



پادگیری تفویت

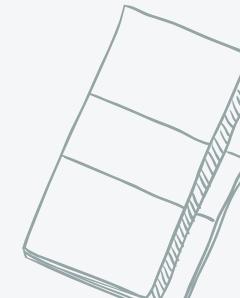
آرش عبدی هجراندوست

arash.abdi.hejrandoost@gmail.com

دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

نیم سال اول ۱۴۰۲-۱۴۰۳





یادگیری تقویتی

REINFORCEMENT LEARNING - RL

✖ یادگیری با نظارت: مجموعه‌ای از داده‌های برجسته خورده

جفت‌های ○

- موقعیت - تصمیم ■
- مسگر - عمل ■
- وودی - فروختی ■

آموزش سنتی ○

دستورالعملی : هرگاه، ... آنگاه ... ■

هرچند دارای تعمیم به آموزش‌های داده نشده است ■

✖ یادگیری تقویتی:

یادگیری با پاداش/جریمه ○

به جای معلم و پیش آموزش: تجربه و آموزش در میان اجرا ○

چند حثال

بازی شطرنج ✗

نگاه نظاری: ○

موقعیت ← مرکز

پایگاه داده‌ای از جفت‌های فوق از مرکات فرد بزنده در بازی‌های های اساتید بزرگ شطرنج

مجم خیلی کم در مقایسه با نیاز (10^8 - کل موقعیت‌ها: 10^{40})

موقعیت جدید؟ به کاری می‌گذیم ولی تفسیری از آن نداریم!

فوتبال (باتی) ✗

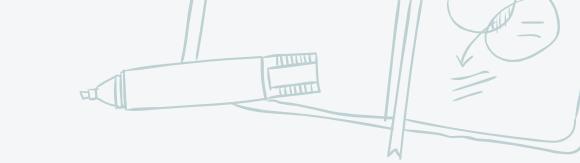
آموزش به حیوانات (سیرک و ...)

مهارت‌های ورزشی (فیلیپینی، ووپایی، سربالا بودن هنگام پا به توب بودن)

مهارت‌های اجتماعی ✗

آموزش

پرورش



یادگیری تقویتی اما: X

- با جهان پیرامون تعامل دارد.
- دائماً جایزه (سیگنال تقویتی) دریافت می‌کند.
- اعمالش را و دریافتش را از جهان به روز می‌کند.

شباهت با (MDP) X

- هدف: افزایش میانگین پاداش‌ها
- سیستم در RL، در جهانی از جنس MDP به سر می‌برد.
- توصیف جهان است: RL (وش یادگیری رفتار در آن MDP)

یادگیری تقویتی، یادگیری با نظارت نیست، هرچند عملاً دارای نوعی از نظارت است X

چند نکته دیگر

✖ پاداش در انتهای دنباله‌ای از اعمال

✖ تأثیرگذار در اعمال قبلی

○ در نوبت‌های بعدی اجرا

○ در ادامه اجرای فعلی اگر ...

✖ مناسب برای محیط کاملاً جدید و ناشناخته

○ بازی جدید

✖ تامین پاداش ارزان‌تر از تامین جفت نمونه-برچسب است

○ معلوم است هدف چیست = پاداش

○ برای بیان آن نیاز به خبرگی نیست (نمونه با برچسب غلط، کم احتمال‌تر است)

○ در مثال‌های فوق

✖ می‌توان صرفاً در انتهای پاداش نداد (Sparse Rewards)

○ مرکت‌های میانی هم پاداش داشته باشند ← (امت‌تر شدن یادگیری)

پادگیری تقویتی حبتنی بر مدل

MODEL-BASED RL

- ✗ دارای مدل انتقال وضعيت محیط
- ✗ امکان تفسیر پاداش
- ✗ کمک به تصمیم برای (فتار فعلی)

✗ مدل می‌تواند در ابتدا نامعلوم یا معلوم باشد

- شطرنج (مدل معلوم)

✗ در محیط نیمه مشاهده پذیر، از مدل انتقال می‌توان برای تخمین وضعيت (فعلی) هم استفاده کرد

یادگیری تقویتی بدون مدل

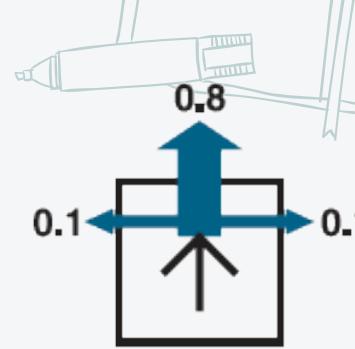
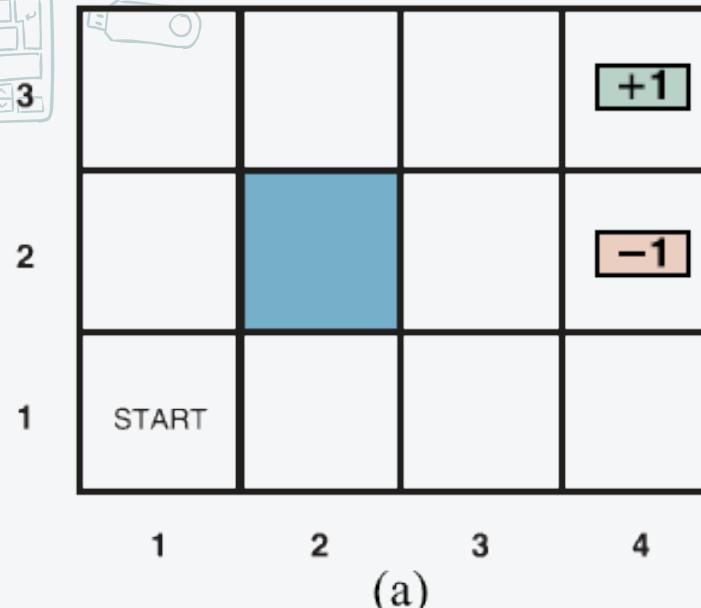
MODEL-FREE RL

- ✖ بدون مدل، و مستقیم‌تر عمل را تعیین می‌کند.
 - یادگیری تابع عمل-سودمندی (action-utility)
 - Q-learning
 - Q-function
- در یک موقعیت خاص
 - جستجوی خط مشی (Policy search)
 - به جای یادگیری سود حاصل از اعمال (که با کمک آنها بتوان عمل مناسب را انتخاب کرد)، مستقیماً برای هر وضعیت، عمل مناسب (ا) یاد می‌گیرد:
 - در این موقعیت، این کار را بگن
 - چرا؟ ارتش چرا ندارد!

پادگیری تقویتی انفعالی

PASSIVE RL

- با یک خطا مشی ثابت (s) اعمال تعیین می‌شوند
- هدف: پادگیری سودمندی (s)
 - امیدریاضی جمع پاداش اگر از وضعیت S با خطا مشی π مرگت کنیم.
- عملای ارزیابی خطا مشی است
 - بدون داشتن مدل انتقال وضعیت



احتمال نتیجه مطلوب با
[Up; Up; Right; Right; Right] X

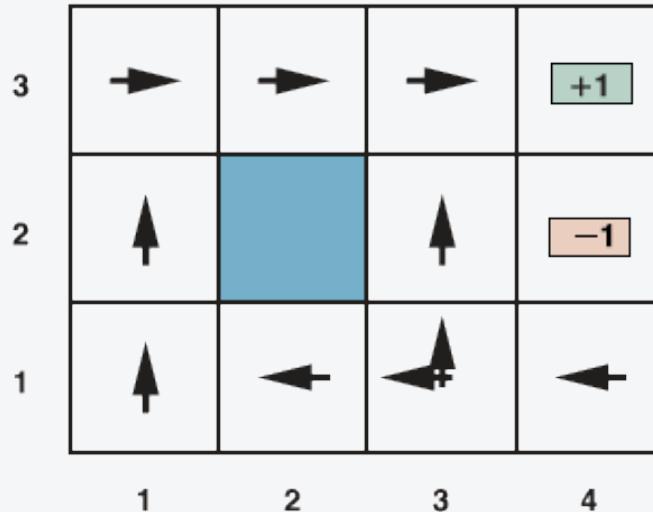
بات متمرک X
احتمال حرکت اشتباه X
برخورد با دیوار/مانع = ماندن در جای خود X

فرض مارکف: احتمال رسیدن به وضعیت بعدی صرفاً تابع وضعیت فعلی
است، نه وضعیت‌های گذشته X

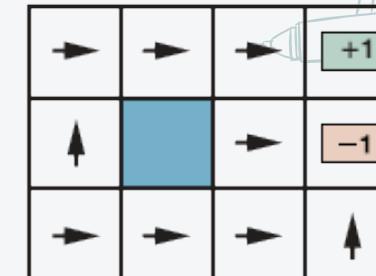
هر حرکت (جز حرکت به وضعیت هدف) پاداش 0.04 - دارد (جریمه) 
حرکت به هدف، پاداش 1+ یا 1 - دارد. 

تعريف MDP: 

مساله تصمیمگیری متوالی (sequential)، در یک محیط کاملا مشاهده پذیر و تصادفی، با داشتن مدل انتقال مارکف و اعمال پاداش 

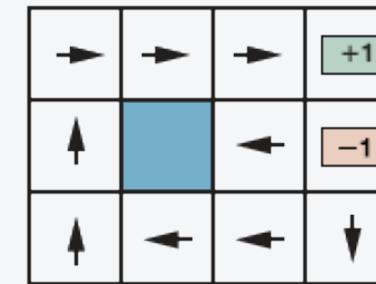


(a)

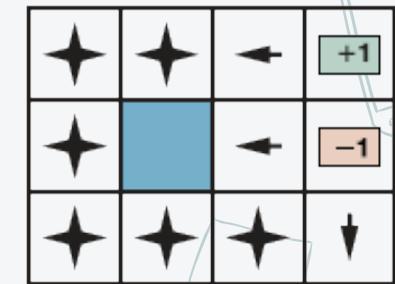


$$r < -1.6497$$

$$-0.7311 < r < -0.4526$$



$$-0.0274 < r < 0$$

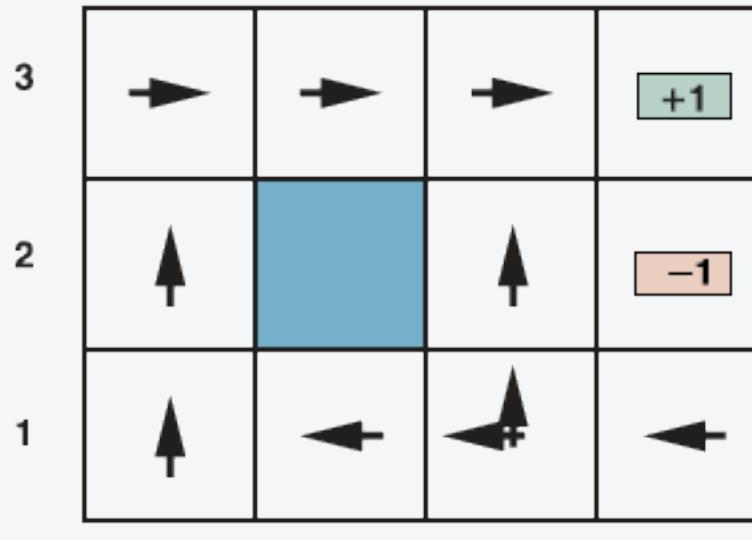
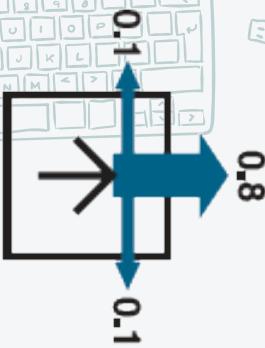


$$r > 0$$

(b)

هزینه انتقال و ضعیت
جهت ها : فط مشی بهینه

یادگیری سودمندی $U^\pi(s)$ در RL ازفعالی X



(a)

3	0.8516	0.9078	0.9578
2	0.8016		0.7003
1	0.7453	0.6953	0.6514

1 2 3 4

(b)



یادگیری سودمندی (s) $U^\pi(s)$ در RL انفعالی X

- ✖ پند اجرا (trial) داریم (حرکت از نقطه 1,1 تا سیدن به هدف)
- با یک خط مشی ثابت و مشخص (مثلث اسلاید قبل)
- ✖ وضعیت جاری، و پاداش (سیدن به وضعیت جاری حس می‌شود).
- ✖ مثلا

$$\begin{array}{ccccccccc}
 (1, 1) & \xrightarrow[\text{Up}]{-.04} & (1, 2) & \xrightarrow[\text{Up}]{-.04} & (1, 3) & \xrightarrow[\text{Right}]{-.04} & (1, 2) & \xrightarrow[\text{Up}]{-.04} & (1, 3) & \xrightarrow[\text{Right}]{-.04} & (2, 3) & \xrightarrow[\text{Right}]{-.04} & (3, 3) & \xrightarrow[\text{Right}]{-.04} & (4, 3) \\
 (1, 1) & \xrightarrow[\text{Up}]{-.04} & (1, 2) & \xrightarrow[\text{Up}]{-.04} & (1, 3) & \xrightarrow[\text{Right}]{-.04} & (2, 3) & \xrightarrow[\text{Right}]{-.04} & (3, 3) & \xrightarrow[\text{Right}]{-.04} & (3, 2) & \xrightarrow[\text{Up}]{-.04} & (3, 3) & \xrightarrow[\text{Right}]{-.04} & (4, 3) \\
 (1, 1) & \xrightarrow[\text{Up}]{-.04} & (1, 2) & \xrightarrow[\text{Up}]{-.04} & (1, 3) & \xrightarrow[\text{Right}]{-.04} & (2, 3) & \xrightarrow[\text{Right}]{-.04} & (3, 3) & \xrightarrow[\text{Right}]{-.04} & (3, 2) & \xrightarrow[\text{Up}]{-.04} & (4, 2) & \xrightarrow[\text{Up}]{-1} & (4, 3)
 \end{array}$$

نحویں سو دھنڈی:

✗ هدف یافتن مجموع با شروع از وضعیت S است

$$U^\pi(s) = E \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(S_t), S_{t+1}) \right]$$

✗ γ ضریب جریمه (دیرکرد) است (فعلاً)

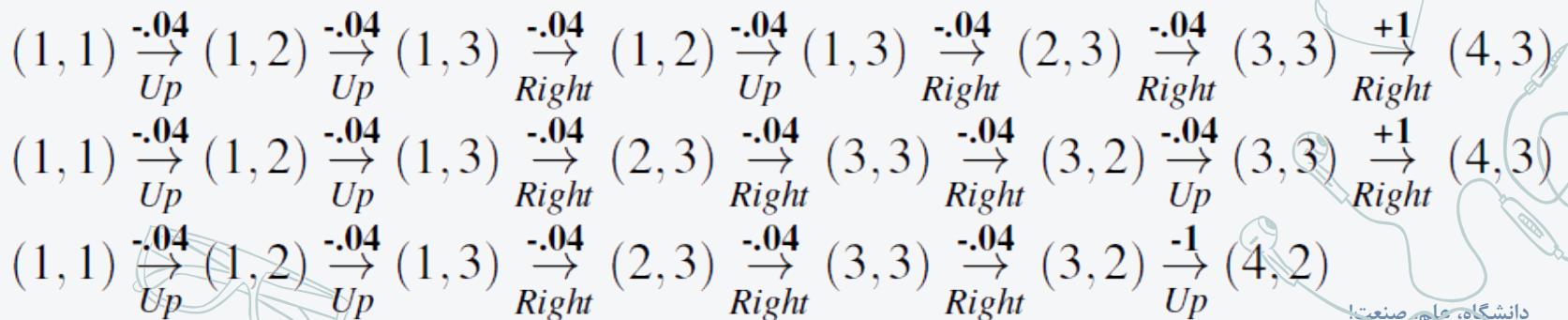
تغییر سودمندی با شیوه مستقیم

✗ پند بار اجرا (از شروع تا پایان)

✗ هر اجرا = یک یا پند نمونه برای هر یک از وضعیت‌های مسیر
 هر نمونه = سودمندی از اینجا به بعد (reward-to-go) اگر از وضعیت S شروع کنیم.

✗ سرشماری و میانگین‌گیری در کل نمونه‌ها

✗ تعداد اجرای بیشتر ← همگرایی به امید ریاضی مذکور



با شیوه مستقیم، RL تبدیل شده است به یادگیری با نظارت (SL) ○ نمونه ها (وضعیت‌ها) و برچسب (جمع سودمندی)

خوب یا بد؟ مساله این است! ✗

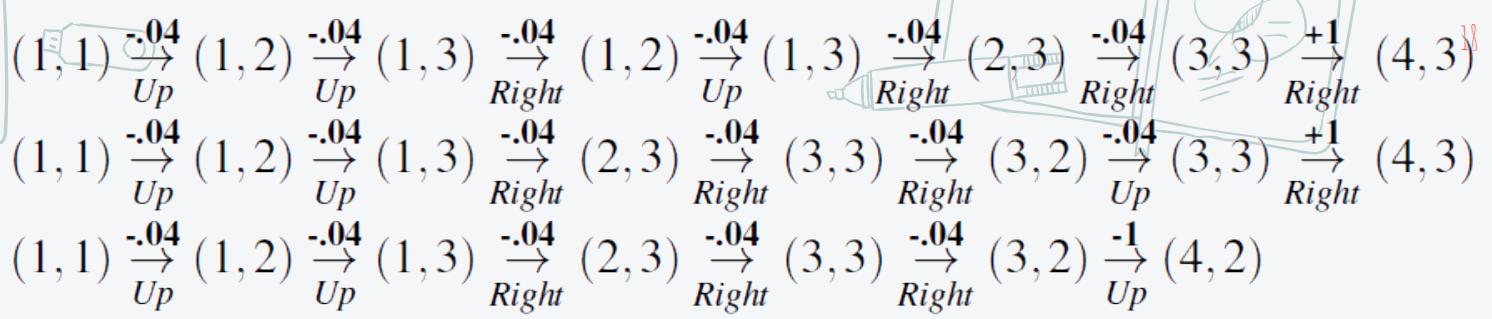
(وش‌های زیاد و متنوع برای یادگیری این تابع سودمندی (در SL) ✗

اما: ارتباط بین وضعیت‌ها کنار گذاشته شده ✗

سود وضعیت، حاصل از پاداش (فتن) به وضعیت بعدی و سود وضعیت بعدی
بود، این اطلاعات سودمندند و باعث یادگیری بهتر ○

$$U_i(s) = \sum_{s'} P(s' | s, \pi_i(s)) [R(s, \pi_i(s), s') + \gamma U_i(s')].$$

هتللا ... بریم اسلاید بعدی جا باشه ... ✗



	→	→	→	+1
↑			↑	-1
↑	←	↖	←	
1	2	3	4	

(a)

3	0.8516	0.9078	0.9578	+1
2	0.8016		0.7003	-1
1	0.7453	0.6953	0.6514	0.4279

(b)

مثالاً در اجراهای فوق

خانه (3,2) در اجرای دوه برای اولین بار ملاقات می‌شود
■ تجربه ای درباره آن نداریم که SL چیز قابل اعتنایی

بتواند بگوید

اما خانه مجاور به وی، (3,3) است که در اجرای اول مشاهده شده و سود زیادی دارد
پس: (3,2) جای خوبی است، زیرا همسایه خوبی دارد، اما شیوه مستقیم محاسبه سودمندی، تا وقتی به انتهای اجرا نرسد، خوبی آن را درنخواهد یافت.

ضمناً شیوه مستقیم، کند همگرا می‌شود.

ضمناً (!) خیلی از وضعیت‌ها کم تکرار و فضای بزرگتر از آن است که شیوه مستقیم بتواند آن را یاد بگیرد.

قبل از اداصه:

)

هر MDP از مجموعه‌های زیر تشکیل شده است:

S : وضعیت‌ها (و وضعیت شروع)

A : اعمال ممکن در هر وضعیت

R : پاداش هر عمل در هر وضعیت

P : مدل انتقال (فتن از هر وضعیت با هر عمل به وضعیت دیگر (احتمال))

(وش‌های حل MDP عمدتاً از جنس برنامه‌ریزی پویا هستند.

dynamic programming

(Divide and Conquer) ■

هل MDP ○

یافتن مقدار (سودمندی) در هر وضعیت ■

...)

BELLMAN معادله

X معادله Bellman برای بیان سودمندی هر وضعيت:

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

تقسیم و غلبه



.....)

الگوریتم برای حل MDP

- × الگوریتمی تکراری
- × مقداردهی اولیه برای سودمندی وضعیت ها
- × همگرایی مقادیر در چرخه های الگوریتم (محض مولا)
- × استفاده از تابع زیر برای محاسبه سود هر وضعیت-عمل

function Q-VALUE(*mdp*, *s*, *a*, *U*) **returns** a utility value
return $\sum_{s'} P(s' | s, a) [R(s, a, s') + \gamma U[s']]$

function Q-VALUE(mdp, s, a, U) **returns** a utility value
return $\sum_{s'} P(s' | s, a) [R(s, a, s') + \gamma U[s']]$

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function

inputs: mdp , an MDP with states S , actions $A(s)$, transition model $P(s' | s, a)$
 rewards $R(s, a, s')$, discount γ

ϵ , the maximum error allowed in the utility of any state

local variables: U, U' , vectors of utilities for states in S , initially zero
 δ , the maximum relative change in the utility of any state

repeat

$U \leftarrow U'; \delta \leftarrow 0$

for each state s **in** S **do**

$U'[s] \leftarrow \max_{a \in A(s)} \text{Q-VALUE}(mdp, s, a, U)$
if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$

until $\delta \leq \epsilon(1 - \gamma)/\gamma$

return U

.....)

در هر چرخه عمل "Bellman اعمال می‌شود:

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

یادآوری: معادله بلمن:

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

در دوستی، مقدار بهینه U ها باید یک ضرب پیدا شود.

در اولی، در هر لحظه یه مقدار فعلی برای U ها وجود دارد و به صورت تدریجی بهبود پیدا می‌کند.

.....)

الگوریتم برای حل MDP

خط مشی را به ووز می‌کند X

الگوریتمی تکراری X

دو گام در هر تکرار: X

ارزیابی خط مشی ○

محاسبه سودمندی وضعیت‌ها در یک خط مشی ثابت ■

بهبود خط مشی ○

محاسبه خط مشی جدید با یک قدم نگاه به جلو (باداشتن سودمندی هر وضعیت) ■

function POLICY-ITERATION(*mdp*) **returns** a policy

inputs: *mdp*, an MDP with states S , actions $A(s)$, transition model $P(s' | s, a)$

local variables: U , a vector of utilities for states in S , initially zero

π , a policy vector indexed by state, initially random

repeat

$U \leftarrow \text{POLICY-EVALUATION}(\pi, U, mdp)$

unchanged? \leftarrow true

for each state s **in** S **do**

$a^* \leftarrow \underset{a \in A(s)}{\text{argmax}} \text{ Q-VALUE}(mdp, s, a, U)$

if $\text{Q-VALUE}(mdp, s, a^*, U) > \text{Q-VALUE}(mdp, s, \pi[s], U)$ **then**
 $\pi[s] \leftarrow a^*$; *unchanged?* \leftarrow false

until *unchanged?*

return π

.....)

ارزیابی خط حشنه در POLICY ITERATION

- ✗ تابع Policy-Evaluation چگونه محاسبه شود؟
- ✗ الگوریتم Bellman معادله Value Iterartion را حل می‌کند
 - بیشینه‌گیری از اعمال ممکن در هر وضعیت
 - در اینجا خط مشی ثابت (فقط یک عمل ممکن است) و کار ساده‌تر است
 - نسخه ساده شده (وابط Bellman) :

$$U_i(s) = \sum_{s'} P(s' | s, \pi_i(s)) [R(s, \pi_i(s), s') + \gamma U_i(s')].$$

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

به جای نسخه اصلی:

n وضعیت = n متغیر = n معادله سودمندی

هر یک = ترکیب خطی متغیرهای همسایه

با (وشاهی خطی در زمان n^3 قابل حل است (زمان زیاد)



MODIFIED POLICY ITERATION

به جای حل دقیق ارزیابی فقط مشی طبق روابط فقط
تقریب ارزیابی با نسخه ساده شده «ب» (ووزرسانی «Bellman»)

$$U_{i+1}(s) \leftarrow \sum_{s'} P(s' | s, \pi_i(s)) [R(s, \pi_i(s), s') + \gamma U_i(s')],$$

ساده = فقط مشی ثابت بدون بیشینه گیری
نسخه اصلی (وابط به (ووزرسانی «Bellman»)

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

Simplified Value Iteration

Modified Policy Iteration

: به جای حل دقیق با معادلات فقط، از روش تقریبی simplified value iteration استفاده می‌کند

ادامه

✗ تخدمین سودمندی با شیوه مساقیم

- یادگیری با نظارت
- در نظر نگرفتن (وابط بین وضعيت‌ها

✗ خطا مشی ثابت

- در ادامه ...

نخستین سودمندی با برنامه ریزی پویای تطبیق پاافته ADAPTIVE DYNAMIC PROGRAMMING

خط مشی ثابت 

یادگیری مدل انتقال (وقتی مدل انتقال را نداریم) 

حل MDP با برنامه ریزی پویا 

$$U_i(s) = \sum_{s'} P(s' | s, \pi_i(s)) [R(s, \pi_i(s), s') + \gamma U_i(s')].$$

حل معادلات فطی 

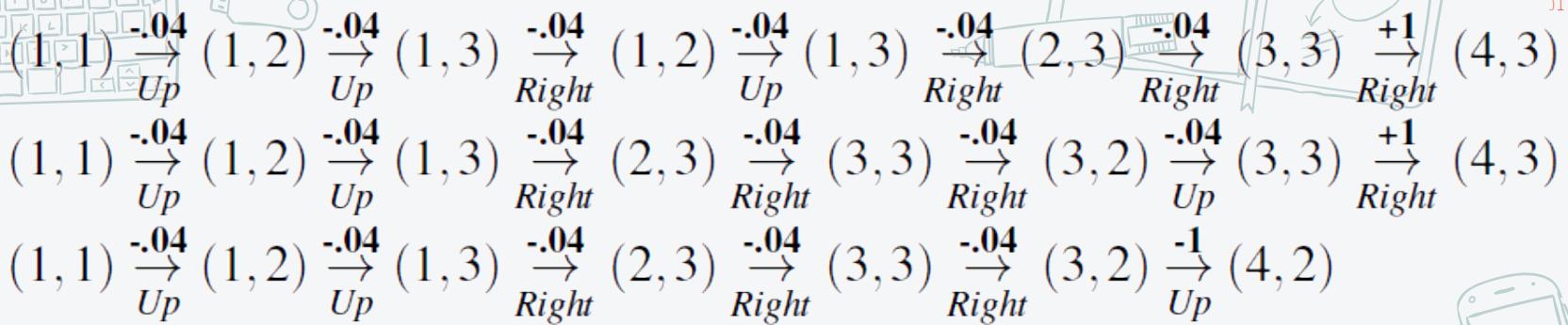
ب استفاده از Simplified Value Iteration 

- یادگیری مدل انتقال تدریجی است
- بعد از هر تغییر، تفمین سودمندی (با value iteration) به وز می‌شود

مقدار اولیه سودمندی به جای صفر، مقادیر پرفه قبلی
همگرایی سریع تر value iteration

- یادگیری مدل انتقال بانظارت است

۹۰ درصدی: ...
خروجی: ...



سرشماری و درصد وقوع انتقال در trial ها
برای یادگیری مدل انتقال

متلا وضعيت (3,3) و عمل (3,3) مثلا وضعيت

function PASSIVE-ADP-LEARNER(*percept*) returns an action

inputs: *percept*, a percept indicating the current state s' and reward signal r

persistent: π , a fixed policy

mdp, an MDP with model P , rewards R , actions A , discount γ

U , a table of utilities for states, initially empty

$N_{s'|s,a}$, a table of outcome count vectors indexed by state and action, initially zero

s, a , the previous state and action, initially null

if s' is new **then** $U[s'] \leftarrow 0$

if s is not null **then**

increment $N_{s'|s,a}[s, a][s']$

$R[s, a, s'] \leftarrow r$

add a to $A[s]$

$P(\cdot | s, a) \leftarrow \text{NORMALIZE}(N_{s'|s,a}[s, a])$

$U \leftarrow \text{POLICY EVALUATION}(\pi, U, mdp)$

$s, a \leftarrow s', \pi[s']$

return a

به دو زمانی دو هر چهار

چرا تا آفر چرفه / trial صبر نکنیم

امکان همراهی و همگرایی ساده تر، وقتی از

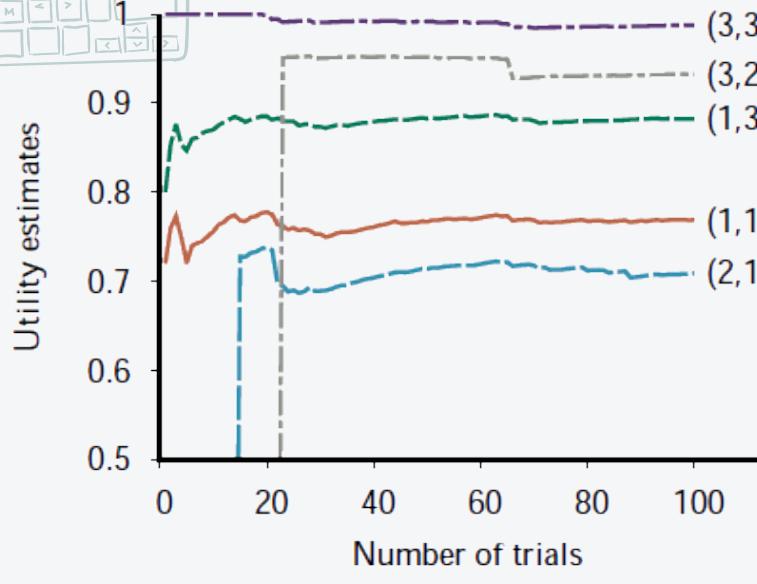
مقادیر سودمندی مرحله قبل به عنوان مقدار

اولیه سودمندی در چرفه بعد استفاده شود.

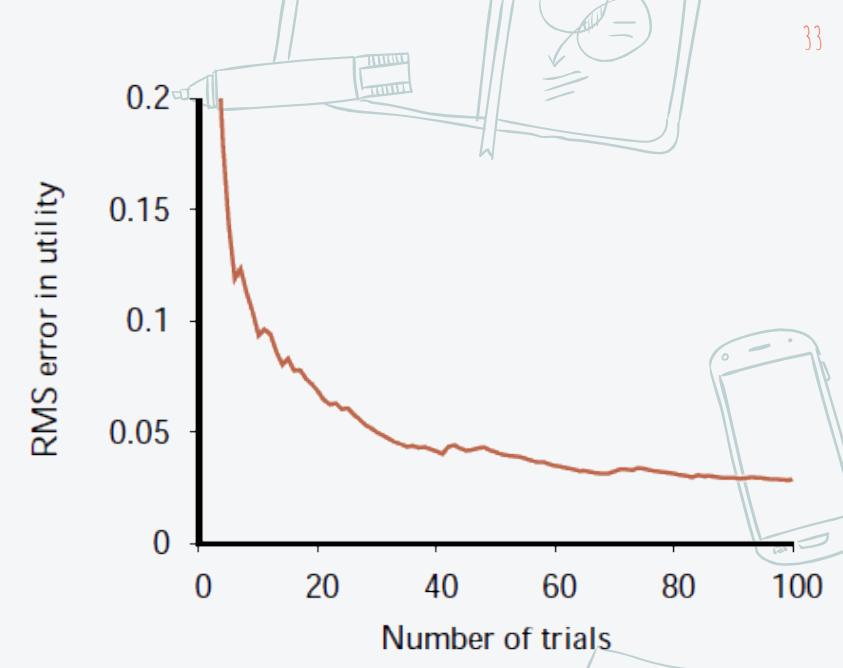
ازیابی خط میشی با

Simplified Value Iteration

یا حل معادلات خطی



(a)



(b)

دانشگاه، علوم صنعتی
33

U(1,1) RMS (root-mean-square) : b
میانگین در چندین اجرای مختلف

یادگیری براساس اختلاف حوقت

TEMPORAL-DIFFERENCE

ADP در محیط‌های بزرگ، ممکن است غیر ممکن باشد X

- سختی یادگیری تابع انتقال

TD حل تناقض بین وضعيت‌های همسایه در X

- فرضی فقط به یک همسایه برویم همیشه

به جای X

$$U_i(s) = \sum_{s'} P(s' | s, \pi_i(s)) [R(s, \pi_i(s), s') + \gamma U_i(s')].$$

داریم: X

$$U^\pi(s) \leftarrow U^\pi(s) + \alpha [R(s, \pi(s), s') + \gamma U^\pi(s') - U^\pi(s)].$$

function PASSIVE-TD-LEARNER(*percept*) **returns** an action

inputs: *percept*, a percept indicating the current state s' and reward signal r

persistent: π , a fixed policy

s , the previous state, initially null

U , a table of utilities for states, initially empty

N_s , a table of frequencies for states, initially zero

if s' is new **then** $U[s'] \leftarrow 0$

if s is not null **then**

increment $N_s[s]$

$U[s] \leftarrow U[s] + \alpha(N_s[s]) \times (r + \gamma U[s'] - U[s])$

$s \leftarrow s'$

return $\pi[s']$

بادگیری تقویتی فعال (ACTIVE REINFORCEMENT LEARNING)

در هر مرحله تصمیم می‌گیرد که عملی انجام شود

مثلا با استفاده از الگوریتم ADP می‌توان مراحل زیر را دنبال کرد:

- به روز رسانی مدل انتقال بعد از هر عمل و ارزیابی مجدد فط مشی (به روز رسانی ارزیابی قبلی)
- بادگیری مدل انتقال کامل برای همه اعمال ممکن در هر وضعیت
- انتخاب عمل بهینه:

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

یا از policy iteration یا value iteration استفاده کرد:

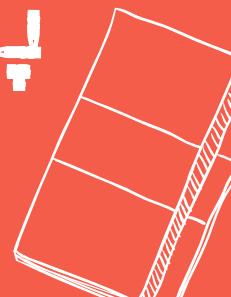
اولی: فقط مقدار سودمندی‌ها به روز شود و عمل بعدی **حریصانه** انتخاب شود

دومی: فقط مشی بهینه هم بعد از به روز کردن سودمندی‌ها به روز شود و عمل

بعدی طبق **فط مشی بهینه** شده به سادگی انتخاب شود.

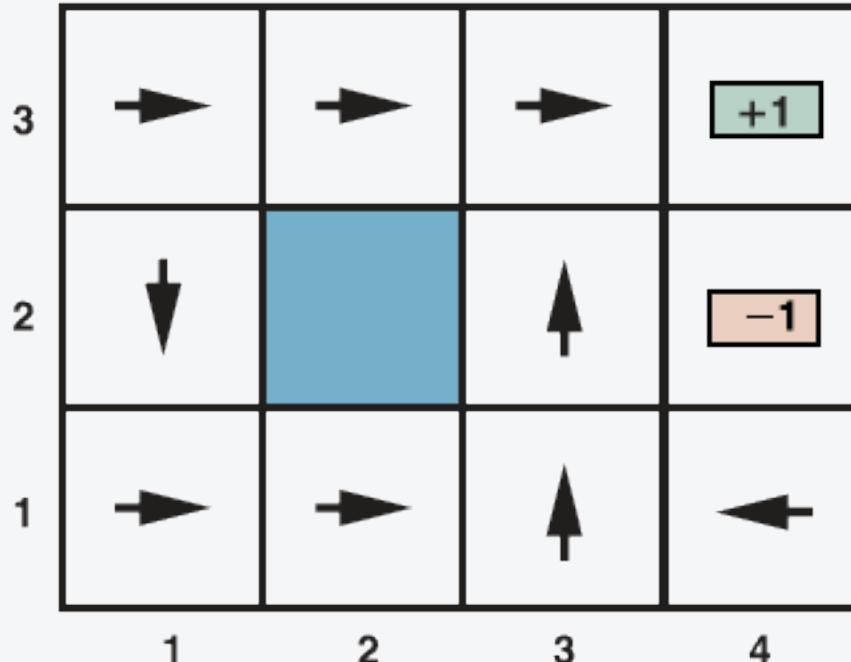
خطا هایی بهینه

پیروی با سریع‌تر؟ حساله این است!





حاله اکتشاف - EXPLORATION



حریمانه بعد از اکتشافات کافی

حریمانه بودن X
بیشترین سود ○
یا اکتشاف X

اعتماد به گذشته X
یا امید به آینده? X



چگونه مکتشفا شویم؟ (در ۲۰ دقیقه! ☺)

انجام عمل تصادفی به جای عمل بھینه، با احتمالی متناسب با محدودس زمان X

همگرا می شود

ممکن است کند باشد

اختصاص وزن بیشتر به اعمال تجربه نشده (متناسب با میزان جدیدیتش!) X

در عین حال، اگر تجربه های اندک منفی بودند، پرهیز کنیم.

عاقل از یک سوراخ دوبار گزیده نمی شود. Agent

تابع سودمندی اکتسافی: X

استفاده در یک عامل مبتنی بر ADP **فعال** value iteration

$$U^+(s) \leftarrow \max_a f \left(\sum_{s'} P(s' | s, a) [R(s, a, s') + \gamma U^+(s')], N(s, a) \right)$$

بیشترین پاداش ممکن: R^+

$$f(u, n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise,} \end{cases}$$

TEMPORAL-DIFFERENCE Q-LEARNING

- ✖ نسخه ADP فعال به صورت افتلاف زمانی (TD)
- ✖ پرهیز از داشتن مدل انتقال
 - په باید گردی
 - یادگیری تابع سودمندی-عمل به جای تابع سودمندی
 - چرا؟

$$Q(s, a) = \sum_{s'} P(s' | s, a) [R(s, a, s') + \gamma \max_{a'} Q(s', a')]$$

$$Q(s, a) \leftarrow Q(s, a) + \alpha [R(s, a, s') + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

عدم نیاز به مدل = سادگی = امکان پذیری در محیط های پیچیده

همچنین: نداشتن ابزار برای نگاه به آینده

- نمیشود چند عمل جلوت را پیش بینی کرد

function Q-LEARNING-AGENT(*percept*) **returns** an action

inputs: *percept*, a percept indicating the current state s' and reward signal r

persistent: Q , a table of action values indexed by state and action, initially zero

N_{sa} , a table of frequencies for state-action pairs, initially zero

s, a , the previous state and action, initially null

if s is not null **then**

increment $N_{sa}[s, a]$

$$Q[s, a] \leftarrow Q[s, a] + \alpha(N_{sa}[s, a])(r + \gamma \max_{a'} Q[s', a'] - Q[s, a])$$

$s, a \leftarrow s', \arg\max_{a'} f(Q[s', a'], N_{sa}[s', a'])$

return a

$$f(u, n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise,} \end{cases}$$



یادگیری مان تقویت شد

