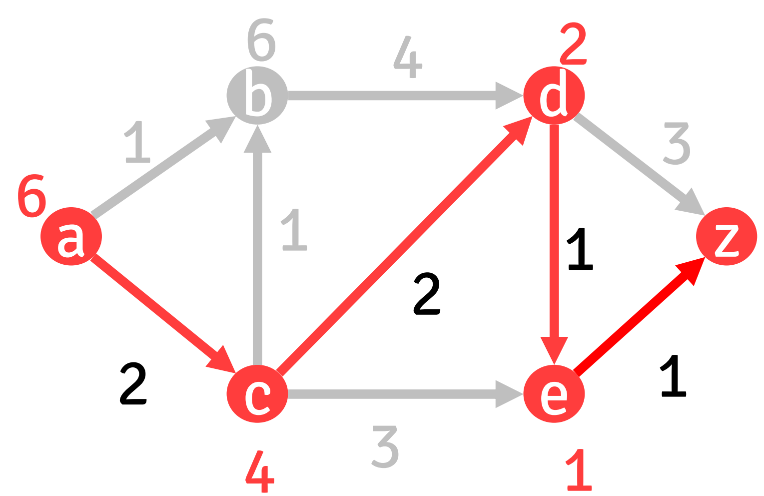
$$V^*(x) = \max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a)\, V^*(y) \right\}$$

Dijkstra:

Cost-to-go = immediate cost
+ future cost-to-go

Bellman:

Value = immediate reward
+ expected future value



Dynamic Programming for discounted rewards

1. basic definitions
2. value functions and optimal policies
3. the Bellman equations
4. value iteration and policy iteration

DYNAMIC PROGRAMMING

Dynamic programming

=

computing value functions through repeated use of the "Bellman operators"

THE BELLMAN OPERATOR

Bellman operator T^{π}:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^{\pi} f \in \mathbb{R}^X$:

$$g(x) = (T^{\pi} f)(x) = r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x)) f(y)$$

THE BELLMAN OPERATOR

Bellman operator T^{π}:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^{\pi} f \in \mathbb{R}^X$:

$$g(x) = (T^{\pi} f)(x) = r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x)) f(y)$$

r.h.s. of BE

THE BELLMAN OPERATOR

Bellman operator T^{π}:
maps a function $f \in \mathbb{R}^{X}$ to another function $g = T^{\pi} f \in \mathbb{R}^{X}$:

$$g(x) = (T^{\pi} f)(x) = r\big(x, \pi(x)\big) + \gamma \sum_{y} P(y|x, \pi(x)) f(y)$$

r.h.s. of BE

The Bellman Equations:

$$V^{\pi}(x) = r\big(x, \pi(x)\big) + \gamma \sum_{y} P(y|x, \pi(x))\, V^{\pi}(y)$$

THE BELLMAN OPERATOR

Bellman operator T^π:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^\pi f \in \mathbb{R}^X$:

$$g(x) = (T^\pi f)(x) = r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x)) f(y)$$

r.h.s. of BE

The Bellman Equations:

$$V^\pi = T^\pi V^\pi$$

THE BELLMAN OPERATOR

Bellman operator T^{π}:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^{\pi} f \in \mathbb{R}^X$:

$$g(x) = (T^{\pi} f)(x) = r\big(x, \pi(x)\big) + \gamma \sum_y P(y|x, \pi(x)) f(y)$$

r.h.s. of BE

The Bellman Equations:

$$V^{\pi} = T^{\pi} V^{\pi}$$

V^{π} is the fixed point of T^{π}

POLICY EVALUATION USING THE BELLMAN OPERATOR

Idea: repeated application of T^{π} on any function V_0 should converge to V^{π}…

POLICY EVALUATION USING THE BELLMAN OPERATOR

Idea: repeated application of T^π on any function V_0 should converge to V^π…

…and it works!!

Power iteration

Input: arbitrary $V_0 : X \to \mathrm{R}$ and π

For $k = 1,2, \ldots,$ compute

$$V_{k+1} = T^\pi V_k$$

POLICY EVALUATION USING THE BELLMAN OPERATOR

Idea: repeated application of T^π on any function V_0 should converge to V^π…

…and it works!!

Power iteration

Input: arbitrary $V_0 : X \to \mathrm{R}$ and π

For $k = 1,2, \ldots ,$ compute

$$V_{k+1} = T^\pi V_k$$

Theorem: $\lim_{k \to \infty} V_k = V^\pi$

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion
  $$V_{k+1} = r + \gamma P^{\pi} V_k$$

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion

$$V_{k+1} = r + \gamma P^{\pi} V_k = r + \gamma P^{\pi}(r + \gamma P^{\pi} V_{k-1})$$

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion

$$
\begin{aligned}
V_{k+1} &= r + \gamma P^{\pi} V_k = r + \gamma P^{\pi}(r + \gamma P^{\pi} V_{k-1}) \\
&= r + \gamma P^{\pi} r + (\gamma P^{\pi})^2 r + \cdots + (\gamma P^{\pi})^k r
\end{aligned}
$$

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion

$$\begin{aligned} V_{k+1} &= r + \gamma P^{\pi} V_k = r + \gamma P^{\pi}(r + \gamma P^{\pi} V_{k-1}) \\ &= r + \gamma P^{\pi} r + (\gamma P^{\pi})^2 r + \cdots + (\gamma P^{\pi})^k r \\ &= \sum_{t=0}^{k} (\gamma P^{\pi})^k r \end{aligned}$$

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion

$$
\begin{aligned}
V_{k+1} &= r + \gamma P^{\pi} V_k = r + \gamma P^{\pi}(r + \gamma P^{\pi} V_{k-1}) \\
&= r + \gamma P^{\pi} r + (\gamma P^{\pi})^2 r + \cdots + (\gamma P^{\pi})^k r \\
&= \sum_{t=0}^{k} (\gamma P^{\pi})^k r \\
&= (I - \gamma P^{\pi})^{-1} \cdot \big(I - (\gamma P^{\pi})^k\big) r
\end{aligned}
$$

Geometric sum!
(von Neumann series)

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion

$$
\begin{aligned}
V_{k+1} &= r + \gamma P^{\pi} V_k = r + \gamma P^{\pi}(r + \gamma P^{\pi} V_{k-1}) \\
&= r + \gamma P^{\pi} r + (\gamma P^{\pi})^2 r + \cdots + (\gamma P^{\pi})^k r \\
&= \sum_{t=0}^{k} (\gamma P^{\pi})^k r \\
&= (I - \gamma P^{\pi})^{-1} \cdot \left(I - (\gamma P^{\pi})^k\right) r \\
&\to (I - \gamma P^{\pi})^{-1} r \qquad\qquad (k \to \infty)
\end{aligned}
$$

Geometric sum!
(von Neumann series)

$(\gamma P^{\pi})^k \to 0$

CONVERGENCE OF POWER ITERATION: PROOF SKETCH

- Power iteration can be written as the linear recursion

$$
\begin{aligned}
V_{k+1} &= r + \gamma P^\pi V_k = r + \gamma P^\pi (r + \gamma P^\pi V_{k-1}) \\
&= r + \gamma P^\pi r + (\gamma P^\pi)^2 r + \cdots + (\gamma P^\pi)^k r \\
&= \sum_{t=0}^{k} (\gamma P^\pi)^k r \\
&= (I - \gamma P^\pi)^{-1} \cdot \left(I - (\gamma P^\pi)^k\right) r \\
&\to (I - \gamma P^\pi)^{-1} r \qquad\qquad (k \to \infty)
\end{aligned}
$$

Geometric sum!
(von Neumann series)

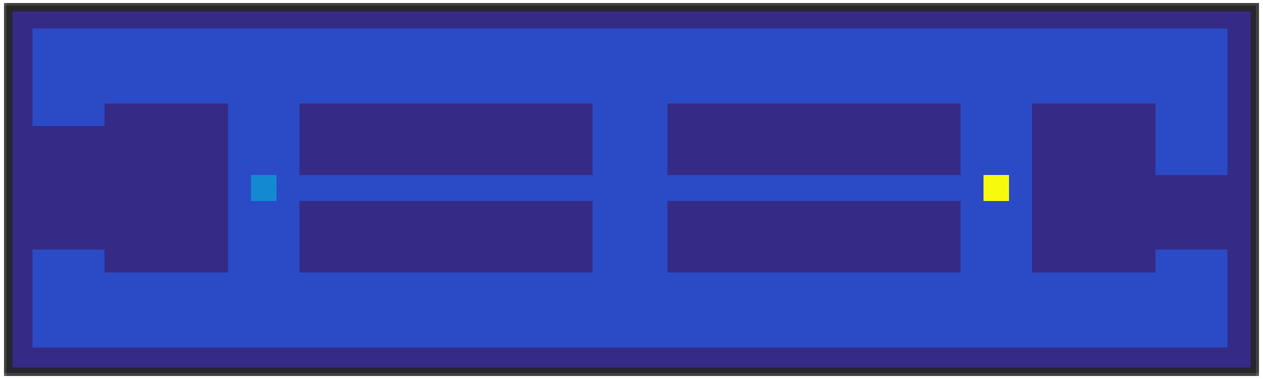
$(\gamma P^\pi)^k \to 0$

- The value function V^π satisfies

$$V^\pi = r + \gamma P^\pi V^\pi \Leftrightarrow V^\pi = (I - \gamma P^\pi)^{-1} r$$

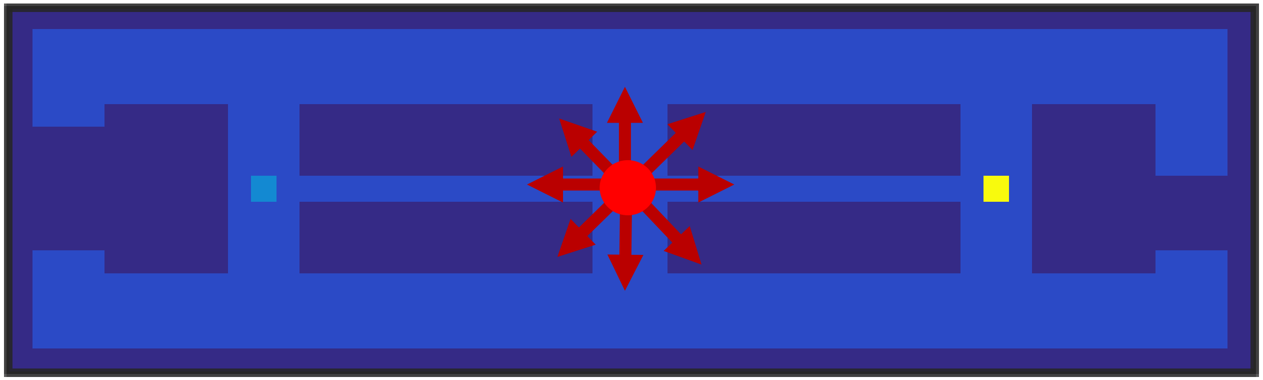
POWER ITERATION IN ACTION

Gridworld MDP



POWER ITERATION IN ACTION

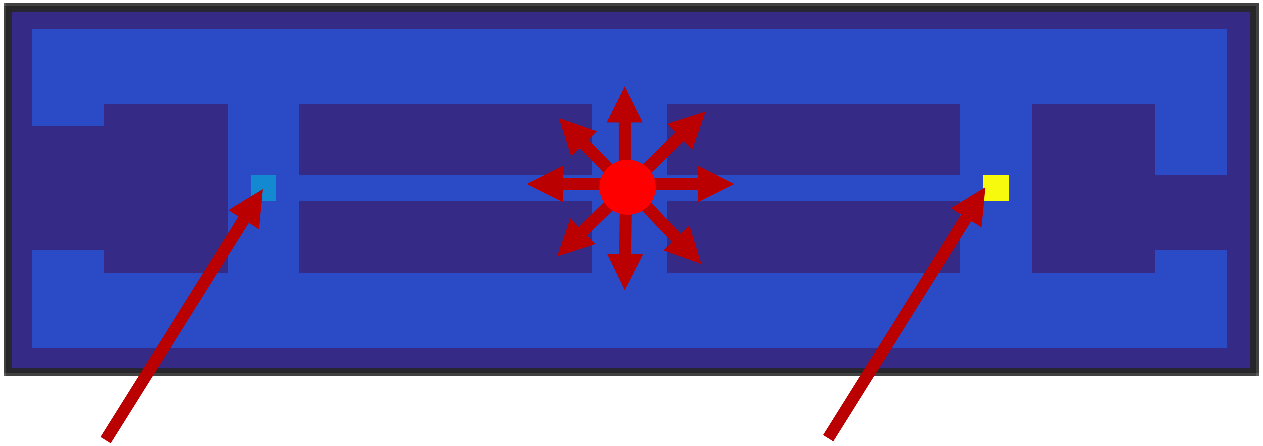
Gridworld MDP



- State: location on the grid
- Actions: try to move in one of 8 directions or stay put
- Transition probabilities:
  - move successfully w.p. $p = 0.5$
  - otherwise move in neighboring direction

POWER ITERATION IN ACTION

Gridworld MDP

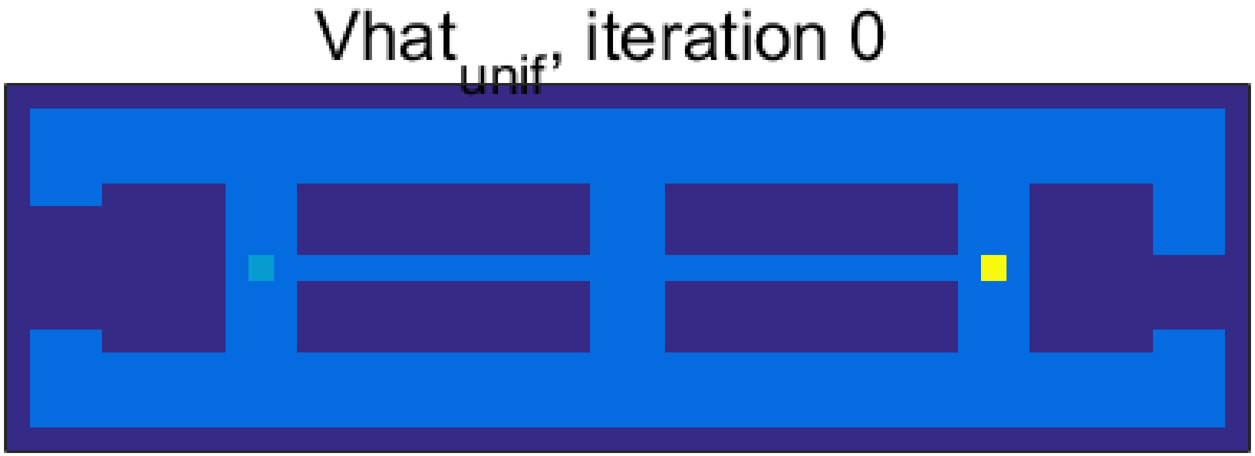


Reward: +100

Reward: +500

- State: location on the grid
- Actions: try to move in one of 8 directions or stay put
- Transition probabilities:
  - move successfully w.p. $p = 0.5$
  - otherwise move in neighboring direction

POWER ITERATION IN ACTION

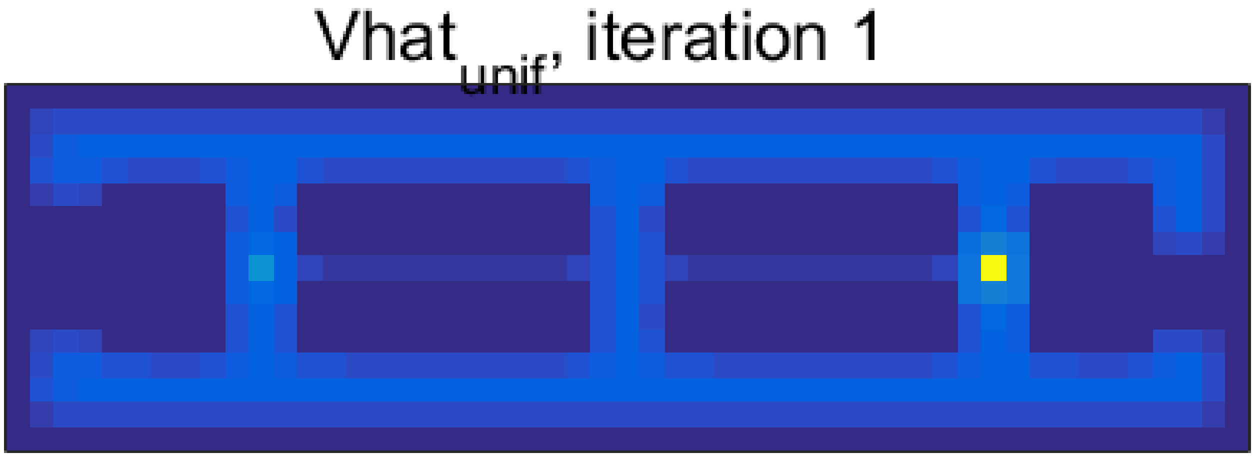


Uniform policy:

$$\pi(a|x) = \frac{1}{9}$$

for all actions $a \in \{1,2,\dots,9\}$

POWER ITERATION IN ACTION

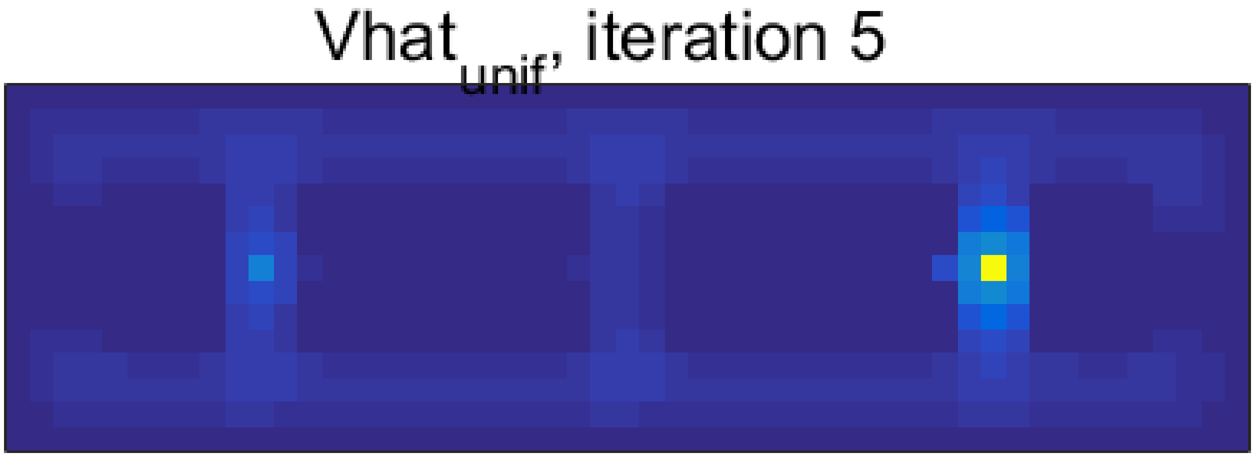


Uniform policy:

$$\pi(a|x) = \frac{1}{9}$$

for all actions $a \in \{1,2, \ldots, 9\}$

POWER ITERATION IN ACTION

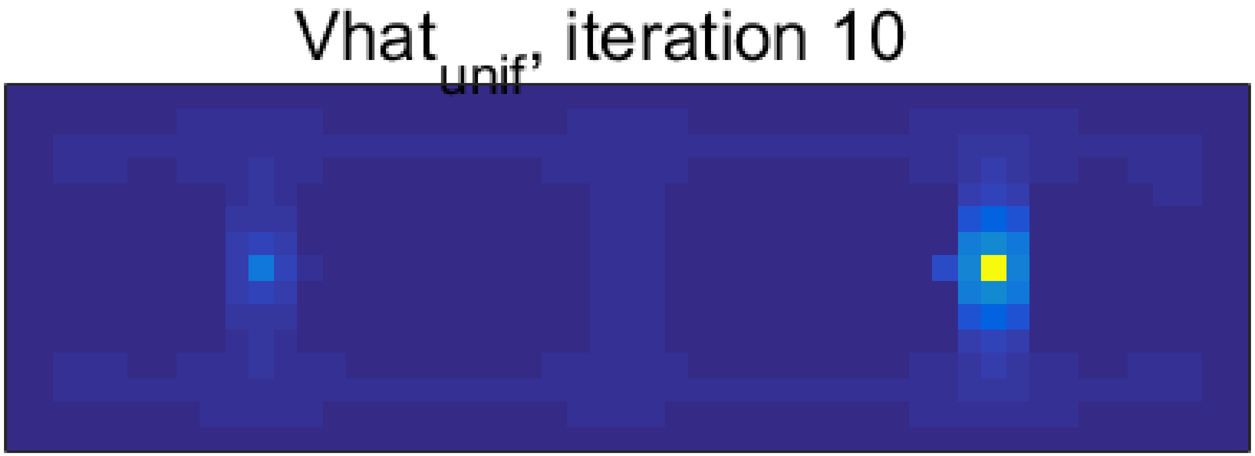


Uniform policy:

$$\pi(a|x) = \frac{1}{9}$$

for all actions $a \in \{1,2,\dots,9\}$

POWER ITERATION IN ACTION

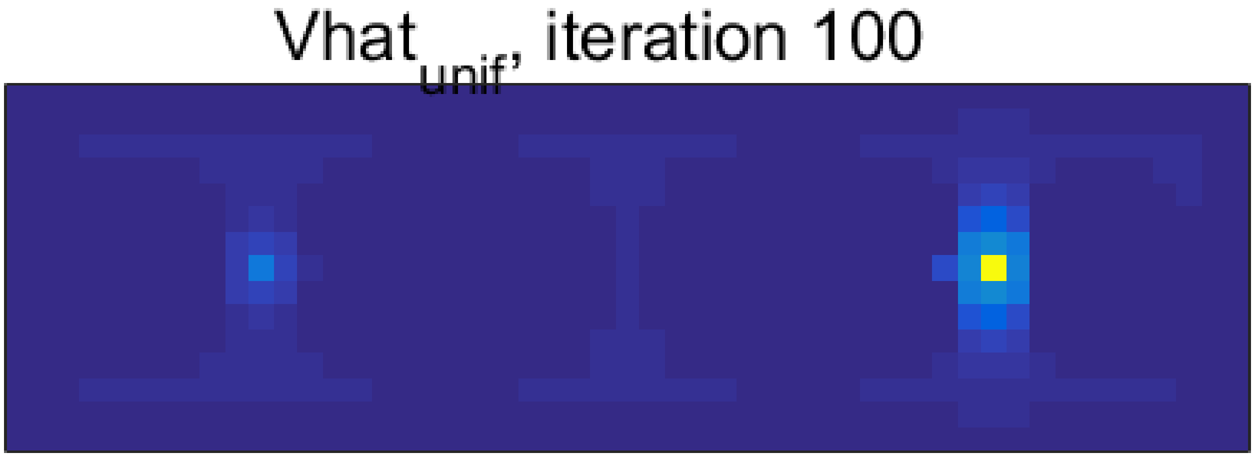


Uniform policy:

$$\pi(a|x) = \frac{1}{9}$$

for all actions $a \in \{1,2, \dots, 9\}$

POWER ITERATION IN ACTION

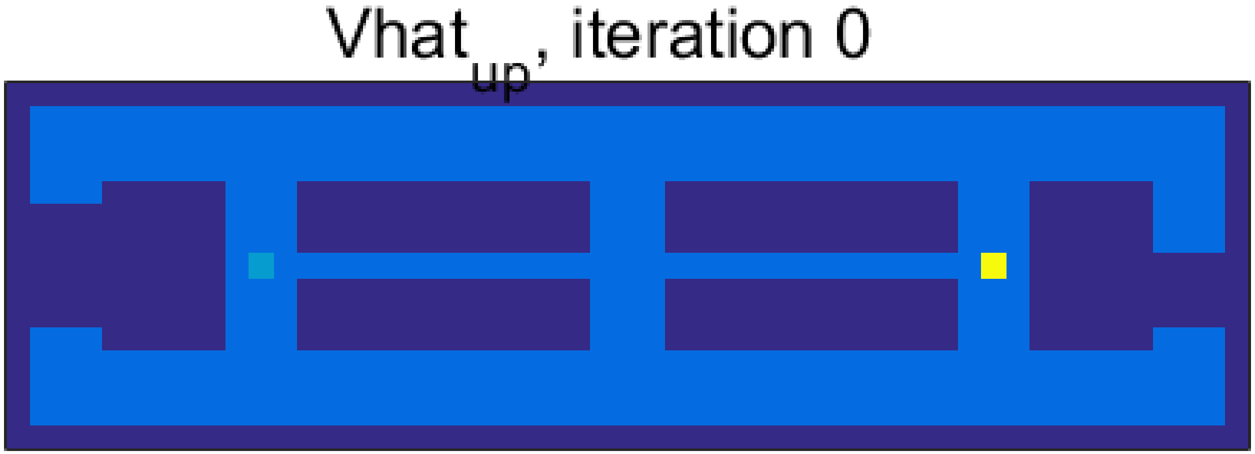


Uniform policy:

$$\pi(a|x) = \frac{1}{9}$$

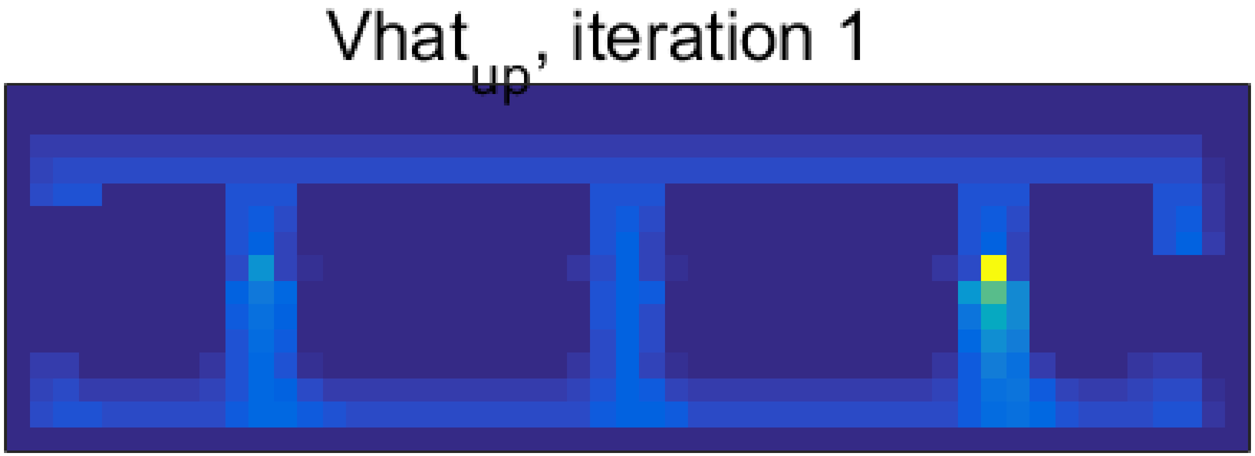
for all actions $a \in \{1,2,\dots,9\}$

POWER ITERATION IN ACTION



"Upwards" policy:
$\pi(\text{up}|x) = 1$

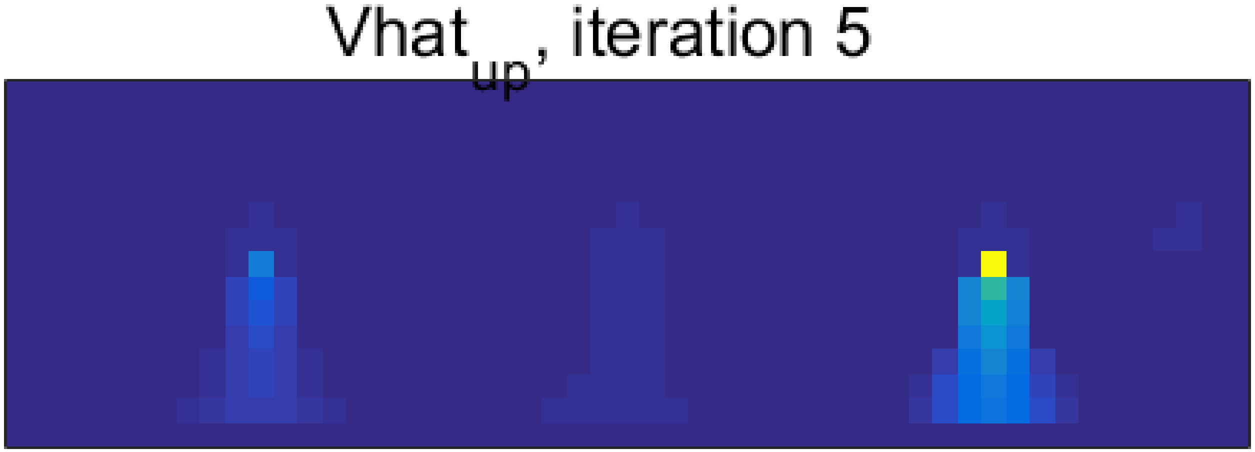
POWER ITERATION IN ACTION



"Upwards" policy:

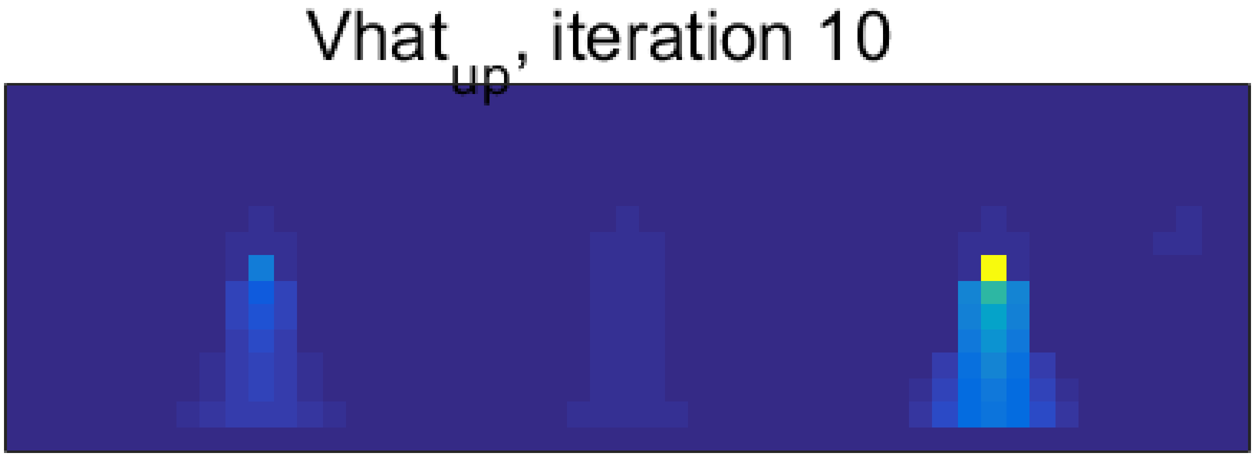
$\pi(\mathrm{up}|x) = 1$

POWER ITERATION IN ACTION



"Upwards" policy:
$\pi(\mathrm{up}|x) = 1$

POWER ITERATION IN ACTION



“Upwards” policy:

$\pi(\mathrm{up}|x) = 1$

THE BELLMAN OPTIMALITY OPERATOR

Bellman optimality operator T^*:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^* f \in \mathbb{R}^X$:

$$g(x) = (T^* f)(x) = \max_a \{r(x,a) + \gamma \sum_y P(y|x,a) f(y)\}$$

THE BELLMAN OPTIMALITY OPERATOR

Bellman optimality operator T^*:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^* f \in \mathbb{R}^X$:

$$g(x) = (T^* f)(x) = \max_a \{r(x,a) + \gamma \sum_y P(y|x,a) f(y)\}$$

r.h.s. of BOE

THE BELLMAN OPTIMALITY OPERATOR

Bellman optimality operator T^*:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^* f \in \mathbb{R}^X$:

$$g(x) = (T^* f)(x) = \max_a \{ r(x,a) + \gamma \sum_y P(y|x,a) f(y) \}$$

r.h.s. of BOE

The Bellman Optimality Equations:

$$V^*(x) = \max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a)\, V^*(y) \right\}$$

THE BELLMAN OPTIMALITY OPERATOR

Bellman optimality operator T^*:
maps a function $f \in \mathbb{R}^X$ to another function $g = T^* f \in \mathbb{R}^X$:

$$g(x) = (T^* f)(x) = \max_a \{r(x,a) + \gamma \sum_y P(y|x,a) f(y)\}$$

r.h.s. of BOE

The Bellman Optimality Equations:

$$V^* = T^* V^*$$

V^* is the fixed point of T^*

VALUE ITERATION

Idea: repeated application of T^* on any function V_0 should converge to V^*...

VALUE ITERATION

Idea: repeated application of T^* on any function V_0 should converge to V^*…

…and it works!!

Value iteration

Input: arbitrary function $V_0 \colon X \to \mathbf{R}$

For $k = 1,2, \ldots ,$ compute

$$V_{k+1} = T^* V_k$$

VALUE ITERATION

Idea: repeated application of T^* on any function V_0 should converge to V^*…

…and it works!!

Value iteration

Input: arbitrary function $V_0 \colon X \to \mathbf{R}$

For $k = 1,2,\ldots,$ compute

$$V_{k+1} = T^* V_k$$

Theorem: $\lim_{k\to\infty} V_k = V^*$

THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a **contraction**

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \leq \gamma \|V - V'\|_\infty$$

THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \leq \gamma \|V - V'\|_\infty$$
- repeated application gives
$$\|V_{k+1} - V^*\|_\infty = \|T^*V_k - T^*V^*\|_\infty$$

THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \le \gamma \|V - V'\|_\infty$$
- repeated application gives
$$\begin{aligned}\|V_{k+1} - V^*\|_\infty &= \|T^*V_k - T^*V^*\|_\infty \\ &\le \gamma \|V_k - V^*\|_\infty\end{aligned}$$

THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \leq \gamma \|V - V'\|_\infty$$
- repeated application gives
$$\begin{aligned}\|V_{k+1} - V^*\|_\infty &= \|T^*V_k - T^*V^*\|_\infty \\ &\leq \gamma \|V_k - V^*\|_\infty \\ &\leq \gamma^2 \|V_{k-1} - V^*\|_\infty\end{aligned}$$

THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \leq \gamma \|V - V'\|_\infty$$
- repeated application gives
$$\begin{aligned} \|V_{k+1} - V^*\|_\infty &= \|T^*V_k - T^*V^*\|_\infty \\ &\leq \gamma \|V_k - V^*\|_\infty \\ &\leq \gamma^2 \|V_{k-1} - V^*\|_\infty \\ &\leq \cdots \leq \gamma^k \|V_0 - V^*\|_\infty \end{aligned}$$

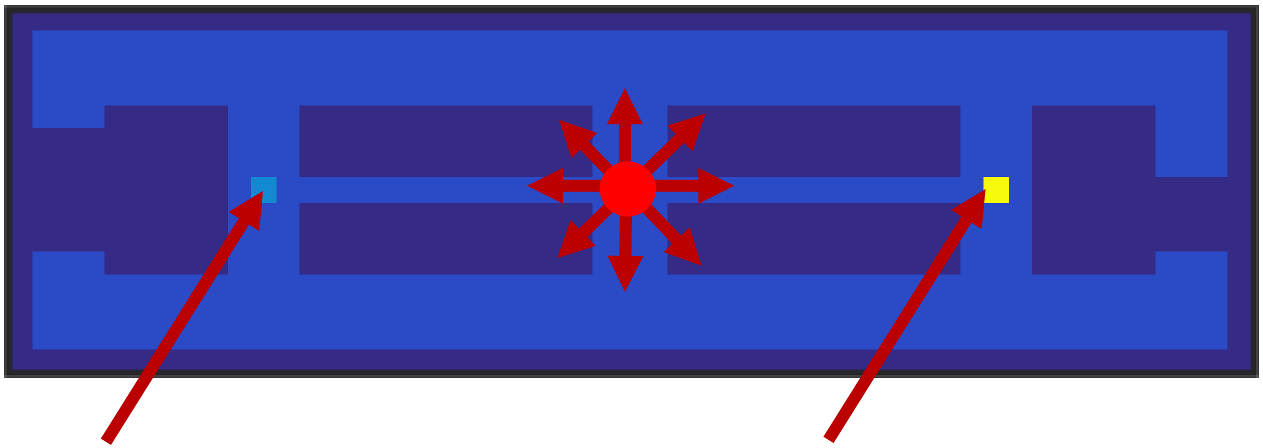
THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \le \gamma\|V - V'\|_\infty$$
- repeated application gives
$$\begin{aligned}\|V_{k+1} - V^*\|_\infty &= \|T^*V_k - T^*V^*\|_\infty \\ &\le \gamma\|V_k - V^*\|_\infty \\ &\le \gamma^2\|V_{k-1} - V^*\|_\infty \\ &\le \cdots \le \gamma^k\|V_0 - V^*\|_\infty\end{aligned}$$
- thus
$$\lim_{k\to\infty}\|V_{k+1} - V^*\|_\infty = 0$$

VALUE ITERATION IN ACTION

Gridworld MDP

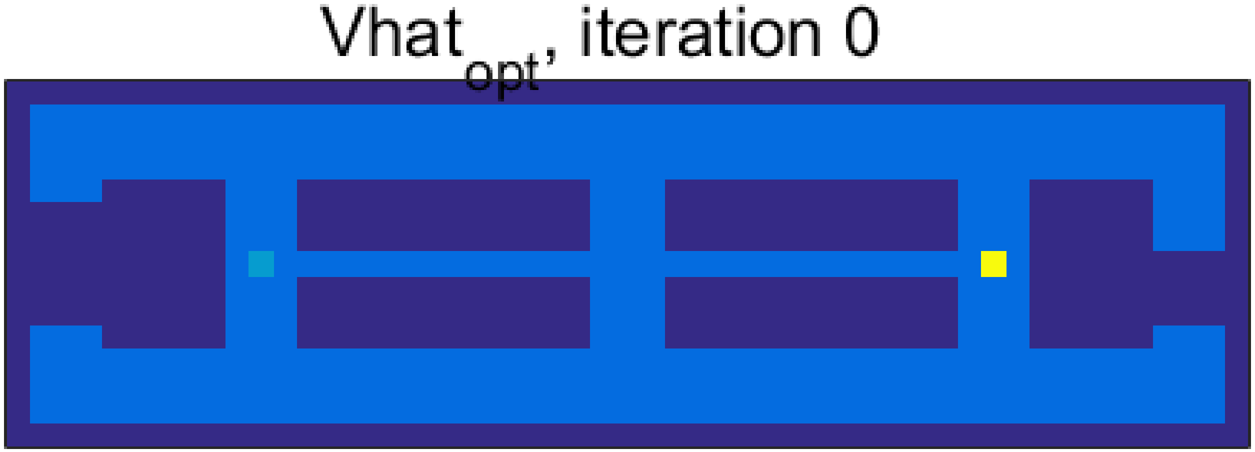


Reward: +100

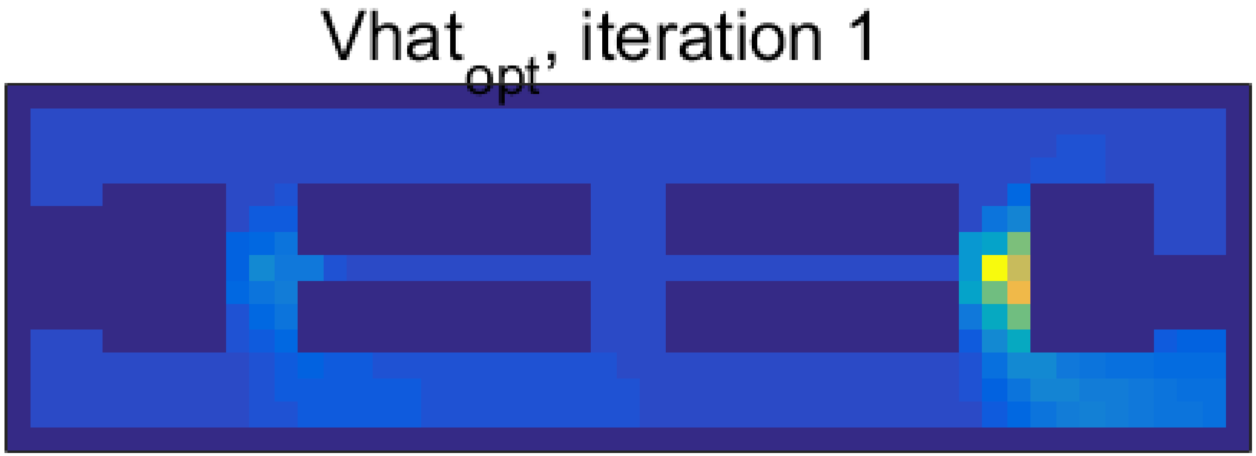
Reward: +500

- State: location on the grid
- Actions: try to move in one of 8 directions or stay put
- Transition probabilities:
    - move successfully w.p. $p = 0.5$
    - otherwise move in neighboring direction

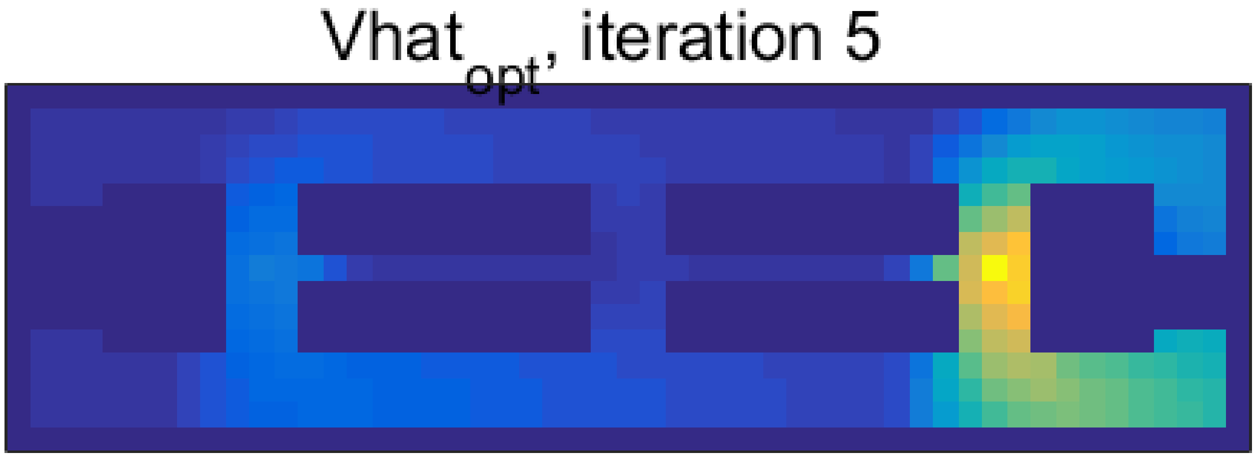
VALUE ITERATION IN ACTION



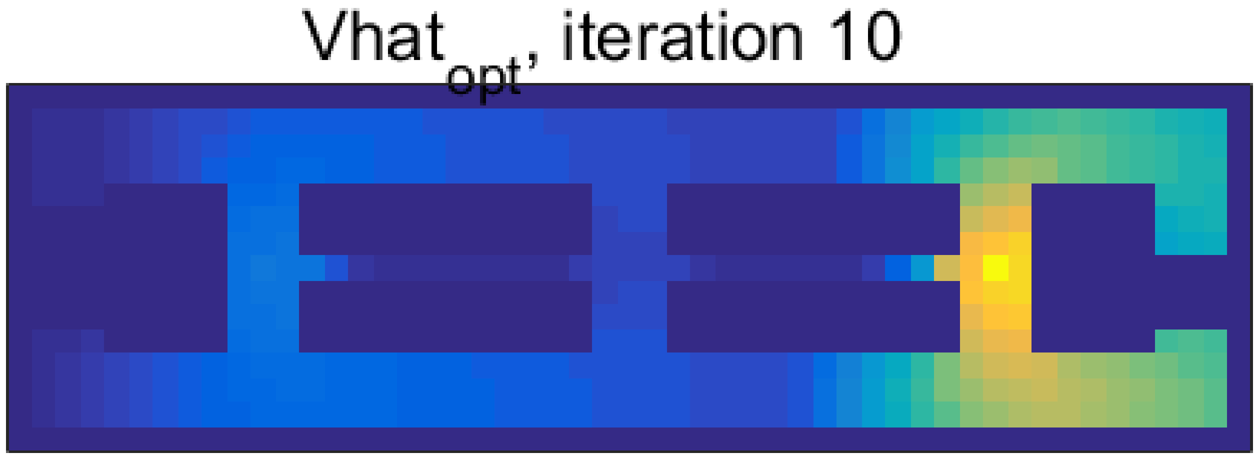
VALUE ITERATION IN ACTION



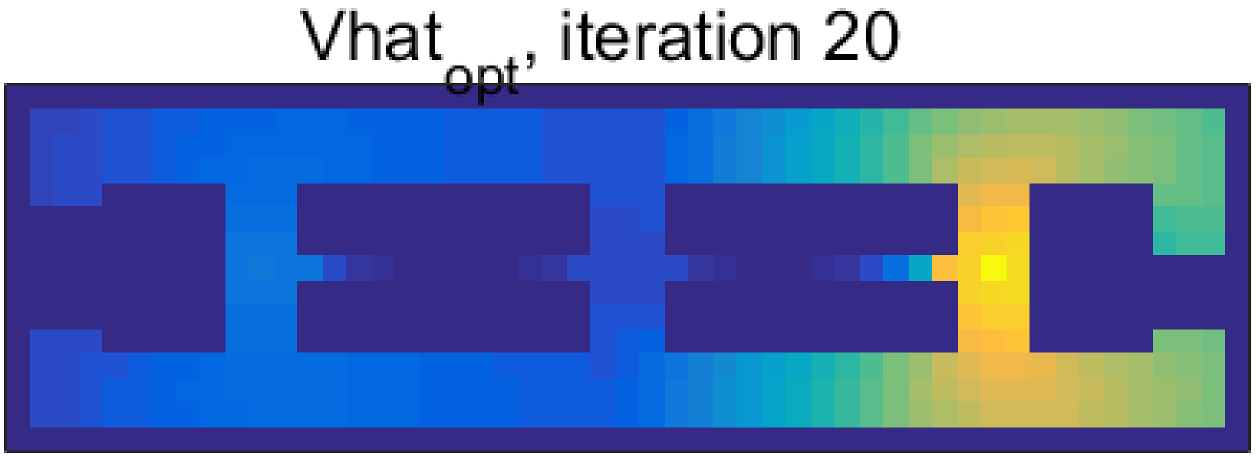
VALUE ITERATION IN ACTION



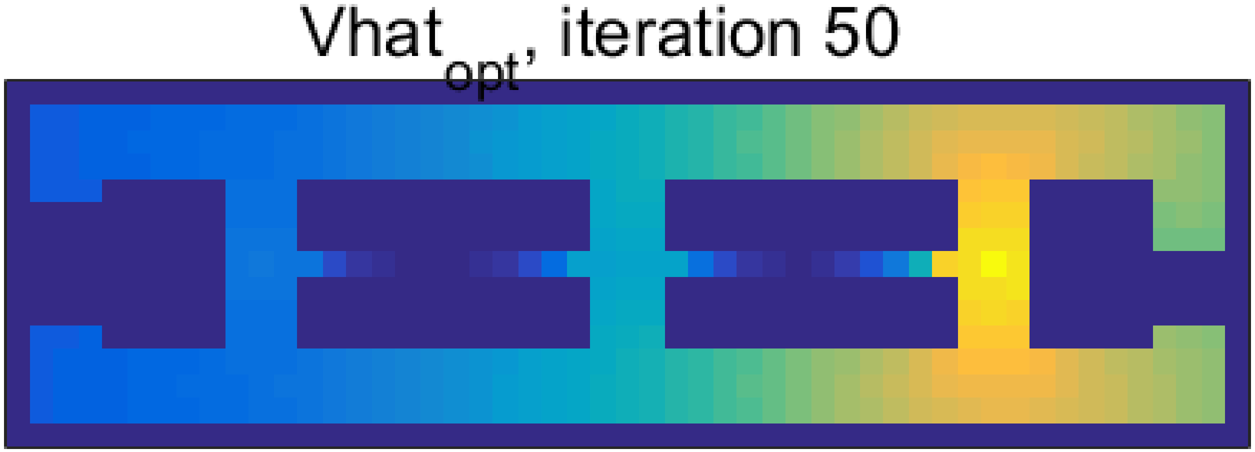
VALUE ITERATION IN ACTION



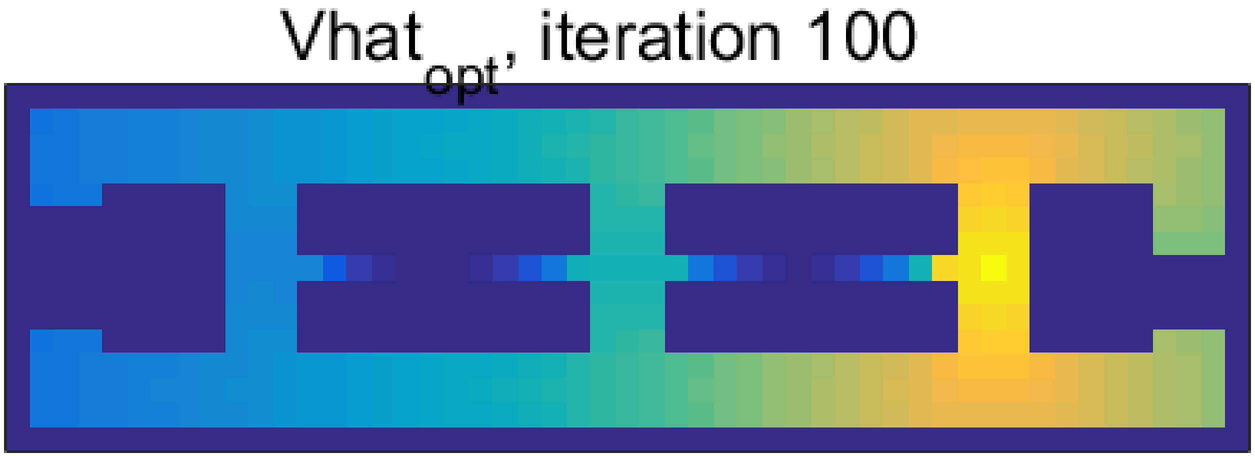
VALUE ITERATION IN ACTION



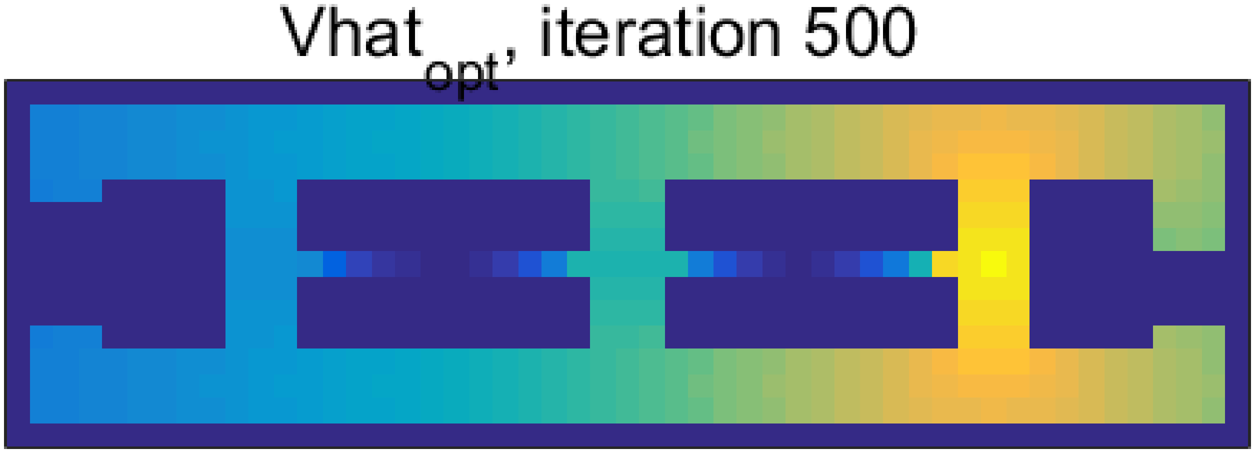
VALUE ITERATION IN ACTION



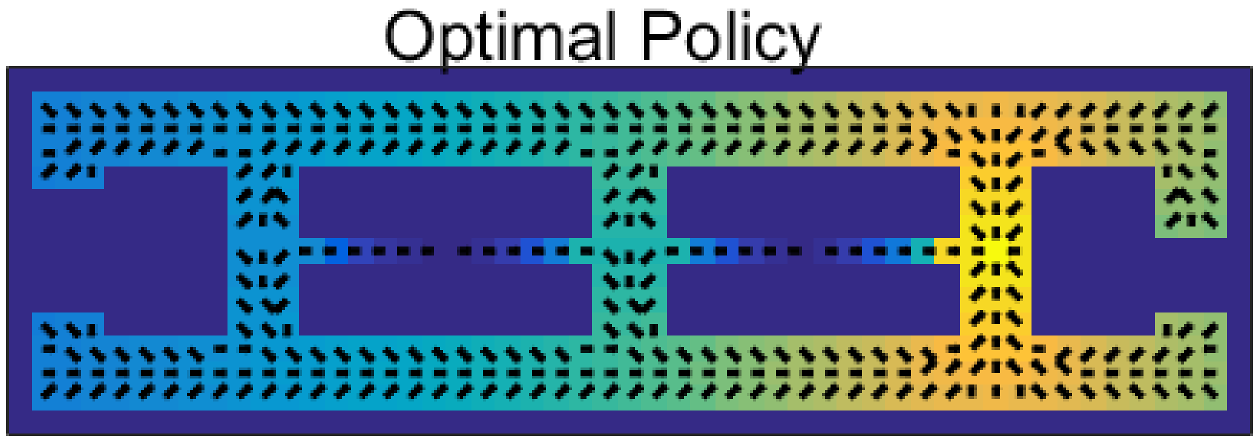
VALUE ITERATION IN ACTION



VALUE ITERATION IN ACTION



VALUE ITERATION IN ACTION



POLICY ITERATION

Greedy policy with respect to V:

$$\pi_V(x) = \arg\max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a) V(x) \right\}$$

POLICY ITERATION

Recall: $\pi^* = \pi_{V^*}$

Greedy policy with respect to V:

$$\pi_V(x) = \arg\max_a \left\{ r(x,a) + \gamma \sum_y P(y|x,a) V(x) \right\}$$

POLICY ITERATION

Recall: $\pi^* = \pi_{V^*}$

Greedy policy with respect to V:

$$\pi_V(x) = \arg\max_a \{r(x,a) + \gamma \sum_y P(y|x,a)V(x)\}$$

Policy Iteration

Input: arbitrary function $V_0 : X \to \mathbf{R}$

For $k = 0,1,\dots,$ compute

$$\pi_k = \pi_{V_k}, \qquad V_{k+1} = V^{\pi_k}$$

POLICY ITERATION

Recall: $\pi^* = \pi_{V^*}$

Greedy policy with respect to V:

$$\pi_V(x) = \arg\max_a \{r(x,a) + \gamma \sum_y P(y|x,a)V(x)\}$$

Policy Iteration

Input: arbitrary function $V_0: X \to \mathbf{R}$

For $k = 0,1, \dots,$ compute

$$\pi_k = \pi_{V_k}, \qquad V_{k+1} = V^{\pi_k}$$

Theorem: $\lim_{k\to\infty} V_k = V^*$

THE CONVERGENCE OF VALUE ITERATION: PROOF SKETCH

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \leq \gamma \|V - V'\|_\infty$$
- repeated application gives
$$\begin{aligned} \|V_{k+1} - V^*\|_\infty &= \|T^*V_k - T^*V^*\|_\infty \\ &\leq \gamma \|V_k - V^*\|_\infty \\ &\leq \gamma^2 \|V_{k-1} - V^*\|_\infty \\ &\leq \cdots \leq \gamma^k \|V_0 - V^*\|_\infty \end{aligned}$$
- thus
$$\lim_{k \to \infty} \|V_{k+1} - V^*\|_\infty = 0$$

THE CONVERGENCE OF ~~VALUE~~ policy ITERATION: PROOF SKETCH

Just replace T^* with the operator
$B^*: f \mapsto (T^{\pi_f})^\infty$

Key idea: T^* is a contraction

- for any two functions V and V', we have
$$\|T^*V - T^*V'\|_\infty \le \gamma \|V - V'\|_\infty$$
- repeated application gives
$$\begin{aligned}\|V_{k+1} - V^*\|_\infty &= \|T^*V_k - T^*V^*\|_\infty \\ &\le \gamma \|V_k - V^*\|_\infty \\ &\le \gamma^2 \|V_{k-1} - V^*\|_\infty \\ &\le \cdots \le \gamma^k \|V_0 - V^*\|_\infty\end{aligned}$$
- thus
$$\lim_{k\to\infty} \|V_{k+1} - V^*\|_\infty = 0$$

EPILOGUE

from
Dynamic Programming
to
Reinforcement Learning

FROM DYNAMIC PROGRAMMING TO VALUE-BASED REINFORCEMENT LEARNING

Policy iteration:

$$V_k$$

FROM DYNAMIC PROGRAMMING TO VALUE-BASED REINFORCEMENT LEARNING

Policy iteration:

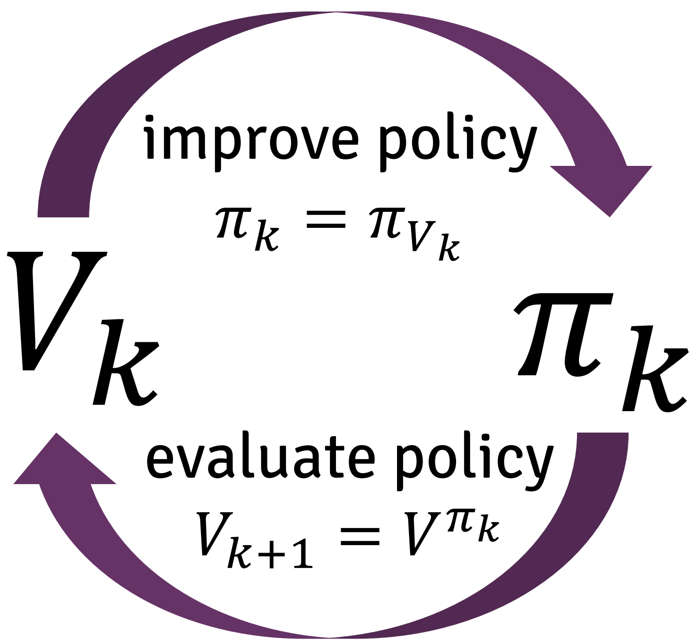
improve policy

$$\pi_k = \pi_{V_k}$$

$$V_k \qquad \pi_k$$

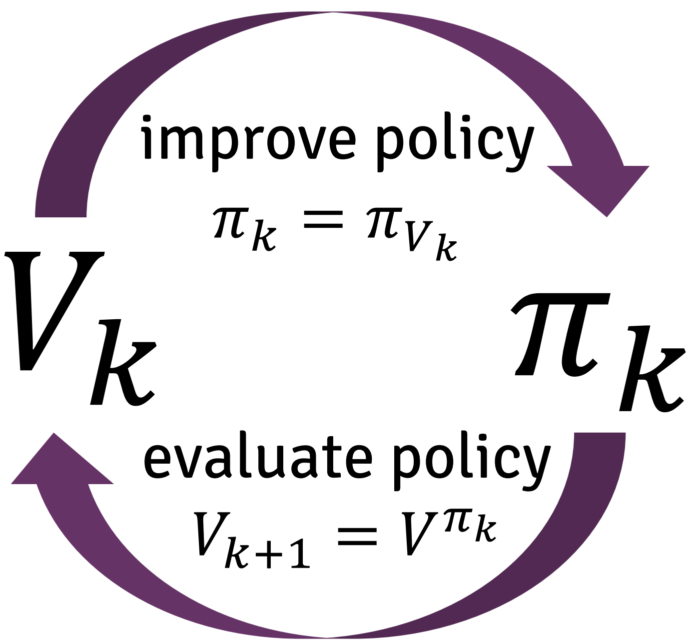
FROM DYNAMIC PROGRAMMING TO VALUE-BASED REINFORCEMENT LEARNING

Policy iteration:



FROM DYNAMIC PROGRAMMING TO VALUE-BASED REINFORCEMENT LEARNING

Policy iteration:



Approximate policy iteration: