به نام خدا

Iran University
of Science and
Technology

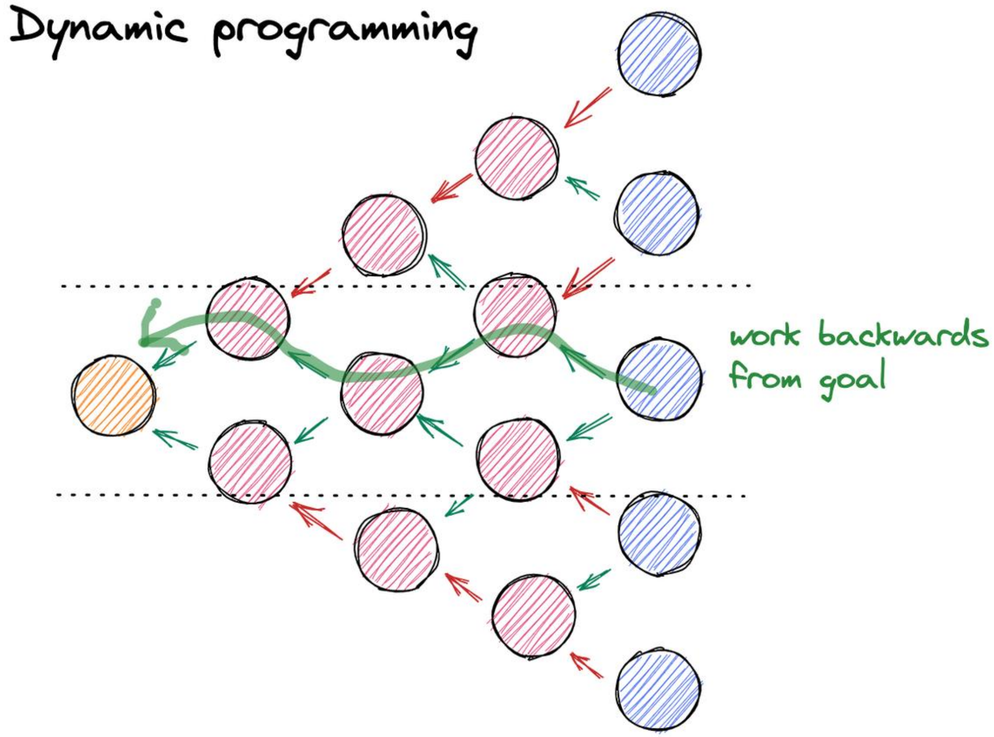
یادگیری تقویتی در کنترل

دکتر سعید شمقدری

دانشکده مهندسی برق
گروه کنترل

نیمسال اول ۱۴۰۵–۱۴۰۴

Dynamic Programming



برنامه ریزی پویا

روش های برنامه ریزی پویای کلاسیک:

نیاز به دانستن مدل کامل محیط
بیان مسئله در قالب MDP محدود
حجم محاسبات بالا

محاسبه پالیسی بهینه

کاربرد برنامه ریزی پویا در یادگیری تقویتی:

تقریب تابع Value
محاسبه پالیسی بهینه (در طول زمان)

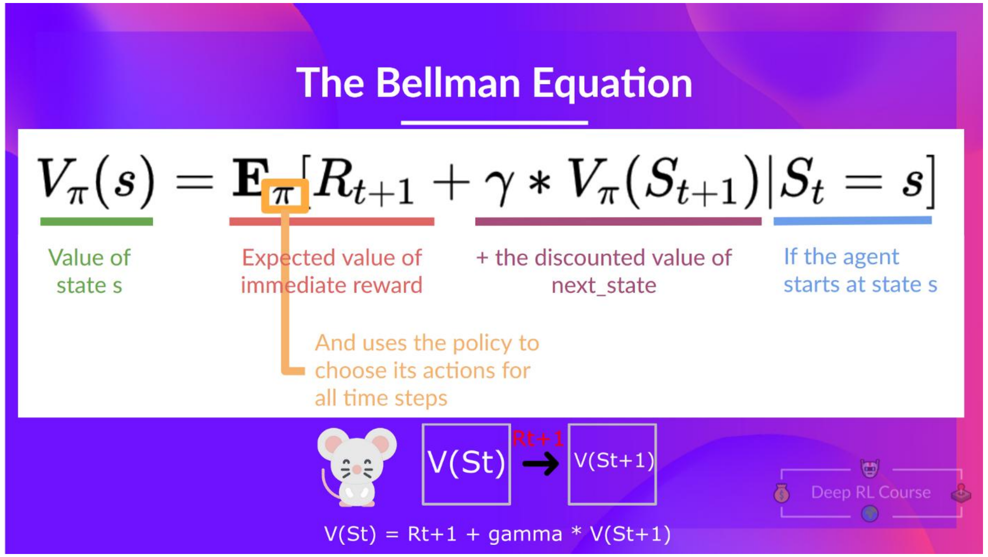
سیستم‌های پیوسته و گسسته

معادله بهینگی State Value بلمن

$$v_*(s) = \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right].$$

نحوه تعیین V_* ؟
نحوه تعیین پالیسی بهینه؟

یادآوری!



Lecture 3

معادله بهینگی Action Value بلمن

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \;\Big|\; S_t = s, A_t = a\Big] \\
&= \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma \max_{a'} q_*(s', a')\Big].
\end{aligned}
$$

نحوه تعیین q_*؟
نحوه تعیین پالیسی بهینه؟

تعیین V_π و q_π با استفاده از Dynamic Programming

تقریب تابع Value با استفاده از الگوریتم‌های Iterative بر اساس معادله بلمن

اثبات همگرایی به تابع Value واقعی

با فرض پالیسی π : تعیین $V_\pi(s)$ با استفاده از معادله بلمن !!!

تقریب $V_\pi(s)$: Policy Evaluation

با داشتن تقریبی از $V_\pi(s)$: تعیین پالیسی بهتر π'

Policy Improvement

هدف: برای یک پالیسی π معین، محاسبه V_π

یادآوری: معادله بلمن

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_\pi(s')\Big]
\end{aligned}
$$

معادله بلمن:

ارتباط «تابع Value در نقطه s» با «تابع Value در نقطه s'»

فضای متریک و نقطه ثابت باناخ

بیا

A ***metric space*** is an ordered pair (X, d) consists of an *underlying set* X and a real-valued function $d(x, y)$, called *metric*, defined for $x, y \in X$ such that for any $x, y, z \in X$ the following conditions are satisfied:

1. $d(x, y) \geq 0$ [*non-negativity*]
2. $d(x, y) = 0 \Leftrightarrow x = y$ [*identity of indiscernibles*]
3. $d(x, y) = d(y, x)$ [*symmetry*]
4. $d(x, y) \leq d(x, z) + d(z, y)$ [*triangle inequality*]

فضای متریک و نقطه ثابت باناخ

تعریف (Contraction)

Let (X, d) be a metric space and $f: X \to X$. We say that f is a *contraction*, or a *contraction mapping*, if there is a real number $k \in [0,1)$, such that

$$d(f(x), f(y)) \leq kd(x, y)$$

for all x and y in X, where the term k is called a *Lipschitz coefficent* for f.

فضای متریک و نقطه ثابت باناخ

قضیه (نگاشت انقباضی)

Let (X, d) be a complete metric space and let $f: X \rightarrow X$ be a contraction. Then there is one and only one fixed point x^* such that

$$f(x^*) = x^*.$$

Moreover, if x is any point in X and $fn(x)$ is inductively defined by

$$f_2(x) = f\big(f(x)\big), f_3(x) = f\big(f_2(x)\big), \dots, f_n(x) = f\big(f_n - 1(x)\big),$$

then $f_n(x) \rightarrow x^*$ as $n \rightarrow \infty$.

نگرش نقطه ثابت باناخ

$\boldsymbol{V_\pi}$ is a fixed point of the Bellman equation: $V_\pi(s) = T(V_\pi(s'))$

ارزیابی سیاست: روش

انتخاب اولیه برای V_π : v_0 دلخواه

بروز رسانی تقریب تابع Value با قانون زیر:

$$
\begin{aligned}
v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_k(s')\Big]
\end{aligned}
$$

Iterative Policy Evaluation

$k \to \infty \quad \longrightarrow \quad V_k \to V_\pi$

الگوریتم

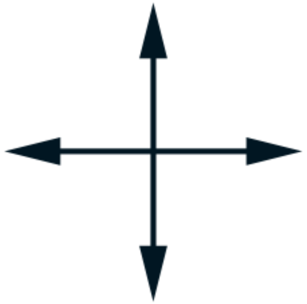
Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

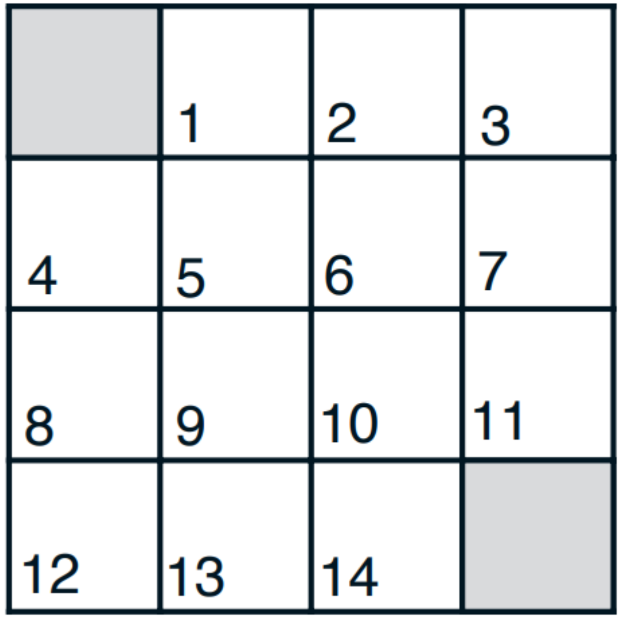
Loop:
    $\Delta \leftarrow 0$
    Loop for each $s \in \mathcal{S}$:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

مثال

Gridworld



actions



$$R_t = -1$$

on all transitions

پالیسی π : random walk

مقداردهی اولیه : $V_0 = 0$

Gridworld

مثال

$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$$V_{k+1}(s) = \sum \frac{1}{4}(-1 + \gamma V_k(s'))$$

مثال

Gridworld

$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

بهبود سیاست

Theorem

Let π and π' be any pair of deterministic policies such that, for all $s \in S$,

$$q_\pi(s, \pi'(s)) \geq v_\pi(s)$$

Then the policy π' must be as good as, or better than, π. That is, it must obtain greater or equal expected return from all states $s \in S$:

$$v_{\pi'}(s) \geq v_\pi(s)$$

$$q_\pi(s, \pi'(s)) \geq v_\pi(s)$$

مفهوم؟؟

بهبود سیاست

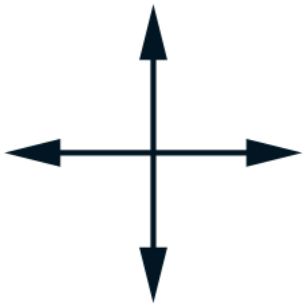
$$
\begin{aligned}
v_\pi(s) &\leq q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] \\
&= \mathbb{E}_{\pi'}[R_{t+1}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

انتخاب بهترین سیاست

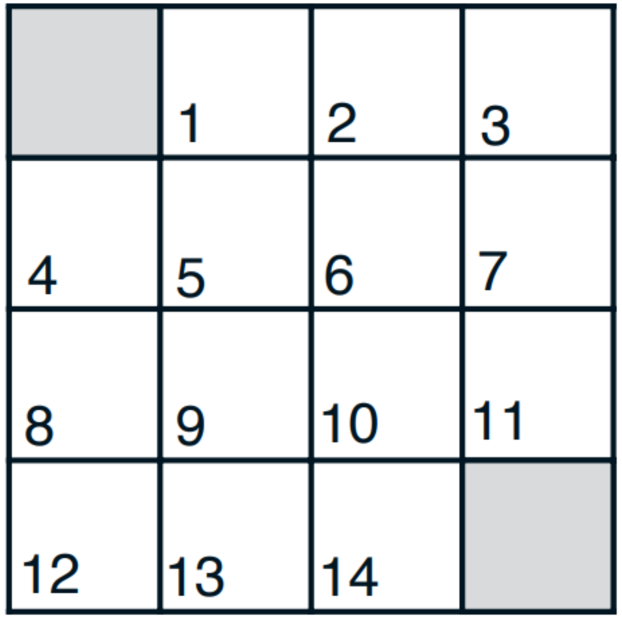
$$
\begin{aligned}
\pi'(s) &\doteq \underset{a}{\arg\max}\, q_\pi(s, a) \\
&= \underset{a}{\arg\max}\, \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \underset{a}{\arg\max} \sum_{s', r} p(s', r \mid s, a) \Big[r + \gamma v_\pi(s') \Big],
\end{aligned}
$$

Gridworld

مثال



actions



$$R_t = -1$$

on all transitions

پالیسی π : random walk

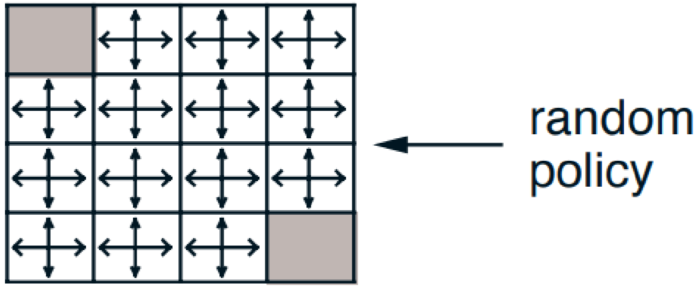
مقداردهی اولیه : $V_0 = 0$

Gridworld

مثال

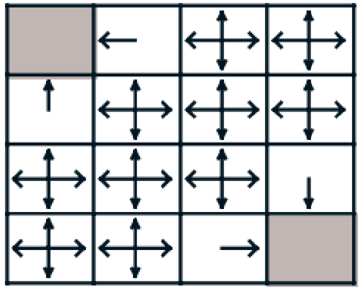
$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



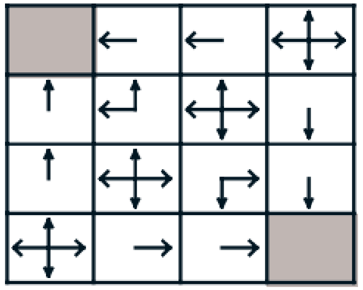
$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



Gridworld

مثال

$k = 3$

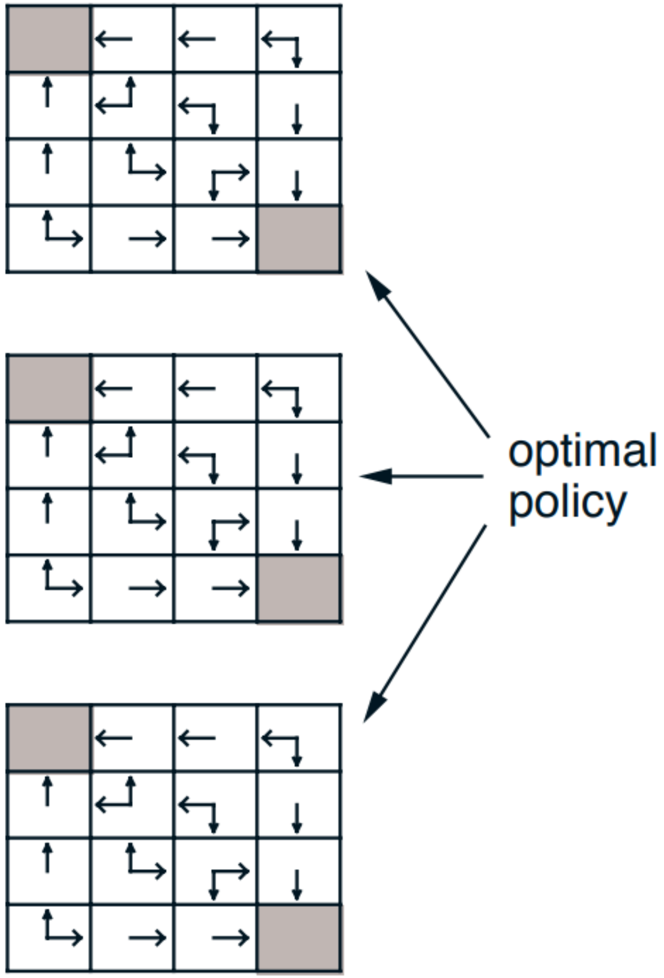
0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



Gridworld

مثال

$k = 3$

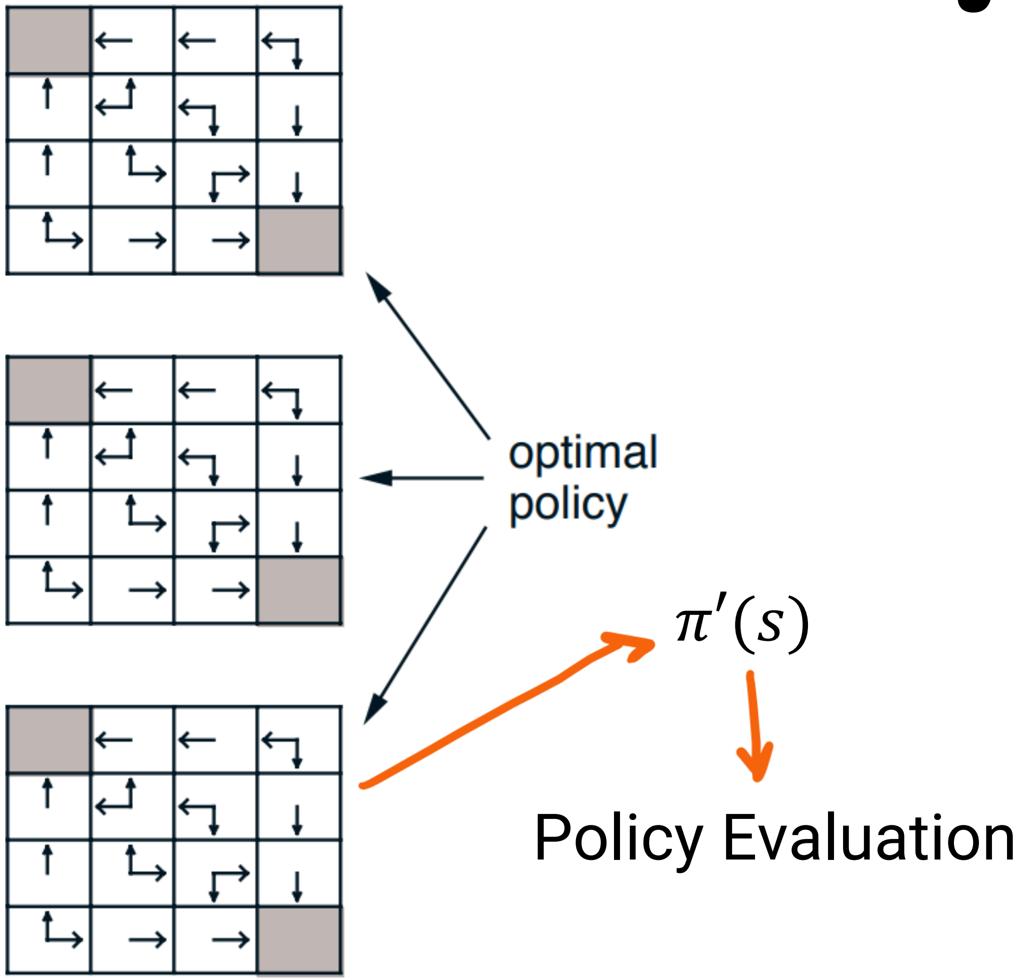
0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

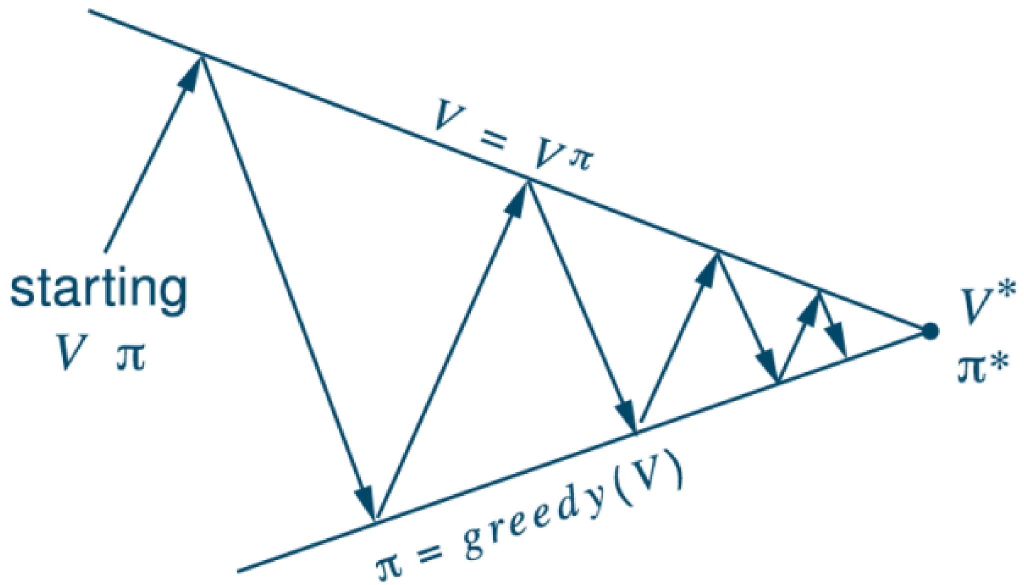
0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

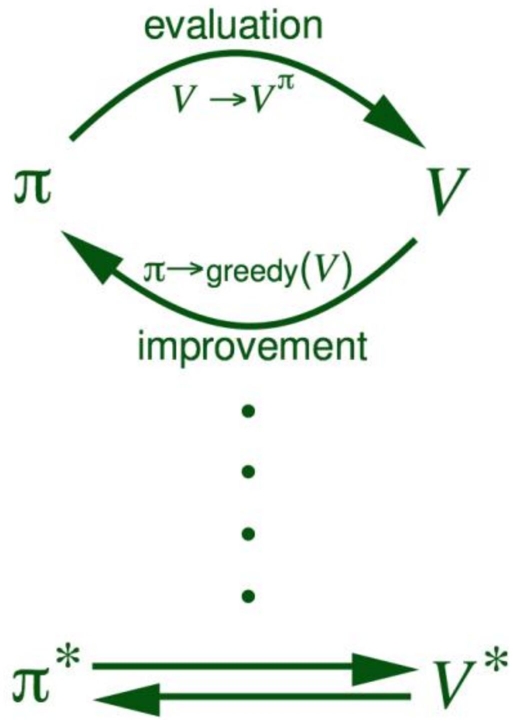


تکرار سیاست

ترکیب ارزیابی سیاست و بهبود سیاست:

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





تضمین همگرایی؟

الگوریتم

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

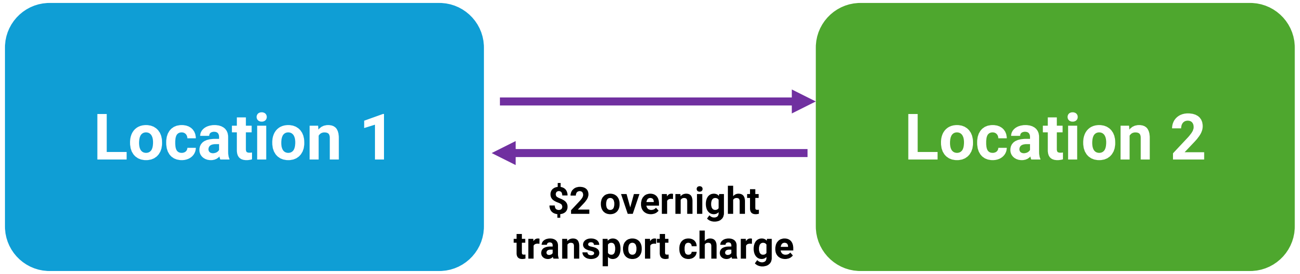
1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

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

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
   $\quad$ 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

Example: Jack’s Car Rental

مثال

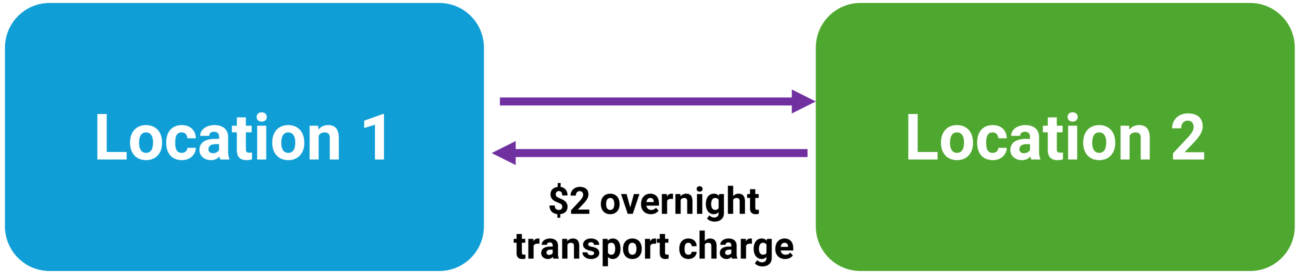


- $10 flat rental fee
- Random rental rates (Poisson)
- Random return rates (Poisson)
- 20 car capacity per location

- States?
- Actions?

Example: Jack’s Car Rental

مثال



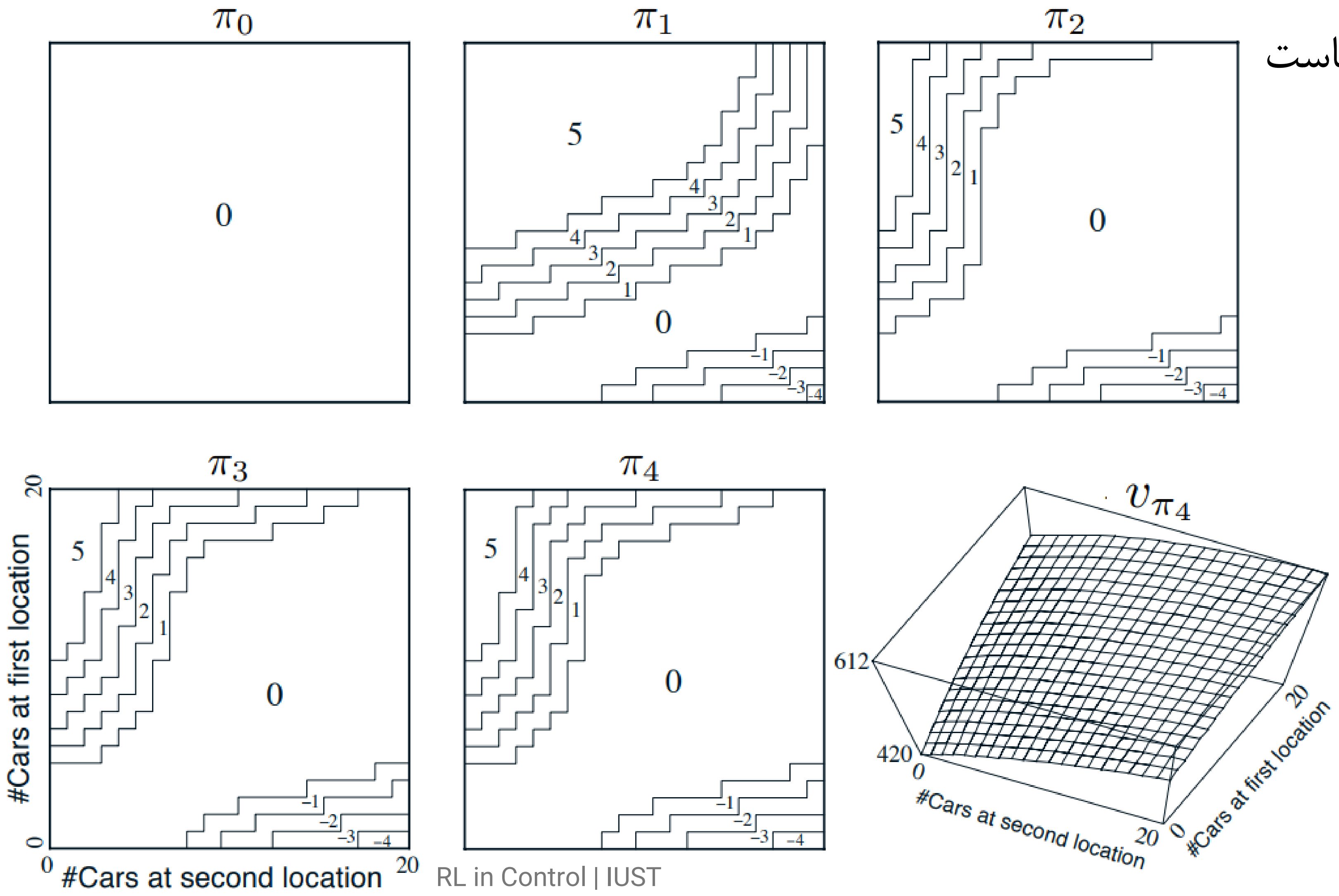
- $10 flat rental fee
- Random rental rates (Poisson)
- Random return rates (Poisson)
- 20 car capacity per location

- States: Two locations, maximum of 20 cars at each
- Actions: Move up to 5 cars between locations overnight
- Reward: $10 for each car rented (must be available)

Example: Jack’s Car Rental

مثال

تکرار سیاست



Gridworld

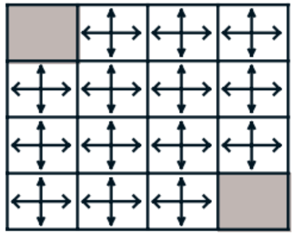
مثال

پالیسی اولیه : توزیع یکنواخت

محاسبه π' به ازای هر محاسبه V_π

$k = 0$

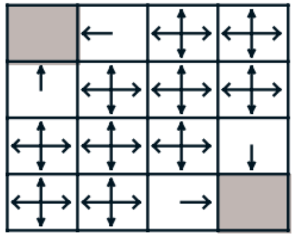
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



random policy

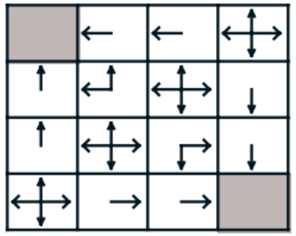
$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

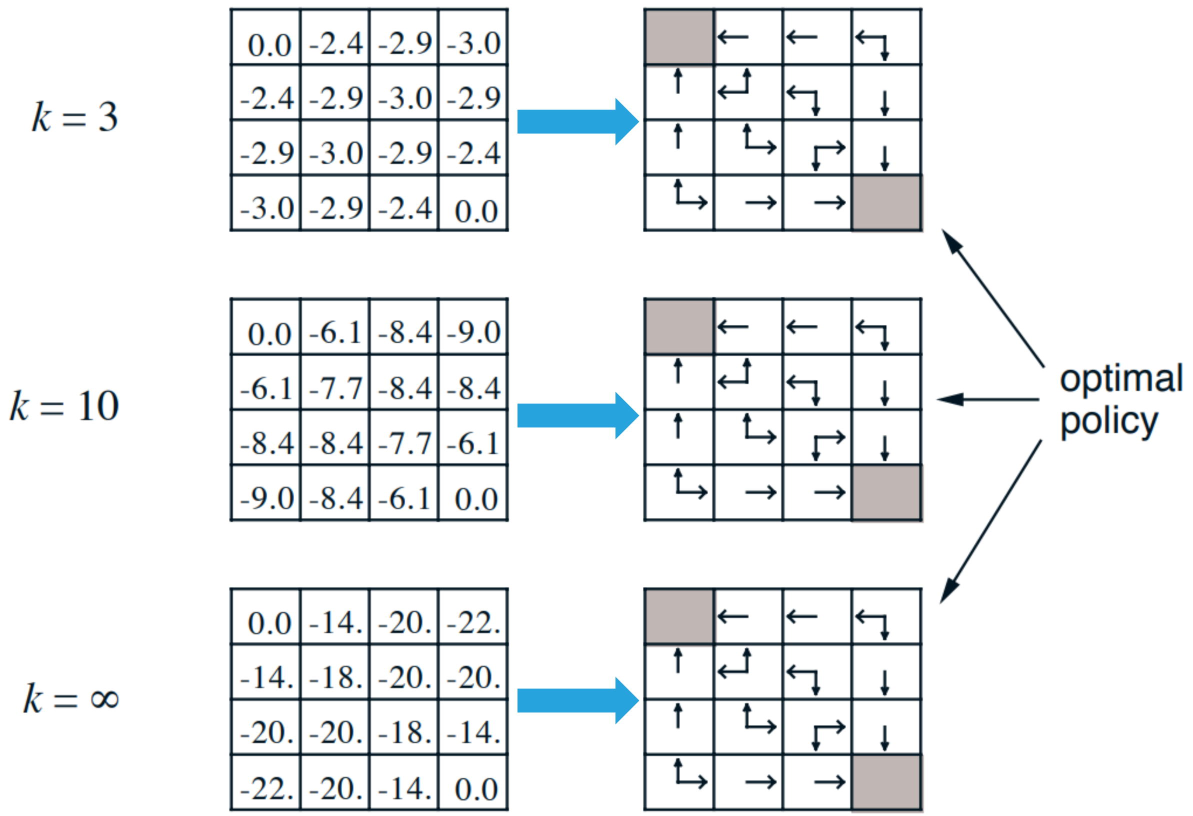


Gridworld

مثال

پالیسی اولیه : توزیع یکنواخت

محاسبه π' به ازای هر محاسبه V_π



بروزرسانی تقریب V_π درهر sample time:

$$v_{k+1}(s) \doteq \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a]$$

$$= \max_a \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_k(s')\Big],$$

محاسبه π' به ازای هر تقریب V_π

الگوریتم

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $\quad v \leftarrow V(s)$
| $\quad V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
| $\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$

Example: Gambler’s Problem

مثال

تعداد کل سکه ها :۱۰۰
بازی:
احتمال شیر : ۰.۴
سهم گذاری بخشی از ۱۰۰ سکه
شیر: دوبرابر شدن سهم
خط: از دست دادن سهم
اتمام بازی : رسیدن به ۱۰۰ یا صفر سکه

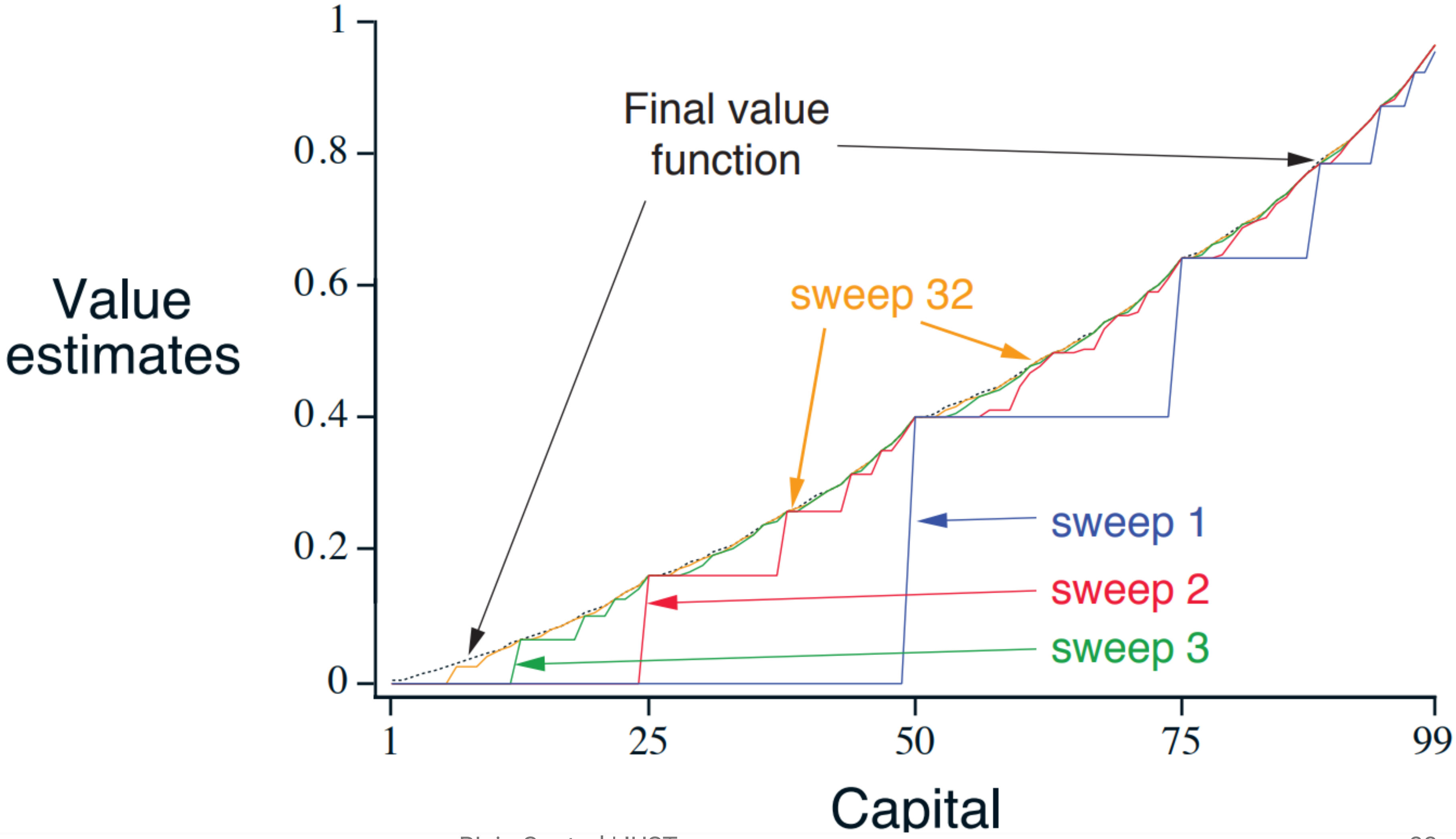
Action Space? $$a \in \{0, 1, \ldots, \min(s, 100 - s)\}$$

State Space? $$s \in \{1, 2, \ldots, 99\}$$

Reward? Reward: Goal:+1 any transition:-1

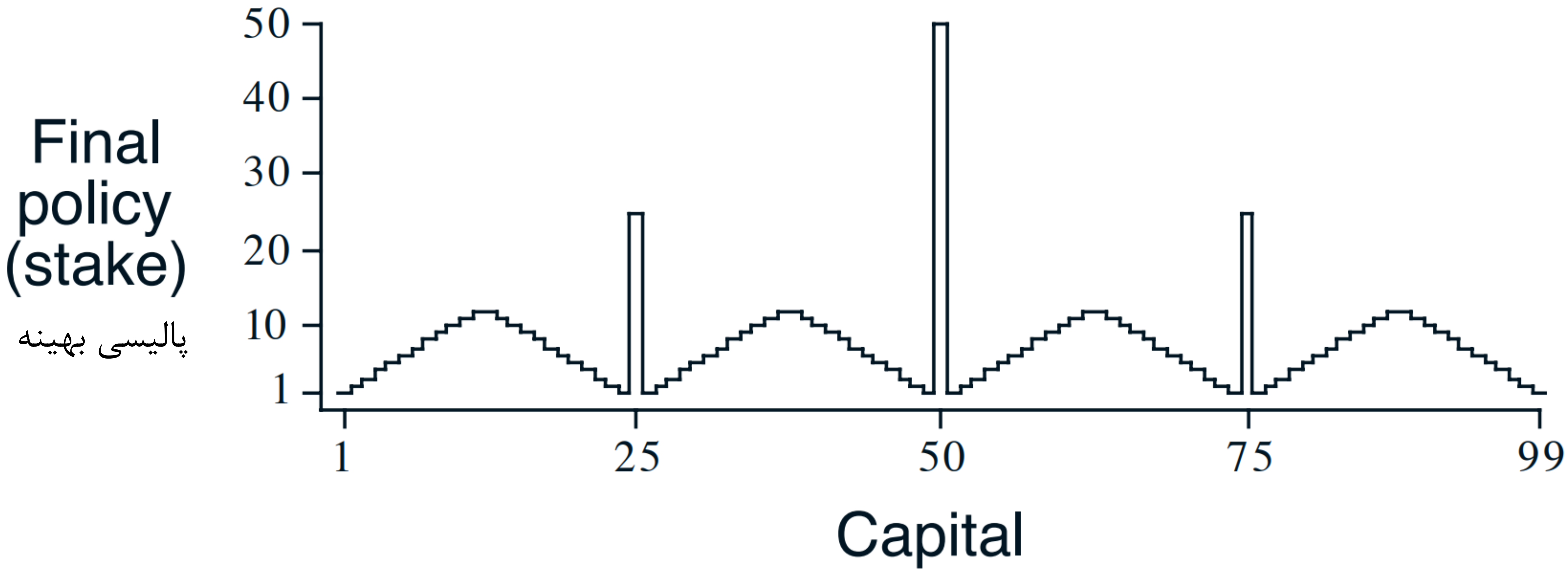
Example: Gambler's Problem

مثال



Example: Gambler's Problem

مثال



Asynchronous Dynamic Programming

بروز رسانی همه حالت ها برای فضای حالت بزرگ؟؟ حافظه زیاد!

راه حل:

In-Place Iteration Dynamic Programming

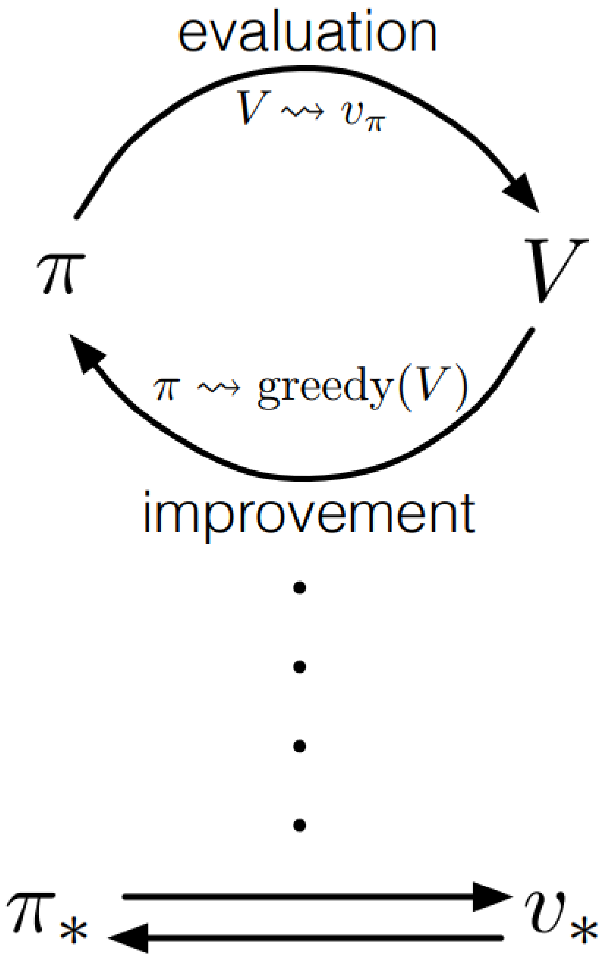
قابلیت جدید:

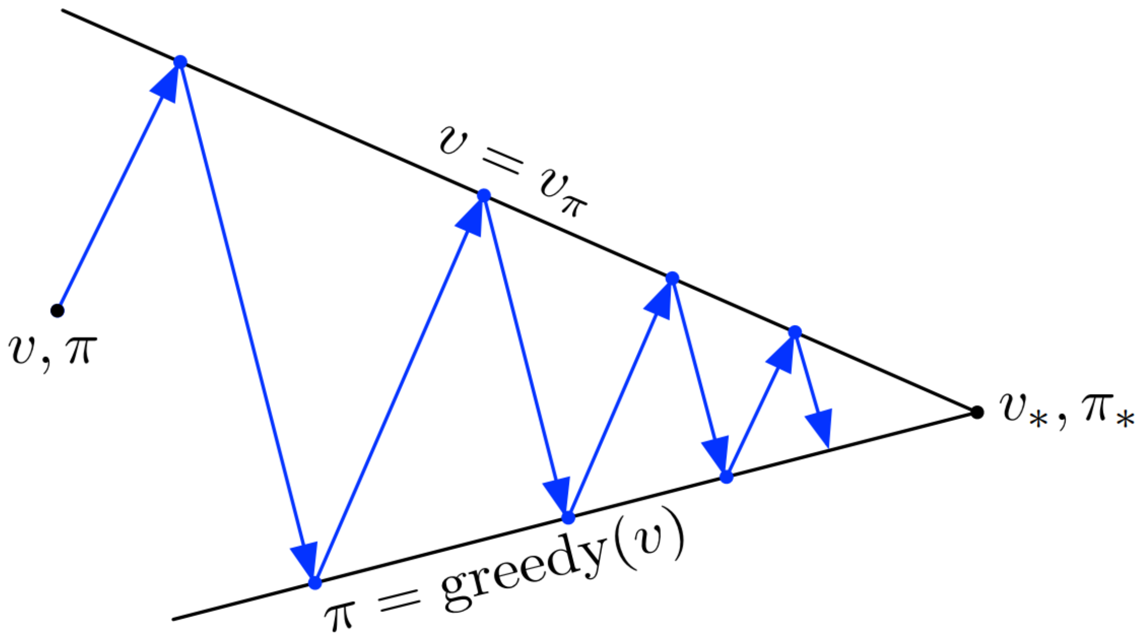
بروز رسانی بخش های مهم فضای حالت
بروز رسانی کم یا عدم بروز رسانی بخش های کم اهمیت
بروز رسانی ارزش حالتهای که به آنها برخورد میکنیم

چالش همگرایی و بهینگی ؟!

Generalized Policy Iteration

VI or PI
Random Policy
Initial Value
....





چالش همگرایی و بهینگی ؟!

Efficiency of Dynamic Programming

n حالت
k عمل

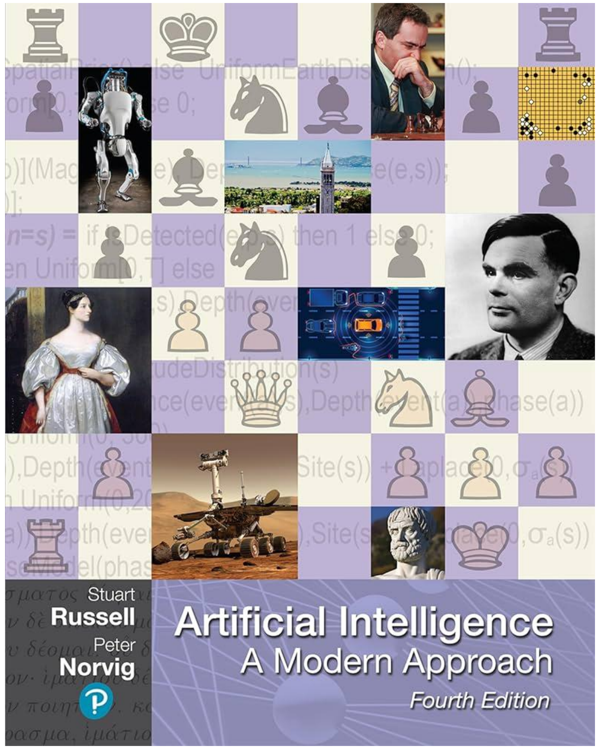
فضای پالیسی؟

curse of dimensionality

LP & DP

نگاهی دیگر در اثبات همگرایی

مرجع مطالب این بخش



Artificial Intelligence
A Modern Approach

By Stuart **Russell**,
and Peter **Norvig**

نگاهی دیگر در اثبات همگرایی

نمایش دیگر بازگشت و معادله بلمن

A Utility of a state sequence is:

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

With discounted rewards, the utility of an infinite sequence is finite.

$$U_h([s_0, s_1, s_2, \ldots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = R_{\max}/(1-\gamma)$$

نگاهی دیگر در اثبات همگرایی

نمایش دیگر بازگشت و معادله بلمن

The expected utility obtained by executing π starting in s is given by

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^{t} R(S_t)\right]$$

Now, out of all the policies the agent could choose to execute starting in s, one (or more) will have higher expected utilities than all the others.

$$\pi_s^* = \underset{\pi}{\operatorname{argmax}}\, U^{\pi}(s)$$

نگاهی دیگر در اثبات همگرایی

نمایش دیگر بازگشت و معادله بلمن

The utility of a state is the immediate reward for that state plus the expected discounted utility of the next state, assuming that the agent chooses the optimal action. That is, the utility of a state is given by

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

This is called the Bellman equation, after Richard Bellman (1957).

نگاهی دیگر در اثبات همگرایی

الگوریتم تکرار ارزش

We start with arbitrary initial values for the utilities, calculate the right-hand side of the equation, and plug it into the left-hand side—thereby updating the utility of each state from the utilities of its neighbors. We repeat this until we reach an equilibrium. Let $U_i(s)$ be the utility value for state s at the ith iteration. The iteration step, called a Bellman update, looks like this:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_i(s')$$

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

The basic concept used in showing that value iteration converges is the notion of a contraction. Roughly speaking, a contraction is a function of one argument that, when applied to two different inputs in turn, produces two output values that are “closer together,” by at least some constant factor, than the original inputs.

نگاهی دیگر در اثبات همگرایی

دو نکته مهم!

1. A contraction has only one fixed point; if there were two fixed points they would not get closer together when the function was applied, so it would not be a contraction.
2. When the function is applied to any argument, the value must get closer to the fixed point (because the fixed point does not move), so repeated application of a contraction always reaches the fixed point in the limit.

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

View the Bellman update

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_i(s')$$

as an operator B that is applied simultaneously to update the utility of every state. Let U_i denote the vector of utilities for all the states at the ith iteration. Then the Bellman update equation can be written as:

$$U_{i+1} \leftarrow B\, U_i$$

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

Next, we need a way to measure distances between utility vectors. We will use the max norm, which measures the “length” of a vector by the absolute value of its biggest component:

$$||U|| = \max_{s} |U(s)|$$

With this definition, the “distance” between two vectors, is the maximum difference between any two corresponding elements.

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

Let U_i and U_i' be any two utility vectors. Then we have

$$||B\,U_i - B\,U_i'|| \leq \gamma\,||U_i - U_i'||$$

That is, the Bellman update is a contraction by a factor of γ on the space of utility vectors.

Hence, from the properties of contractions in general, it follows that value iteration always converges to a unique solution of the Bellman equations whenever $\gamma < 1$.

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

We can also use the contraction property to analyze the rate of convergence to a solution. In particular, we can replace U_i' with the true utilities U, for which $BU = U$. Then we obtain the inequality

$$||B\,U_i - U|| \leq \gamma\,\underbrace{||U_i - U||}_{\text{error}}$$

We see that the error is reduced by a factor of at least γ on each iteration. This means that value iteration converges exponentially fast.

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

We can calculate the number of iterations required to reach a specified error bound ϵ as follows:

- First, recall from slide 38, that the utilities of all states are bounded by

$$\pm R_{\max}/(1-\gamma)$$

This means that the maximum initial error

$$||U_0 - U|| \leq 2R_{\max}/(1-\gamma)$$

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

Suppose we run for N iterations to reach an error of at most ϵ. Then, because the error is reduced by at least γ each time, we require

$$\gamma^N \cdot 2R_{\max}/(1-\gamma) \leq \epsilon$$

Taking logs, we find

$$N = \lceil \log(2R_{\max}/\epsilon(1-\gamma))/\log(1/\gamma) \rceil$$

iterations suffice.

نگاهی دیگر در اثبات همگرایی

تحلیل همگرایی

The good news is that, because of the exponentially fast convergence, N does not depend much on the ratio ϵ/R_{max}. The bad news is that N grows rapidly as γ becomes close to 1. We can get fast convergence if we make γ small, but this effectively gives the agent a short horizon and could miss the long-term effects of the agent's actions.