

535514: Reinforcement Learning

Lecture 4 – Value Iteration, Policy Iteration, and Regularized MDPs

Ping-Chun Hsieh

March 4, 2024

This Lecture: We Discuss 3 Surprising Facts of MDPs!

1. Value iteration (VI) can find an optimal policy

(VI → Value-based RL, e.g., Q-learning)

2. Policy iteration (PI) can also find an optimal policy

(PI → Policy-based RL, e.g., Policy Gradient, PPO, ...)

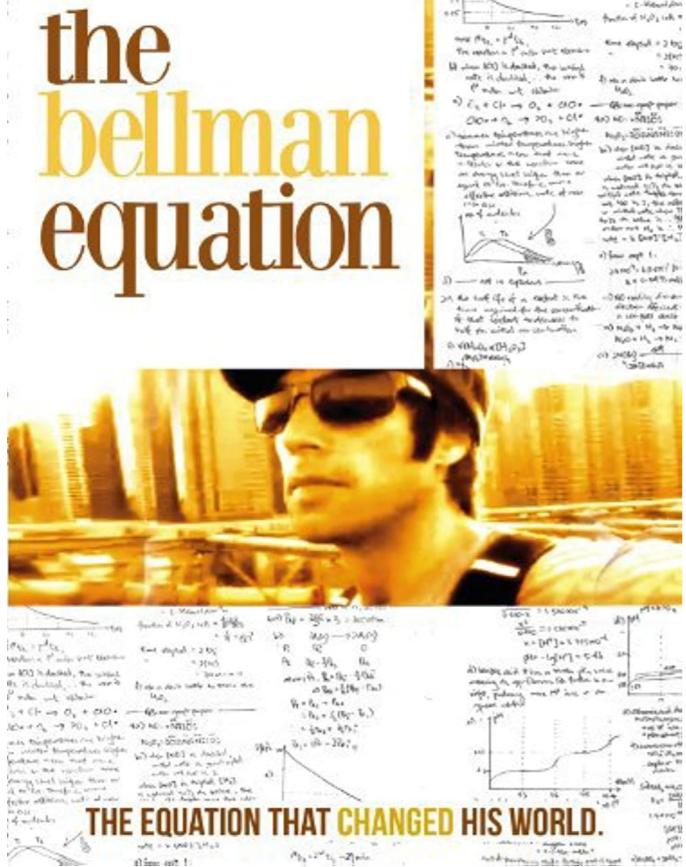
3. “Existence” of an optimal policy for MDPs

Bellman Equations: Some Historical Accounts



Richard Bellman
(1920-1984)

A seminal paper on “Value Iteration”
(published in 1957)



R. Bellman is a member of the Manhattan Project

A documentary released in 2011



A Markovian Decision Process

RICHARD BELLMAN

1. Introduction. The purpose of this paper is to discuss the asymptotic behavior of the sequence $\{f_N(i)\}$, $i = 1, 2, \dots, M$, $N = 1, 2, \dots$, generated by the non-linear recurrence relations

$$(1) \quad f_N(i) = \max_{\mathbf{q}} \left[b_i(\mathbf{q}) + \sum_{j=1}^M a_{ij}(\mathbf{q}) f_{N-1}(j) \right], \quad N = 1, 2, \dots,$$
$$f_0(i) = c_i, \quad i = 1, 2, \dots, M.$$

Although these equations are non-linear, they possess certain quasi-linear properties which permit a more thorough discussion than might be imagined upon first glance.

As we shall discuss below, this question arises from the consideration of a dynamic programming process. A related process gave rise to an equation of the above form which was discussed in [1], under a particular set of assumptions concerning the functions $b_i(\mathbf{q})$ and the matrices $A(\mathbf{q}) = (a_{ij}(\mathbf{q}))$. Here we shall impose restrictions of a quite different type. Any complete discussion of relations of the foregoing type is at least as detailed as a corresponding discussion of the linear case, and, as in the linear case, the assumptions made determine the techniques employed to a considerable extent.

We shall discuss elsewhere some interesting quadratically non-linear recurrence relations which arise from specializing the forms of $b_i(\mathbf{q})$ and $a_{ij}(\mathbf{q})$. These are related to the types of differential equations discussed in [3], being essentially a particular type of discrete version.

It will be clear from what follows that similar techniques can be utilized to treat the determination of the asymptotic behavior of the sequences defined by

Bellman Equations and *Dynamic Programming*

Why using the name “dynamic programming”?

RICHARD BELLMAN ON THE BIRTH OF
DYNAMIC PROGRAMMING

STUART DREYFUS

University of California, Berkeley, IEOR, Berkeley, California 94720, dreyfus@ieor.berkeley.edu

What follows concerns events from the summer of 1949, when Richard Bellman first became interested in multistage decision problems, until 1955. Although Bellman died on March 19, 1984, the story will be told in his own words since he left behind an entertaining and informative autobiography, *Eye of the Hurricane* (World Scientific Publishing Company, Singapore, 1984), whose publisher has generously approved extensive excerpting.

During the summer of 1949 Bellman, a tenured associate professor of mathematics at Stanford University with a developing interest in analytic number theory, was consulting for the second summer at the RAND Corporation in Santa Monica. He had received his Ph.D. from Princeton in 1946 at the age of 25, despite various war-related activities during World War II—including being assigned by the Army to the Manhattan Project in Los Alamos. He had already exhibited outstanding ability both in pure mathematics and in solving applied problems arising from the physical world. Assured of a successful conventional academic career, Bellman, during the period under consideration, cast his lot instead with the kind of applied mathematics later to be known as operations research. In those days applied practitioners were regarded as distinctly second-class citizens of the mathematical fraternity. Always one to enjoy controversy, when invited to speak at various university mathematics department seminars, Bellman delighted in justifying his choice of applied over pure mathematics as being motivated by the real world's greater challenges and mathematical demands.

Following are excerpts, taken chronologically from Richard Bellman's autobiography. The page numbers are given after each. The excerpt section titles are mine. These excerpts are far more serious than most of the book, which is full of entertaining anecdotes and outrageous behaviors by an exceptionally *human* being.

Stuart Dreyfus

BELLMAN'S INTRODUCTION TO MULTISTAGE
DECISION PROCESS PROBLEMS

what RAND was interested in. He suggested that I work on multistage decision processes. I started following that suggestion" (p. 157).

CHOICE OF THE NAME DYNAMIC PROGRAMMING

"I spent the Fall quarter (of 1950) at RAND. My first task was to find a name for multistage decision processes.

"An interesting question is, 'Where did the name, dynamic programming, come from?' The 1950s were not good years for mathematical research. We had a very interesting gentleman in Washington named Wilson. He was Secretary of Defense, and he actually had a pathological fear and hatred of the word, research. I'm not using the term lightly; I'm using it precisely. His face would suffuse, he would turn red, and he would get violent if people used the term, research, in his presence. You can imagine how he felt, then, about the term, mathematical. The RAND Corporation was employed by the Air Force, and the Air Force had Wilson as its boss, essentially. Hence, I felt I had to do something to shield Wilson and the Air Force from the fact that I was really doing mathematics inside the RAND Corporation. What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word, 'programming.' I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying—I thought, let's kill two birds with one stone. Let's take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. It also has a very interesting property as an adjective, and that is it's impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible. Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities" (p. 159).

EARLY ANALYTICAL RESULTS

"We had a very interesting gentleman in Washington named Wilson. He was Secretary of Defense, and he actually had a pathological fear and hatred of the word, *research*. I'm not using the term lightly; I'm using it precisely. His face would suffuse, he would turn red, and he would get violent if people used the term, *research*, in his presence. You can imagine how he felt, then, about the term, *mathematical*. The RAND Corporation was employed by the Air Force, and the Air Force had Wilson as its boss, essentially. Hence, I felt I had to do something to shield Wilson and the Air Force from the fact that I was really doing mathematics inside the RAND Corporation. What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word, 'programming.' I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying....Let's take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense ...Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities."

2-Minute Review: What We Learned in Lecture 3

- ▶ Optimal value function $V^*(s)$:
$$V^*(s) := \max_{\pi \in \Pi} V^\pi(s)$$
- ▶ Optimal action-value function $Q^*(s, a)$:
$$Q^*(s, a) := \max_{\pi \in \Pi} Q^\pi(s, a)$$
- ▶ Optimal policy π^* :
$$\pi^* \geq \pi, \text{ for all } \pi \in \Pi$$

\uparrow

$$V^{\pi^*}(s) \geq V^\pi(s), \text{ for all } s \in S.$$
- ▶ Existence of an optimal policy (to be proved):
$$\text{For any finite MDP, there exists at least one } \pi^*.$$

Review: Bellman Optimality Equations

(1) V^* written in Q^*

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

(2) Q^* written in V^*

$$Q^*(s, a) = R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V^*(s')$$

(3) V^* written in V^*

$$V^*(s) = \max_{a \in \mathcal{A}} R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V^*(s')$$

(4) Q^* written in Q^*

$$Q^*(s, a) = R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \left(\max_{a \in \mathcal{A}} Q^*(s', a) \right)$$

How to Solve the Bellman Optimality Equation?

$$V^*(s) = \max_{a \in \mathcal{A}} \left(R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V^*(s') \right)$$

- ▶ **Question:** Solve this by linear algebra?
- ▶ The “max” operation makes it non-linear
- ▶ We need to resort to iterative methods:
 - ▶ Value iteration
 - ▶ Policy iteration

1. Value Iteration (VI)

From Bellman Optimality Equation to “Bellman Optimality Backup Operator”

- ▶ Recall: Bellman optimality equation

$$V^*(s) = \max_{a \in \mathcal{A}} \left(R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V^*(s') \right)$$

- ▶ Define: Bellman optimality backup operator T^* : $\mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|}$

$$T^*(V) := \max_{a \in \mathcal{A}} \left(R^a + \gamma P^a V \right)$$

(Comparison: IPE operator $T^\pi(V) := R^\pi + \gamma P^\pi V$)

Value Iteration: Pseudo Code

Step 1. Initialize $k = 0$ and set $V_0(s) = 0$ for all states

Step 2. Repeat the following until convergence:

$$V_{k+1} \leftarrow T^*(V_k)$$

- ▶ Equivalently: for each state s

$$V_{k+1}(s) \leftarrow \max_{a \in \mathcal{A}} \left(R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V_k(s') \right)$$

- ▶ **Remark:** Complexity per iteration is $O(|\mathcal{S}|^2|\mathcal{A}|)$
- ▶ **Remark:** Intermediate value functions V_k 's may not correspond to any policy

Example: Shortest Path

$$V_{k+1}(s) \leftarrow \max_{a \in \mathcal{A}} \left(R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V_k(s') \right)$$

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

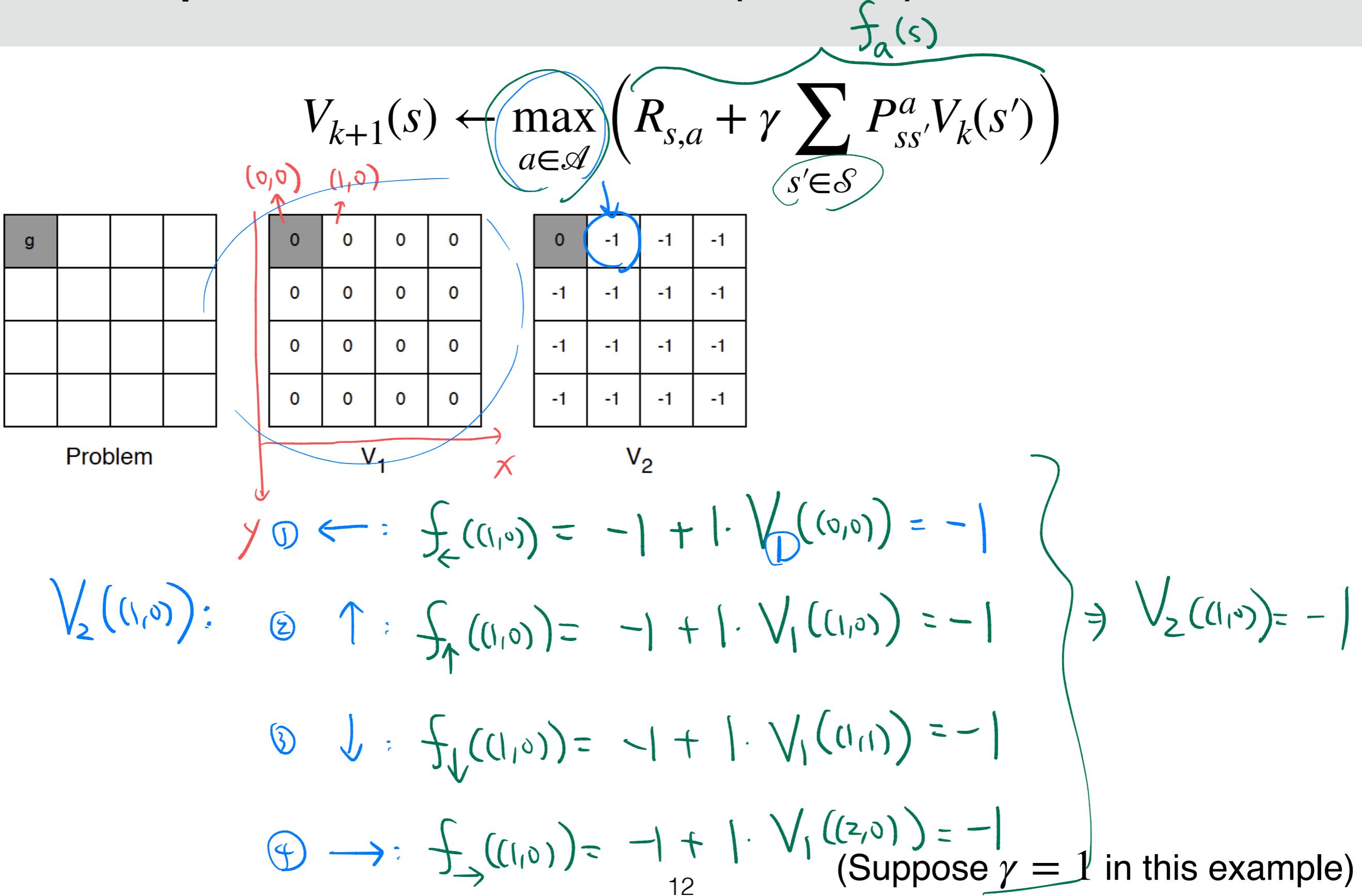
0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

Example: Shortest Path (Cont.)



Convergence of Value Iteration

- ▶ **Theorem (VI converges on V^*):** For any initial $\underline{V_0} \in \mathbb{R}^{\overbrace{|\mathcal{S}|}}$, Value Iteration achieves that $\underline{V_k} \rightarrow \overbrace{V^*}$, as $k \rightarrow \infty$.
- ▶ **Question:** How to show this?

Convergence of Value Iteration

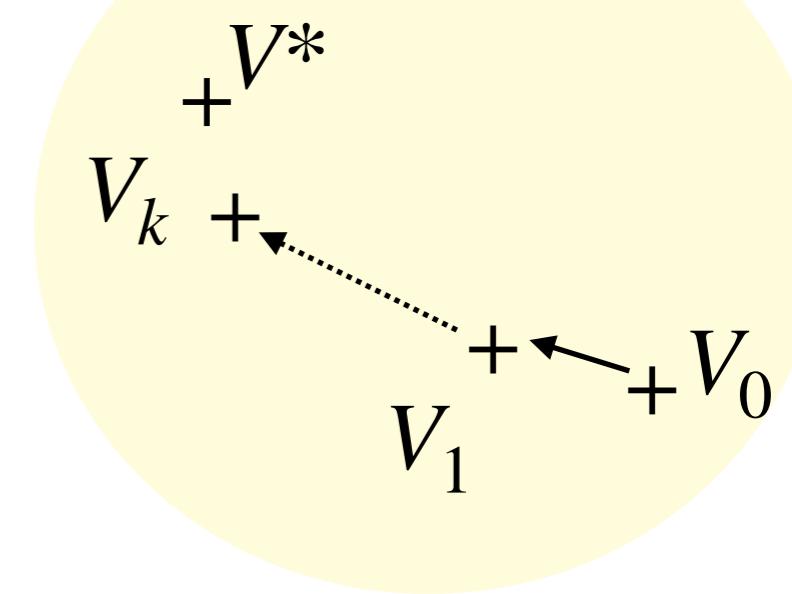
- **Theorem (VI converges on V^*):** For any initial $V_0 \in \mathbb{R}^{|\mathcal{S}|}$, Value Iteration achieves that $V_k \rightarrow V^*$, as $k \rightarrow \infty$.
- **Proof:** We prove convergence by the following 3 steps

(B1): Show that T^* is a *contraction operator*

(B2): Under a contraction operator, $\{V_k\}$ shall converge to the unique fixed point (why?)

(B3): Since V^* is a fixed point, then $V_k \rightarrow V^*$ due to uniqueness

Complete metric vector space $\mathbb{R}^{|\mathcal{S}|}$



(B1): T^* is a γ -Contraction Operator on V

- Bellman optimality backup operator: $T^*(V) := \max_{a \in \mathcal{A}} (R^a + \gamma P^a V)$

$$\|T^*(V) - T^*(\hat{V})\|_\infty$$

by the def. $= \max_s |T^*(V)(s) - T^*(\hat{V})(s)|$

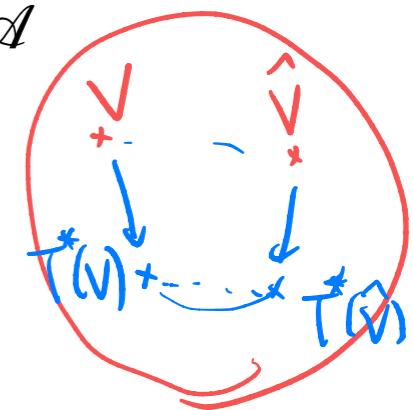
Loc norm $= \max_s \left| \max_a \left(R_{s,a} + \gamma \sum_{s'} P_{ss'}^a V(s') \right) - \max_{a'} \left(R_{s,a'} + \gamma \sum_{s'} P_{ss'}^{a'} \hat{V}(s') \right) \right|$

*by T^** $\leq \max_s \max_a \left| \left(R_{s,a} + \gamma \sum_{s'} P_{ss'}^a V(s') \right) - \left(R_{s,a} + \gamma \sum_{s'} P_{ss'}^a \hat{V}(s') \right) \right|$

$\leq \max_s \max_a \left| \gamma \sum_{s'} P_{ss'}^a (V(s') - \hat{V}(s')) \right|$

$\leq \gamma \|V - \hat{V}\|_\infty$

Therefore, T^* is a γ -contraction operator ($\gamma < 1$)



$$U_a = \begin{bmatrix} U_{a_1} \\ U_{a_2} \\ U_{a_3} \end{bmatrix}$$

$$U_b = \begin{bmatrix} U_{b_1} \\ U_{b_2} \\ U_{b_3} \end{bmatrix}$$

$$\left| \max_{i=1,2,3} U_{a_i} - \max_{i=1,2,3} U_{b_i} \right|$$

$$\leq \max_{i=1,2,3} |U_{a_i} - U_{b_i}|$$

(B2): T^* Converges to the Unique Fixed Point

- ▶ T^* is a γ -contraction operator in a complete metric space
- ▶ By Banach Fixed Point Theorem, T^* converges to the unique fixed point
- ▶ Note that V^* is one fixed point of T^* (why?)
by Bellman Optimality equation
- ▶ Therefore, V^* is the unique fixed point of T^*

(B3): $V_k \rightarrow V^*$ as $k \rightarrow \infty$

- Let's put everything together!

Let V_0 be the initialization.

$$\|\underbrace{T^*(V_k)}_{\text{``}V_{k+1}\text{''}} - \underbrace{T^*(V^*)}_{\text{``}V^*\text{''}}\|_\infty \stackrel{\text{by (B1)}}{\leq} \gamma \cdot \|V_k - V^*\|_\infty$$

$$\Rightarrow \|V_k - V^*\|_\infty \leq \gamma^k \|V_0 - V^*\|_\infty \quad \text{uniqueness by (B2)}$$

By letting $k \rightarrow \infty$, we have $\lim_{k \rightarrow \infty} \|V_k - V^*\|_\infty = 0$

Discussion: Issues With Value Iteration

score

- ▶ **Question 1:** What would happen if T^* has multiple fixed points?

optimal

$V_k \rightarrow V^{**}$, where V^{**} may not be V^*

- ▶ **Question 2:** In how many iterations will VI converge?

trainig steps

You need "infinitely many" iterations.

- ▶ **Question 3:** By applying VI, could we find an optimal policy?

$V^* \rightarrow Q^* \rightarrow \pi^*$

- ▶ **Question 4:** By using VI, could we directly prove the existence of an optimal policy?

Not yet.

(VI only shows how to find V^* ,
but we still need to find a π^* such that $V^{\pi^*} = V^*$)

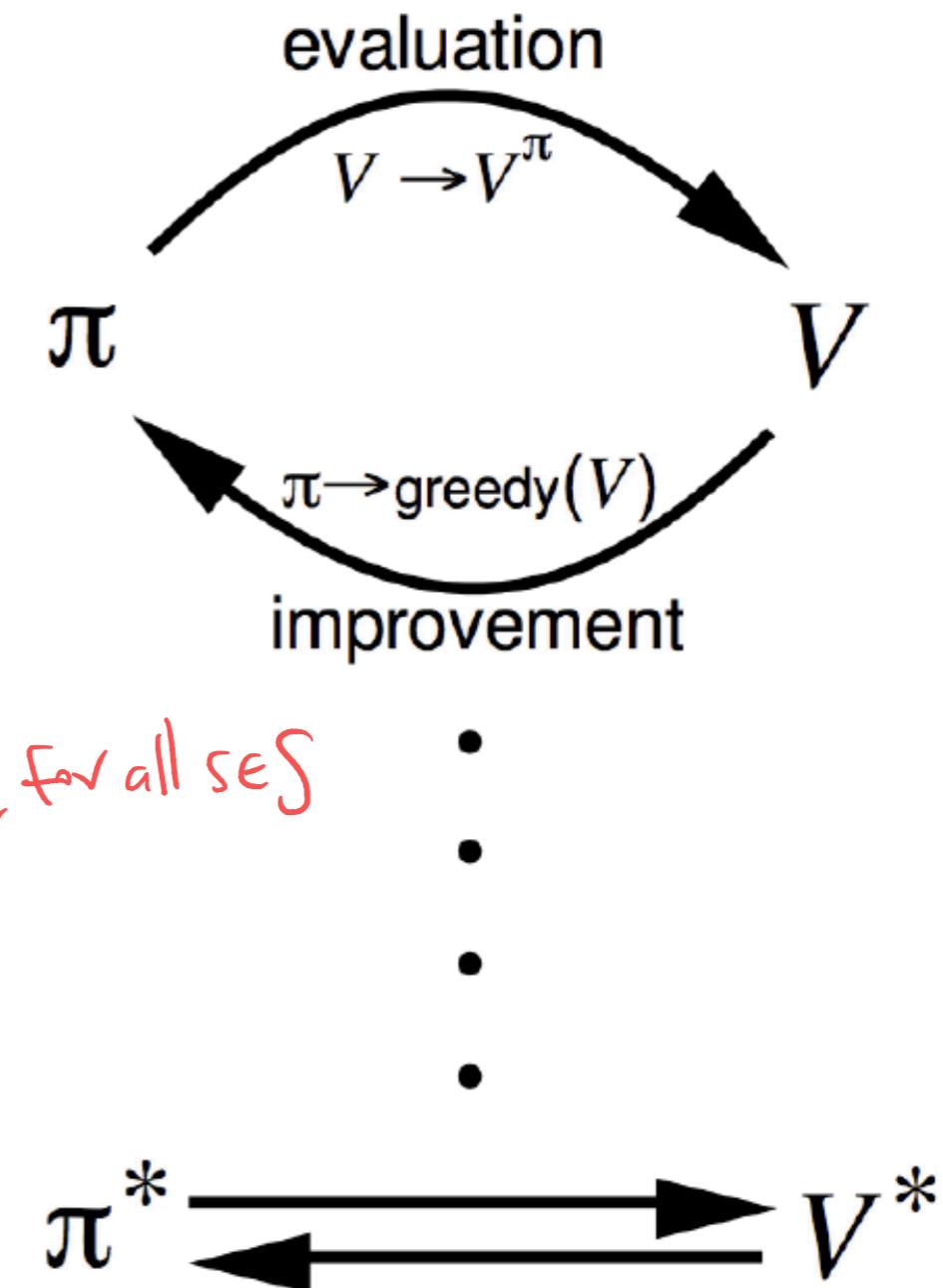
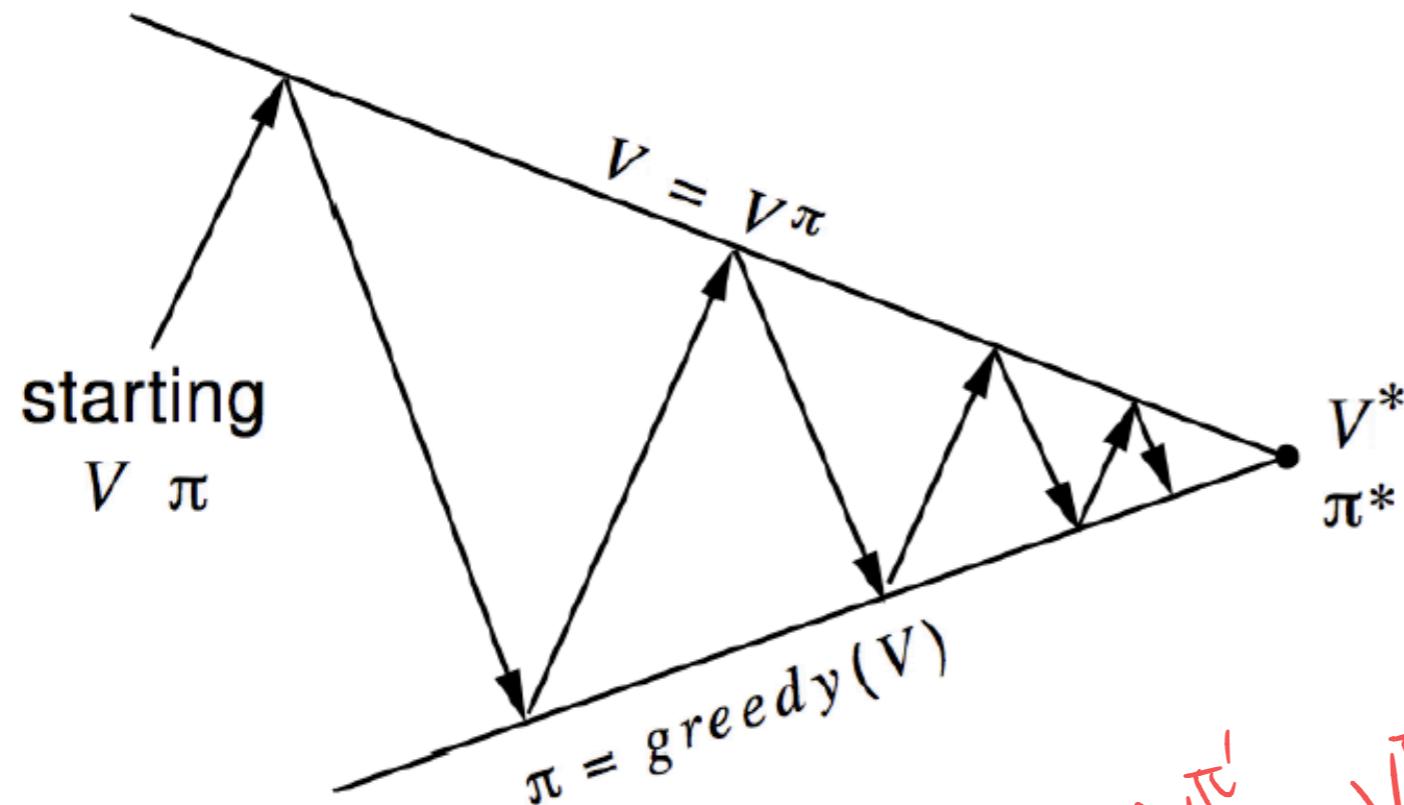
Discussion: Asymptotic Convergence vs Convergence Rate

VI enjoys the following types of convergence:

- ▶ **Asymptotic Convergence:** $V_k \rightarrow V^*$, as $k \rightarrow \infty$
- ▶ **Convergence Rate:** $\|V_k - V^*\|_\infty \leq \gamma^k \cdot \|V_0 - V^*\|_\infty$
- ▶ **Question:** Which one is stronger?

2. Policy Iteration (PI)

"Policy" Iteration: Generic Procedure



- {
- Policy evaluation Estimate v_π
 - Iterative policy evaluation
 - Policy improvement Generate $\pi' \geq \pi$
 - Greedy policy improvement
- •
•

Policy Iteration: Pseudo Code

(We focus on deterministic policies)

Step 1. Initialize $k = 0$ and set $\pi_0(s)$ arbitrarily for all states

Step 2. While k is zero or $\pi_k \neq \pi_{k-1}$:

- ▶ Derive V^{π_k} via **policy evaluation** for π_k (iterative/non-iterative)
- ▶ Derive π_{k+1} by greedy **one-step policy improvement**

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

One-Step Policy Improvement

- Given V^{π_k} , compute $\underline{Q^{\pi_k}(s, a)}$:

$$Q^{\pi_k}(s, a) = R(s, a) + \gamma \sum_s P(s' | s, a) V^{\pi_k}(s')$$

- Derive the new policy π_{k+1} : For all states s ,

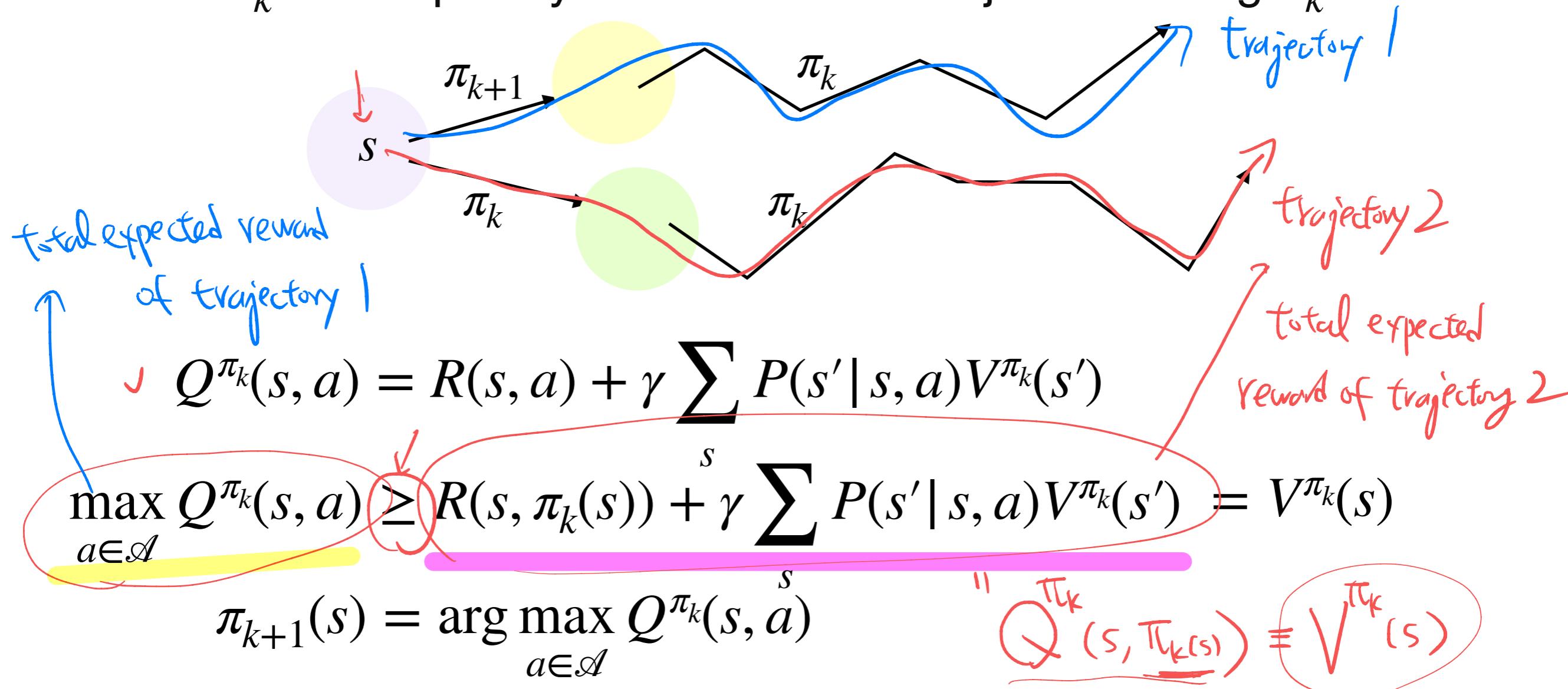
$$\pi_{k+1}(s) = \arg \max_{a \in \mathcal{A}} Q^{\pi_k}(s, a)$$

... "one-step greedy policy improvement"

- Note: We will use $R(s, a)$ and $R_{s,a}$ interchangeably

Why is One-Step Policy Improvement Reasonable?

- ▶ **Question:** Suppose we take $\pi_{k+1}(s)$ for one step and then follow π_k subsequently. Is this better than just following π_k ?



- ▶ **Question:** But how about following π_{k+1} all the way?

Monotonic Improvement in Policy

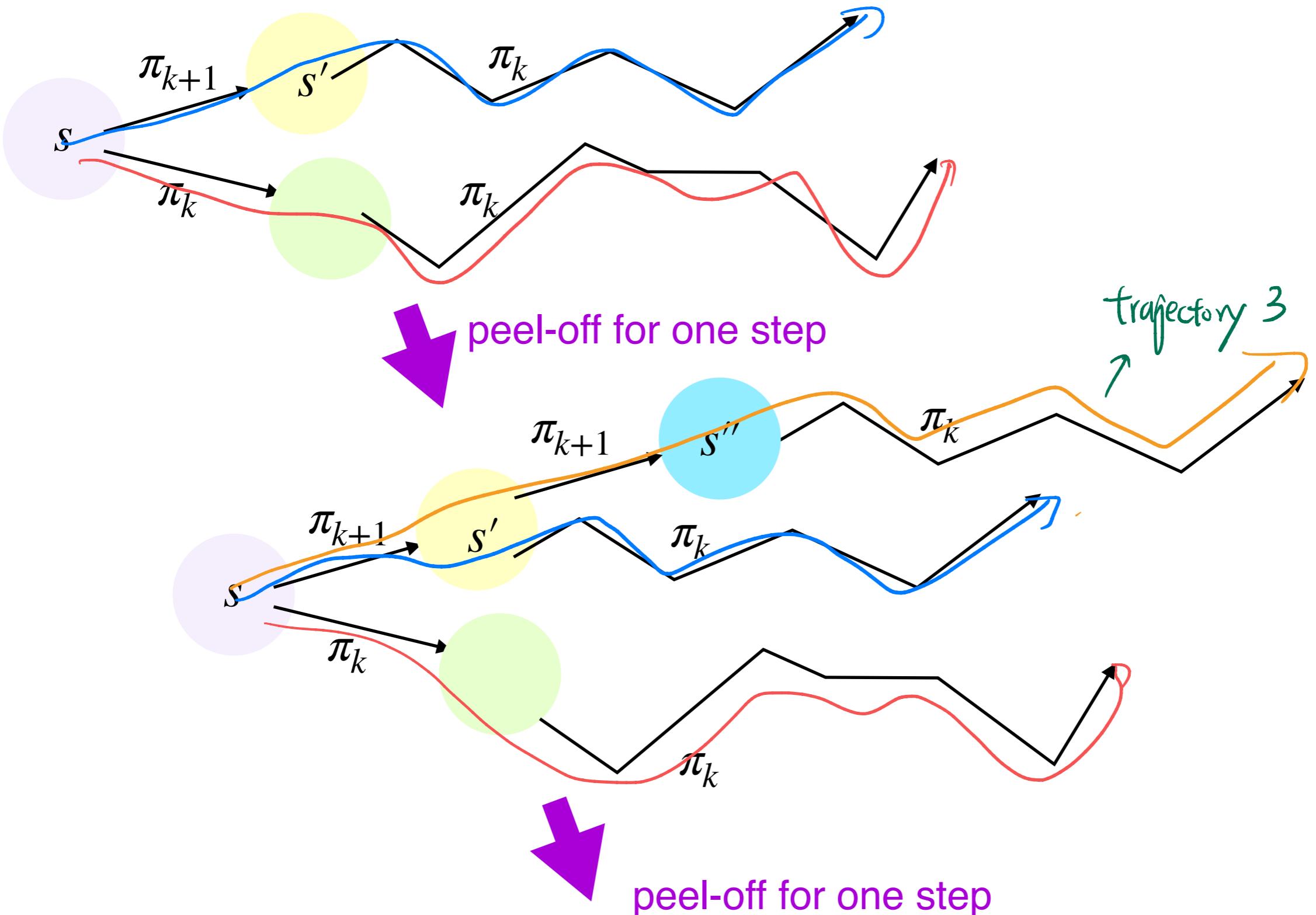
- ▶ Recall: Partial ordering of policies

$$\pi \geq \pi' \quad \text{if} \quad V^\pi(s) \geq V^{\pi'}(s), \forall s$$

- ▶ Question: Do we have $\pi_{k+1} \geq \pi_k$?

- ▶ **Theorem (Monotonic Policy Improvement):** Under the one-step policy improvement step, we have $V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s)$ for all $s \in \mathcal{S}$ and hence $\pi_{k+1} \geq \pi_k$.

Proof Idea: “Peeling off”



Proof: Monotonic Policy Improvement

$$\begin{aligned} V^{\pi_k}(s) &\leq \max_{a \in \mathcal{A}} Q^{\pi_k}(s, a) && \dots \text{by P.24} \\ &= \max_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s'} P(s' | s, a) V^{\pi_k}(s') && \dots \text{by Bellman expectation equation} \\ &= R(s, \pi_{k+1}(s)) + \gamma \sum_{s'} P(s' | s, \pi_{k+1}(s)) V^{\pi_k}(s') && \text{by def. of } \bar{\pi}_{k+1} \\ &\leq R(s, \pi_{k+1}(s)) + \gamma \sum_{s'} P(s' | s, \pi_{k+1}(s)) \max_{a' \in \mathcal{A}} Q^{\pi_k}(s', a') && \text{by P.24} \\ &= R(s, \pi_{k+1}(s)) + \gamma \sum_{s'} P(s' | s, \pi_{k+1}(s)) \\ &\quad \times \left(R(s', \pi_{k+1}(s')) + \gamma \sum_{s''} P(s'' | s', \pi_{k+1}(s')) V^{\pi_k}(s'') \right) && \text{total expected reward of trajectory 3} \\ &= V^{\pi_{k+1}}(s) && \Rightarrow V^{\bar{\pi}_k}(s) \leq V^{\bar{\pi}_{k+1}}(s) \end{aligned}$$

Discussions: Policy Iteration

- ▶ **Question 1:** Will policy iteration terminate in finitely many iterations?
-

- ▶ Yes, in at most $|\mathcal{A}|^{|\mathcal{S}|}$ iterations (assume $|\mathcal{A}|, |\mathcal{S}|$ are finite)

(There are $|\mathcal{A}|^{|\mathcal{S}|}$ deterministic policies)

(If $\pi_{k+1} \neq \pi_k$, then they must differ by at least 1 entry)

(Monotonic policy improvement: $\pi_{k+1} \geq \pi_k$)

Discussions: Policy Iteration (Cont.)

- ▶ **Question 2:** If we have $\pi_{k+1} = \pi_k$, what shall we expect about π_{k+2} ?

- ▶ **Recall:**

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

$$\pi_{k+1}(s) = \arg \max_a Q^{\pi_k}(s, a)$$

$$\pi_{k+2}(s) = \arg \max_a Q^{\pi_{k+1}}(s, a)$$

Therefore, policy iteration can terminate when $\pi_{k+1} = \pi_k$

Moreover, $\pi_{k+1} = \pi_k$ implies **Bellman optimality equation** is satisfied by π_k :

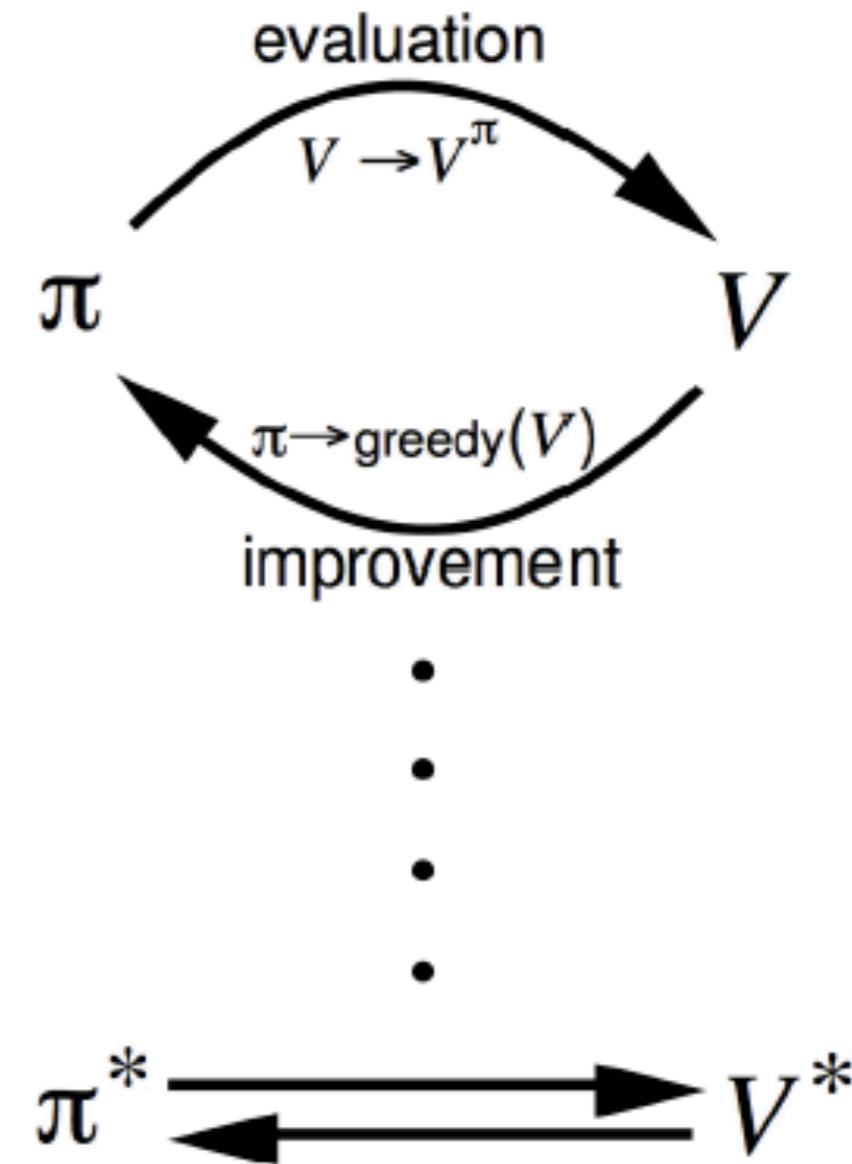
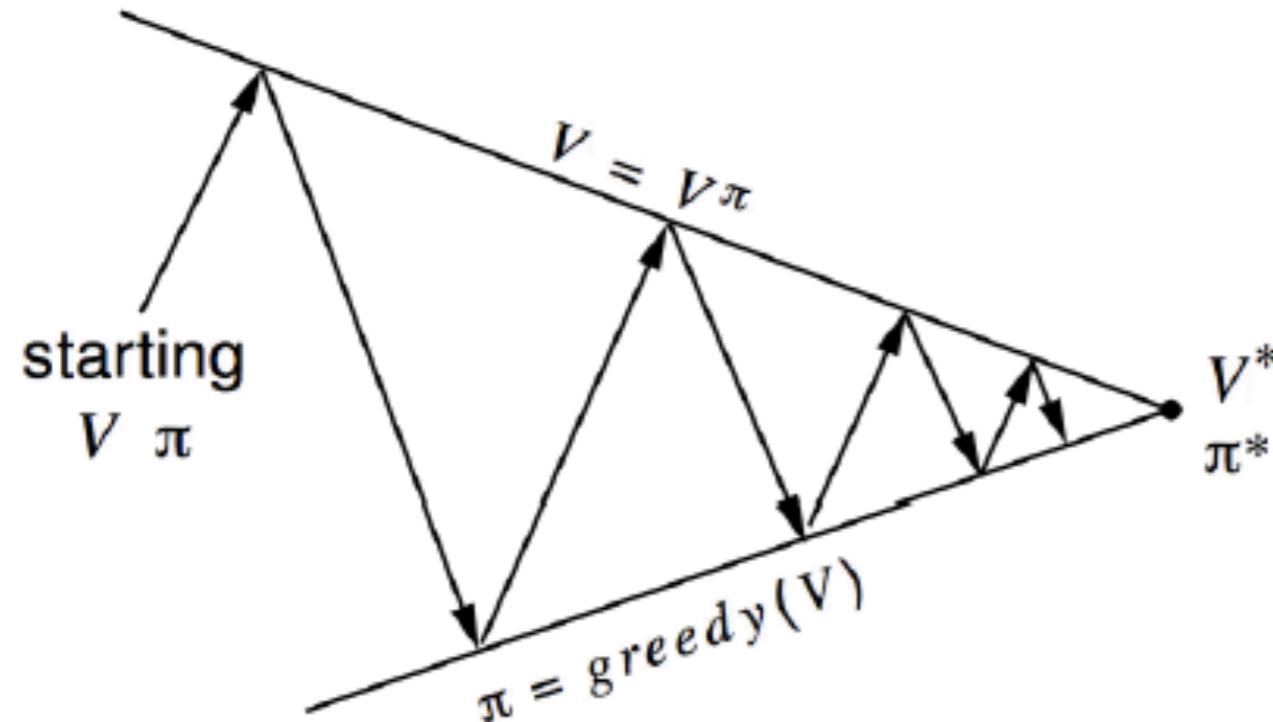
$$V^{\pi_k}(s) = \max_a Q^{\pi_k}(s, a)$$

Because V^* is the unique fixed point

Therefore, π_k must be a (deterministic) optimal policy (**Why?**)

Hence, PI proves the existence of an optimal policy

Extension: Generalized Policy Iteration



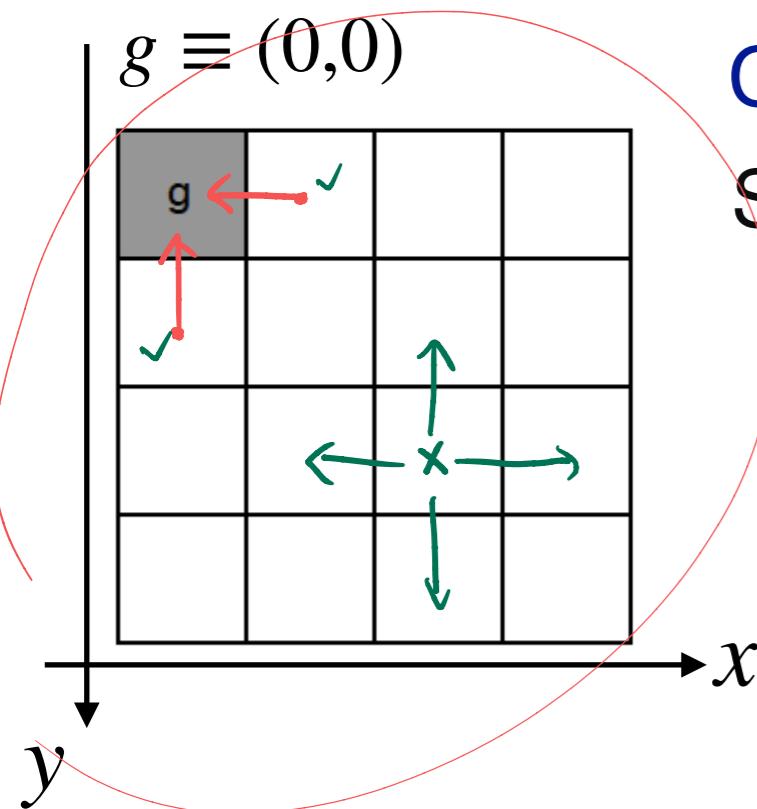
- ✓ Policy evaluation Estimate v_π
Any policy evaluation algorithm
- ✓ Policy improvement Generate $\pi' \geq \pi$
Any policy improvement algorithm
- ▶ Remark: Policy gradient methods can be interpreted as an instance of generalized policy iteration (discussed in Lectures 5-7)

Summary

Problem	Bellman Equation	Algorithm
Prediction	Bellman Expectation Equation	Iterative Policy Evaluation <i>(IPE)</i>
Control	Bellman Expectation Equation + Greedy Policy Improvement	Policy Iteration <i>(PI)</i>
Control	Bellman Optimality Equation	Value Iteration <i>(VI)</i>

Extension: Regularized MDPs

Motivation: *Reward Shaping* for Faster Learning

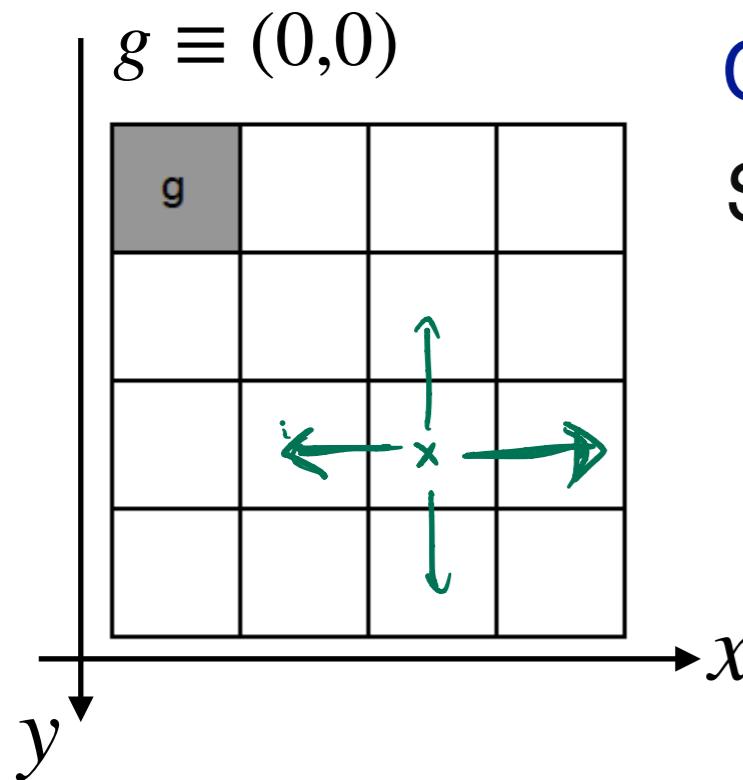


Consider our favorite “shortest path” problem
Suppose the reward function of the environment is

- $R((1,0), \leftarrow) = R((0,1), \uparrow) = \underline{\underline{1}}$
- $R(s, a) = 0, \text{ otherwise}$

Is this problem easy to learn? *Sparse rewards*

Motivation: "Reward Shaping" for Faster Learning



Consider our favorite “shortest path” problem

Suppose the reward function of the environment is

- $R((1,0), \leftarrow) = R((0,1), \uparrow) = 1$
- $R(s, a) = 0$, otherwise

Is this problem easy to learn?

What if we augment the reward function (denoted by \tilde{R}) as follows:

- $\tilde{R}((1,0), \leftarrow) = \tilde{R}((0,1), \uparrow) = 1$
- $\tilde{R}(s, a) = -\|s - g\|_1$, otherwise

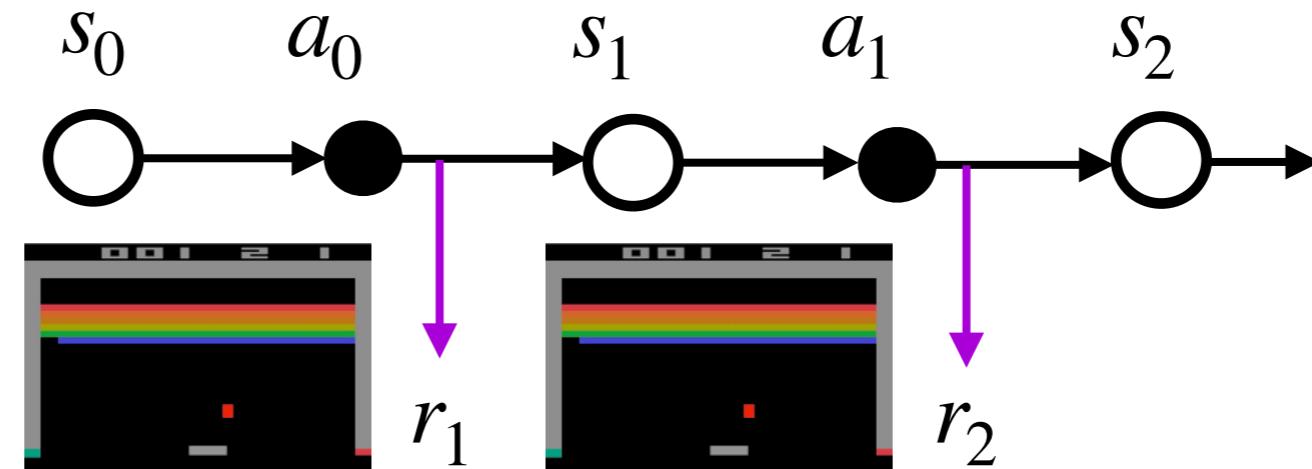
Question: Does this lead to the same optimal policy? Yes

Question: Is this problem easier to learn? Yes (reward is no longer sparse)

Example: Entropy as "Intrinsic Reward"

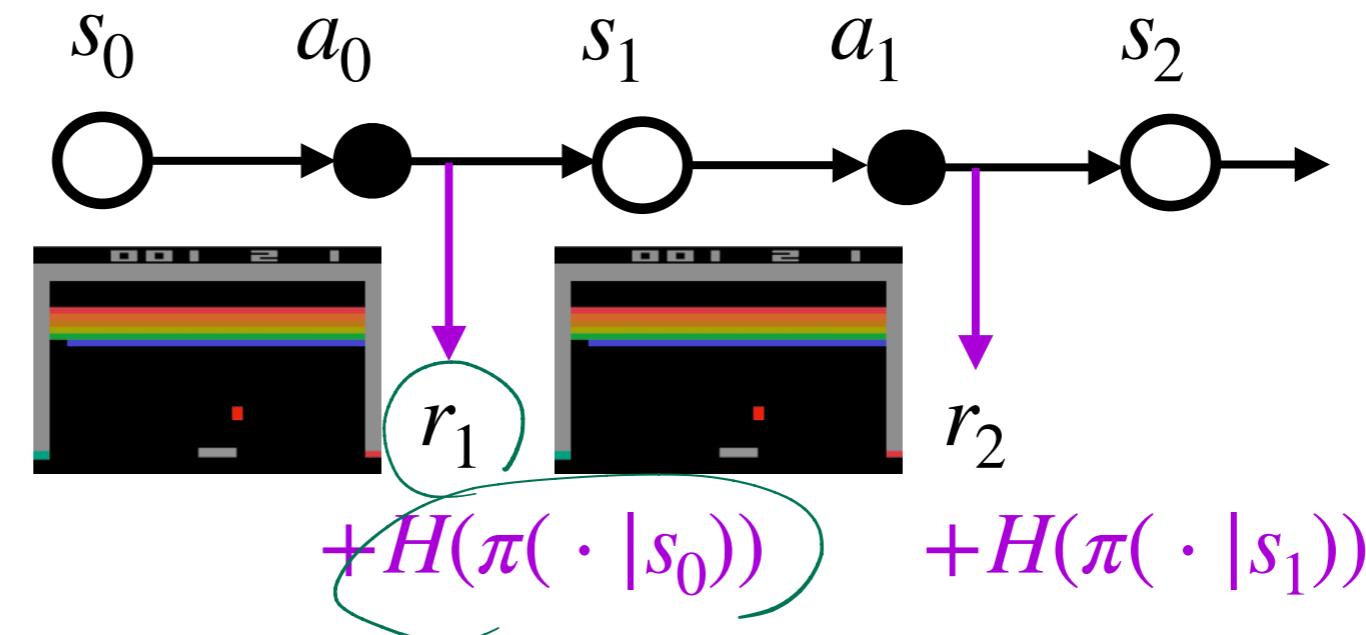
► Extrinsic Rewards & Intrinsic Rewards

Standard MDP



Extrinsic rewards

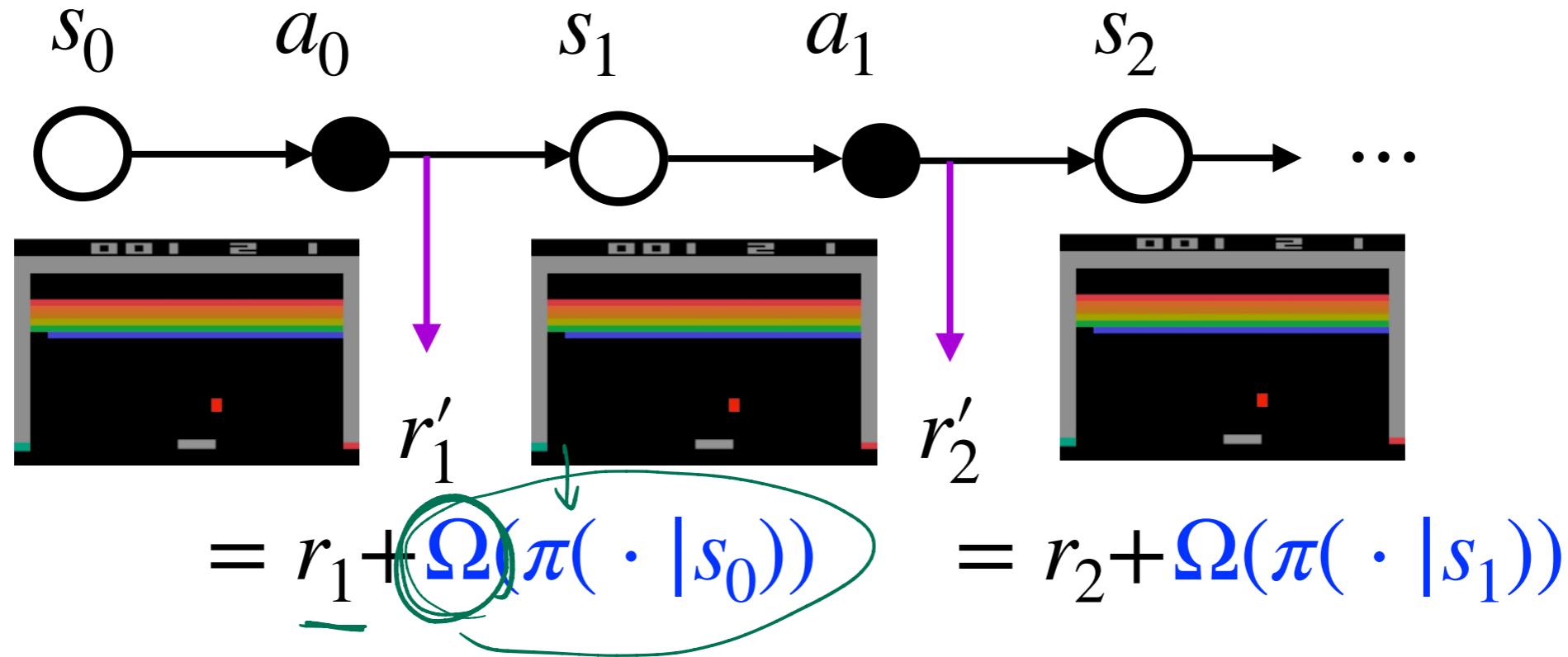
MDP with
entropy bonus



Entropy as intrinsic rewards for better exploration

More Generally: *Regularized MDPs*

Regularized MDP = Standard MDP + Regularized rewards!



- ▶ A regularized MDP can be specified by $(\mathcal{S}, \mathcal{A}, P, R, \Omega, \gamma)$
- ▶ $\Omega(\cdot)$: A function that maps an *action distribution* to a *real number*

Value Functions of *Regularized MDPs*

	Unregularized MDP	Regularized MDP
Return	$G_t := r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots$	
Value function	$V^\pi(s) := \mathbb{E}[G_t s_t = s; \pi]$	
Q function	$Q^\pi(s, a) := \mathbb{E}[G_t s_t = s, a_t = a; \pi]$	
Bellman expectation equations	$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a s) Q^\pi(s, a)$ $Q^\pi(s, a) = R_{s,a} + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a V^\pi(s')$	

Next Question: *How to find V_Ω^π ?*

Regularized Bellman Expectation Operator

- ▶ Regularized Bellman Expectation Operator

$$\begin{aligned}[T_{\Omega}^{\pi} V](s) &:= [T^{\pi} V](s) - \underbrace{\Omega(\pi(\cdot | s))}_{\text{regularization term}} \\ &= \underbrace{R_s^{\pi} - \Omega(\pi(\cdot | s))}_{\text{regularized immediate reward}} + \gamma P_{ss'}^{\pi} V\end{aligned}$$

-
- ▶ Question: Is T_{Ω}^{π} a contraction? Yes! (in L_{∞} -norm)
 - ▶ Therefore, under T_{Ω}^{π} , there is a unique fixed point, which is the regularized value function V_{Ω}^{π}
 - ▶ To find V_{Ω}^{π} , we can use the regularized IPE method

Next Question: *How to define “optimality” for regularized MDPs?*

Regularized Bellman Optimality Operator

- ▶ Recall: Bellman optimality operator for *unregularized* MDPs

$$[T^*V](s) = \max_{a \in \mathcal{A}} R_s^a + \gamma P_s^a V = \max_{\pi} \underbrace{R_s^\pi + \gamma P_s^\pi V}_{=[T^\pi V](s)}$$

- ▶ Regularized Bellman optimality equations

$$[T_\Omega^* V](s) = \max_{\pi} \{ [T_\Omega^\pi V](s) \} \leftarrow \text{a greedy step!}$$

- ▶ Useful Facts

1. T_Ω^* is also a contraction map (in L_∞ -norm)
2. There is a unique fixed point of T_Ω^*
3. We define regularized optimal value function V_Ω^* as the fixed point of T_Ω^*

(See HW1 problem for more details)

Regularized Q-functions and Policy Iteration

Regularized
Q-function

$$Q_{\Omega}^{\pi}(s, a) := R_s^a + \gamma E_{s' \sim P(\cdot | s, a)}[V_{\Omega}^{\pi}(s')]$$

Regularized optimal
Q-function

$$Q_{\Omega}^{*}(s, a) := R_s^a + \gamma E_{s' \sim P(\cdot | s, a)}[V_{\Omega}^{*}(s')]$$

- ▶ **Property:** $[T_{\Omega}^{\pi} V_{\Omega}^{\pi}](s) = V_{\Omega}^{\pi}(s) = \langle \pi(\cdot | s), Q_{\Omega}^{\pi}(s, \cdot) \rangle - \Omega(\pi(\cdot | s))$
- ▶ **Question:** Now we are ready for “regularized policy iteration”. How?

(See HW1 problem for more details)

Regularized Policy Iteration (Regularized PI)

Regularized Policy Iteration

1. Initialize $k = 0$ and set $\pi_0(\cdot | s)$ arbitrarily for all states
2. While k is zero or $\pi_k \neq \pi_{k-1}$:
 - ▶ Derive $V_\Omega^{\pi_k}$ and $Q_\Omega^{\pi_k}$ via policy evaluation
 - ▶ Derive π_{k+1} by greedy policy improvement:

$$\pi_{k+1}(\cdot | s) = \arg \max_{\pi} \left\{ \langle \pi(\cdot | s), Q_\Omega^{\pi_k}(s, \cdot) \rangle - \Omega(\pi(\cdot | s)) \right\}$$

Regularized PI + Entropy Regularizer

- ▶ **Soft Policy Iteration:** A special case of regularized PI with negative entropy

$$\Omega(\pi(\cdot | s)) := \sum_a \pi(a | s) \log \pi(a | s)$$

- ▶ **Theorem:** Under Soft Policy Iteration, we have

$$\begin{aligned}\pi_{k+1}(\cdot | s) &= \arg \max_{\pi} \left\{ \langle \pi(\cdot | s), Q_{\Omega}^{\pi_k}(s, \cdot) \rangle - \Omega(\pi(\cdot | s)) \right\} \\ &= \frac{\exp(Q_{\Omega}^{\pi_k}(s, \cdot))}{\sum_{a \in \mathcal{A}} \exp(Q_{\Omega}^{\pi_k}(s, a))}\end{aligned}$$

- ▶ **Proof:** HW1 problem