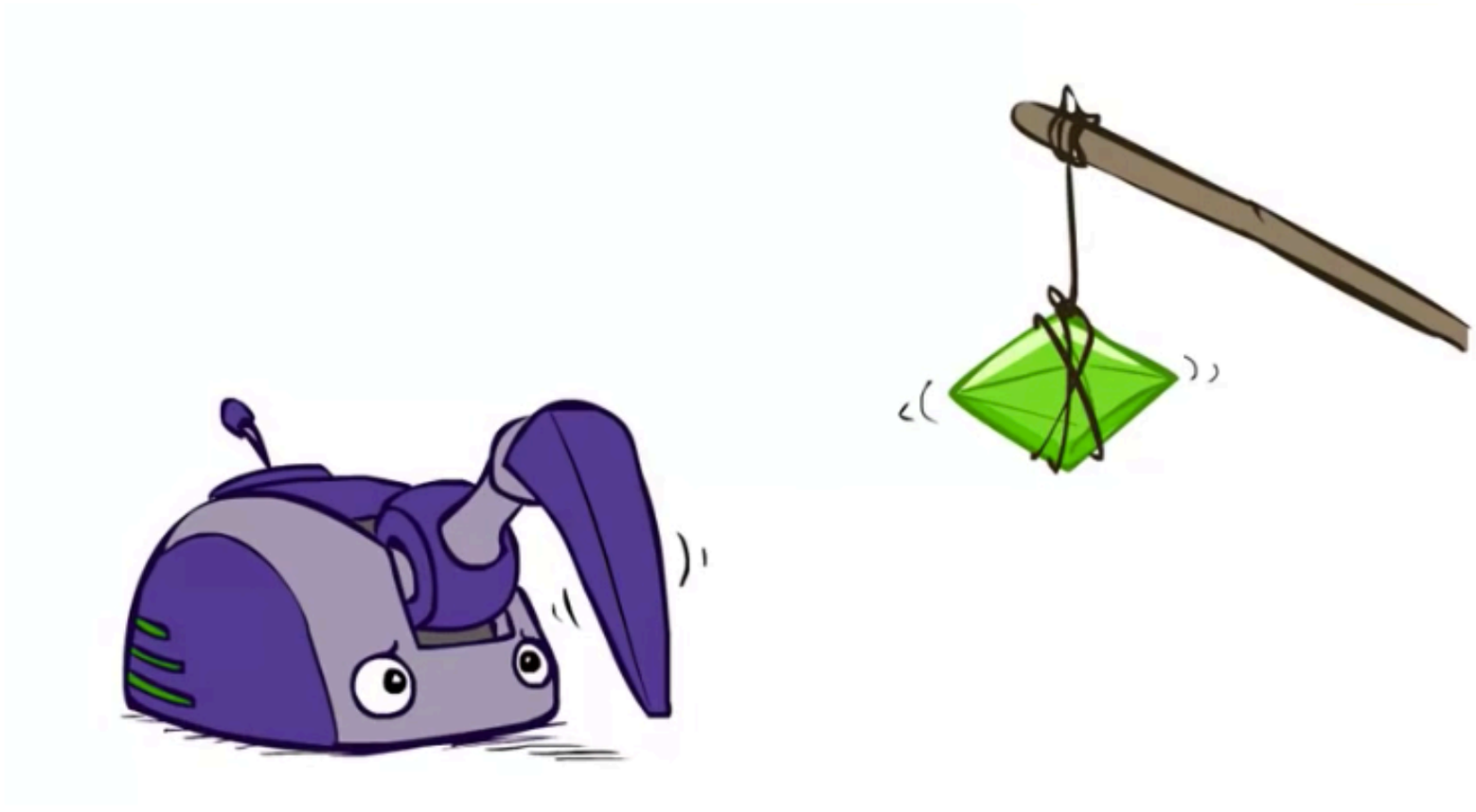
Reinforcement learning

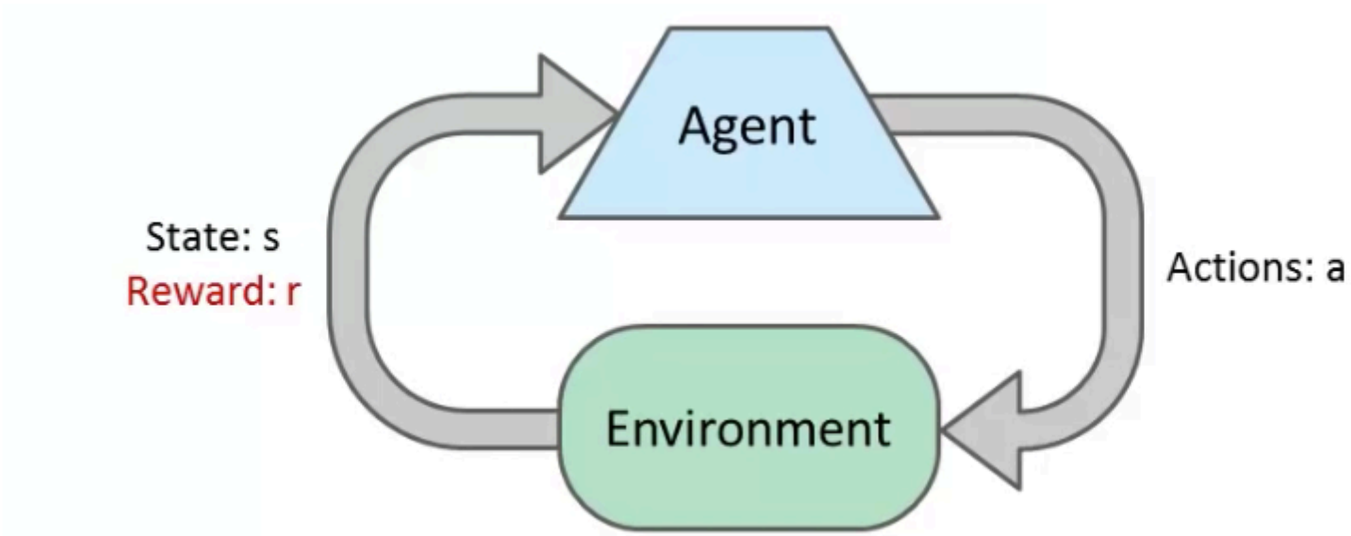
Q learning, SARSA

Technical University of Košice
Department of Computers and Informatics

Reinforcement learning



Reinforcement learning

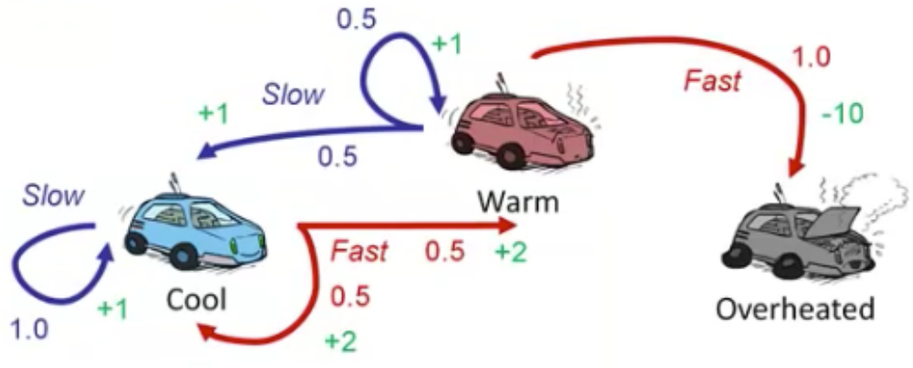


- Základné predpoklady
- Agent interaguje s prostredím a získava po každý akcií odmenu
- Úžitok agenta je definovaný získanými diskontovaými odmenami
- Cieľ: vykonávať akcie maximalizujúce jeho očakávaný úžitok
- Agent nemá vedomosti o modele. Učí sa na základe vzorkovania (samplovania) prostredia

Príklady

Reinforcement learning

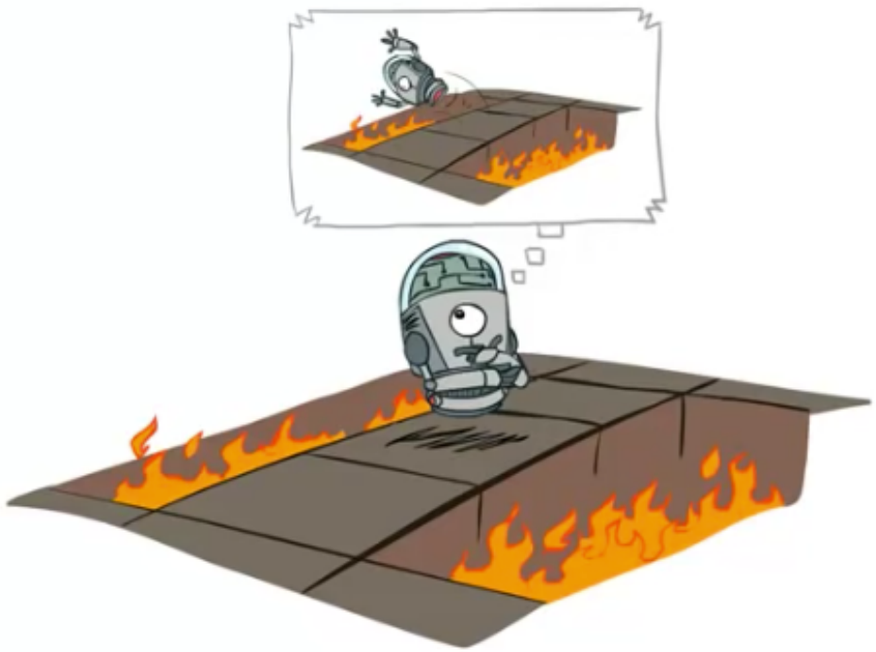
- Uvažujeme stále základný model MDP
  - Množinu stavov $s \in \mathcal{S}$
  - Množinu akcií nad každým stavom a, $a \in \mathcal{A}$
  - Dynamiku modelu $p(s', r \mid s, a)$



  - Odmenu, resp odmeňovaciu funkciu $R(s, a, s')$
  - Cieľ: nájdenie optimálneho strategického profilu $\pi_*(s)$

Nepoznáme dynamiku modelu a odmeňovaciu funkciu!!!!!!

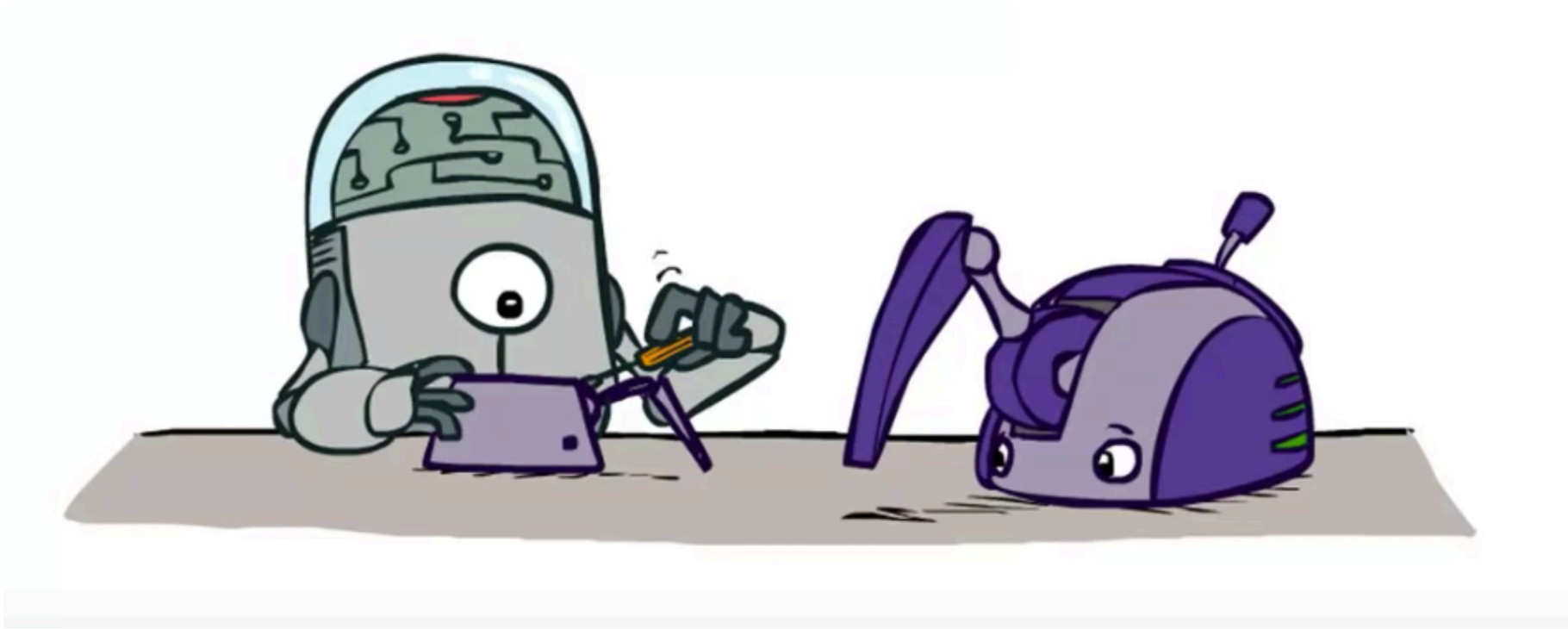
Offline (MDP) vs online (RL)



Riešenie offline (dynamika je známa)

Riešenie online (zisťujeme parametre modelu)

Model-based learning (učenie na báze modelu)



Model-based learning (učenie na báze modelu)

- Učenie na báze modelu:

  Postupné učenie parametrov modelu na základe interakcie

  Aplikácia dosiaľ uvedených algoritmov na naučenom modeli

- Krok 1. Učenie

  Určenie dynamiky modelu, rátaj pravdepodobnosť výskytu stavu s′ pre každé *s* a *a*

  Normovanie vedie k získaniu aproximovaných hodnôt $p(s', r \mid s, a)$ a $\hat{r}(s, a, s')$

Model-based learning (učenie na báze modelu)

- Učenie na báze modelu:

  Postupné učenie parametrov modelu na základe interakcie

  Aplikácia dosiaľ uvedených algoritmov na naučenom modeli

- Krok 1. Učenie

  Určenie dynamiky modelu, rátaj pravdepodobnosť výskytu stavu s′ pre každé *s* a *a*

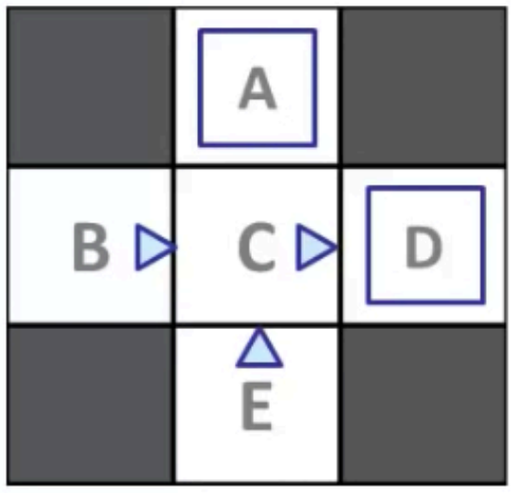
  Normovanie vedie k získaniu aproximovaných hodnôt $p(s', r \mid s, a)$ a $\hat{r}(s, a, s')$

- Krok 2. Implementácia algoritmov

  Value iteration, Policy iteration, resp. Policy evaluation

Model-based learning (učenie na báze modelu)

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

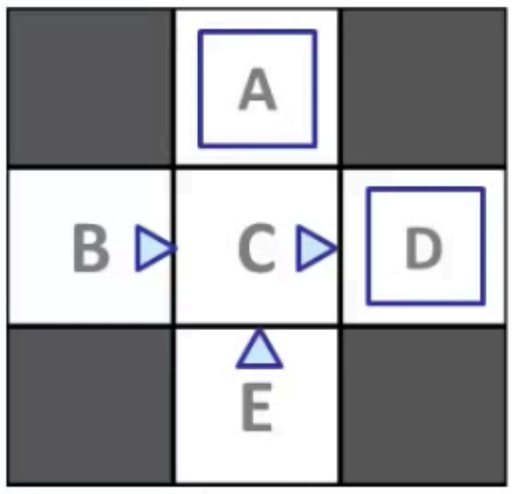
E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Model-based learning (učenie na báze modelu)

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Learned Model

$\hat{T}(s, a, s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
...

$\hat{R}(s, a, s')$

R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
...

Value Iteration, Policy Improvement, etc

Nevýhoda: zistenie parametrov modelu trvá čas

Model based vs model free učenie

Cieľ: zistenie vekovej skupiny nášho predmetu

Predpoklad: perfektná vedomosť hustoty pravdepodobnosti: $E(A) = \sum_a P(a)a$

Ak predpoklad nie je splnený, zbieraj vzorky študentov $a_1, a_2, a_3, \ldots, a_N$

Výpočet (je nutné vypočítať *P(a)* pre Model Based))

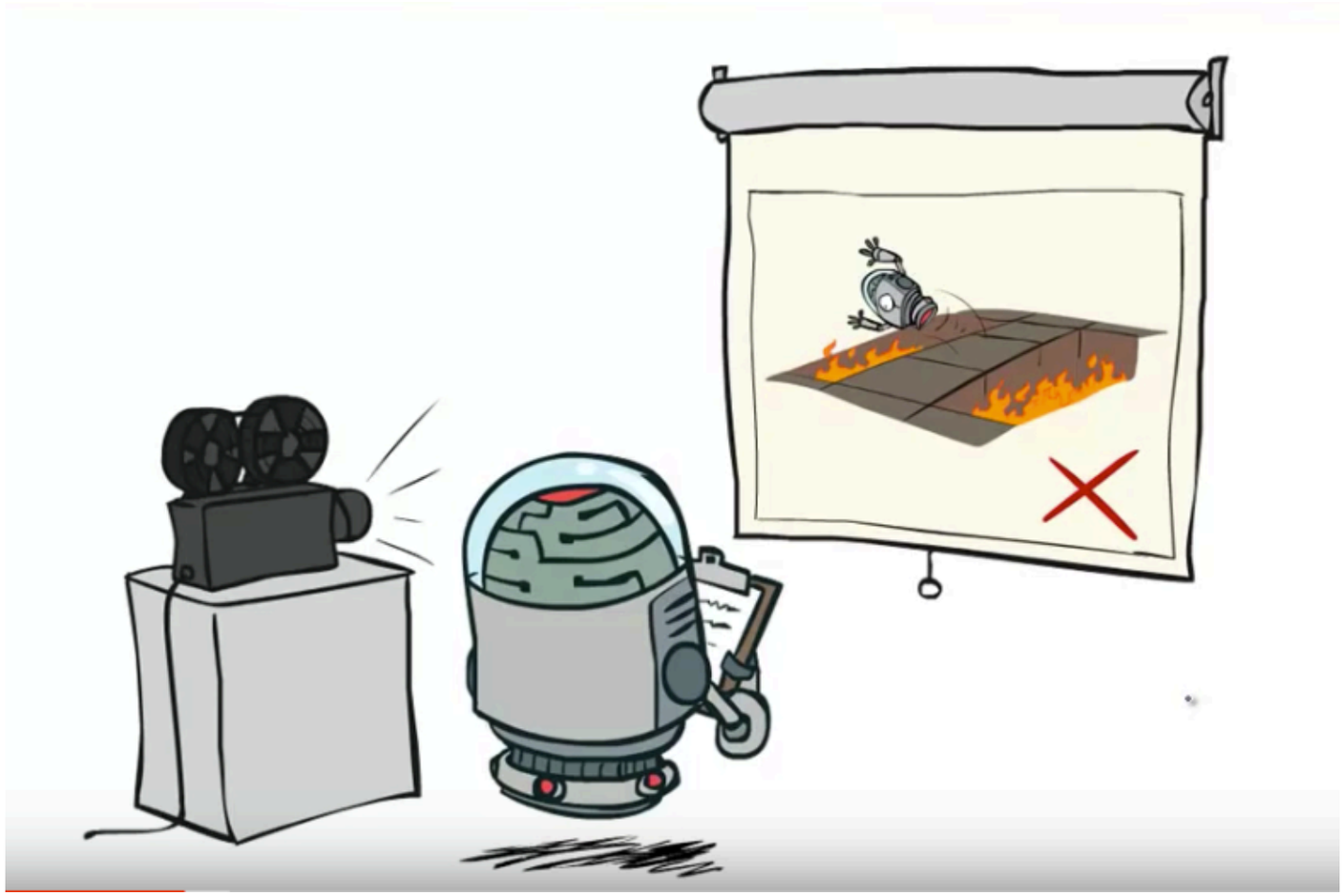
$\hat{P}(a) = \frac{\text{num}(a)}{N}$, kde *N* je počet vzoriek študentov množiny A

$E(A) \approx \sum_a \hat{P}(a)a$

Model free

$E(A) \approx \frac{1}{N}\sum_i a_i$

Pasívny reinforcement learning (prediction problem)



Pasívny reinforcement learning (prediction problem)

Zjednodušená úloha: evaluácia daného strategického profilu

Vstup: fixný strategický profil $\pi(s)$

$R(s, a, s')$ a $p(s', r \mid s, a)$ sú neznáme

Cieľ: určenie hodnôt jednotlivých stavov (i.e. $V(s), \forall s$)

V tomto prípade:

Agent vykonáva akcie free ride na základe definovaného strat. profilu

Strategický profil je konštantný

Vykonáva akcie na základe strategického profilu a ohodnocuje jednotlivé stavy

Priama evaluácia

- Cieľ: určenie hodnoty jednotlivých stavov na základe $\pi(s)$

- Myšlienka: Aritmetický priemer pozorovaných hodnôt (model free)

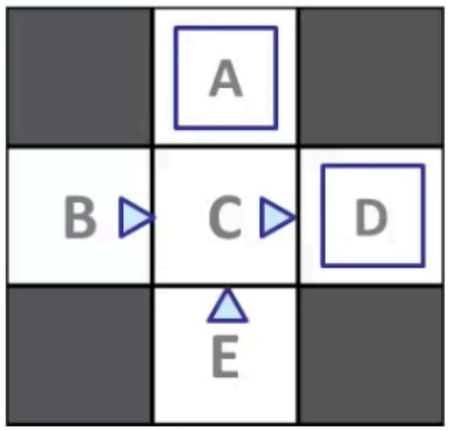
  - Vykonávaj akcie na základe $\pi(s)$

  - V každom časovom okamihu zapíš sumu diskotovaných odmien pre daný stav

  - Spriemerni (priama evaluácia)

Priama evaluácia

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

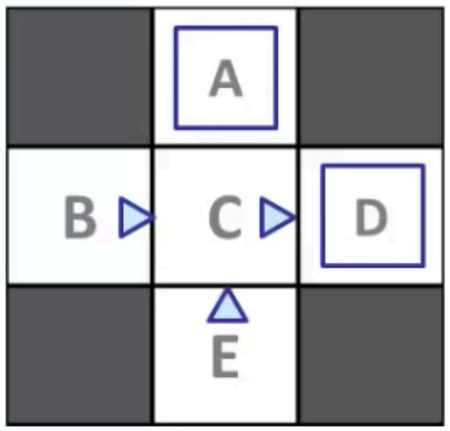
E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Priama evaluácia

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

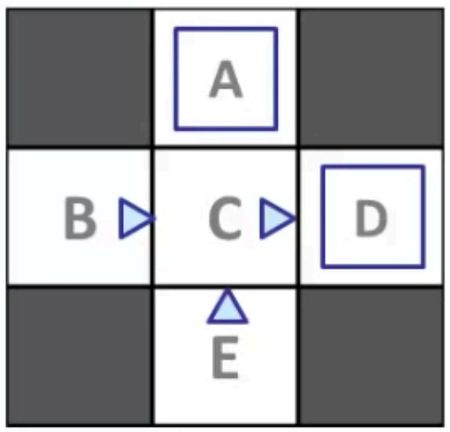
E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values

$$V_\pi(C) = \frac{9 + 9 + 9 - 11}{4}$$

Priama evaluácia

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

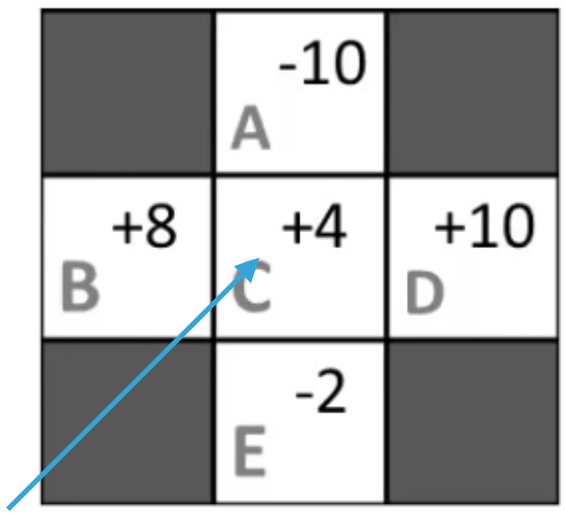
E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values

$$V_\pi(C) = \frac{9 + 9 + 9 - 11}{4}$$



Strácame súvislosť medzi stavmi! E nereflektuje na hodnotu C

Neuplatňujeme Bellmana, ale je to jednoduché no neefektívne

Policy evaluation?

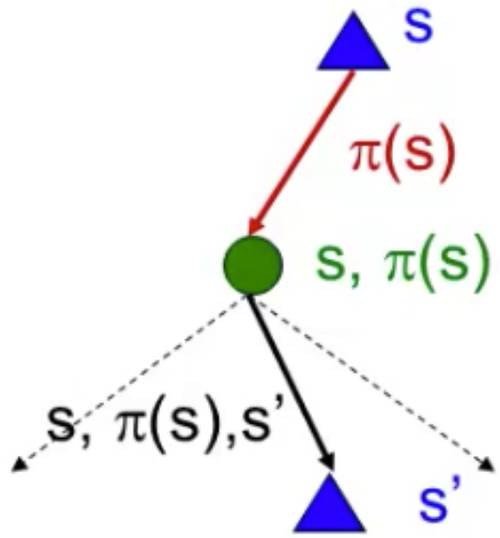
Úvaha: Využitie policy evaluation pre konštantný strategický profil

Pre každý časový okamih interakcie agenta s prostredím urobme update hodnoty stavu

$$V_\pi^0(s) = 0$$

$$V_\pi^{k+1}(s) \leftarrow \sum_{s'} T\left(s, \pi(s), s'\right) \left[R\left(s, \pi(s), s'\right) + \gamma V_\pi^k(s')\right]$$

Nepoznáme to, riešenie: vzorkovanie



Otázka: Nepoznáme váhy $T(s, \pi(s), s')$, ako môžeme uplatniť daný rekurzívny vzťah?

Policy evaluation by sampling?

Cieľ: Snaha zlepšiť presnosť ohodnotenia stavu $V(s)$ spriemerovaním jednotlivých pozorovaní

$$V_{\pi}^{k+1}(s) \leftarrow \sum_{s'} T\left(s, \pi(s), s'\right)\left[R\left(s, \pi(s), s'\right) + \gamma V_{\pi}^{k}(s')\right]$$

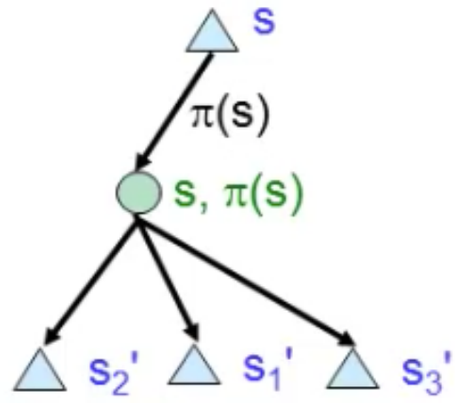
Riešenie: Vzorkovanie (samplovanie) výstupov zo stavov s' a spriemernovanie

$\pi(s)$, resp alternatívne akcia *a* v pripade determ. str.profilu

$$sample_1 = R\left(s, \pi(s), s_1'\right) + \gamma V_k^{\pi}\left(s_1'\right)$$

$$sample_2 = R\left(s, \pi(s), s_2'\right) + \gamma V_k^{\pi}\left(s_2'\right)$$

$$sample_3 = R\left(s, \pi(s), s_n'\right) + \gamma V_k^{\pi}\left(s_n'\right)$$



Policy evaluation by sampling?

Cieľ: Snaha zlepšiť presnosť ohodnotenia stavu $V(s)$ spriemerovaním jednotlivých pozorovaní

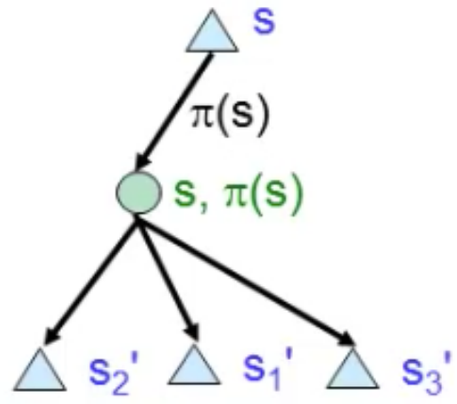
$$V_{\pi}^{k+1}(s) \leftarrow \sum_{s'} T\left(s, \pi(s), s'\right)\left[R\left(s, \pi(s), s'\right) + \gamma V_{\pi}^{k}(s')\right]$$

Riešenie: Vzorkovanie (samplovanie) výstupov zo stavov s' a spriemernovanie

$$sample_1 = R\left(s, \pi(s), s_1'\right) + \gamma V_k^{\pi}\left(s_1'\right)$$

$$sample_2 = R\left(s, \pi(s), s_2'\right) + \gamma V_k^{\pi}\left(s_2'\right)$$

$$sample_3 = R\left(s, \pi(s), s_n'\right) + \gamma V_k^{\pi}\left(s_n'\right)$$



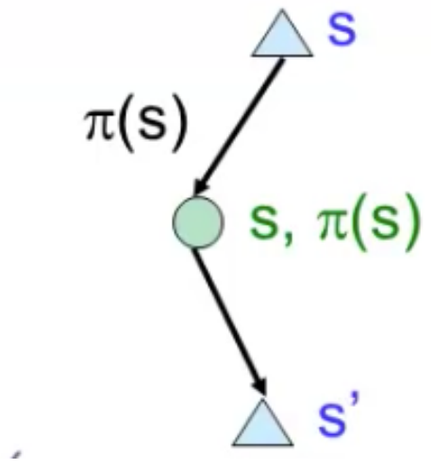
$$V_{\pi}^{k+1} = \frac{1}{N}\sum_{i} sample_i$$

Problém: MDP: nevieme sa vracať v čase

$$V_k^{\pi}(s)$$

Temporal difference learning

- Cieľ: učenie na základe každej interakcie s modelom
  - Update $V_\pi^k(s)$ po jednotlivých prechodoch *(s,a,s',r)*
  - Stavy s' , ktorých pravdepodobnosť výskytu $p(s', r \mid s, a)$, resp. $T(s', \pi(s), s)$ je vyššia budú ovplyvňovať $V_\pi^k(s)$ viac



Vzorka *V(s)*: $sample_1 \leftarrow R\left(s, \pi(s), s_1'\right) + \gamma V_k^\pi\left(s_1'\right)$

Vzorka *V(s)*: $V_\pi(s) \leftarrow (1-\alpha)V_\pi(s) + \alpha * sample$

resp: $V_\pi(s) \leftarrow V_\pi(s) + \alpha(sample - V_\pi(s))$

Kĺzavý priemer (exponential moving average)

- Kĺzavý priemer pozorovaní:

$$\bar{x}(n) = (1-\alpha) \,.\, \bar{x}_{n-1} + \alpha \,.\, x_n$$

- Posledné pozorované vzorky majú väčšiu váhu, ako tie ktoré boli pozorované v minulosti

$$\bar{x}(n) = \frac{x_n + (1-\alpha) \,.\, x_{n-1} + (1-\alpha)^2 \,.\, x_{n-2} + \ldots}{1 + (1-\alpha) + (1-\alpha)^2 + \ldots}$$

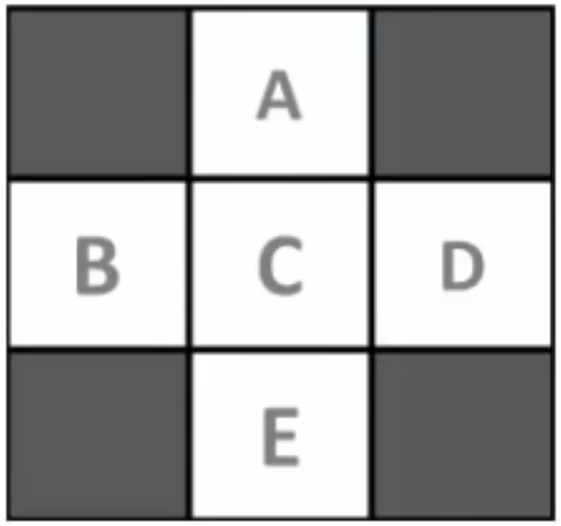
- Snaha: Prvé pozorovania boli značne vychýlené, ich vplyv sa bude s časom zmenšovať.

- Cieľ: Znižovanie parametra α nám zaručuje konvergenciu uvedených algoritmov

$$\alpha(n) = \frac{1}{n} \qquad\qquad \alpha(n) \leftarrow \alpha(n) * 0.99$$

Príklad TD

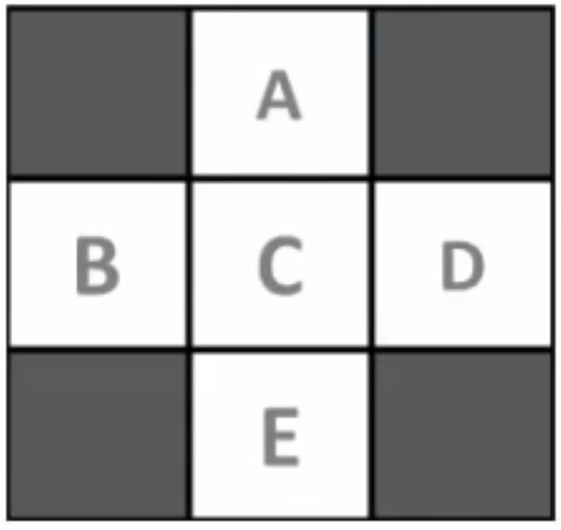
States



Assume: $\gamma = 1$, $\alpha = 1/2$

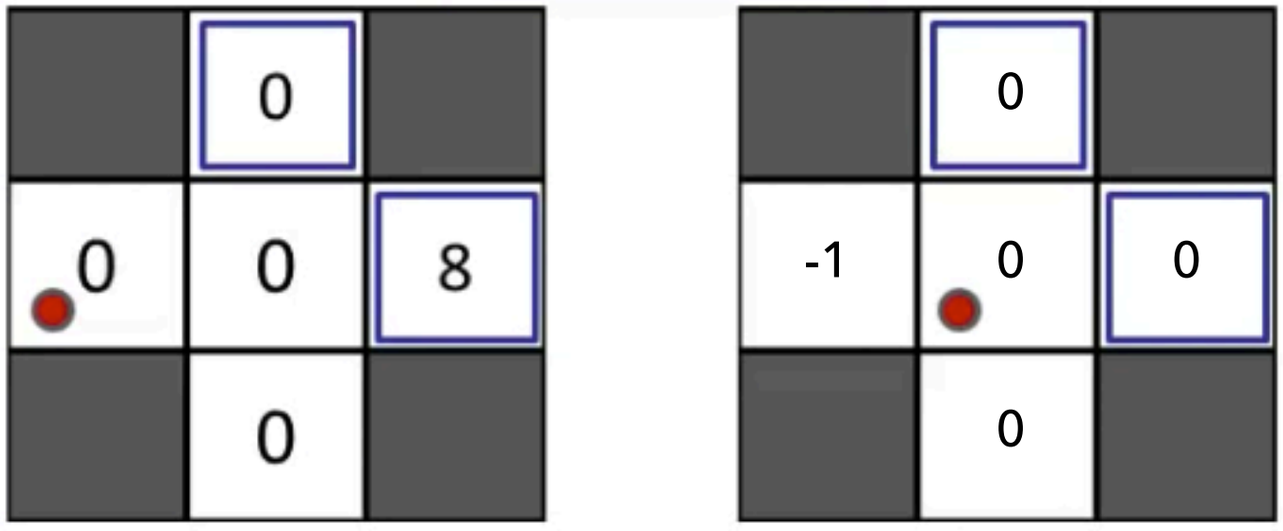
Príklad TD

States



Observed Transitions

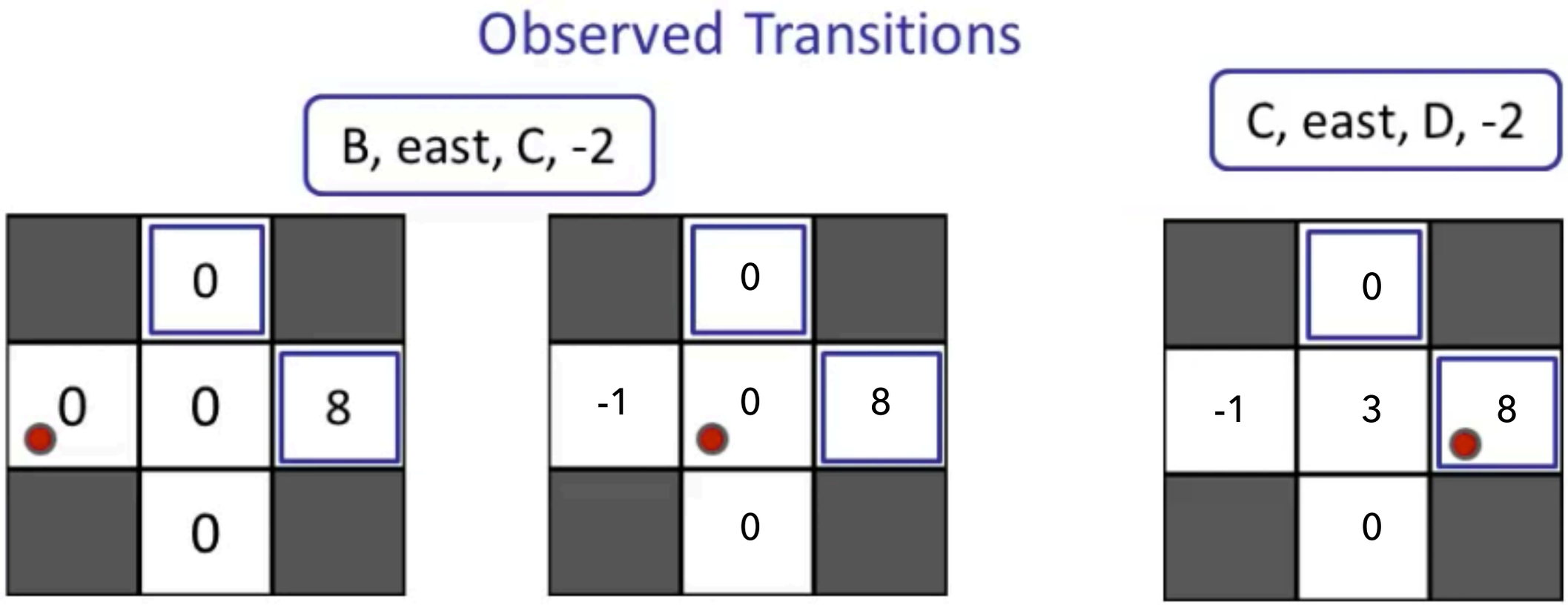
B, east, C, -2



Assume: $\gamma = 1$, $\alpha = 1/2$

$$V_\pi(s) \leftarrow (1-\alpha)V_\pi(s) + \alpha * (R(s,\pi(s),s') + \gamma \,.\, V_\pi(s')$$

Príklad TD



$$V_\pi(s) \leftarrow (1-\alpha)V_\pi(s) + \alpha * (R(s,\pi(s),s') + \gamma \,.\, V_\pi(s')$$

Temporal difference: problémy

TD je vynikajúci prístup na ohodnocovanie konkrétneho strategického profilu $\pi(s)$ využitím

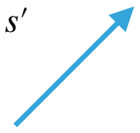
základných Bellman rovníc a ich iteratívneho riešenia (na báze model free prístupu)

Problém: ako zlepšiť strategický profil (hodnotové funkcie stavov nám nehovoria nič o akciách)?

$$\pi(s) = \arg\max Q(s,a), \forall a \in \mathcal{A}$$

Nemožné: POLICY EXTRACTION sa použiť nedá

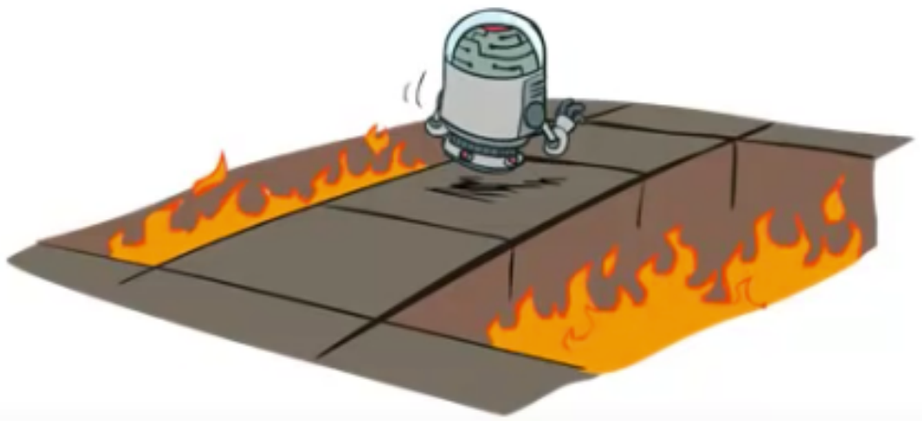
$$Q(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V(s')]$$



Nápad: učenie Q hodnôt

Aktívny RL (control based RL)

- Schopnosť zlepšovať strategický profil správanie agenta v neznámom prostredí
  - Nevieme prechodové pravdepodobnosti $p(s', r \mid s, a)$
  - Nevieme tvar odmeňovacej funkcie *R(s,a,s')*
  - Agent vyberá akcie a snaží sa ich zlepšiť za účelom maximalizácie očakávaného úžitku
- Agent rozhoduje o akciách v prostredí, nie daný strategický profil ako pri TD
  - Explorácia vs. Exploitation - problém ako pri banditoch



Aktívny RL (control based RL)

Recap: Value iteration

Inicializujme $V^0(s) = 0, \forall s$

Iteratívnym prístupom updatujme jednotlivé hodnoty pre všetky stavy *s*

$$V^{k+1}(s) = \max_a \sum_{s'} \pi(s, a, s')[R(s, a, s') + \gamma(V^k(s')]$$

Iteratívnym prístupom updatujme jednotlivé hodnoty pre všetky stavy *s*

Problém: Našou snahou je zistiť Q hodnoty

Inicializujme $Q^0(s, a) = 0, \forall s, a$

Iteratívnym prístupom updatujme jednotlivé hodnoty $Q^k(s, a)$ postupne, na základe interakcií s modelom

Aktívny RL (control based RL)

Recap: Value iteration

Inicializujme $V^0(s) = 0, \forall s$

Iteratívnym prístupom updatujme jednotlivé hodnoty pre všetky stavy *s*

$$V^{k+1}(s) = \max_a \sum_{s'} \pi(s,a,s')[R(s,a,s') + \gamma(V^k(s')]$$

jednoduchý sampling ako pri TD nestačí,

máme MAX

Iteratívnym prístupom updatujme jednotlivé hodnoty pre všetky stavy *s*

Problém: Našou snahou je zistiť Q hodnoty

Inicializujme $Q^0(s,a) = 0, \forall s, a$

Iteratívnym prístupom updatujme jednotlivé hodnoty $Q^k(s,a)$ postupne, na základe interakcií s modelom

Aktívny RL (control based RL)

Recap: Value iteration

Inicializujme $V^0(s) = 0, \forall s$

Iteratívnym prístupom updatujme jednotlivé hodnoty pre všetky stavy *s*

$$V^{k+1}(s) = \max_a \sum_{s'} \pi(s,a,s')[R(s,a,s') + \gamma(V^k(s')]$$

→ jednoduchý sampling ako pri TD nestačí,

→ máme MAX

Iteratívnym prístupom updatujme jednotlivé hodnoty pre všetky stavy *s*

Problém: Našou snahou je zistiť Q hodnoty

Inicializujme $Q^0(s,a) = 0, \forall s, a$

Iteratívnym prístupom updatujme jednotlivé hodnoty $Q^k(s,a)$ postupne, na základe interakcií s modelom

neevidujeme max

$$Q^{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^k(s')]$$

$$Q^{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q^k(s',a')]$$

Q learning

Q learning: updatovanie Q hodnôt jednotlivých stavov na základe interakcie s modelom

$$Q^{k+1} \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

Algoritmus: učenie Q(s,a) postupne, ako prechádzame modelom

1) získanie vzorky (s,a,s',r)

2) Načítanie odhadu hodnoty Q(s,a)

3) Výpočet nového samplu v tvare $sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$

4) Aplikácia kĺzavého priemeru na nový update Q pre vykonanú akciu *a* v stave *s* hodnoty v tvare

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(sample)$$

Q learning

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$
    $S \leftarrow S'$
  until S is terminal

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha\left[R_{t+1} + \gamma \arg\max_{A'} Q\left(S_{t+1}, A'_{t+1}\right) - Q\left(S_t, A_t\right)\right]$$

DEMO Q learning

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha\left[R_{t+1} + \gamma \arg\max_{A'} Q\left(S_{t+1}, A'_{t+1}\right) - Q\left(S_t, A_t\right)\right]$$

SARSA learning

Cieľ: transformácia TD na SARSA, náhradou V(s) za Q(s,a) podľa $\pi(s)$

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha\left[R_{t+1} + \gamma Q\left(S_{t+1}, A_{t+1}\right) - Q\left(S_t, A_t\right)\right]$$

pozor, nerobíme max (neaproximujeme $Q_*(s, a)$)

Tvrdenie: SARSA umožnuje nájsť optimálny strategický profil $\pi_*(s)$, za predpokladu, že všetky stavy v MDP sú navštívene dostatočne veľakrát a parameter ϵ klesá k nule dostatočne pomaly.

SARSA learning

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\mathit{terminal}, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

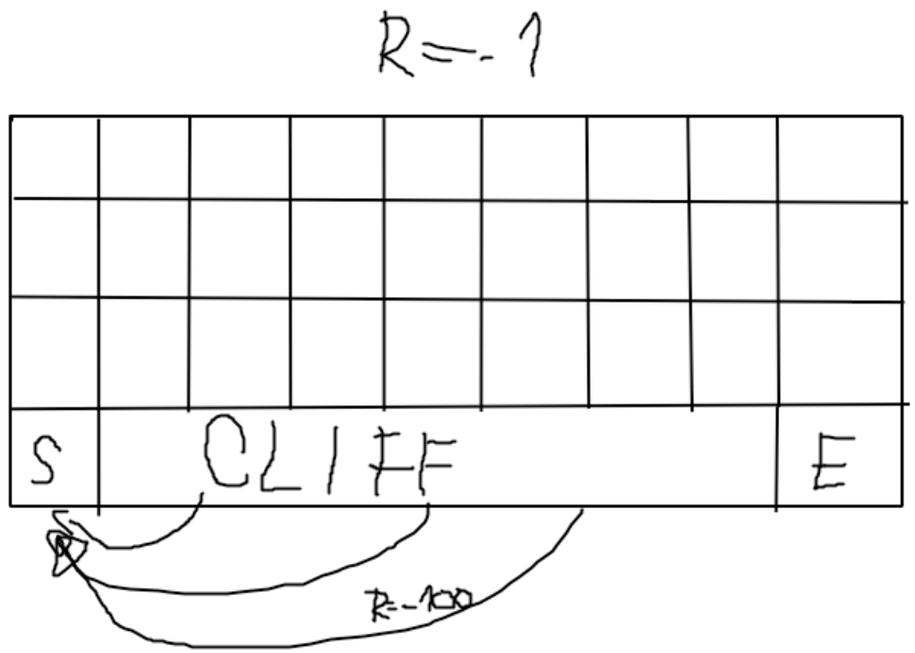
Kvíz

Optimálny strategický profil pre

konštantné $\epsilon = 0.2$

SARSA vs Q learning

Aký je očakávaný užítok pre optimálne strategické profily?



Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

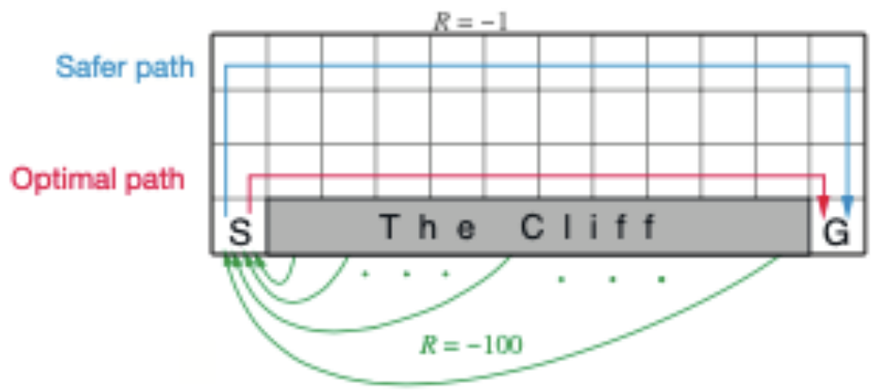
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

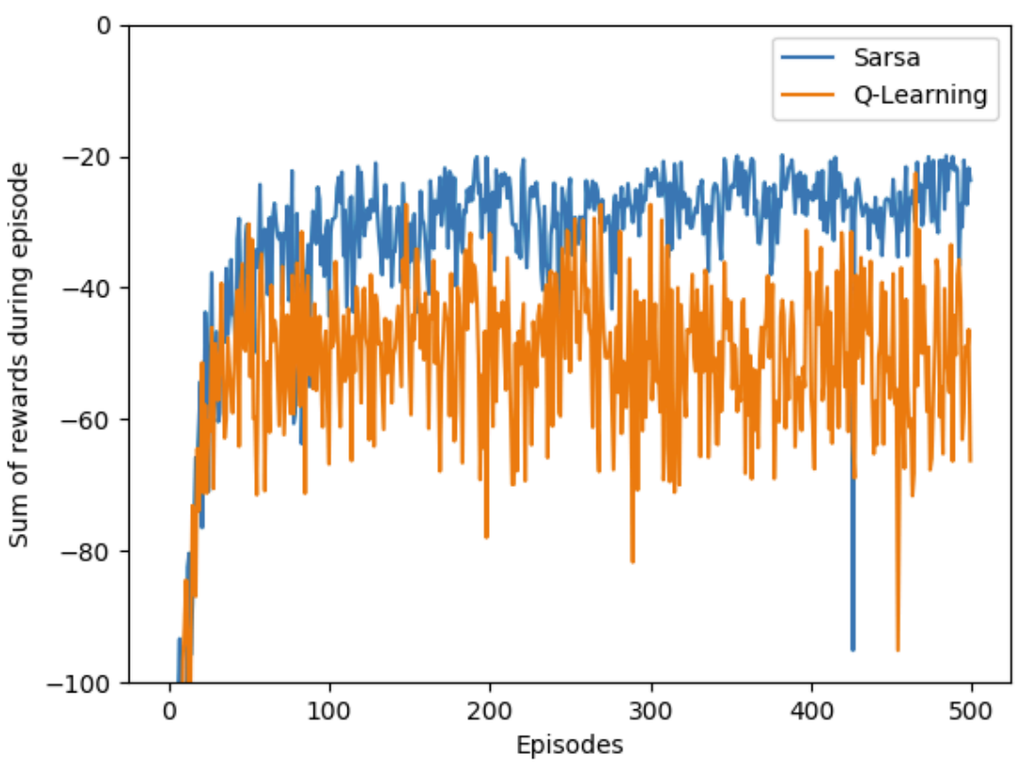
Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

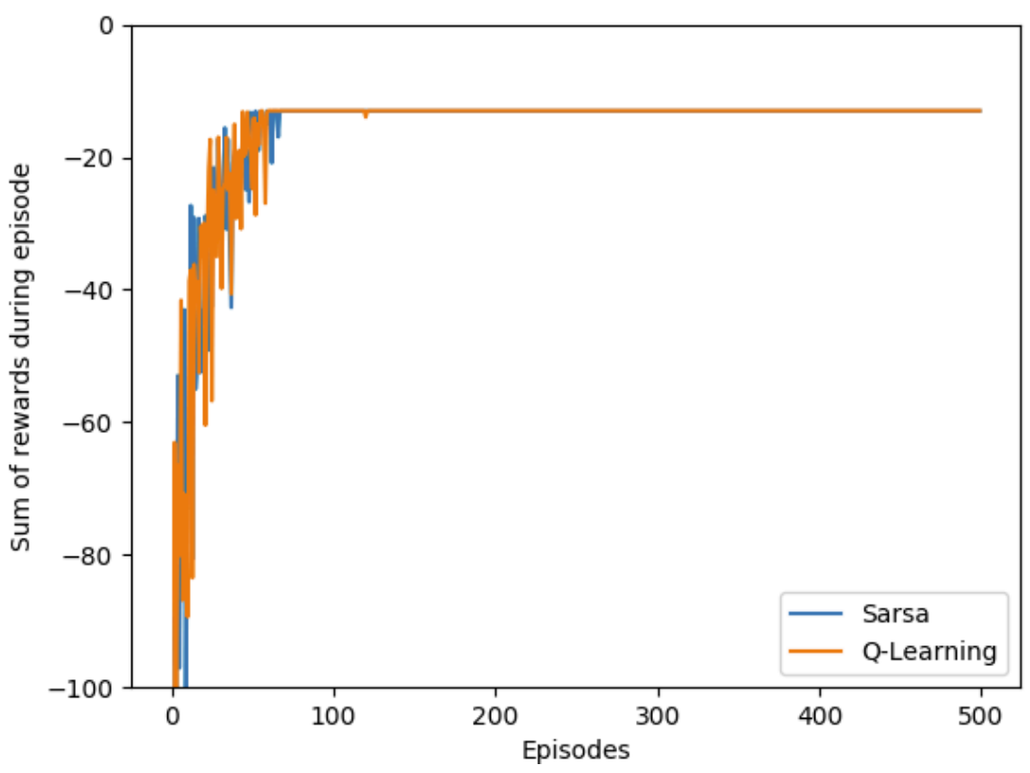
SARSA vs Q learning (príklad Python, 6.6)



$\epsilon = 0.2$



$\epsilon \leftarrow \epsilon * 0.99$



SARSA a Q learning

- Tvrdenie: Q learning umožnuje získanie optimálneho strategického profilu bezohľadu aký je strategický profil agenta (náhodný, pseudonáhodný, greedy, atď).
- Táto skutočnosť podmieňuje nazývať Q learning ako OFF-POLICY algoritmus
- Predpoklady:
  - Musíme dostatočne dlho spoznávať model (fáza exploration)
  - Správna rovnováha v parametri learning rate. Nesmie klesať príliš rýchlo, ale ani veľmi pomaly
  - Teoreticky pri Q learningu nie je podstatné ako navštevujeme stavy, ani akým strat. profilom sa riadime.

SARSA a Q learning

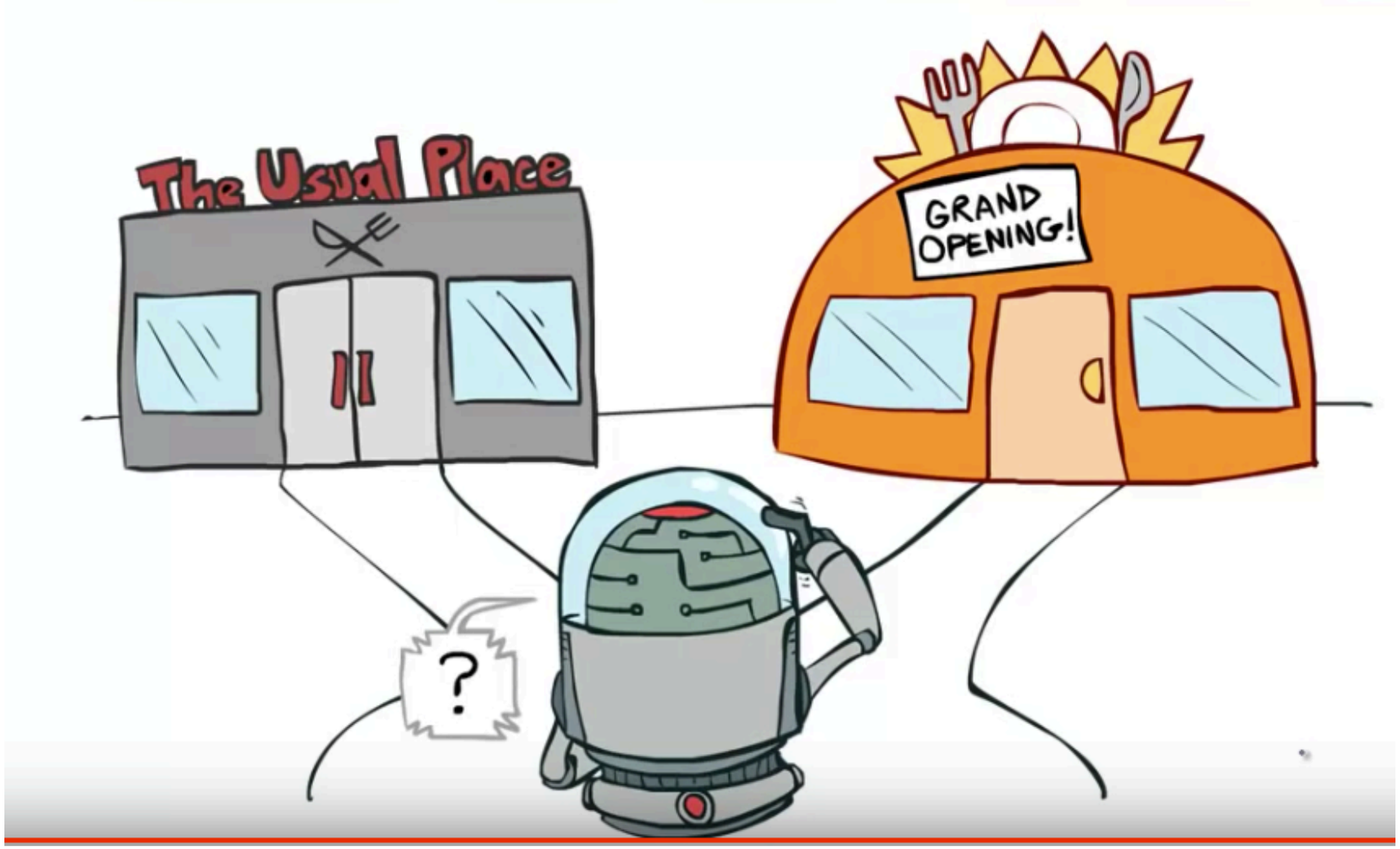
SARSA algoritmus: on policy algoritmus, update Q hodnôt jednotlivých stavov je vykonávaný podľa aktuálneho strategického profilu

Je viac SAFE ako Q learning.

Pri vhodne klesajúcej miere explorácie dosahujú v tabuľkových formách oba algoritmy rovnaké výsledky

Exploration vs Exploitation

Bádanie v modeli, resp využívanie modelu



Exploration vs Exploitation

Vo všeobecnosti existuje niekoľko prístupov podporujúcich exploráciu

Jednoduché riešenie: ϵ greedy algoritmus

v každom časovom okamihu si hoď mincou

S malou *p()* ϵ vykonaj náhodnú akciu

S veľkou *p() 1-*ϵ, vykonaj akciu podľa súčasného strategického profilu

Úvaha: náhodné akcie spôsobujú po čase zníženie výkonnosti agenta

Po natrénovaní modelu môžu náhodné akcie agenta dostať do stavov pre neho nevýhodných

Riešenie: znižujme parameter ϵ (analógia so simulovaným žíhaním)