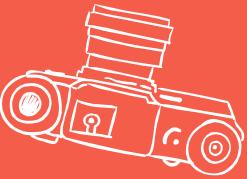
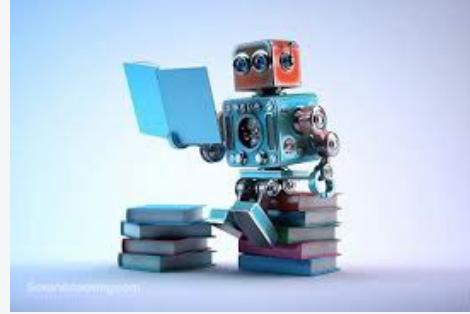


بە نام خەل





پادگاری ماشین

آرش عبدی هجراندوست

arash.abdi.hejrandoost@gmail.com

دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

نیم سال اول ۱۴۰۲-۱۴۰۳



پادگیری تقویتی

Reinforcement Learning- RL

✖ پادگیری با نظارت: مجموعه‌ای از داده‌های برچسب فورده

- جفت‌های

- موقعیت - تصمیم

- حسگر - عمل

- وودی - فروجی

- آموزش سنتی

- دستورالعملی : هرگاه، ... آنگاه ...

- هرچند دارای تعمیم به آموزش‌های داده نشده است

✖ پادگیری تقویتی:

- پادگیری با پاداش/جریمه

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

چند مثال

بازی شطرنج

نگاه نظاری:

موقعیت ← مرکز

پایگاه دادهای از جفت‌های فوق از مرگات فرد بزنده در بازی‌های های اساتید

بزرگ شطرنج

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

موقعیت جدید؟ یه کاری می‌کنیه ولی تفسیری از آن نداریم!

فوتبال (باتی)

آموزش به حیوانات (سیرک و ...)

مهارت‌های ورزشی (فیلیپینی 😊)

مهارت‌های اجتماعی

تربيت

آموزش

زندگی

✖ یادگیری تقویتی اما:

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

✖ شباهت با (MDP) Markov decision process

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

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

چند نکته درباره پاداش

- 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 خیلی 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 شبکه بر مدل

Model-based RL

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

✖ 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 (فیلترینگ)

یادگیری تقویتی بدون مدل

Model-free RL

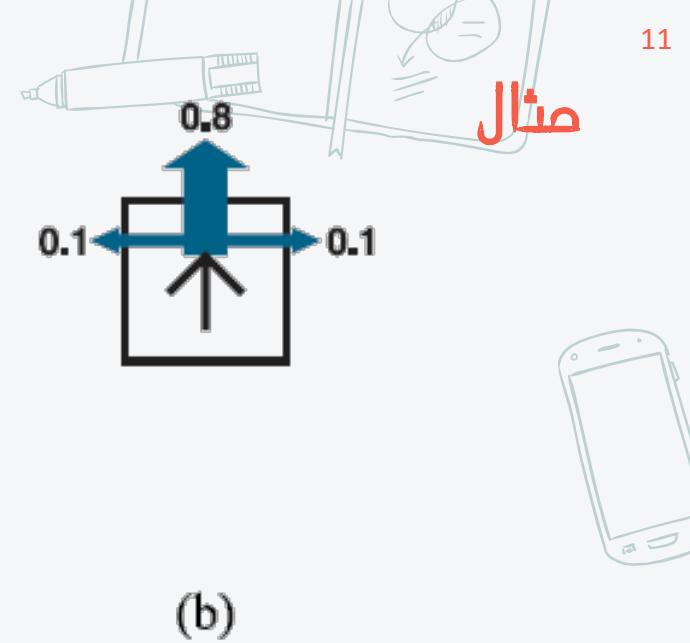
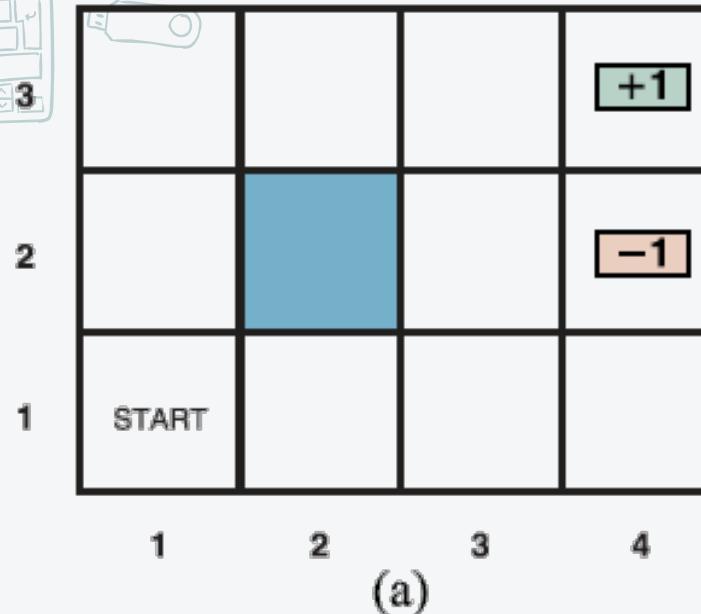
- بدون مدل، و مستقیم‌تر عمل را تعیین می‌کند.**
- **یادگیری تابع عمل-سودمندی (action-utiliity)**
- **Q-learning**
- **Q-function : مجموع سود حاصل (تا انتهای) از انجام یک عمل خاص در یک موقعیت خاص**
- **جستجوی خط مشی (Policy search)**
- **به جای یادگیری سود حاصل از اعمال (که با کمک آنها بتوان عمل مناسب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

پادگیری تقویتی انفعالی

Passive RL

- ✖ با یک خطا مشی ثابت (s) π اعمال تعیین می‌شوند
- ✖ هدف: پادگیری سودمندی (s) $U^\pi(s)$
- امیدریاضی جمع پاداش اگر از وضعیت s با خطا مشی π حرکت کنیم.
- ✖ عملاً ارزیابی خطا مشی است
- بدون داشتن مدل انتقال وضعیت

مثال



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

برخورد با دیوار/مانع = ماندن در جای خود

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

بات متمرک

احتمال حرکت اشتباه

✗

✗

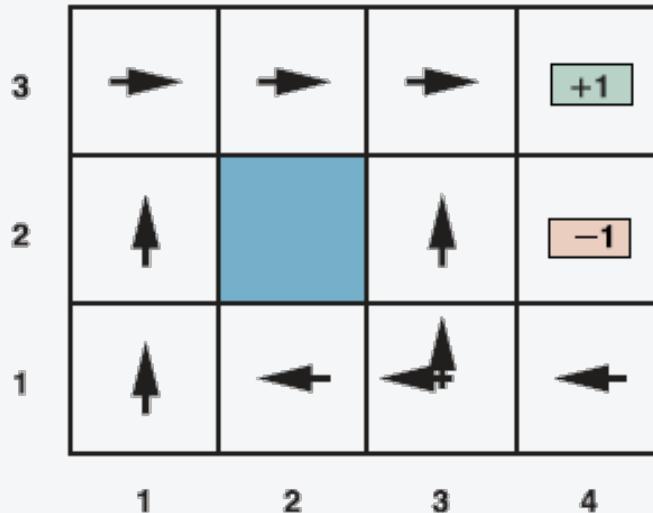
✗

✗

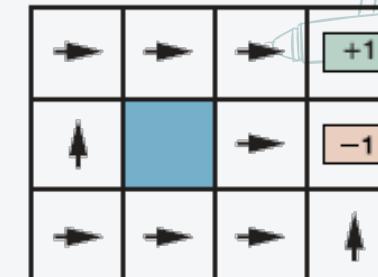
هر مرکت (جز مرکت به وضعیت هدف) پاداش 0.04- دارد (جریمه) X
مرکت به هدف، پاداش 1+ یا 1- دارد. X

تعريف MDP

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

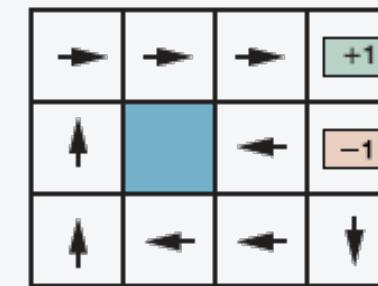


(a)

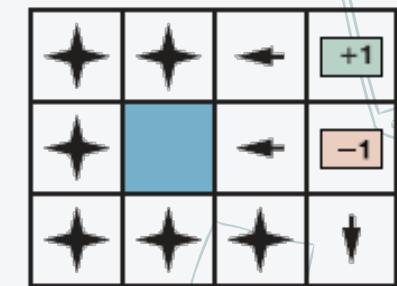


$$r < -1.6497$$

$$-0.7311 < r < -0.4526$$



$$-0.0274 < r < 0$$

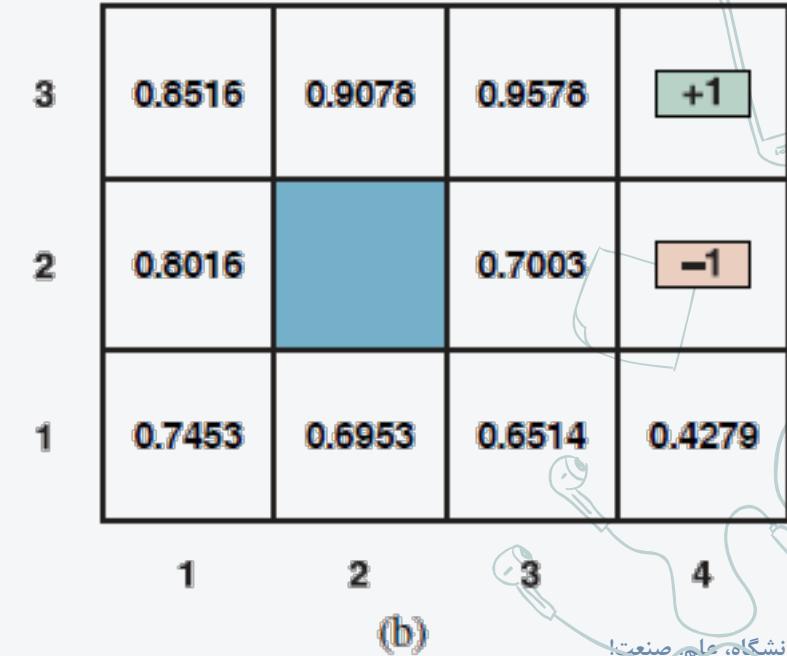
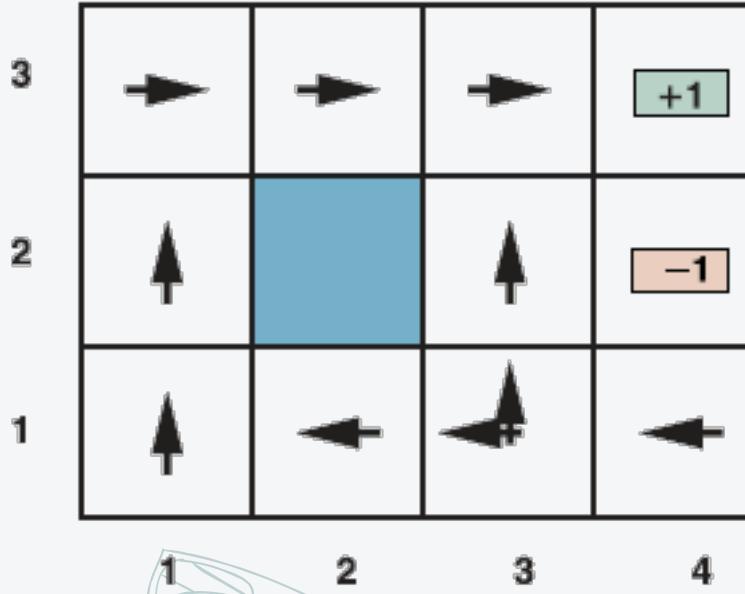
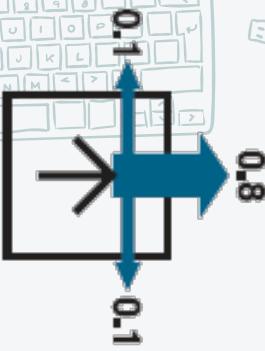


$$r > 0$$

(b)

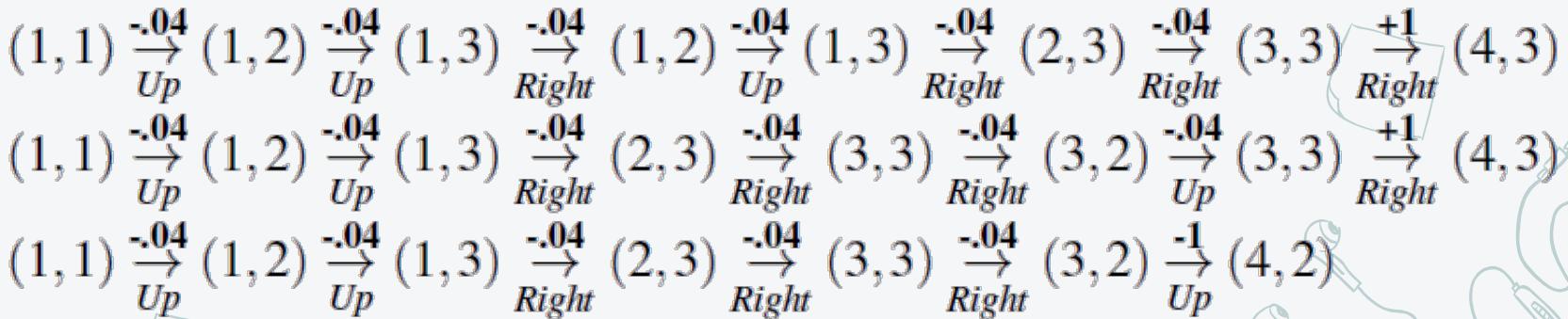
هزینه انتقال و ضعیت
جهت ها : فط مشی بهینه

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



پادگیری سودمندی (s) $U^\pi(s)$ در RL انتقالی

- چند اجرا (trial) داریم (مرکت از نقطه ۱,۱ تا (سیدن به هدف با یک خط مشی ثابت و مشخص (مثلًا اسلاید قبل) وضعيت جاری، و پاداش (سیدن به وضعيت جاری می‌شود.**
- مثلًا** 



تخصیص سودمندی

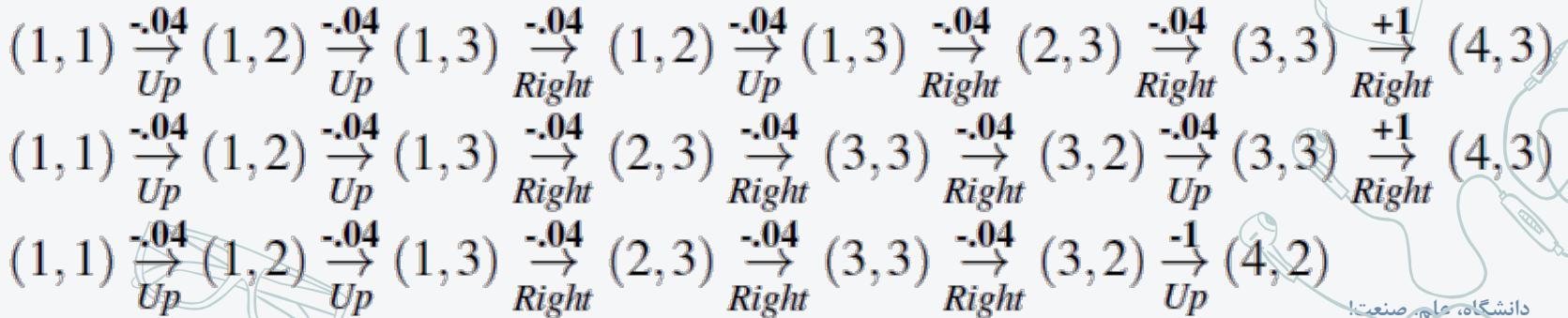
✗ هدف یافتن مجموع با شروع از وضعیت s است

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

✗ γ ضریب جریمه (دیرکرد) است (فعلاً)

تاختین سودمندی با شیوه مسنتفیم

- ✖ چند بار اجرا (از شروع تا پایان)**
- ✖ هر اجرا = یک یا چند نمونه برای هر یک از وضعیت‌های مسیر**
 - هر نمونه = سودمندی از اینجا به بعد (reward-to-go) اگر از وضعیت 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 نظرات (SL) (SL) نمونه ها (وضعیت ها) و برچسب (جمع سودمندی)

خوب یا بد؟ مساله این است!

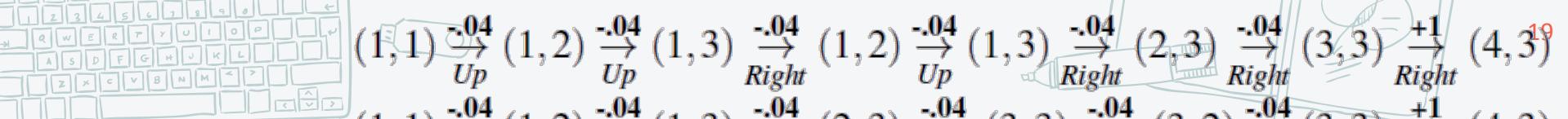
(وش های زیاد و متنوع برای یادگیری این تابع سودمندی (در SL)

اما: ارتباط بین وضعیت ها 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)

مثلاً در اجراهای فوق

- خانه (3,2) در اجرای دوم برای اولین بار ملاقات می‌شود
- تجربه ای درباره آن نداریم که SL پیز قابل اعتمادی بتواند بگوید

اما خانه همچنان (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 0)
- 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
- تقسیم و غلبه (Divide and Conquer)
- هل MDP ؟
- یافتن مقدار (سودمندی) در هر وضعیت

...)

معادله Bellman

X معادله Bellman برای بیان سودمندی هر وضعیت:

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

تقسیم و غلبه



.....)

الگوریتم Value Iteration برای حل MDP

- ✖ الگوریتمی تکراری
- ✖ مقداردهی اولیه برای سودمندی وضعيت ها
- ✖ همگرایی مقادیر در چرخه های الگوریتم (محضه اولا)
- ✖ استفاده از تابع زیر برای محاسبه سود هر وضعيت-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 ل ها باید یک ضرب پیدا شود.

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

.....)

الگوریتم Policy Iteration برای حل MDP

خط مشی را به ووز می‌کند X

الگوریتمی تکراری X

دو گام در هر تکرار X

ارزیابی خط مشی ○

محاسبه سودمندی وضعيت‌ها در یک خط مشی ثابت ■

بهبود خط مشی ○

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

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 Q-VALUE(*mdp*, s , a^* , U) $>$ Q-VALUE(*mdp*, s , $\pi[s]$, U) **then**

$\pi[s] \leftarrow a^*$; *unchanged?* \leftarrow false

until *unchanged?*

return π

.....)

ارزیابی خط متش در Policy Iteration

- ✖ تابع Policy-Evaluation پکونه محاسبه شود؟
- ✖ الگوریتم Value Iterartion ا هل می‌گند
 - بیشینه‌گیری اوی اعمال ممکن در هر وضعیت
- ✖ در اینجا خط متش ثابت (فقط یک عمل ممکن است) و کار ساده‌تر است
 - نسخه ساده شده (وابط Bellman :

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

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

n وضعیت = n متغیر = n معادله سودمندی

هریک = ترکیب خطی متغیرهای همسایه

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

.....)

Modified Policy Iteration

به جای حل دقیق ارزیابی خط مثبت طبق روابط خطی X
 «Bellman تقریب ارزیابی با نسخه ساده شده به روزرسانی O

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

ساده = خط مثبت ثابت بدون بیشینه گیری O
 نسخه اصلی روابط به روزرسانی O: Bellman

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

Value Iteration X

Modified Policy Iteration X

ادامه

✗ تخدمین سودمندی با شیوه مساقیم

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

✗ فطا مشی ثابت

- در ادامه ...






نحوین سودمندی با

برنامه ریزی پویای تطبیق پافته

Adaptive Dynamic Programming

خط مشی ثابت 

یادگیری مدل انتقال 

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 

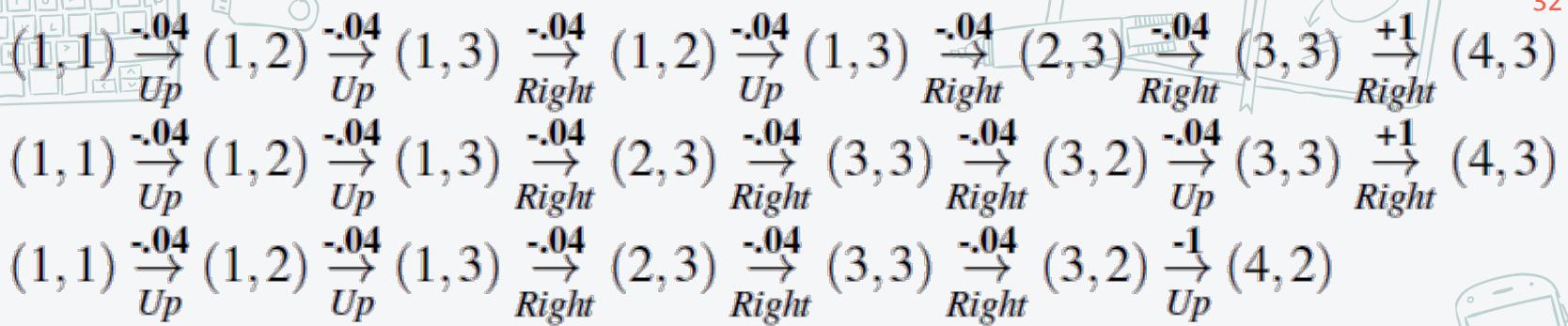
Modified Value Iteration از 

- ✖ یادگیری مدل انتقال تدریجی است**
- ✖ بعد از هر تغییر، تفمین سودمندی (با v) به (وز) می‌شود**

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

- ✖ یادگیری مدل انتقال یادگیری با نظرارت است**

- ۹۰ درصدی: ...
- فروجی: ...



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

متلا وضعيت (3,3) و عمل Right

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 $A[s]$

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

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

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

return a

به دو زمانی دو هر قده

چرا تا آفر چرفه/trial صبر نکنیم

امکان همراهی و همگرایی ساده تر، وقتی از

مقادیر سودمندی مرحله قبل به عنوان مقدار

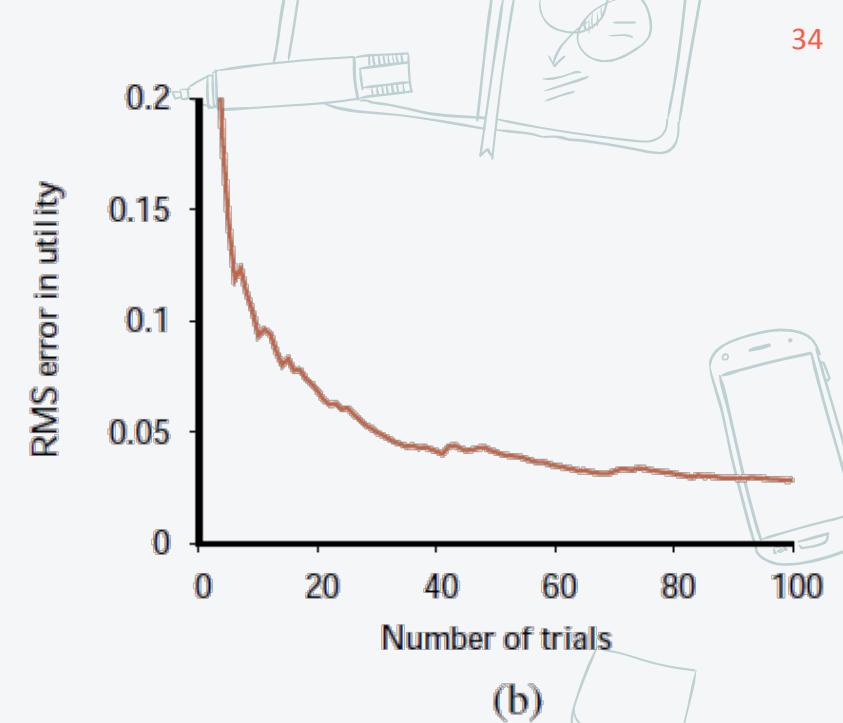
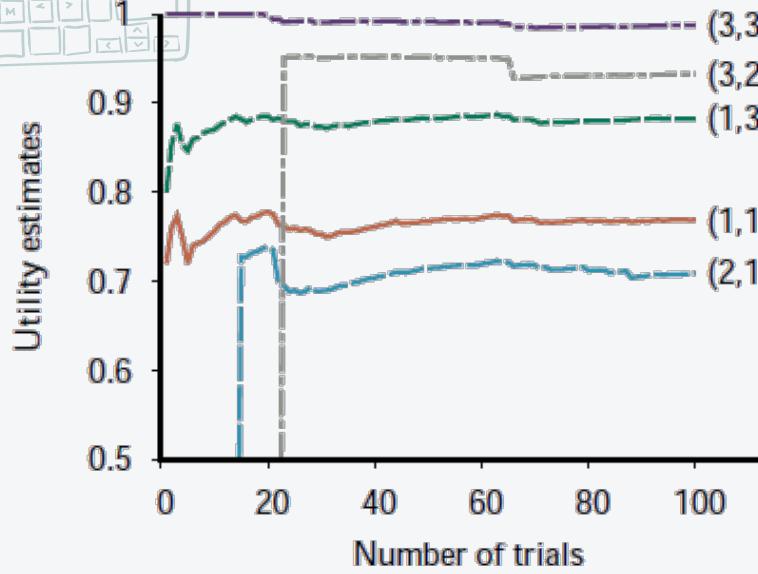
اولیه سودمندی در چرفه بعد استفاده شود.

ازیابی خط میشی با

Modified Value Iteration

یا حل معادلات خطی





U(1,1) root-mean-square : b X

یادگاری براساس اختلاف حوقت

Temporal-Difference

- ✗ حل تناقض بین وضعيت های همسایه
- فرضی فقط به یک همسایه برویم همیشه

✗ به جای

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

✗ داریم:

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

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 $\pi[s']$

پادگیری تقویتی فعال (Active Reinforcement Learning)

در هر مرحله تصمیم می‌گیرد که عملی انجام شود

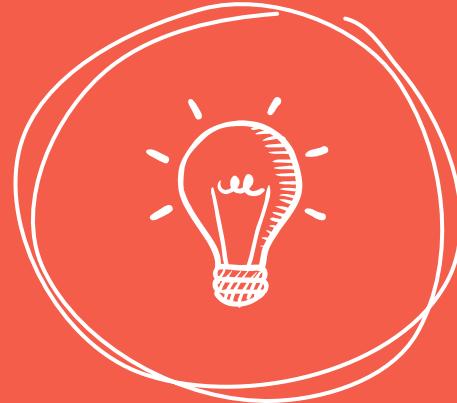
با استفاده از الگوریتم ADP:

- به روز رسانی مدل انتقال بعد از هر عمل و ارزیابی مجدد فقط مشی (به روز رسانی آن
- پادگیری مدل انتقال کامل برای همه اعمال ممکن در هر وضعیت
- انتخاب عمل بهینه:

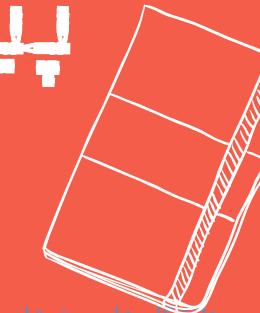
$$U(s) = \max_{a \in A(s)} \sum_{s'} P(s' | s, a) [R(s, a, s') + \gamma U(s')].$$

استفاده از policy iteration یا value iteration

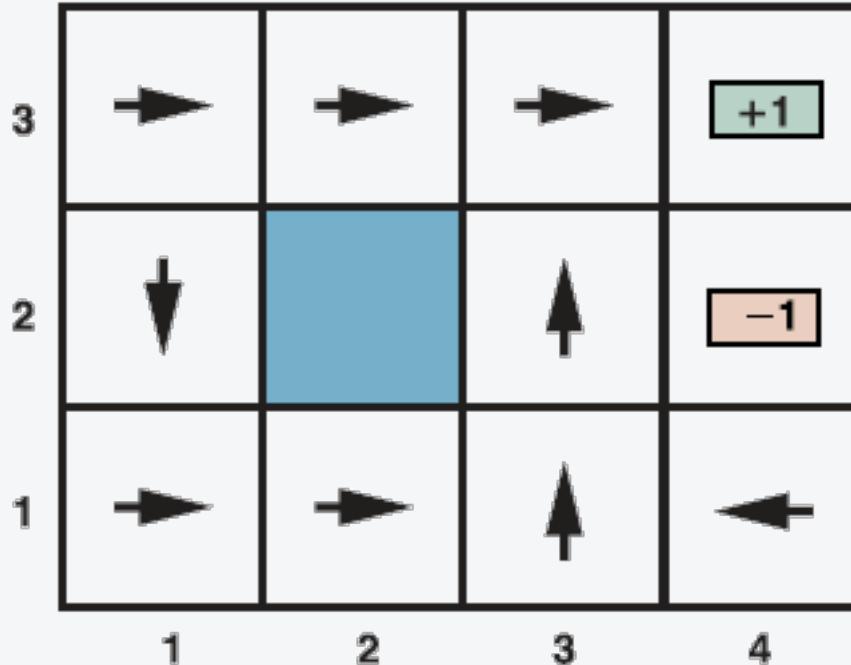
- اولی: فقط مقدار سودمندی‌ها به روز شود و عمل بعدی مریضانه انتخاب شود
- دومنی: فقط مشی بهینه هم بعد از به روز کردن سودمندی‌ها به روز شود و عمل بعدی طبق خطا مشی بهینه شده به سادگی انتخاب شود.



خطا هشیب به بینه
پیروی با سریجیج؟ حساله این است!



حاله اکتشاف - Exploration



حریمانه بعد از اکتشافات کافی

حریمانه بودن
بیشترین سود
یا اکتشاف

اعتماد به گذشته
یا امید به آینده؟



چگونه مکتشف شویم؟

(در ۰.۲ دقیقه! ☺)

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

همگرا نمی‌شود

ممکن است کند باشد

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

در عین حال، اگر تجربه‌های اندک منفی بودند، پرهیز کنیم.

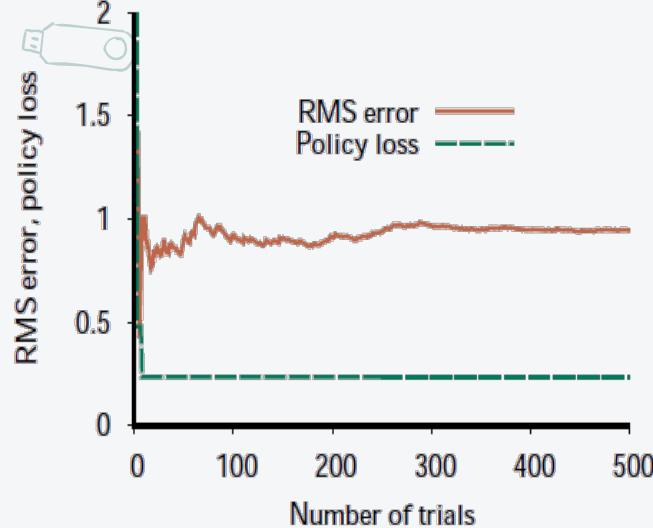
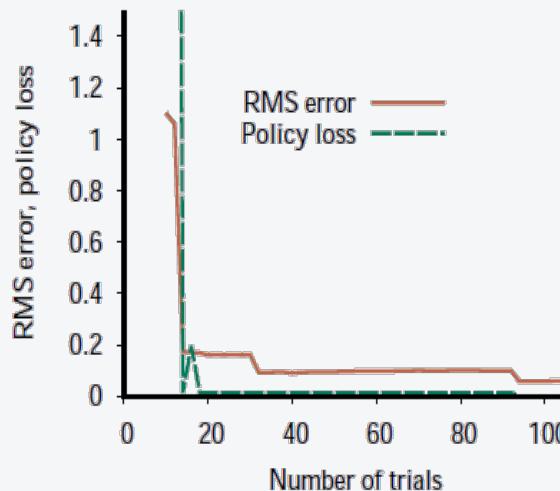
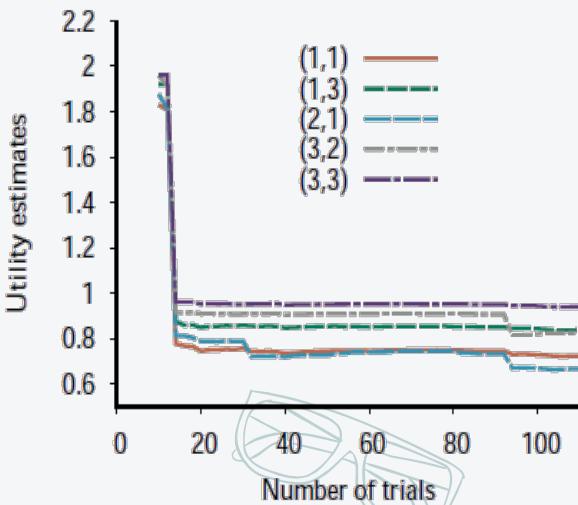
آدم عاقل از یک سوراخ دوبار گزیده نمی‌شود.

X تابع سودمندی اکتشافی:

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

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

R⁺: بیشترین پاداش ممکن



خطای همه اوضاع (!)

Policy Loss
خطای سود در وضعیت (1,1)

نمودار بالا: مریضانه

سرعت همگرایی
روش اکتسافی

اکتشاف اصن

اکتشاف منجر به مرگ یا هزینه بزر یا منجر به وضعیت‌های بی‌بازگشت

آیا ریسک اکتشاف پذیرفتی است؟

شطرنج؟

شبیه ساز (اندگی)؟

یا خود راندگی در محیط واقعی؟ (همچنان اعمال خطییر انجام بدهیم)

الگوریتم ADP بر اساس تفمین مدل انتقال وضعیت با بیشترین احتمال (likelihood) عمل می‌کند.

بهتر است، همه مدل‌های محتمل در نظر گرفته شوند

و عملی انتخاب شود که در همه آنها (وزن دار) سود بیشتری بدهد

عمل ایمن

اکتشاف اصن

با پادگاری تقویتی صبتنی برابر

h : فرضیه مدل درست X

$P(h|e)$ X

- محاسبه بر اساس قاعده بیز با داشتن مشاهدات (انتقال وضعيت‌ها)
- تبدیل اطلاعات تشخیصی به اطلاعات علی

انتساب خط مشی X

$$\pi^* = \operatorname{argmax}_\pi \sum_h P(h | e) U_h^\pi$$

$$\pi^* = \operatorname{argmax}_{\pi} \sum_h P(h | \mathbf{e}) U_h^\pi$$

چرا امن؟

مدل‌های مختلف در رابطه په هستند؟

فقط مشیی را دنبال کن که نه فقط بر اساس محتمل ترین مدل انتقال، بیشترین سود را دارد، بلکه بر اساس متوسط وزن دار همه مدل‌های محتمل بیشترین سود را دارد.

محباطانه

کمتر حریصانه (بر اساس محتمل ترین مدل انتخاب نکن، سایر مدلها هم محتملند)

اکتشاف امن

با تئوری کنترل پایدار

$$\pi^* = \operatorname{argmax}_{\pi} \min_h U_h^\pi.$$

تعدادی فرضیه قابل اعتنا 

صرف نظر از احتمال هر یک 

سیاستی را انتخاب کن که با بدترین مدل انتقال، بیشترین سود را بدهد. 

بدترین مدل : مدلی که با آن سیاست کمترین سود را دارد

سیاست های دیگر احتمالا سود بیشتری می دهند، اما ممکن هم هست که سود کمتری بدهند (اگر مدل انتقال بدی هرچند با احتمال کم، وجود داشته باشد).

در سیاست π^* ، بدترین حالت (= کمترین سود در بدترین مدل)، بهتر از بدترین حالت در سایر سیاست ها است. 

Temporal-difference Q-learning

- ✖ نسخه ADP فعال به صورت اختلاف زمانی (TD)
- ✖ پرهیز از داشتن مدل انتقال
 - په باید کرد؟
 - 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

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

SARSA

state, action, reward, state, action

پسرعموی Q-Learning

استفاده از: 

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

به جای: 

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

به جای بیشینه گیری روی اعمال، صبر کنیم عمل بعدی a' عملاً در الگوریتم انجام شود و بعد به روز کنیم: شاید برای اکتشاف یا ... عمل بهینه انتخاب نشود.

تداوم فطمشی در به روزرسانی سود: وقتی اکتشاف داریم، فرض نکنیم در قدم بعدی بیشینه سود دنبال می‌شود: **on-policy** یا **off-policy**

تعصیم RL

- × وقتی تعداد وضعيت‌ها فیلی زیاد باشد
- × تقریب تابع سودمندی ($\text{وو}(\text{دی}) = \text{وضعيت}$, $\text{فروجی} = \text{سودمندی}$)
- × ساده‌ترین حالت (تَرکِیب **خطی** ویژگی‌هایی مستخرج از وضعيت‌ها):

$$\hat{U}_\theta(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \dots + \theta_n f_n(s).$$

- مثلاً ویژگی‌ها (f ‌ها) = موقعیت خانه‌ها در مثال‌های قبل (یا هر ویژگی مربوط به خانه‌ها)
- تقریب صفحه (خطی) برای سودمندی
- × به جای نگهداری همه حالات، یادگارفتن تعدادی پارامتر تقریب زننده تابع سودمندی
- × با تقریب تابع، در وضعيت‌های دیده نشده هم تفہیم سود داریم

یادگیری تقویتی عمیق

- حالا شد !**
- به جای تقریب فطی، یادگیری عمیق**
- یا حتی کم عمق !!
- استفراج خودکار و یا زگی از وضعیت ها

- ناییدار است** (تغییرات محیط نسبت به فضای Train میتواند کارایی (ا خیلی کاهش دهد)
- کمتر در صنعت استفاده شده، اما در فضاهای پژوهشی، مورد علاقه است.**

چند مبحث ریزه صیزه

Reward Shaping X
 دادن شبیه پاداش

یادگیری تقویتی سلسله مراتبی X

- شکستن دنباله به زیر دنباله های کوچکتر
- تا وقتی که به احتمال قابل حل باشد

یادگیری تقویتی معکوس (inverse) X

- به جای یادگیری فقط مشی از روی پاداش، پاداش (هدف) یادگرفته شود از روی فقط مشی استاد!
- به جای آنکه عمل انعام دهد و بعد بر اساس پاداش/جریمه دریابد که عمل خوب بوده یا نه، عمل خوب/بد گفته شود و عامل دریابد که چرا این عمل خوب/بد بوده و مبنای پاداش چیست.

شاگردی در کارگاه، بدون لزوماً پاداش و جریمه مستقیم ■

والدین نسبت به فرزند: هم RL و هم RL معکوس ■

و هم ترکیب هر دو (میگوید درس بفوان(فقط مشی)، و وقتی فهاند چایزه (پاداش) هم میگیرد) ■

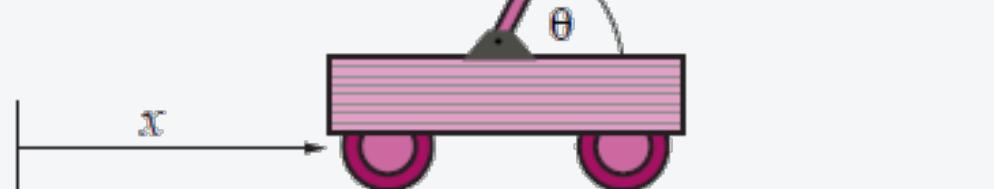
مثال پیوسته

(موقعیت، زاویه، سرعت چرخها، سرعت میله)

گام ثابت)
با طول گام متغیر در نقاط مختلف فضا)
(شبکه عصبی)

وضعیت پیوسته
ا) اهل؟

- گسسته سازی (با طول
- گسسته سازی تطبیقی (
- تقریب تابع پیوسته





یادگیری مان تقویت شد

