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

Finite Markov Decision Processes

فرایندهای تصمیم مارکوف

وابستگی state در هر لحظه، فقط به state لحظه قبل و action اعمال شده

Definition

A state S_t is *Markov* if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \ldots, S_t]$$

محاسبه تراژکتوری state

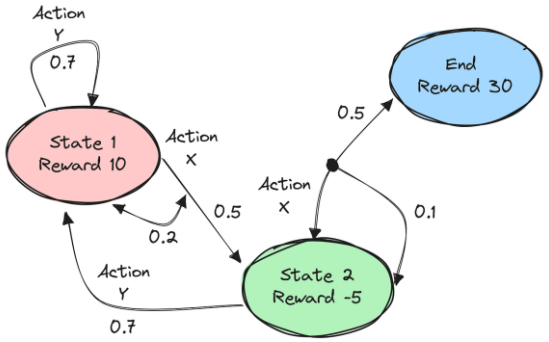
تاثیر action بر reward لحظه ای و آینده

تابع Value:

Multi-armed bandit : $q_*(a)$

MDP in general : $q_*(s,a)$ or $V_*(s)$

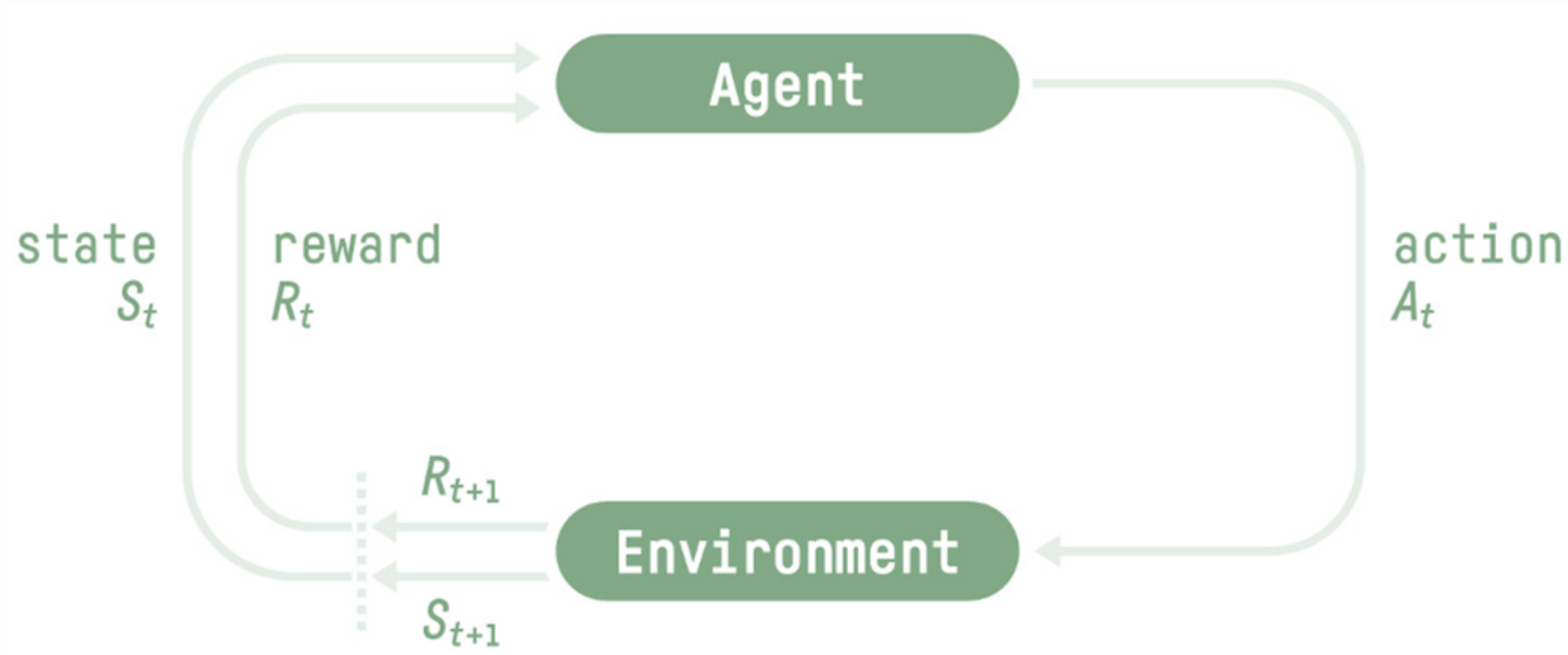
RL and MDP?



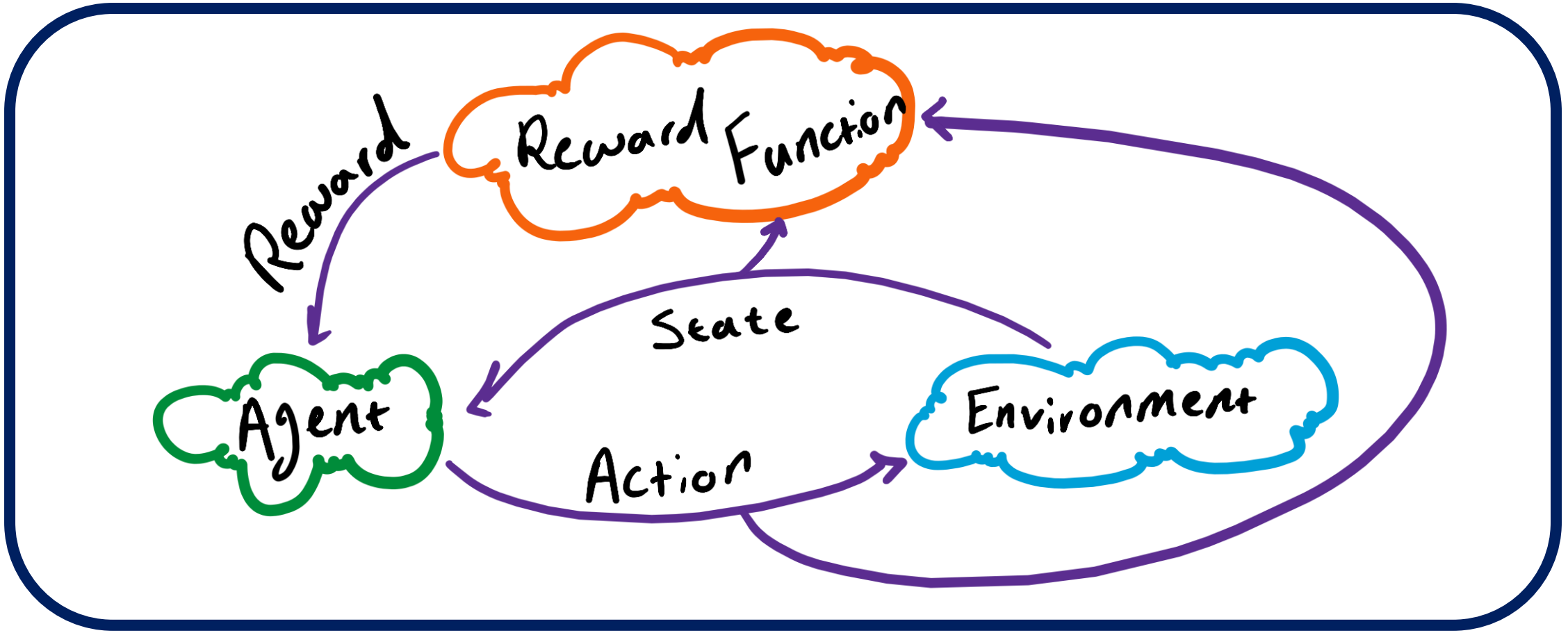
عامل و محیط؟

عامل؟

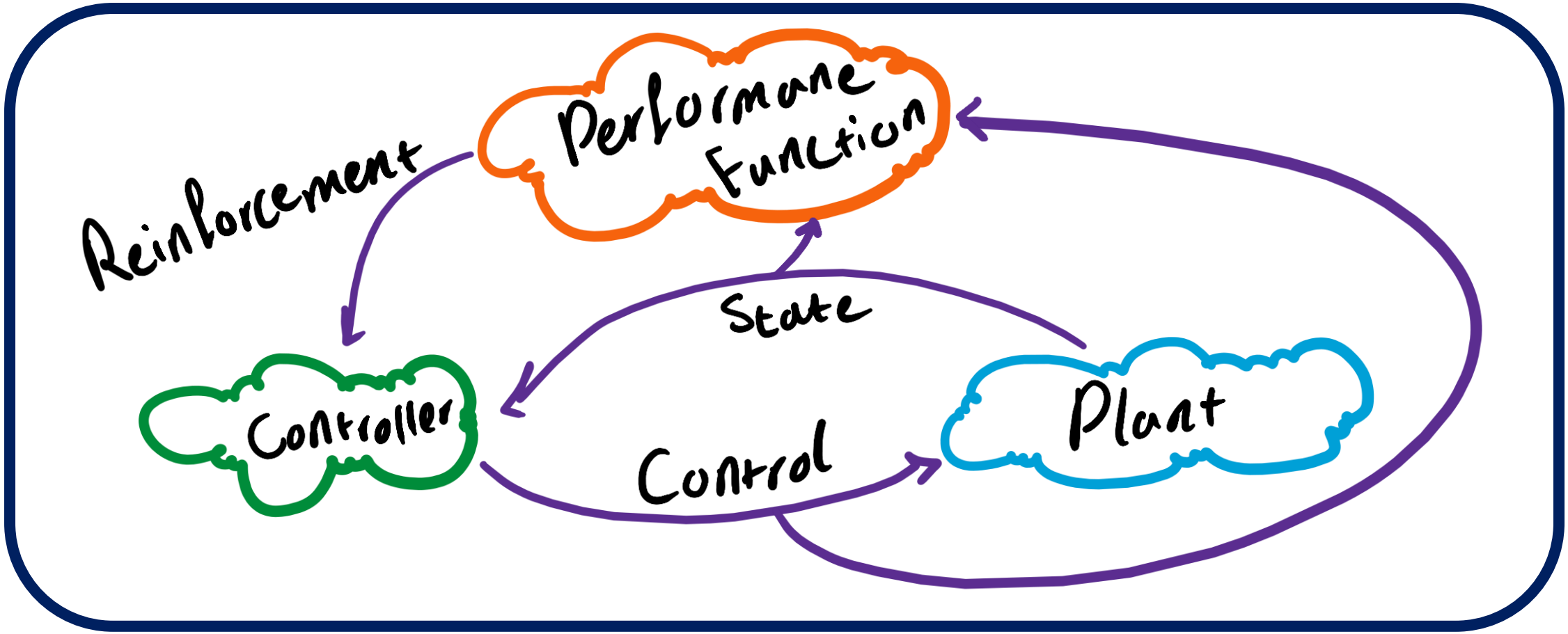
محیط؟



یادگیری تقویتی در ادبیات هوش مصنوعی



یادگیری تقویتی در ادبیات مهندسی



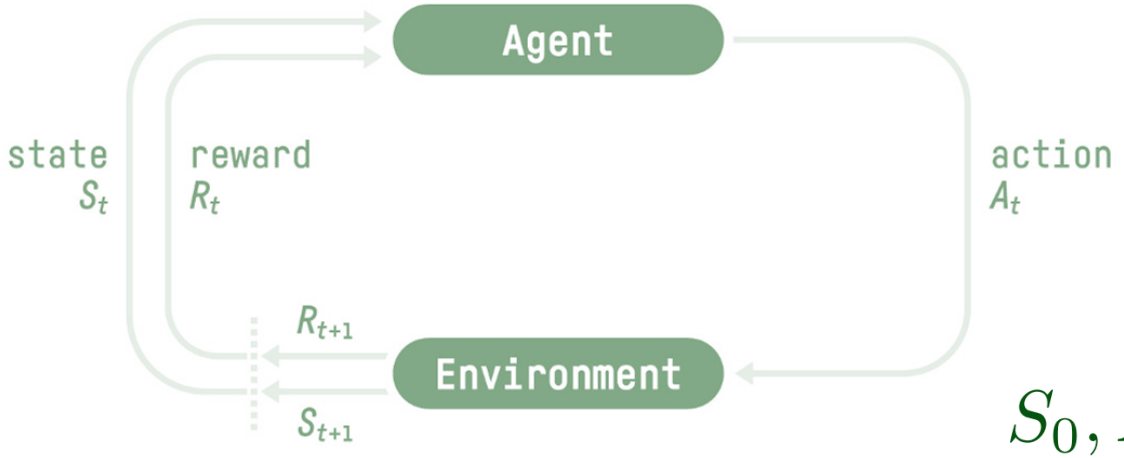
در هر لحظه $t = 0,1,2,3, \ldots \ldots$:

دریافت state محیط توسط agent به صورت $S_t \in \mathcal{S}$

انتخاب action مبتنی بر یک policy به صورت $A_t \in \mathcal{A}(s)$

پاسخ به A_t: تغییر حالت محیط به S_{t+1} و دریافت پاداش $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

A_t, S_t: random variables



تراژکتوری سیستم:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

دینامیک MDP

Definition

The function *p* defines the dynamics of the MDP:

$$p(s', r \,|\, s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$$

Computing the probability of transitioning to s' as the next state and receiving reward r

$$s', s \in \mathcal{S},\ r \in \mathcal{R},\ \text{and}\ a \in \mathcal{A}(s)$$

$$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$$

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \,|\, s, a) = 1,\ \text{for all}\ s \in \mathcal{S}, a \in \mathcal{A}(s)$$

P: توصیف کامل دینامیک محیط

تعیین توابع مختلف گذر حالت از تابع p:

$$p(s'|s,a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r|s, a)$$

$$p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1])$$

تعیین expected reward برای (state-action):

$$r(s,a) \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a)$$

$$r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

تعیین توابع مختلف گذر حالت از تابع p:

تعیین expected reward برای (state-action-next state):

$$r(s,a,s') \doteq \mathbb{E}[R_t \mid S_{t-1}=s, A_{t-1}=a, S_t=s'] = \sum_{r\in\mathcal{R}} r \frac{p(s',r\mid s,a)}{p(s'\mid s,a)}$$

$$r : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$$

اگر از روی تابع چگالی احتمال توام بخواهیم تابع چگالی احتمال یک متغیر را با دانستن متغیر دیگر بدست آوریم به آن تابع احتمال شرطی می‌گویند.

$$P(s',r \mid s,a) \text{ is known so } p(r \mid s,a,s') = \frac{P(s',r|s,a)}{P(s'|s,a)}$$

حال اگر E[Rt] را بخواهیم، حساب کنید:

$$E[R_t \mid s_{t-1}=s, R_{t-1}=a, s_t=s'] = \sum_{r\in R} rP(r \mid s,a,s')$$

تعیین Action و State

Action? Low-level Control → High-level Control

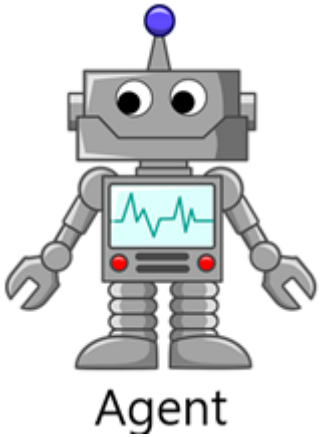
ولتاژ موتور یک بازوی ربات
حرکت ربات به چپ/راست

تصمیمی برای رسیدن به هدف

State? Low-level State → High-level State (sensor output)

موقعیت بازوی ربات
متحرک یا ثابت بودن ربات

نیازمند دانستن برای رسیدن به هدف



مرز بین عامل و محیط

ربات

حیوان

محیط ← غیر agent (توسط agent قابل تغییر نباشد)

اطلاعات agent از محیط؟

کامل

کم(نحوه محاسبه پاداش، نحوه تغییر حالتها)

هیچ

مرز بین محیط و agent؟ حد کنترل agent (قابل تغییر درمقاصد مختلف)

مدلسازی مسئله RL در فریمورک MDP

Action: سیگنال کنترل agent

State: تاثیر سیگنال agent

Reward: سیگنال نشان دهنده هدف agent

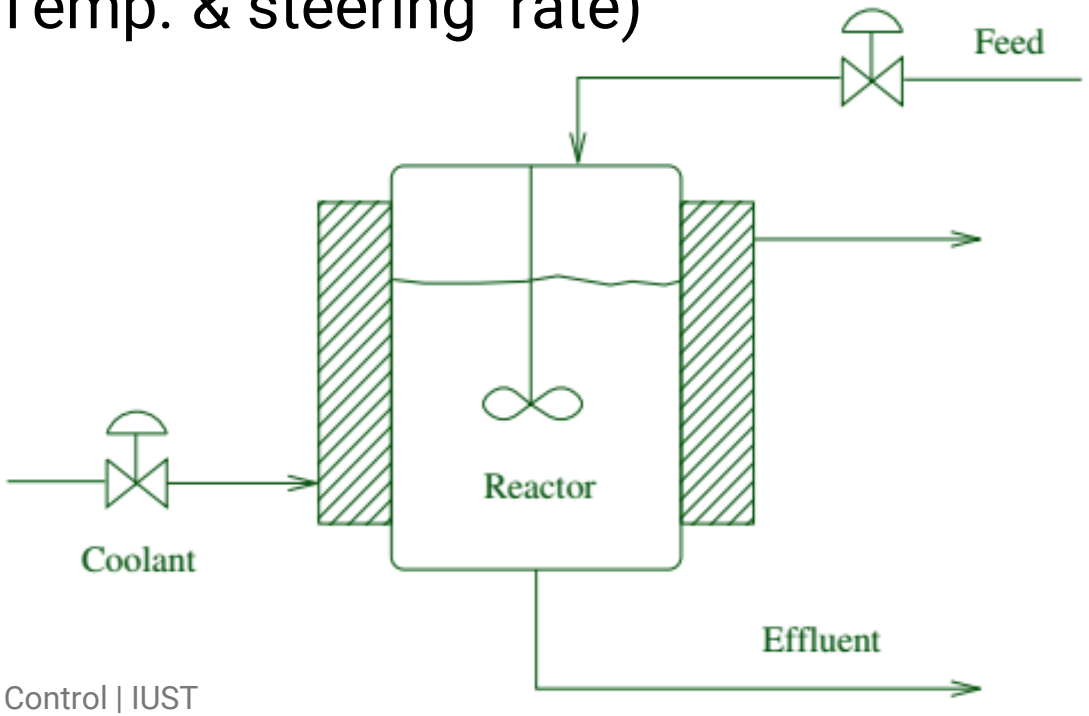
کارایی RL ←→ انتخاب مناسب action و state

مثال

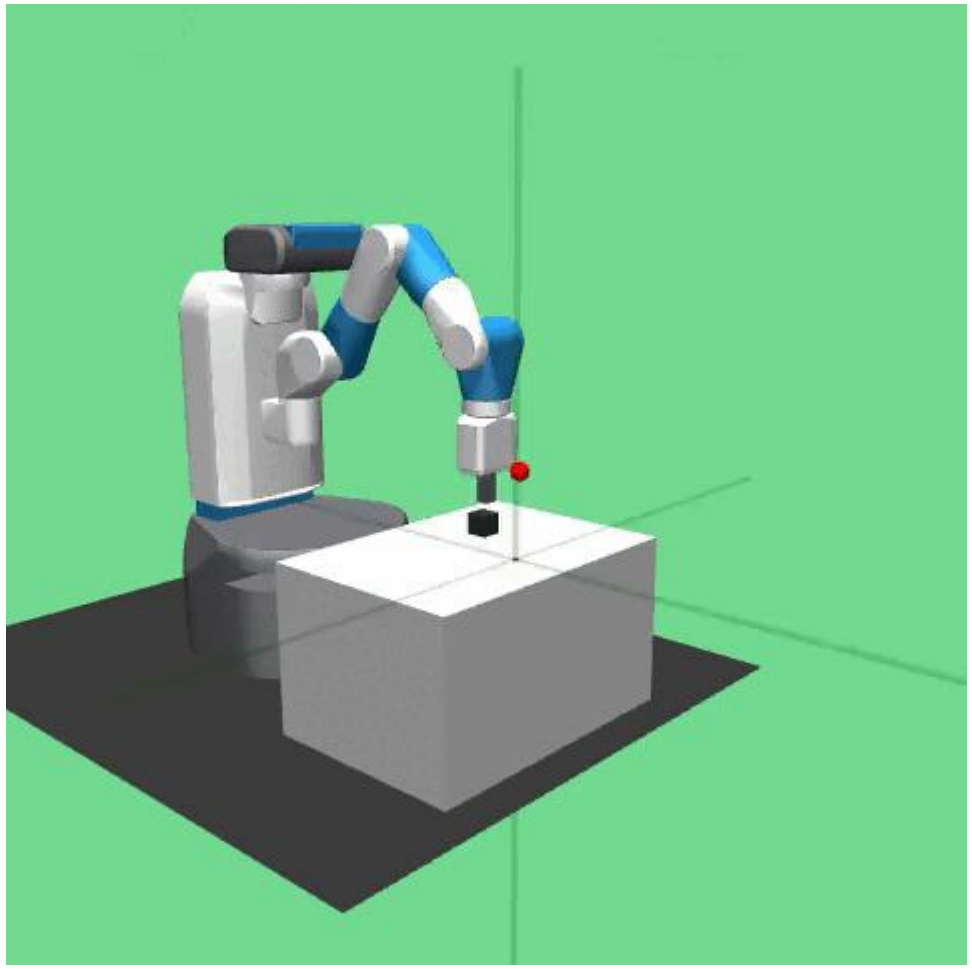
Bioreactor

States: Temperature , Concentrations

Action: (control valve) → set points (Temp. & steering rate)
Reward: chemical product rate



Pick and Place



مثال

هدف: حرکت سریع و نرم!

Action؟

State؟

Reward؟

Action؟ ولتاژ موتور هر مفصل

State؟ زاویه و سرعت زاویه ای

Reward؟ هر حرکت مطلوب کامل 1+

هر برخورد ضربه ای $\epsilon-$

هر سمپل تایم β-

مثال

Action در رانندگی؟؟

گشتاور چرخ (در دوجهت)
ترمز، گاز، فرمان
فرامین دست و پای راننده
رانندگی در کدام مسیر

کدام را ترجیح می دهید؟

مثال

ربات جمع آوری زباله:

سنسورها....
عملگرها...
ناوبری و کنترل ....
باتری قابل شارژ
Action, State ؟؟؟؟

تعریف در سطح بالا:
State : سطح شارژ باتری : low-high
Action: {جستجو- برگشت به محل شارژ – انتظار } یا {جستجو-– انتظار }

مثال

دینامیک سیستم:

Search: اگر باتری high باشد، به احتمال α در high میماند و با $1-\alpha$ به low میرود
اگر باتری low باشد، به احتمال βدر low میماند و با $1-\beta$خالی می شود

Wait: مصرف باتری ندارد

Recharge: از low به high میرود

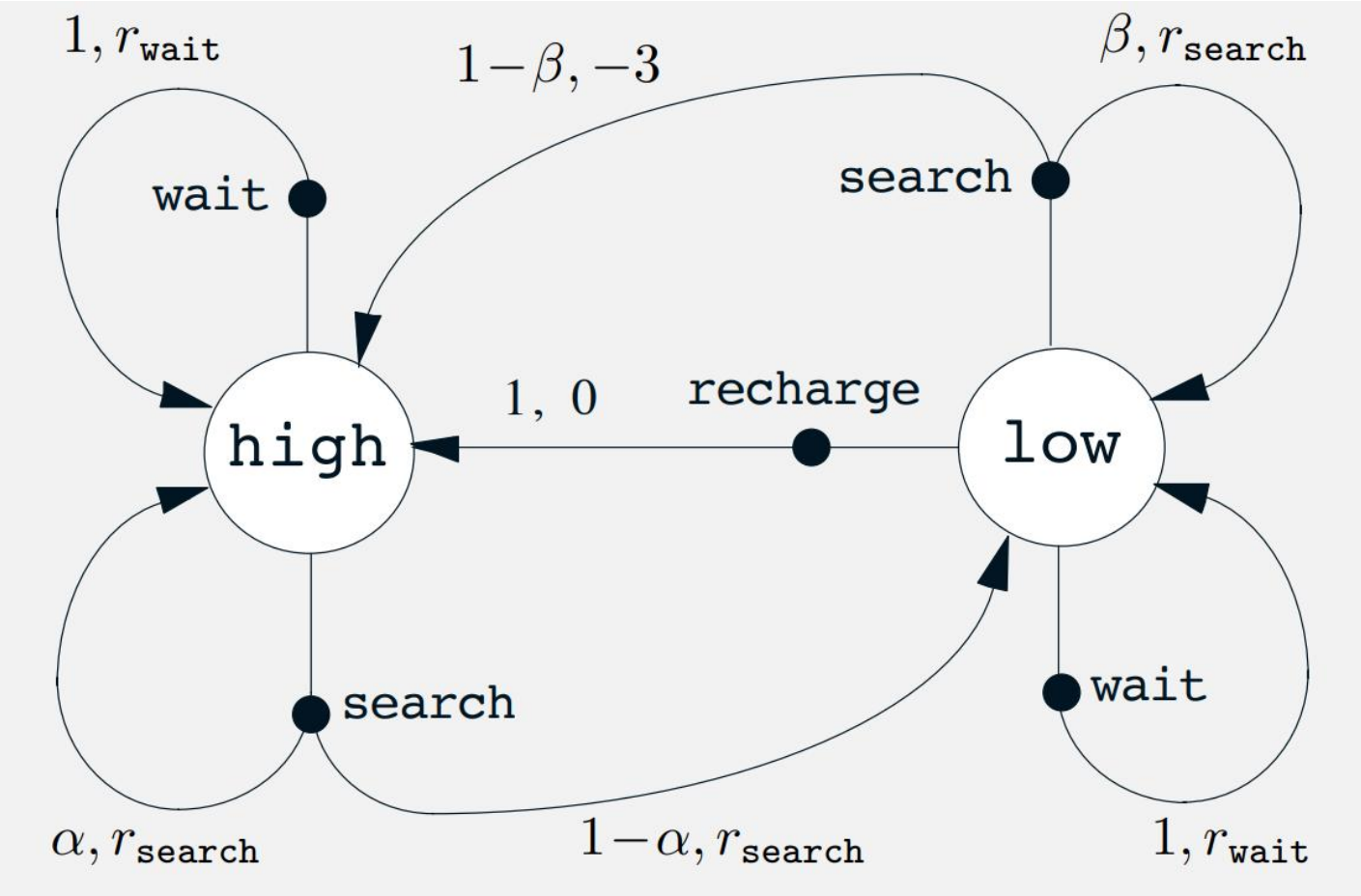
Reward:

زباله در سبد: $+1*(r_{search} \ or \ r_{wait})$

باتری خالی شود: -3

مثال

دیاگرام دینامیک finite MDP برای ربات:



مثال

تابع احتمال transition و reward برای finite MDP:

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
low	search	high	$1-\beta$	-3
low	search	low	β	r_{search}
high	wait	high	1	r_{wait}
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	r_{wait}
low	recharge	high	1	0
low	recharge	low	0	-

معرفی هدف برای عامل

سیگنال عددی reward که با max کردن امید ریاضی آن به هدف برسد
(یا پاداش تجمعی)

محدودیت؟؟

تعریف پاداش

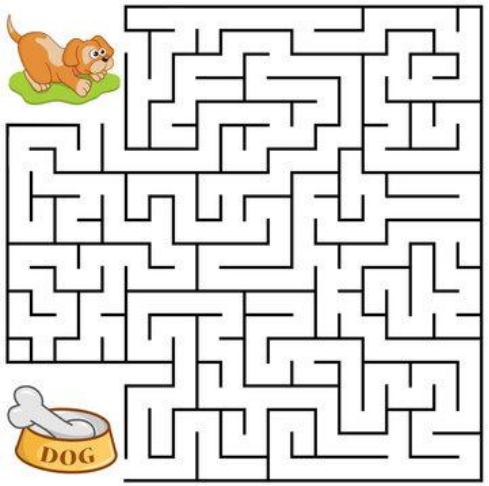
- ❑ معرف آنچه میخواهیم انجام شود (نه چگونه انجام شود)
- ❑ عدم ارائه دانش اولیه با پاداش (مقداردهی اولیه policy یا value)
- ❑ روشی برای ارتباط با ربات

اگر agent پاداش را max کند باید به هدف برسد

مثال

شطرنج

Maze



مثال

زباله : 1+

برخورد مانع : 1-

مثال

بازی Go (یکی از قدیمی‌ترین بازی‌ها و سخت‌ترین بازی‌ها)

عامل: بازیکن

محیط: حریف

حالت: آرایش برد بازی

عمل: قرار دادن مهره

پاداش:

+۱ برای برد

-۱ برای باخت

2016 Milestone: AlphaGo defeats world champion Lee Sedol (4-1).

Return and Episodes

فرموله کردن پاداش دراز مدت
در شکل ساده: جمع پاداش ها از t بعد

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

هدف : max کردن expected return

Episode:
تعاملی بین agent و محیط که پایانی دارد
دارایTerminal state
Continuing Task:
تعاملی بین agent و محیط که پایانی ندارد.

جمع‌بندی

Type of tasks

Episodic: **starting point** and an **ending point** (a terminal state)

Continuing: task that **continue forever** (no terminal state)

Deep RL Course

Discounted Return

Definition

The *return* G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

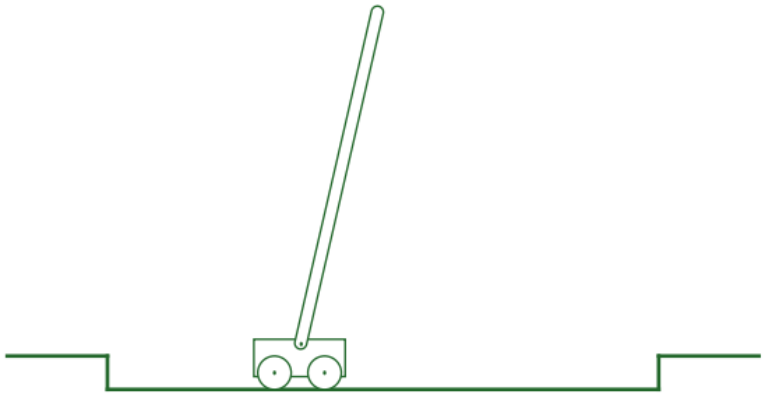
$0 < \gamma < 1 \quad \rightarrow \quad G_t < \infty$

$\gamma \rightarrow 0?$

$\gamma \rightarrow 1?$

پیاده سازی بازگشتی Return

$$
\begin{aligned}
G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\
&= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots \right) \\
&= R_{t+1} + \gamma G_{t+1}
\end{aligned}
$$



مثال

پاندول معکوس مقید

$$\theta_{min} < \theta < \theta_{max}$$

Episodic or Continuing?

Episode:

اجرای هر Task از شروع تا زمان افتادن

Reward:

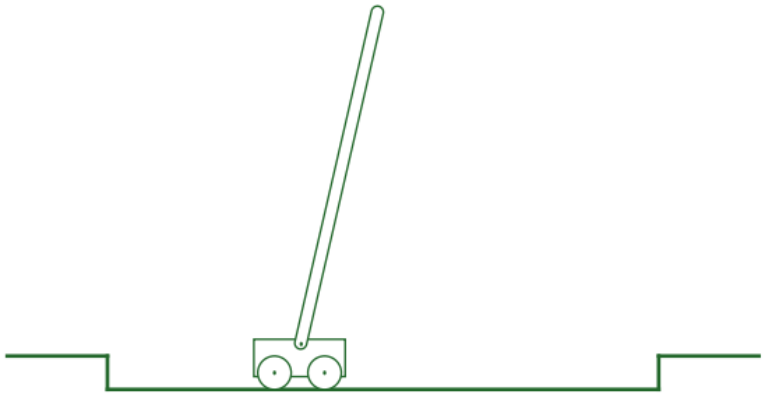
هر stepکه خطا رخ نداده: 1+

Return:

زمان شروع تا خطا

بالانس موفق:

Return → ∞



مثال

پاندول معکوس مقید

$$\theta_{min} < \theta < \theta_{max}$$

Episodic or Continuing?

Continuing:

Taskهای پشت سرهم با شروع بعد از هر خطا از وسط

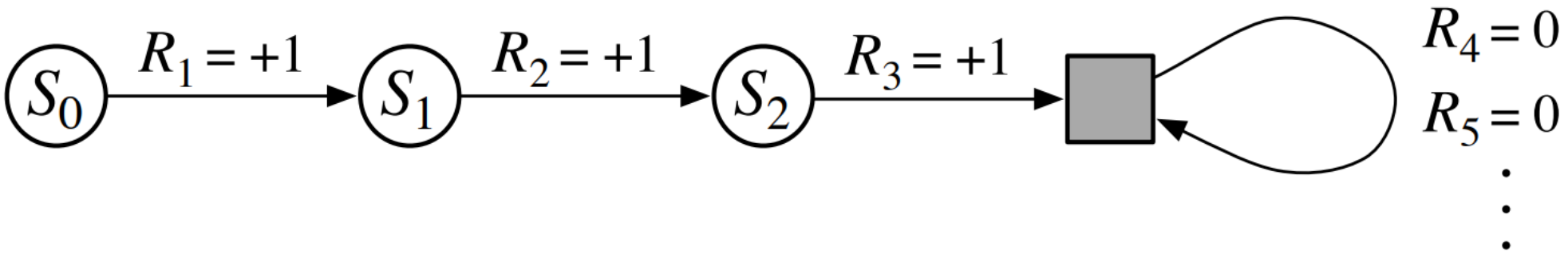
Reward:

هر بار خطا 1-

Discounted Return:

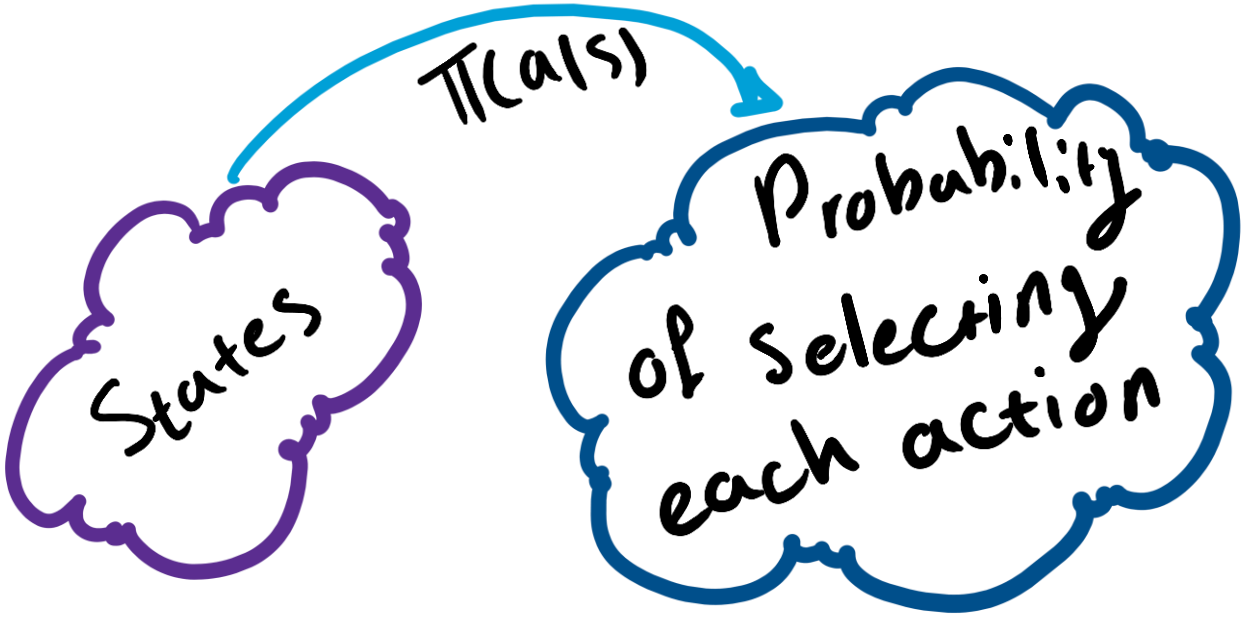
تا $-\gamma^K$ (K: زمان خطا)

نمایش یکسان Episodic و Continuing:



$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

$$T = \infty \text{ or } \gamma = 1 \text{ (but not both)}$$



مسائل Reinforcement Learning

State
State & Action
} Value تخمین تابع

Value: مقدار Expected Return
Policy: روش‌های انتخاب Action

Definition

A *policy* π is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}\left[A_t = a \mid S_t = s\right]$$

توابع ارزش

اگر وقتی در state : s هستیم و پالیسی π انتخاب شده باشد،

Definition

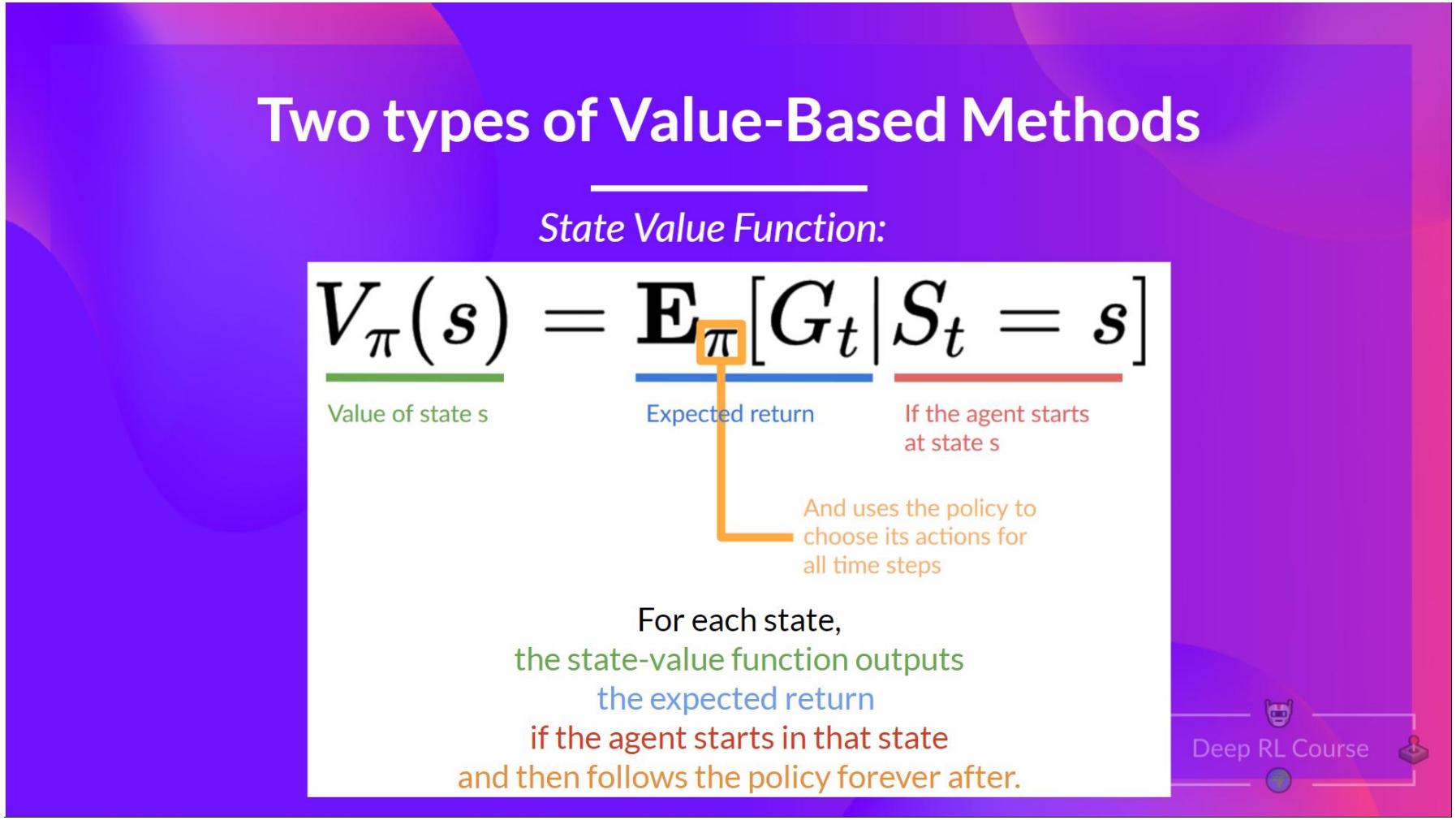
The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$v_\pi(s) = \mathbb{E}_\pi [G_t \mid S_t = s]$$

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

State-value function

توابع ارزش


Two types of Value-Based Methods
State Value Function:
$V_\pi(s) = \mathbf{E}_\pi[G_t | S_t = s]$
Value of state s
Expected return
If the agent starts at state s
And uses the policy to choose its actions for all time steps
For each state,
the state-value function outputs
the expected return
if the agent starts in that state
and then follows the policy forever after.
Deep RL Course

توابع ارزش

اگر وقتی در state : s هستیم و action : a اعمال می‌شود و پس از آن پالیسی π انتخاب شده باشد،

Definition

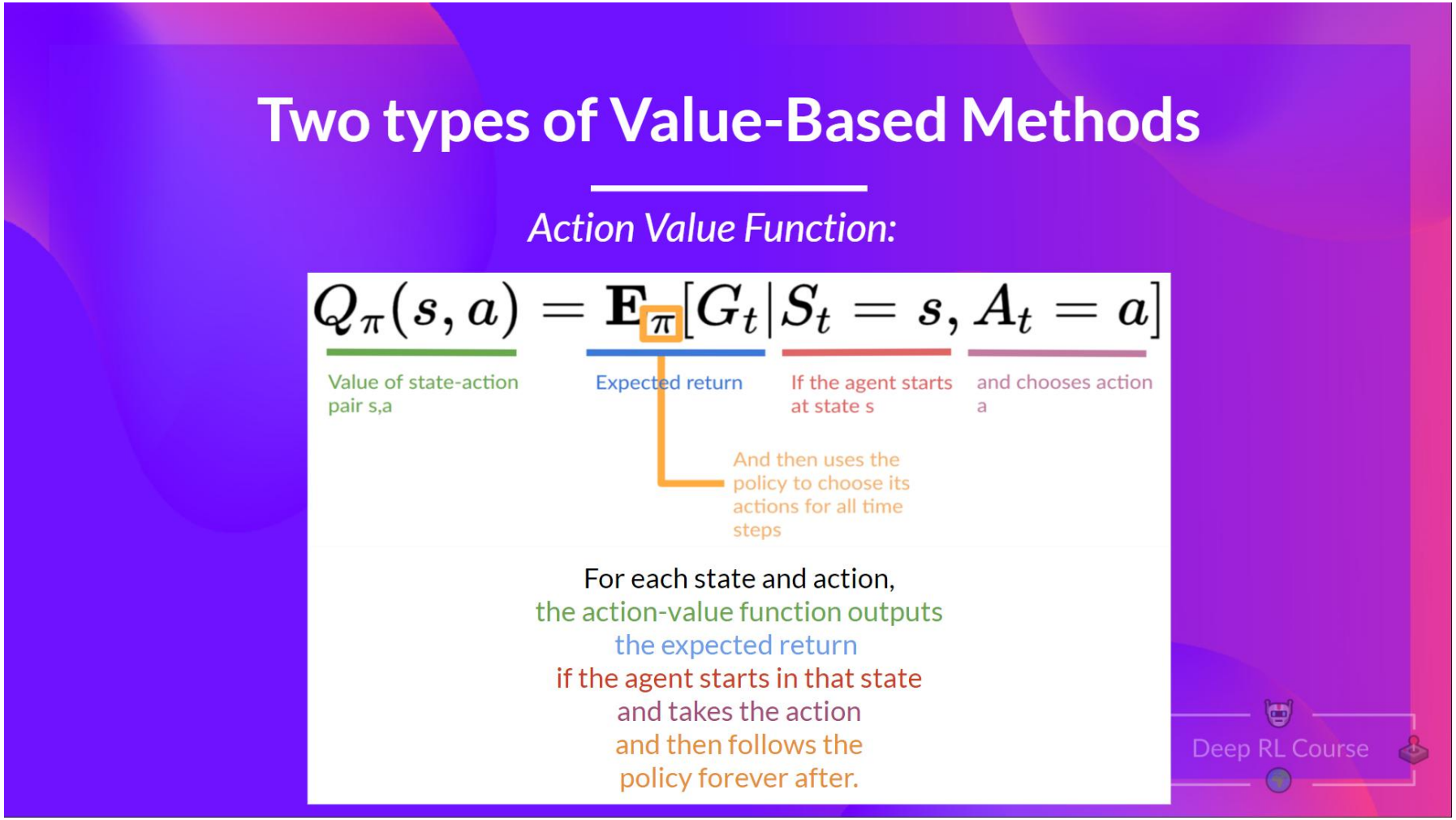
The *action-value function* $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π

$$q_\pi(s, a) = \mathbb{E}_\pi [G_t \mid S_t = s, A_t = a]$$

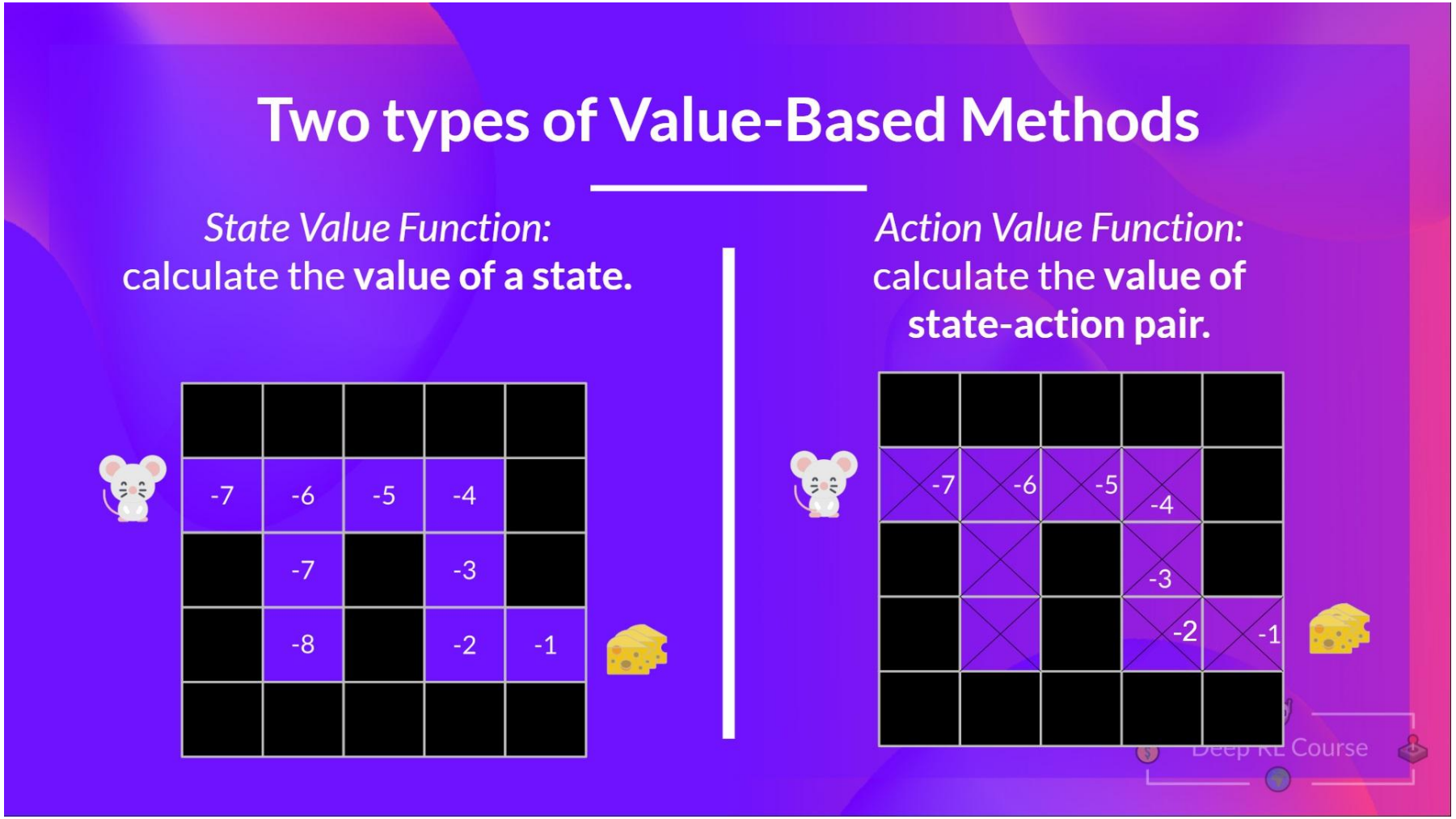
$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t=s, A_t=a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t=s, A_t=a\right]$$

Action-value function

توابع ارزش


Two types of Value-Based Methods
Action Value Function:
$Q_\pi(s,a) = \mathbf{E}_\pi[G_t | S_t = s, A_t = a]$
Value of state-action pair s,a
Expected return
If the agent starts at state s
and chooses action a
And then uses the policy to choose its actions for all time steps
For each state and action,
the action-value function outputs
the expected return
if the agent starts in that state
and takes the action
and then follows the
policy forever after.
Deep RL Course

جمع‌بندی


Two types of Value-Based Methods
State Value Function:
calculate the value of a state.
-7
-6
-5
-4
-7
-3
-8
-2
-1
Action Value Function:
calculate the value of state-action pair.
-7
-6
-5
-4
-3
-2
-1
Deep RL Course

توابع ارزش

تخمین بازگشتی تابع value:

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

توابع ارزش

تخمین بازگشتی تابع value:

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

معادله بلمن

بیان ارتباط بین تخمین تابع state value در t و t+1

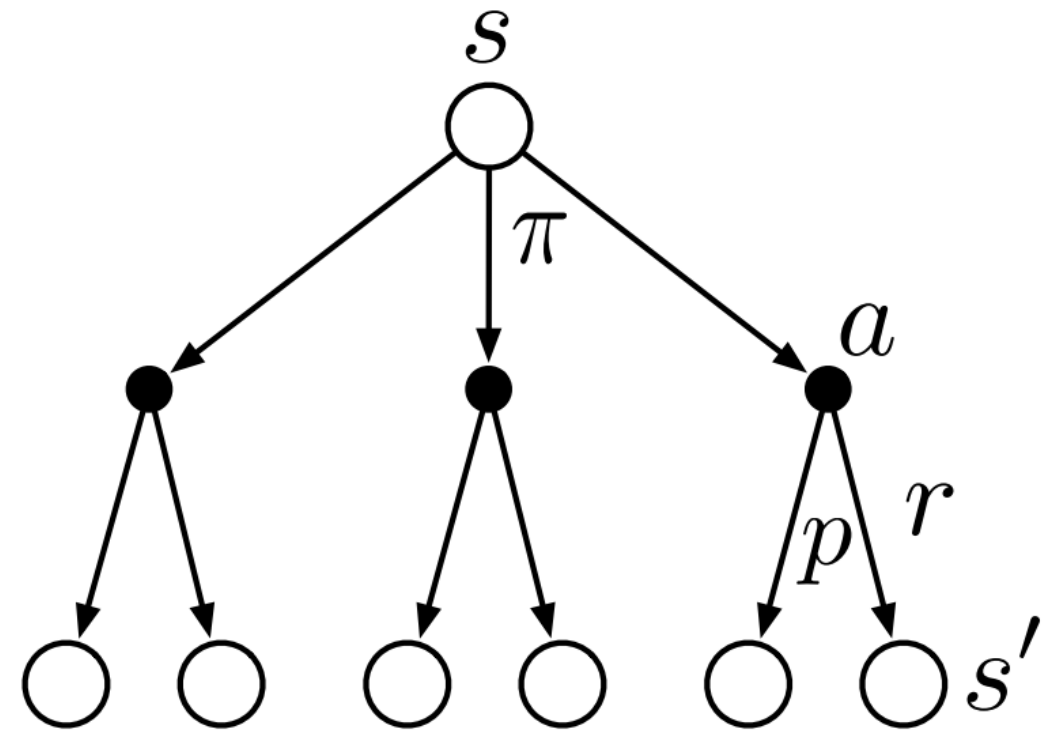
حل یکتای معادله بلمن : V_π

توصیف معادله بلمن

State:

State – Action:

State:



Backup diagram for v_π

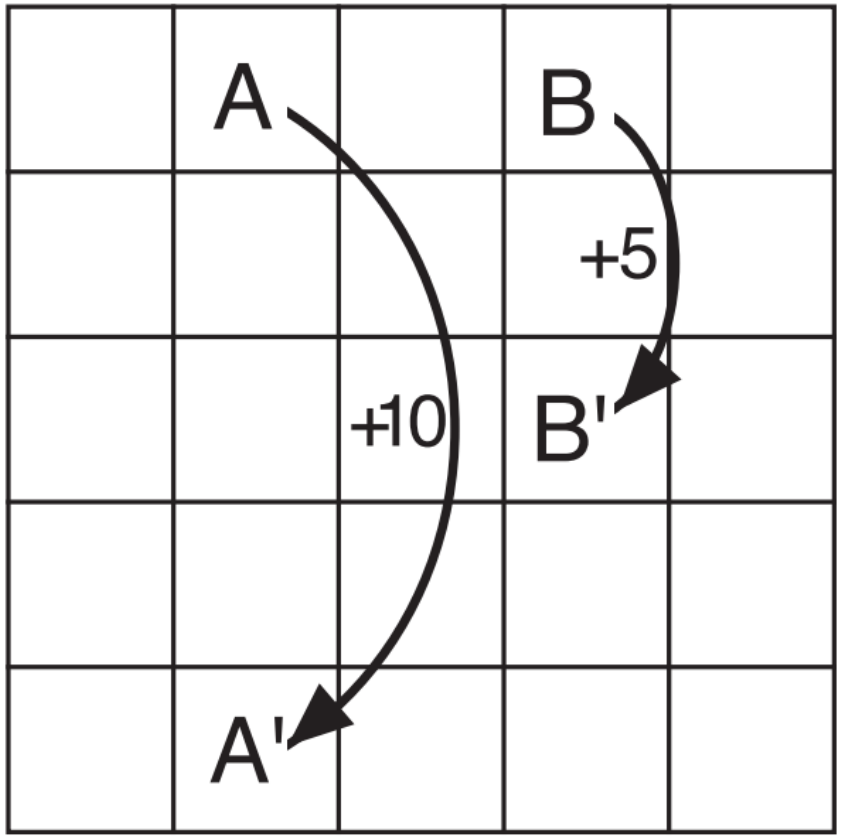
روش محاسبه تابع value از معادله بلمن

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

حل دستگاه معادلات خطی فوق (n معادله n مجهول $V_\pi(s)$ برای همه $s \in S$)

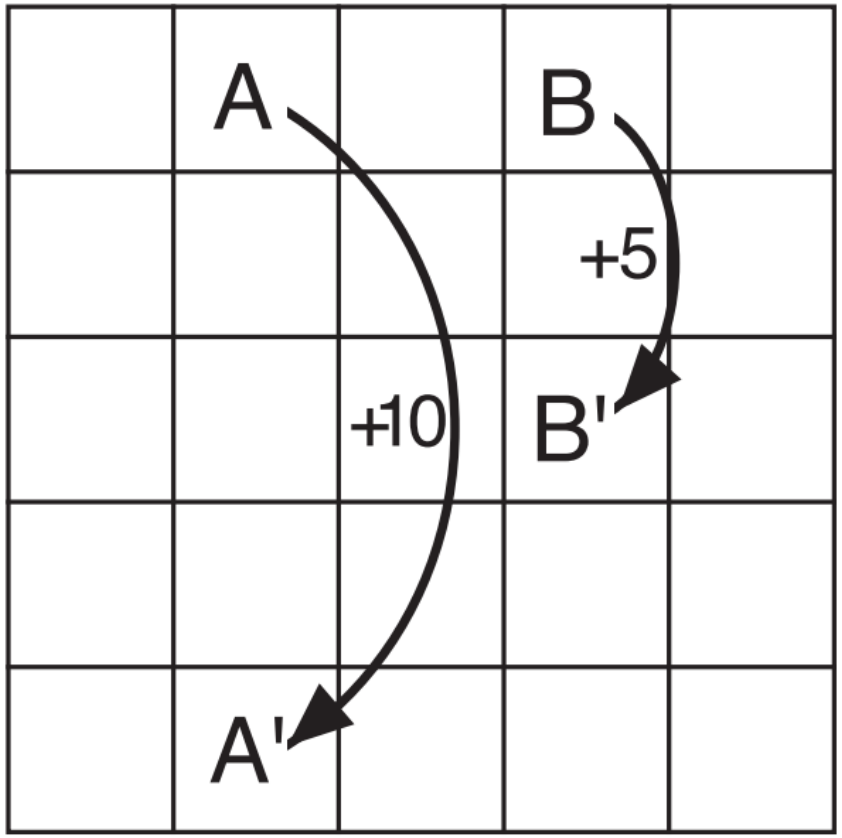
حل یکتای معادله بلمن: V_π

مثال



State: cell
Action: up-down-left-right
Reward:

برخورد با دیواره : -1 (s'=s)
خروج از A : +10 (برای هر اکشن) ← به A'
خروج از B: +5 (برای هر اکشن) ← به B'
سایر 0



مثال

اگر فقط A برویم:

$$G_t = 10 + 10\gamma^5 + 10\gamma^{10} + \cdots = \frac{10}{1-\gamma^5}$$

اگر فقط B برویم:

$$G_t = 5 + 5\gamma^3 + 5\gamma^6 + \cdots = \frac{5}{1-\gamma^3}$$

$$\gamma = 0.9 \quad \rightarrow G_t = ?$$

حل دستگاه معادلات بلمن برای همه سلول‌های جدول (تعیین $V_\pi(s)$ برای هر state)

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

State-value function for the equiprobable random policy

تاثیر γ در $V_\pi(s)$؟

تابع $p(s',r|s,a)$؟؟

تابع $\pi(a|s)$ ؟؟

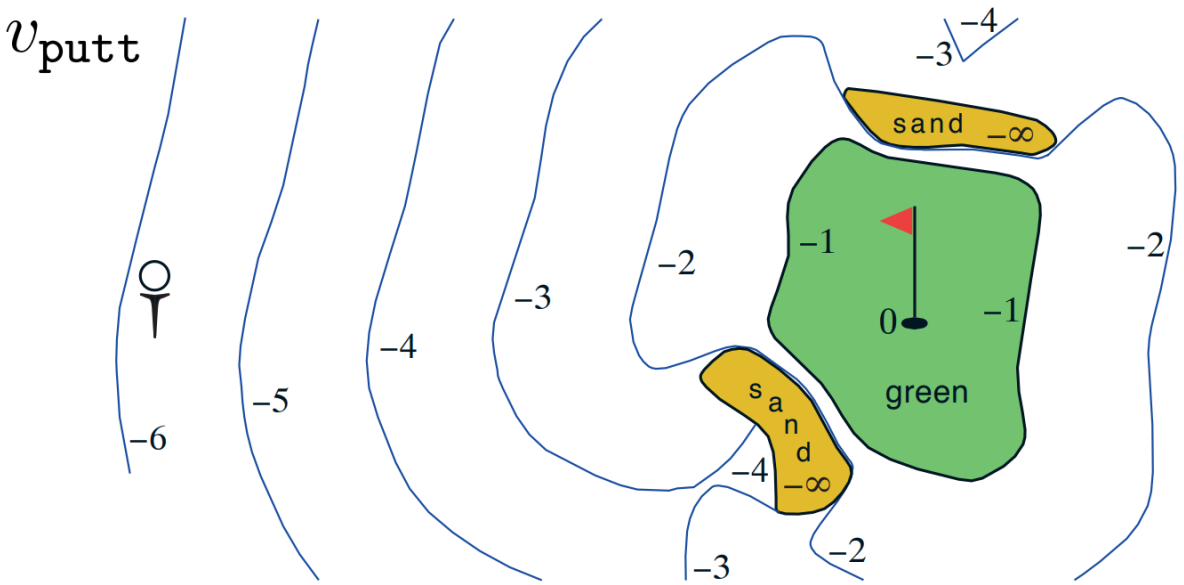
حل دستگاه معادلات بلمن برای همه سلول‌های جدول (تعیین $V_\pi(s)$ برای هر state)

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

State-value function for the equiprobable random policy

این اعداد حل معادله بلمن برای پالیسی رندوم با توزیع یکنواخت است.
ردیف‌های کنار بخاطر برخورد با دیواره و جریمه -۱ منفی شده است.
ارزش A کمتر از ۱۰ است چون به سمت دیواره پرت می‌کند.
ارزش B بیشتر از ۵ است چون به سمت ارزش بیشتر از ۰ پرت می‌کند.

مثال



State: Distance to the hole
Action: Driver – Putter
Reward:

1- برای هر ضربه توپ
Value برای هر state:
منفی تعداد ضربه ها تا حفره

فرض: نشانه گیری دقیق و قطعی ولی برد توپ محدود

Definition

The *optimal state-value function* $v_*(s)$ is the maximum value function over all policies

$$v_*(s) = \max_\pi v_\pi(s)$$

The *optimal action-value function* $q_*(s,a)$ is the maximum action-value function over all policies

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

سیاست بهینه

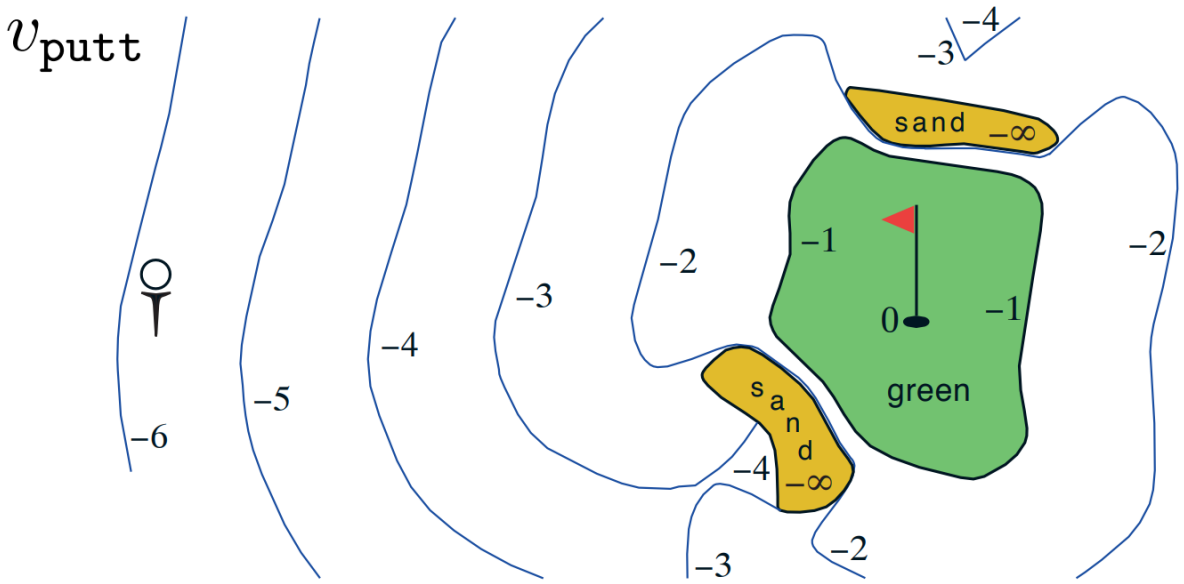
$$\pi \geq \pi'$$

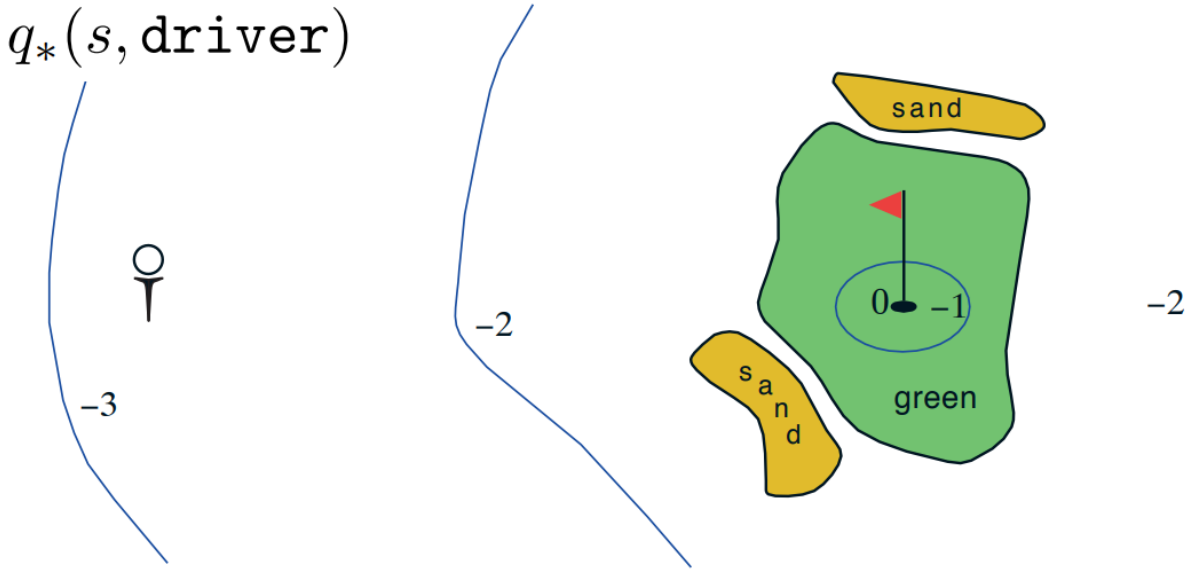
$$v_\pi(s) \geq v_{\pi'}(s)$$

بیان $q_*(s,a)$ بر حسب $v_*(s)$:

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

مثال





فرض: شروع با درایور

سوال : $q_*(s,driver) =?$

اکشن بهینه برای نقاط دور: دو درایور و یک پاتر

معادله بهینگی بلمن برای تابع State Value

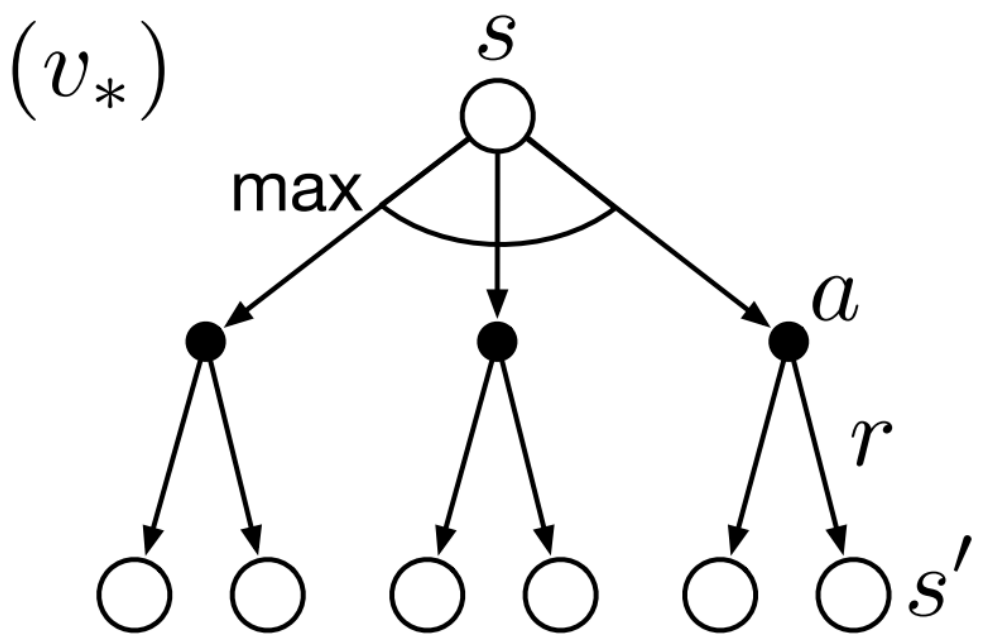
$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big].
\end{aligned}
$$

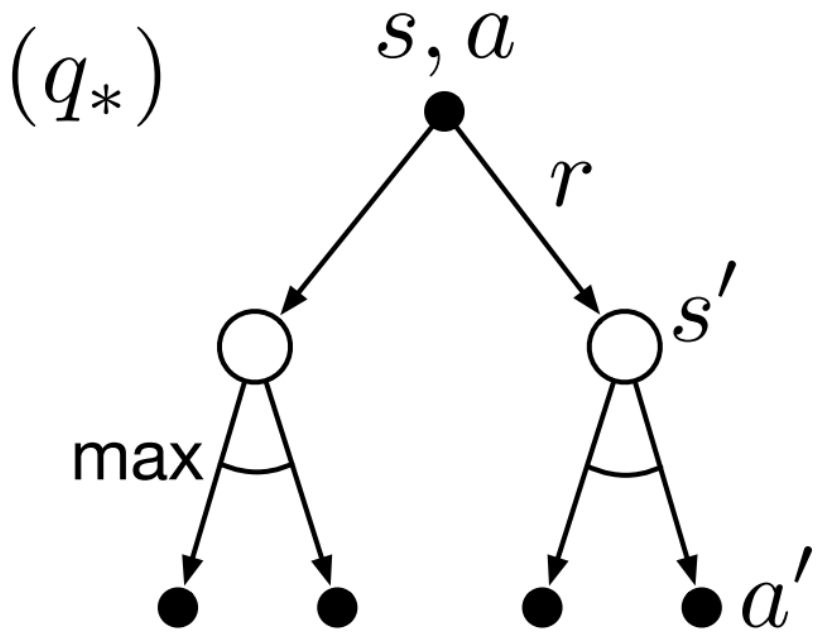
حل دستگاه معادلات غیر خطی فوق (n معادله n مجهول $V_*(s)$ برای همه $s \in S$)

معادله بهینگی بلمن برای تابع Action Value

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

Backup Diagrams برای v_* و q_*





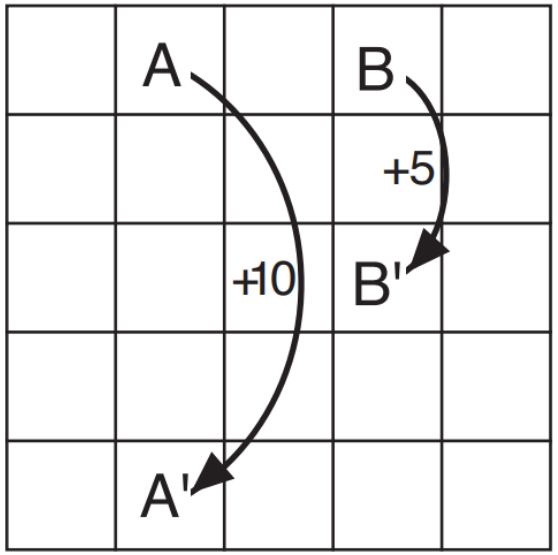
Optimal State Value Function

تعیین پالیسی بهینه به سادگی از روی $V_*(s)$ بدست آمده (سرچ ساده)
پالیسی بهینه : هر پالیسی که بر اساس $V_*(s)$ ، Greedy باشد.
توجه: پالیسی با انتخاب Greedy در دراز مدت هم بهینه است.

Optimal Action Value Function

تعیین پالیسی بهینه از روی $q_*(s,a)$ بدست آمده:
پالیسی که حداکثر $q_*(s,a)$ را دارد! (ساده تر از قبل)

حل دستگاه معادلات بلمن برای همه سلول‌های جدول (تعیین $V_\pi(s)$ برای هر state) Gridworld



Gridworld

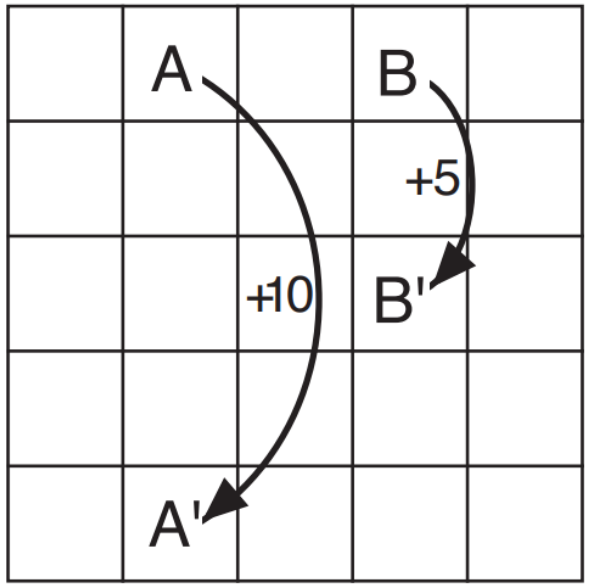
22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*

حل دستگاه معادلات بهینگی بلمن (optimal state- value) با ۲۵ معادله غیرخطی

Gridworld

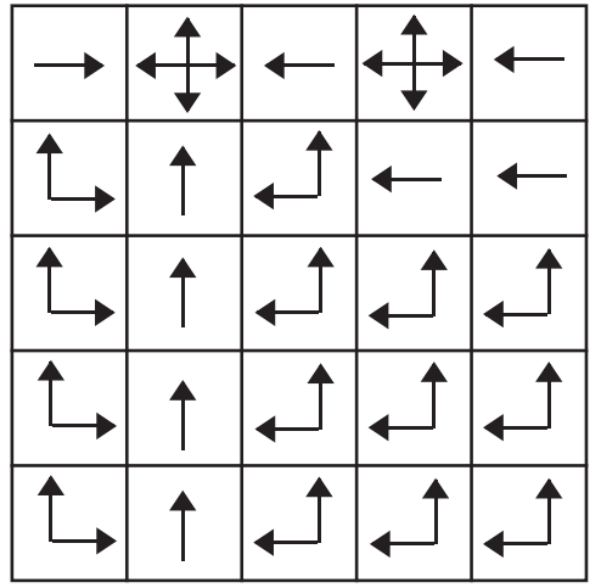
تعیین $\pi_*(s)$ از روی $V_*(s)$:



Gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

تاثیر γ در جدول فوق؟

Recycling Robot

معادلات بهینگی بلمن؟

State ها: Low – High ← محاسبه $V_*(s)$ با max گرفتن روی action های ممکن

Action ها :

search & wait ← High

search & wait & recharge ←Low

S=High:

Action	S'	p	r
Search	High	α	r_s
Search	Low	$1-\alpha$	r_s
Wait	High	1	r_w

Recycling Robot

معادلات بهینگی بلمن؟

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

S=High:

Action	S'	p	r
Search	High	α	r_s
Search	Low	$1-\alpha$	r_s
Wait	High	1	r_w

$$\begin{aligned}
v_*(\mathtt{h}) &= \max\left\{\begin{array}{l} p(\mathtt{h}|\mathtt{h},\mathtt{s})[r(\mathtt{h},\mathtt{s},\mathtt{h}) + \gamma v_*(\mathtt{h})] + p(\mathtt{l}|\mathtt{h},\mathtt{s})[r(\mathtt{h},\mathtt{s},\mathtt{l}) + \gamma v_*(\mathtt{l})], \\ p(\mathtt{h}|\mathtt{h},\mathtt{w})[r(\mathtt{h},\mathtt{w},\mathtt{h}) + \gamma v_*(\mathtt{h})] + p(\mathtt{l}|\mathtt{h},\mathtt{w})[r(\mathtt{h},\mathtt{w},\mathtt{l}) + \gamma v_*(\mathtt{l})] \end{array}\right\} \\
&= \max\left\{\begin{array}{l} \alpha[r_\mathtt{s} + \gamma v_*(\mathtt{h})] + (1-\alpha)[r_\mathtt{s} + \gamma v_*(\mathtt{l})], \\ 1[r_\mathtt{w} + \gamma v_*(\mathtt{h})] + 0[r_\mathtt{w} + \gamma v_*(\mathtt{l})] \end{array}\right\} \\
&= \max\left\{\begin{array}{l} r_\mathtt{s} + \gamma[\alpha v_*(\mathtt{h}) + (1-\alpha) v_*(\mathtt{l})], \\ r_\mathtt{w} + \gamma v_*(\mathtt{h}) \end{array}\right\}.
\end{aligned}$$

For any choice of $r_\mathtt{s}$, $r_\mathtt{w}$, α, β, and γ, with $0 \le \gamma < 1$, $0 \le \alpha, \beta \le 1$
We can calculate $V_*(high)$

Recycling Robot

معادلات بهینگی بلمن؟

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

S=Low:

Action	S'	p	r
Search	Low	β	r_s
Search	High	$1-\beta$	r_s
Wait	Low	1	r_w
Recharge	High	1	0

$$v_*(\mathtt{l}) = \max \left\{ \begin{array}{l} \beta r_{\mathtt{s}} - 3(1-\beta) + \gamma[(1-\beta)v_*(\mathtt{h}) + \beta v_*(\mathtt{l})], \\ r_{\mathtt{w}} + \gamma v_*(\mathtt{l}), \\ \gamma v_*(\mathtt{h}) \end{array} \right\}$$

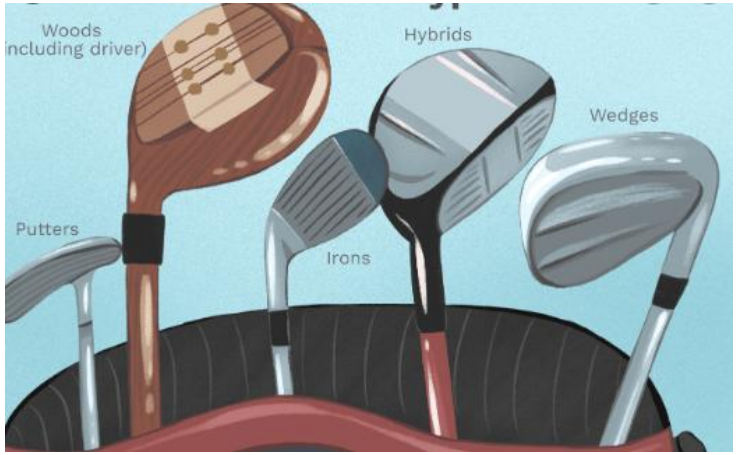
Golf

معادلات بهینگی بلمن؟

$$v_*(s) = \max_a \sum_{s',r} p(s', r | s, a) \left[r + \gamma v_*(s') \right]$$

$$V_*(s) = \max \left[\sum_{r,s'} P\left(s', -1 \mid s, \text{ Putter}\right) \left[-1 + \gamma V_*\left(s'\right) \right] \right.$$

$$\left. , \sum_{r,s'} P\left(s', -1 \mid s, \text{ driver}\right) \left[-1 + \gamma V_*\left(s'\right) \right] \right.$$



جمع‌بندی ...

حل مسئله بهینگی بلمن ← حل مسئله Reinforcement Learning

الزامات:

- دانستن دینامیک محیط : p
- حجم محاسبات!
- خاصیت مارکوف داشتن مسئله

برای backgammon: 10^{20} حالت ! ← محاسبه q_* ,V_*!!!

راه حل: Approximatin

- Dynamic Programming