

Učenje potkrepljivanjem (eng. Reinforcement learning)
Beštak za koga besedice inteligencije na matematičkom jeziku

Nemanja Micević

Markovian proces odlučivanja : (S, A, R, T, P, γ)

S - skup svih stanja

A - skup svih akcija

R - skup svih nagrada ($R \subseteq \mathbb{R}$)

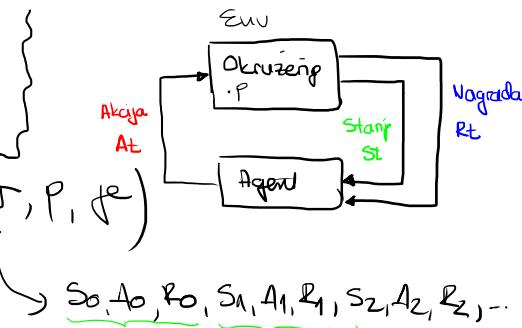
T - transitorija

P - funkcija predstava u okruženju

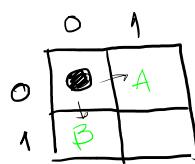
$$P: S \times R \times S \times A \rightarrow \mathbb{R}$$

$$P(s', r | s, a) = P(S_{t+1} = s', R_t = r | S_t = s, A_t = a)$$

γ - faktor učenja $\in [0, 1]$



P je "neba" funkcija, a u praksi je pogodno da je svedeno na verovatnoću



Igrac je u $(0,0)$

Nagrada u tek istoj, +1

dostupne akcije: $a_0 = \downarrow$ $a_1 = \rightarrow$

s_t

$$P((0,1), 1 | (0,0), \rightarrow) = 1 = P(S_{t+1} = (0,1), R_t = 1 | S_t = (0,0), A_t = \rightarrow)$$

$$P((0,1), 1 | (0,0), \downarrow) = 0 = P(S_{t+1} = (0,0), R_t = 1 | S_t = (0,0), A_t = \downarrow)$$

$0 \rightarrow \emptyset / \emptyset \rightarrow \emptyset$

MDP

Markovianog ugovora

prethodna stanja

$$P(S_t, R_{t-1} | \cancel{S_0, A_0, R_0, \dots, S_{t-1}, A_{t-1}}) = P(S_t, R_{t-1} | \cancel{S_{t-1}, A_{t-1}})$$

Ako je poznato trenutno stanje procesa, za zaključivanje o budućnosti

poznavanje prešlosti nje neplaća

nije važno kada smo uđeli došli

ako napustim partiju Šoka u stanju S , onda drugi igrač može da use zahvaljujući

Kada agent interaguje sa okolinom generiše se epizode

Politika $\Pi \rightarrow$ funkcija koja agent donosi odluke.

↳ raspodela verovatnoće

$\Pi(a|s) \rightarrow$ verovatnoća da će agent preduzeti akciju a ako se nađe u stanju S

Mogu biti determinističke i nedenističke

γ - faktor učenja vaga začetaj kratkoročnu i dugoročnu nagradu

$\gamma = 0 \rightarrow$ maksimalni fokus na kratkoročnu nagradu → polarna politika

$$\gamma = 1$$

Sto znači rezultat Markovianog procesa odlučivanja?

Najčešći način kojeg modelizuje očekivanu dobijenu agentu

Dobitak od koraka t

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k} \quad (\gamma = 1)$$

Definicija
MDP-a
(faktor učenja)

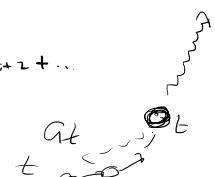
Zavis od politike

Tipičan učenički G.

čemu više stanja nagrade nego daje veće nagrade

$$G_t = \gamma^0 R_t + \gamma^1 R_{t+1} + \gamma^2 R_{t+2} + \dots$$

$$G_t = R_t + \gamma G_{t+1}$$



U praksi na primer $\gamma = 0.9$; sljedeće

$$J = S_0, A_0, R_0, S_1, A_1, R_1, S_2, A_2, R_2$$

$$P(J) = P(S_0) \prod_{t=0}^{\infty} \Pi(A_t | S_t) P(S_{t+1}, R_t | S_t, A_t)$$

Raspodela po svim mogućim transitorijama

Raspodela po mogućim polaznim stanjima

Politika koju praktički agent

verovatnoća da će agent da preduzme At kada dođe u St

Funkcija predstava akcije

Verovatnoća da će da se desi ono što je agent uradio

Očekivana nagrada koju agent dobije

preteći politiku π^t iz stanja s

$$U^{\pi^t}(s) = \mathbb{E}_{\pi^t} [G_t | S_t = s]$$

Funkcija vrednosti stanja

$$q^{\pi^t}(s, a) = \mathbb{E}_{\pi^t} [G_t | S_t = s, A_t = a]$$

Funkcija vrednosti akcije u stanju

U^{π} i g^{π} se mogu povezati

$$U^{\pi}(s) = \mathbb{E}_{\pi} [G_t | S_t = s]$$

$$U^{\pi} \rightarrow g^{\pi} : = \sum_a \underbrace{\pi(a|s)}_{\text{verodljiva akcija}} \mathbb{E}[G_t | S_t = s, A_t = a]$$

$$= \sum_a \underbrace{\pi(a|s)}_{\text{Otežano verodljivo da se izaberu}} \underbrace{g^{\pi}(s, a)}_{\text{akcije } a \text{ u stanju } s}$$

Vise novih odgovara $U^{\pi} \rightarrow g^{\pi}$

kao i obrnuto

$$G_t = R_t + \gamma G_{t+1}$$



$$g^{\pi}(s, a) = \mathbb{E}_{\pi} [G_t | S_t = s, A_t = a]$$

$$g^{\pi} \rightarrow U^{\pi} : = \mathbb{E} [R_t + \gamma G_{t+1} | S_t = s, A_t = a]$$

$$= \sum_{s', r} P(s', r | s, a) [r + \gamma \mathbb{E} [G_t | S_t = s']]$$

$$= \sum_{s', r} P(s', r | s, a) [r + \gamma U^{\pi}(s')]$$

Ovo je u praksi teže dobiti jer zadatak da vam je rezultat kako funkcioniše okruženje

Komentar: Postoji pristupi koji modeluju i druženje

Kako da poređimo 2 politike po kvalitetu?

Ako vam za sve stanja $s \in S$

$$U_{\pi_1}(s) \geq U_{\pi_2}(s), \text{ jasno je da } \pi_1 \text{ bolja od } \pi_2 \\ (\text{ili barem jednaka...})$$

Za MDP postoji optimalna politika π^* koja je bolja (ili jednaka) svim ostalim politikama.

$$U^*(s) = \max_{\pi} U_{\pi}(s) \quad q^*(s, a) = \max_{\pi} q_{\pi}(s, a)$$

želimo π^* radi! → one daje U^* → kada ne bi bile optimalne

$$\text{Vidim: } U^*(s) = \max_a q^*(s, a)$$

Najbolja akcija a
u stanju s

Ako poznamo funkciju g^* , možemo izračunati oan jednu optimalnu politiku.

$$\pi^*(a|s) = \begin{cases} 1, \text{ ako } a \text{ je max } q^*(a|s) \\ 0, \text{ inace} \end{cases} \rightarrow \text{Deterministička politika}$$

⇒ želimo neke način da izračunavamo g^*

$$\begin{aligned} U^*(s) &= \max_a g^*(s, a) \\ &= \max_a \mathbb{E}_{\pi^*} [G_t | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi^*} [R_t + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \max_a \left(\sum_{s', r} P(s', r | s, a) \left[r + \gamma \mathbb{E}_{\pi^*} [G_{t+1} | S_t = s', A_t = a] \right] \right) \\ &= \max_a \sum_{s', r} P(s', r | s, a) [r + \gamma U^*(s')] \end{aligned}$$

Buduća (rekurentna)
funkcija za U^*

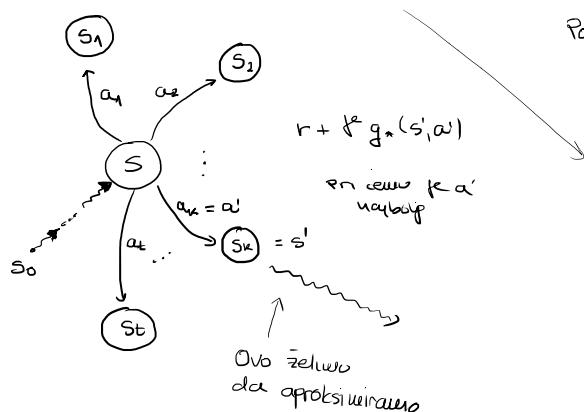
Buduća jednakost za g^* je

$$g^*(s, a) = \sum_{s', r} P(s', r | s, a) \left[r + \max_a g^*(s', a) \right]$$

Poznata polaznog može doći do iterativnog postupka

$$U_{t+1}(s) \leftarrow \max_a \sum_{s', r} P(s', r | s, a) [r + \gamma U_t(s')]$$

$$g_{t+1}(s, a) \leftarrow \sum_{s', r} P(s', r | s, a) \left[r + \gamma \max_a g_t(s', a) \right]$$



Ovo želimo da aproksimiramo

13.2 Učenje u nepoznatom okruženju

Predstavlja iterativni postupak novog ne odgovara jer zahteva poznavanje funkcije okruženja P .

\Rightarrow Agent treba da istraži okruženje

novi element u našem razmišljanju

Explore - exploit dilemma
RL

Podsećaj, belmanova jednačina za π glasi

$$g_{\pi}(s, a) = \sum_{s', a'} p(s', r | s, a) \left[r + \gamma \max_{a'} g_{\pi}(s', a') \right]$$

ekstremne stvare kada
 optimiziramo samo
 poučiti jednu opštiju
 \downarrow
 $(s) \xrightarrow{a} (s')$

$$r + \gamma \max_{a'} g_{\pi}(s', a')$$

linearna apsakulacija pouči
log vagačne varijabli

\Rightarrow novi nivo poznavanja \rightarrow potičeće da je osim

Q learning

$$g(s, a) \leftarrow (1 - \alpha_t) g(s, a) + \alpha_t (r + \gamma \max_{a'} g(s', a'))$$

\downarrow



α_t - korak učenja (eng. learning rate)

r

Popratljiva apsakulacija kroz 'veči' (stevac)

$$\sum_{t=0}^{\infty} \alpha_t = \infty \quad \sum_{t=0}^{\infty} \alpha_t < \infty \quad (\text{Robins-Monro uslov})$$

Kako verovatno staroj, najoj apsakulujuci?
Nadajući parametar α_t .

E politika politika

Politika koja je potpuna "do na Σ ". Postoji neka verovatnoća da politika ne bude potpuna.
Preciznije, politika

$$\Pi_{g_{\Sigma}}(a|s) = \begin{cases} 1-\varepsilon, & \text{ako } a = \arg \max_{a'} g(s, a) \\ \frac{\varepsilon}{|A|-1}, & \text{inace} \end{cases}$$

$$\Pi \rightarrow \Pi_{\varepsilon}$$

$$\varepsilon \rightarrow 0.05$$

$$\varepsilon = 0.99$$

$$\varepsilon_{decay} = 0.95$$

Primer: Neka je s veliko "trouče" na prikaz
 $A = \{ \downarrow, \rightarrow, \uparrow, \leftarrow \}$ step

Ako je samo potpuna
na prikaz

$$\Pi_g(\uparrow, s) = 1$$

$$\Pi_g(\downarrow, s) = 0$$

$$\Pi_g(\rightarrow, s) = 0$$

$$\Pi_g(\leftarrow, s) = 0$$

Ako je $\varepsilon = 0.3$ potpuna

$$\Pi_g(\uparrow, s) = 1 - \varepsilon = 1 - 0.3 = 0.7$$

$$\Pi_g(\downarrow, s) = \frac{\varepsilon}{|A|-1} = \frac{0.3}{4-1} = 0.1$$

$$\Pi_g(\rightarrow, s) = 0.1$$

$$\Pi_g(\leftarrow, s) = 0.1$$

Verovatnoća koja se računava rasporedi na ostale
akcije u odnosu na (dosadšnje) naloge.

$$0.3$$

$$30\%$$

Često ε smanjuje tokom algoritma, tako da nije istraživati. Na primer: $\varepsilon_t = \frac{1}{1+t}$, gde je t broj trenutne iteracije.

Algoritam g -učenja

Uzeti: Broj iteracija N

Uzeti: Apsakulujuća funkcija g

1. Inicijaliziraj s na polazno stanje

2. Inicijaliziraj s na polazno stanje

3. $t \leftarrow 1$

4. ponavljaj

ε politerna politika sa $\varepsilon = \frac{1}{1+t}$

5. predviđaj akciju $a \sim \Pi_{g_{\varepsilon_t}}(a|s)$ i opazi nagradu r i novo stanje s'

6. $\alpha_t \leftarrow \frac{1}{1+t}$ $\alpha_t \leftarrow \frac{1}{N}$

// azurira se korak učenja

7. $g(s, a) \leftarrow (1 - \alpha_t) g(s, a) + \alpha_t (r + \gamma \max_{a'} g(s', a'))$

// ovo se desava "učenje"

8. Ako je s zaučen stanje onda $g(s, a) = \alpha g(s, a) + \alpha (r + \gamma \max_{a'} g(s', a'))$

9. inicijaliziraj s na polazno stanje

10. inace

11. $s \leftarrow s'$

12. $t \leftarrow t+1$

+ Uči jednostavno

- Memoristički (potencijalni) zaključci tabele Q

- Ne može generalizirati sa stanja na stanje

13. dok nije ispunjen uslov $t=N$ $\rightarrow N$ broj epizoda 10 000

14. vrati g kao rešenje

Hotimo li ovu primenu na soli?

teorijalno: da!

Praktično: NE! Možemo li mi čuvati g tabelu u memoriji?

To je matrica dimenzija $|S| \times |A|$

X(ostavio se! :))

13.2.2. Učenje u prostorima stanja i akcija

Do sada, skup stanja i skup akcija su bili konečni.

Ipak postoji situacija kada nisu, ili kada je njihova kardinalnost toliko velika da se ne može reći koliko je.

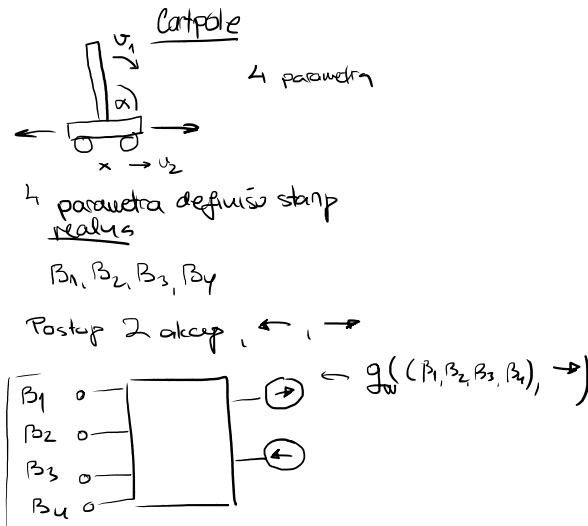
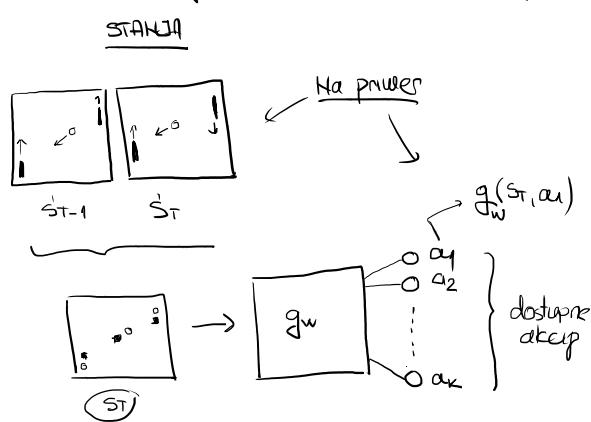
Tada nije moguće primeniti prethodnu pristup.

Mi u stvari želimo da računamo funkciju g_* koju aproksimujemo nekom funkcijom.

Kako je učimo, možemo reći da je parametrizovana, odnosno g_w .

Šta je w u praksi? Na primer, težine neuronke mreže!

Važno pitanje je kako reprezentovati stanja i akcije?



U stvarnosti, mi želimo: nu

$$\min_w \mathbb{E} \left[(g_w(s, a) - g_*(s, a))^2 \right]$$

U praksi, očekujemo menjajuće prosekovanje

$$w \leftarrow w + \frac{\Delta t}{N} \sum_{s, a} (g_w(s, a) - g_*(s, a))^2$$

Kako ovu da rešimo? Gradientni spust!

$$w \leftarrow w + \frac{\Delta t}{N} \sum_{s, a} \underbrace{(g_w(s, a) - g_*(s, a))}_{\text{korak učenja}} \nabla_w g_w(s, a)$$

Na primer, algoritam propagacije unutar za neuronku mreže.

2013, 2015

> Ako pređe da učimo nad 1 uzorkom

$$w \leftarrow w + \Delta t \left(g_w(s, a) - \underbrace{g_*(s, a)}_{\text{Ali mi ne znamo } g_*!} \right) \nabla_w g_w(s, a)$$

Ali mi ne znamo g_* !
⇒ Aproksimiramo je

$$w \leftarrow w + \Delta t \left(g_w(s, a) - \left(r + \max_a g_w(s', a) \right) \right) \nabla_w g_w(s, a)$$

⇒ Ipak, otkao pristup novog teorijskog garanciju → konvergenciju.

Deep Q network

rad iz 2013 godine
koji ilustruje agenta koji igra Atari igru
(+ dosta hokava, trikava...)

implementacione pojednostavljenje prikaz na vezakavu :)

off policy

on policy

$\pi(a_t | s_t)$

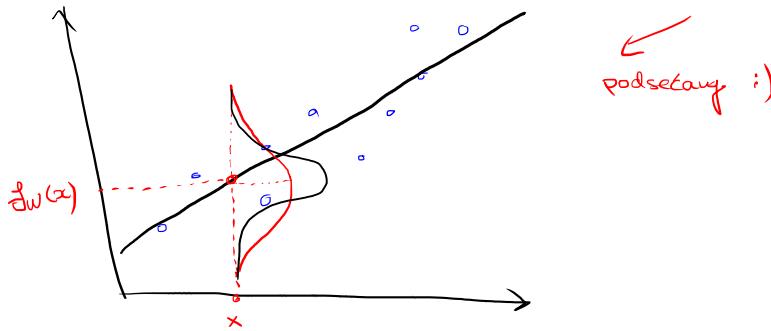
lučko problem sa beskonačnim i neprekidnim akcijama.

$g_w(s) \rightarrow$ gradientni uspon da radi nekako a ? → spor, komplikovan...

Da li možemo da učimo politiku?

Policy gradient i algoritam REINFORCE

Pripremajući da $\pi_w(a|s)$ može se modelirati sa $\mathcal{N}(\hat{\pi}_w(s), \sigma^2)$



Cilj u RL-u:

\hat{C}_t

Maksimizacija očekivane nagrade po svim trajektorijama
u odnosu na raspodjelu trajektorija koju definise jednačst:

$$P(\tau) = P(s_0) \prod_{t=0}^{\infty} \pi_w(a_t | s_t) P(s_{t+1}, r_t | s_t, a_t)$$

Odnosno, mi želimo da maksimizujemo veličinu:

$$J(w) = \mathbb{E}_{\tau \sim P_w(\tau)} [r(\tau)]$$

Kako? Gradient! ^"

Dodatak za trajektoriju τ
zastaviti w ? To su parametri koji definisu politiku ;)

$$\begin{aligned} \nabla_w J(w) &= \nabla_w \mathbb{E}_{\tau \sim P_w(\tau)} [r(\tau)] \\ &= \nabla_w \int r(\tau) P_w(\tau) d\tau \quad \text{raspisivanje očekivanja} \\ &= \int r(\tau) \nabla_w P_w(\tau) d\tau \quad \text{gradient protkoz integral} \\ &= \int r(\tau) \frac{\nabla_w P_w(\tau)}{P_w(\tau)} P_w(\tau) d\tau \quad \text{trik: } \text{pouzdano sa } \lambda = \frac{P_w(\tau)}{P_w(\tau)} \\ &= \int r(\tau) \left[\nabla_w \log P_w(\tau) \right] P_w(\tau) d\tau \quad \text{if } g(x) = \log x \Rightarrow g'(x) = \frac{1}{x} \\ \nabla_w J(w) &= \mathbb{E}_{\tau \sim P_w(\tau)} \left[r(\tau) \nabla_w \log P_w(\tau) \right] \quad \text{zapisivanje nazad u očekivanje} \end{aligned}$$

Sada nose gradijent direktno raspada nesto (P_w) što zavisi od w .

$$\nabla_w \log P_w(\tau) = \nabla_w \left[\log P(s_0) + \sum_{t=0}^T \left(\log \pi_w(a_t | s_t) + \log P(s_{t+1} | s_t, a_t) \right) \right]$$

log ce dati sumu jer P_w je mera probnosti

$$\nabla_w \log P_w(\tau) = \sum_{t=0}^T \nabla_w \log \pi_w(a_t | s_t)$$

I naredno, končno

$$\nabla_w J(w) = \mathbb{E}_{\tau \sim P_w(\tau)} \left[r(\tau) \sum_{t=0}^T \nabla_w \log \pi_w(a_t | s_t) \right]$$

Algortim REINFORCE

Uzor: Broj iteracija N

Izlaz: Aproximacija Jw optimalne politike

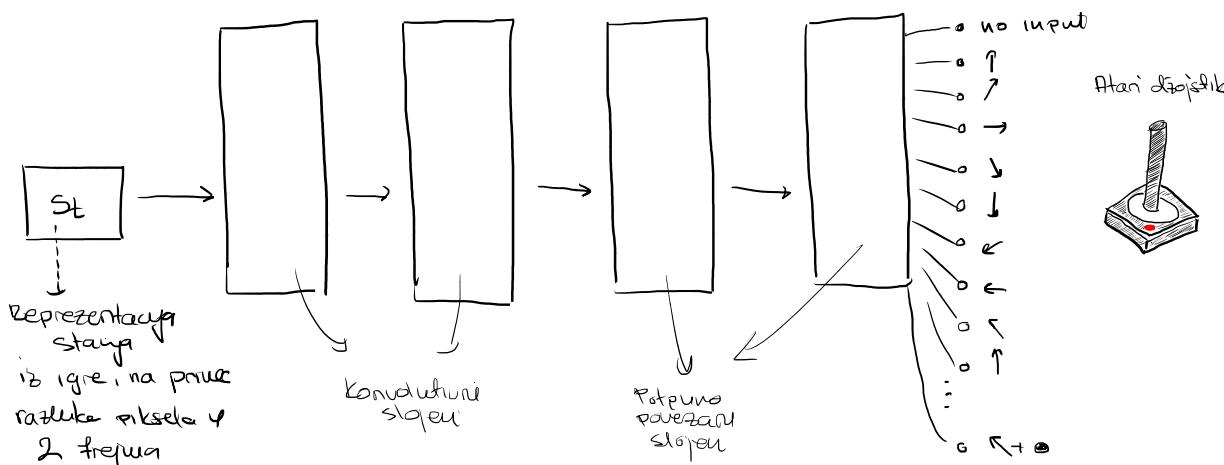
1. Inicijalno inicijaliziraj W
2. $t \leftarrow 1$
3. ponavljaj
4. generisati epizodu T končnjeg politike Jw
5. izracunati nagradu $r(T)$
6. $W \leftarrow W + \alpha \left[r(T) \sum_{t=0}^T \nabla_W \log \pi_{\theta}(a_t | s_t) \right]$
7. $t \leftarrow t+1$
8. dok npr ispunjen uslov $t=N$
9. vrati π_{θ} kao rešenje

Igraju igru...

Saberemo nagrade u celi period

azurirano tezine neuronske mreže (ako je n. mreža)

Deep Q RL



$$W \leftarrow W + \alpha \left[q_w(s, a) - \underbrace{\left(r + \max_a q_w(s', a) \right)}_{\text{Privremeno zadržana mreža}} \right] \nabla_W q_w(s, a)$$

↓
tekucu mrežu

Postoji "memorija" (buffer) u kojoj se svestraju informacije (stanje akcija, nagrada) koje agent stekne tokom igre.
Na osnovu utorka iz ove memorije se vodi trening.

Hvala na pažnji i srećno! :)

u ostatak studija od srca!

