Računarska gimnazija

MATURSKI RAD

Napredne tehnike programiranja

Primena učenja sa podsticajem za rešavanje Atari igara

Autor:
Ognjen Nešković

Mentor:
dr Filip Marić

Beograd, maj 2022.

Sadržaj

Sažetak

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

1

Uvod

1.1 Osnove

Još od davnina ljudi su se nadmetali u igranju igara poput šaha, kineske igre Go , Backgammon itd. U mnogim kulturama oni koji su bili najbolji u ovim igrama bili su izuzetno poštovani, pa bi neki ljudi posvećivali čak i čitav svoj život usavršavanju svoje strategije u nekoj od ovih igara. Stoga, dugo se smatralo da je sposobnost igranja ovih igara nešto jedinstveno za ljude i da je stvaranje mašine ili algoritma sposobnog da pobedi najboljeg šahistu ili drugog profesionalnog igrača gotovo nemoguće.
Ovaj izazov mučio je matematičare stotinama godina i ostao je neprevaziđen sve do 1997. godine kada je konačno računar (Deep blue) prvi put pobedio najboljeg šahistu tog vremena Garija Kasparova (Campbell et al., 2002) . Deep blue je koristio alfa-beta algoritam pretrage, heuristike, ekspertsko znanje i specijalni hardver napravljen kako bi probao što veći broj poteza po sekundi. Postojao je veći broj šahovskih mašina koje su prethodile Deep blue mašini. Kako bi se razvio Deep blue bilo je potrebno puno šahista, programera, vremena i novca. Dakle, jasno je da iako su velik uspeh, Deep blue i njemu slične mašine nisu veoma generalne ili uopšte primenljive van vrlo specijalizovanog domena za koji su napravljene.
Velika prekretnica došla je sa razvojem neuronskih mreža, učenja sa podsticajem i hardvera. Sve ovo omogućilo je da se razviju generalni algoritmi koji bez velike modifikacije mogu naučiti da veoma dobro igraju veliki broj igara (Mnih et al., 2015) . Sledeći veliki korak u mašinskom igranju igara došao je sa programom Alpha Go koji je 2016. godine pobedio svetskog šampiona Li Sedola u igri Go. Alpha Go je koristio učenje sa podsticajem, konvolucione neuronske mreže, Monte Karlo pretragu, takođe delimično je bio treniran na igrama koje su igrali profesionalni Go igrači (Silver et al., 2017) . Godinu dana kasnije - 2017. godine

objavljena je verzija Alpha Go-a koja ne koristi ikakvo ekspertsko znanje - Alpha Go Zero , koja postiže čak bolji rezultat od verzije trenirane sa ekspertskim igrama. Poslednji veliki napredak je program AlphaStar koji igra stratešku video igru StarCraft II gde pobeđuje najbolje igrače ove igre (Vinyals et al., 2019) . Glavna razlika između metoda korišćenih za rešavanje šaha (alfa-beta pretraga) i savremenih metoda je način kako se igra modeluje. U savremenim metodama učenja sa podsticajem problemi koje je potrebno rešiti predstavljaju se kao Markovljevi procesi odlučivanja. Ako se problem ovako prestavi u nekim jednostavnim slučajevima se može rešiti dinamičkim programiranjem, a za složenije slučajeve se rešenje može aproksimirati neuronskim mrežama.

1.2 Motivacija

Igre ili opštije, problemi koji se mogu predstaviti kao Markovljevi procesi odlučivanja su od velikog teoretskog značaja za mašinsko učenje. Pored toga što donekle pružaju uvid u mogućnosti računara i razlike između ljudske i veštačke inteligencije, veliki broj procesa u stvarnosti poput problema optimalne kontrole (upravljanje robotima i mašinama), regulisanje sistema za hlađenje, kompresije videa itd. mogu se modelovati kao Markovljevi procesi odlučivanja. Složene igre pružaju dobar način da se granice savremenih metoda preciznije odrede, što primenu onda čini znatno lakšom.

1.3 Cilj rada

Cilj ovog rada je da se izlože metode učenja sa podsticajem od najjednostavnijih do složenijih. Uz ovo prikazani su problemi koji su rešivi svakom od metoda i problemi koji prevazilaze mogućnosti date metode, pa motivišu upotrebu neke složenije metode.

Glava 2

Metode

2.1 Uvod

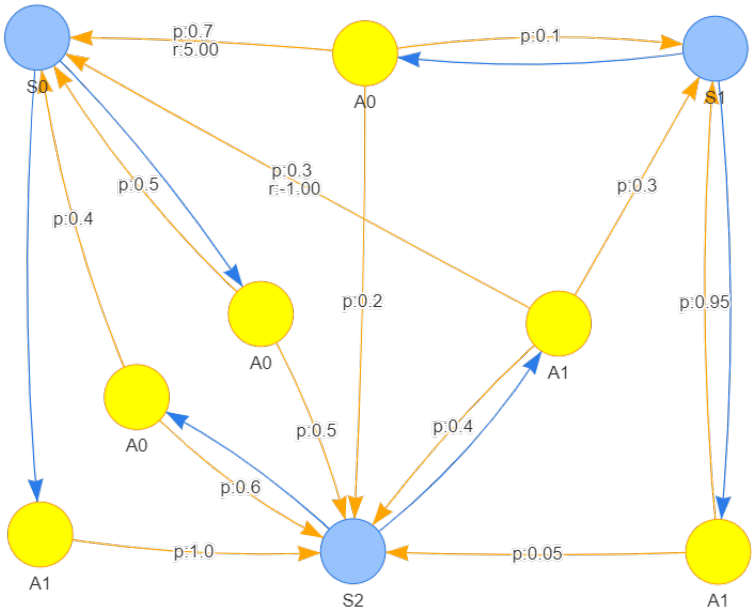
U narednom delu će biti razmatrane varijante Markovljevog procesa odlučivanja i rešenja koja su moguća za te varijante, pa je potrebno definisati Markovljev proces odlučivanja i uvesti potrebnu notaciju. Markovljevi procesi odlučivanja su uopštenje Markovljevih lanaca sa dodatim akcijama, za definiciju markovljevog procesa potrebno je difinisati sledeće:

- Skup stanja S
- Skup akcija A i A_s - skup mogućih akcija u stanju s
- $P_a(s, s') = P(s_{t+1} = s' \mid s_t = s, a_t = a)$ - verovatnoću da se završi u stanju s' ako je u stanju s izvršena akcija a
- $R_a(s, s')$ - nagrada koja se ostvaruje kada se u stanju s izvrši akcija a i završi u stanju s'
- Funkcija (politika) $\pi(s) : S \to A$ koja predstavlja igrača (onog koji donosi odluku), pa za dato stanje s iz skupa stanja S određuje akciju koju treba izvršiti

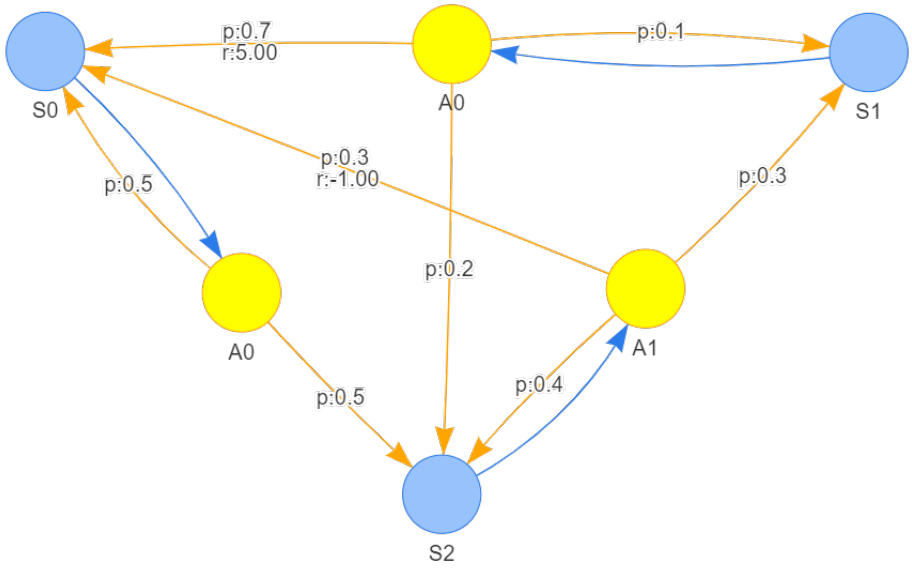
Rešenje Markovljevog procesa odlučivanja je optimalna funkcija π_{opt} :

$$\text{argmax}_{\pi_{\text{opt}}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{\pi_{\text{opt}}(s_t)}(s_t, s_{t+1})\right]$$

Posmatrajmo jednostavan Markovljev proces odlučivanja sa tri stanja S_0, S_1, S_2 i dve moguće akcije A_0, A_1 (slika 1).



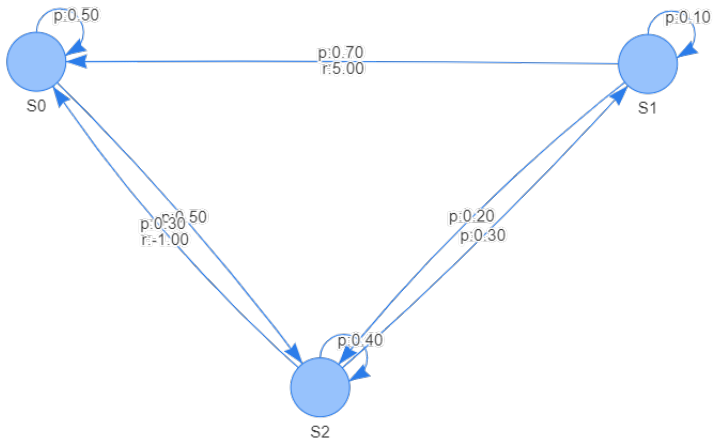
Slika 1: Primer Markovljevog procesa odlučivanja



Slika 2: Markovljev proces pod politikom π

Primetimo da ukoliko se fiksira neka politika π, na primer $\pi(S_0) = A_0, \pi(S_1) = A_0, \pi(S_2) = A_1$ proces sa slike 1 postaje ekvivalentan novom procesu (slika 2).

Dodatno, lako je videti da je tako dobijen proces (slika 2) ekvivalentan sledećem Markovljevom lancu, sa dodatim nagradama (slika 3). Kada je dat ovakav



Slika 3: Ekvivalentan Markovljev lanac sa nagradama

lanac, kao sa slike 3, postavlja se pitanje: kako efikasno odredti očekivanu nagradu u tonm lancu, ako je dato početno stanje ili verovatnoće da svako stanje bude početno. Ako je ovo moguće uraditi, onda je odmah dat efikasan način da se evaluira očekivana nagrada koja će biti ostvarena ako se koristi neka fiksna politika.

2.2 Evaluacija politike (policy evaluation)

Kada je politika fiksirana, traženje očekivane nagrade koja će biti ostvarena ako se prati data politika je ekvivalentna nalaženju očekivane nagrade u nekom markovljevom lancu. Prvo, uvedimo sledeću notaciju:

- S_t - Skup svih sekvenci dužine t. Na primer ako se iz stanja N_0 poseti stanje N_1, pa stanje N_2 itd. pa se na kraju poseti stanje N_t sekvenca bi bila $(N_0, N_1, N_2, \cdots, N_t)$
- $S_{t,u}$ - Skup svih sekvenci stanja dužine t koje se završavaju sa stanjem u.
- $T(u, v)$ - Verovatnoća da se iz stanja u pređe u stanje v.
- $p(s), s \in S_t$ - Verovatnoća da se neka sekvenca dogodi. Ako je data sekvenca $s = (N_0, N_1, N_2, \cdots, N_t)$ onda je $p(s) = T(N_0, N_1) \cdot T(N_1, N_2) \cdots T(N_{t-1}, N_t)$
- $R(u, v)$ - Nagrada kada se iz stanja u završi u stanju v.
- $r(s), s \in S_t$ - Nagrada za datu sekvencu s.

$$r(s) = R(N_0, N_1) + \gamma R(N_1, N_2) + \gamma^2 R(N_2, N_3) \cdots + \gamma^{t-1} R(N_{t-2}, N_{t-1})$$
$$\gamma < 1$$

- $L(u,t)$ - Verovatnoća da se posle t koraka završi u stanju u.

$$L(u,t) = \sum_{s \in S_{t,u}} p(s)$$

- $E_r(t)$ - Očekivana nagrada posle t koraka.

$$E_r(t) = \sum_{s \in S_t} p(s) r(s)$$

Lema 2.1. $L(u,t) = \sum_v T(v,u) L(v,t-1)$

$$\begin{aligned}
L(u,t) &= \sum_{s \in S_{t,u}} p(s) \\
L(u,t) &= \sum_{N_0,N_1,\cdots,N_{t-1}} T(N_0,N_1) \cdot T(N_1,N_2) \cdots T(N_{t-2},N_{t-1}) \cdot T(N_{t-1},u) \\
L(u,t) &= \sum_v \sum_{N_0,N_1,\cdots,N_{t-2}} T(N_0,N_1) \cdot T(N_1,N_2) \cdots T(N_{t-2},v) \cdot T(v,u) \\
L(u,t) &= \sum_v T(v,u) \sum_{N_0,N_1,\cdots,N_{t-2}} T(N_0,N_1) \cdot T(N_1,N_2) \cdots T(N_{t-2},v) \\
L(u,t) &= \sum_v T(v,u) L(v,t-1)
\end{aligned}$$

Ako definišemo da je $L(t)$ vektorska funkcija, a T matrica možemo prethodnu jednačinu zapisati elegantnije kao:

$$L(t) = L(t-1)T$$

Kako je $L(0) = L_0$ gde je L_0 Verovatnoća da se počne iz svakog stanja. Lako se može pokazati da je:

$$L(t) = L_0 T^t$$

Lema 2.2. $E_r(t) = E_r(t-1) + \gamma^{t-1} \sum_u \sum_v T(u,v)R(u,v)L(u,t)$

$$E_r(t) = \sum_{s \in S_t} p(s)r(s)$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} p(s') \sum_v T(u,v)(R(s') + \gamma^{t-1}R(u,v))$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} p(s')(\sum_v T(u,v)R(s') + \gamma^{t-1} \sum_v T(u,v)R(u,v))$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} p(s')(R(s') \sum_v T(u,v) + \gamma^{t-1} \sum_v T(u,v)R(u,v))$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} p(s')(R(s') \cdot 1 + \gamma^{t-1} \sum_v T(u,v)R(u,v))$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} p(s')(R(s') + \gamma^{t-1} \sum_v T(u,v)R(u,v))$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} \left(p(s')R(s') + p(s')\gamma^{t-1} \sum_v T(u,v)R(u,v) \right)$$
$$E_r(t) = \sum_u \sum_{s' \in S_{t-1,u}} p(s')R(s') + \gamma^{t-1} \sum_u \sum_{s' \in S_{t-1,u}} p(s') \sum_v T(u,v)R(u,v)$$
$$E_r(t) = \sum_{s' \in S_{t-1}} p(s')R(s') + \gamma^{t-1} \sum_u \sum_{s' \in S_{t-1,u}} p(s') \sum_v T(u,v)R(u,v)$$
$$E_r(t) = E_r(t-1) + \gamma^{t-1} \sum_u \sum_{s' \in S_{t-1,u}} p(s') \sum_v T(u,v)R(u,v)$$
$$E_r(t) = E_r(t-1) + \gamma^{t-1} \sum_u \left(\sum_v T(u,v)R(u,v) \cdot \sum_{s' \in S_{t-1,u}} p(s') \right)$$
$$E_r(t) = E_r(t-1) + \gamma^{t-1} \sum_u \sum_v T(u,v)R(u,v) \cdot L(u,t)$$

Dodatno možemo primetiti sledeće:

$$\sum_u \sum_v T(u,v)R(u,v) \cdot L(u,t) = \sum_u L(u,t) \sum_v T(u,v)R(u,v)$$

Ukoliko definišemo $L(t)$ kao vektorsku funkciju, tako da je $L(t)_u$ verovatnoća da se u trenutku t bude u stanju u i primetimo da je $\sum_v T(u,v)R(u,v)$ konstantan vektor $\vec{c}$. Možemo zapisati izraz kao:

$$E_r(t) = E_r(t-1) + \gamma^{t-1} \vec{L(t)} \cdot \vec{c}$$

Pa je lako videti da je:

$$E_r(t) = \sum_{i=1}^{t} \gamma^{i-1} \vec{L(i)} \cdot \vec{c}$$

$$E_r(t) = \vec{c} \cdot \sum_{i=1}^{t} \gamma^{i-1} \vec{L(i)}$$

Za aciklične markovljeve lance, odnosno za aperiodične matrice T izračunavanje $E_r(t)$ za dovoljno veliko t je dovoljno za nalaženje očekivane nagrade. Međutim za periodične matrice T je potrebno pokazati da $\lim_{t\to\infty} E_r(t)$ postoji.

Lema 2.3. $\lim_{t\to\infty} E_r(t) = \text{const}$
Ako se u izraz $E_r(t) = \vec{c} \cdot \sum_{i=1}^{t} \gamma^{i-1} L(\vec{i-1})$ zameni definicija za $\vec{L(i)}$ dobija se sledeći izraz:

$$E_r(t) = \vec{c} \cdot \sum_{i=1}^{t} \gamma^{i-1} \vec{L_0} T^{i-1}$$

$$E_r(t) = \vec{c} \cdot \left(\vec{L_0} \cdot \sum_{i=1}^{t} \gamma^{i-1} T^{i-1} \right)$$

Uvedimo matricu $A = \gamma T$. Pa je prethodni izraz jednak:

$$E_r(t) = \vec{c} \cdot \left(\vec{L_0} \cdot \sum_{i=1}^{t} A^{i} \right)$$

Pa je potrebno naći $\lim_{t\to\infty} E_r(t) = \vec{c} \cdot \left(\vec{L_0} \cdot \sum_{t=0}^{\infty} A^t \right)$.
$\sum_{t=0}^{\infty} A^t$ je Nojmanov red primenjn na prostor R^n.
Dovoljan uslov za konvergenciju Nojmanovog reda je da postoji norma tako da važi $||A|| < 1$.
Najlakše je odabrati $||A||_\infty$. Pa je lako videti da:

$$||A||_\infty = \max_i \sum_j |A_{i,j}|$$

$$||A||_\infty = \gamma < 1$$

Pa znamo da suma konvergira, dodatno lako je pokazati da za sumu $S = \sum_{i=0}^{t} A^i$ važi $S(I - A) = I - A^{t-1}$.
Pa ukoliko postoji inverz $(I - A)^{-1}$ (što postoji kada Nojmanov red konvergira) sledi da $S = (I - A^{t-1})(I - A)^{-1}$.
Kada $t \to \infty$ onda je $A^{t-1} = 0$. Pa je $S = I(I - A)^{-1} = (I - A)^{-1}$.
Čime je dokazano da $\lim_{t\to\infty} E_r(t) = \vec{c} \cdot \left(\vec{L_0} \cdot (I - A)^{-1} \right)$.

2.3 Optimizacija politike (policy iteration)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in ante eget massa tincidunt gravida. Sed sit amet eros quis lacus tempor lobortis. Maecenas ut congue sapien. Nunc viverra magna nec convallis maximus. Praesent sed accumsan ante. Ut tincidunt, orci sed malesuada imperdiet, erat neque vestibulum eros, a porta ligula diam at nunc. Fusce at aliquam est. Morbi bibendum erat ut rutrum gravida. Cras tincidunt fermentum ex id condimentum. In scelerisque ut turpis eget faucibus. Vestibulum sollicitudin lacus cursus lacinia ultricies. Cras urna eros, venenatis ut enim ut, lacinia sagittis tortor. Phasellus facilisis viverra finibus. Vestibulum tincidunt fringilla purus ac condimentum. Sed sed egestas eros, ac mattis purus. Curabitur fermentum tempor odio eget eleifend.

2.4 Optimizacija vrednosti (value iteration)

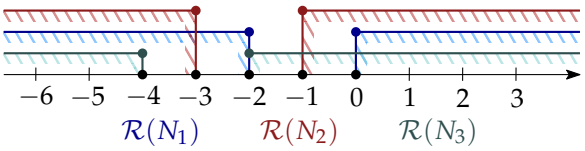
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in ante eget massa tincidunt gravida. Sed sit amet eros quis lacus tempor lobortis. Maecenas ut congue sapien. Nunc viverra magna nec convallis maximus. Praesent sed accumsan ante. Ut tincidunt, orci sed malesuada imperdiet, erat neque vestibulum eros, a porta ligula diam at nunc. Fusce at aliquam est. Morbi bibendum erat ut rutrum gravida. Cras tincidunt fermentum ex id condimentum. In scelerisque ut turpis eget faucibus. Vestibulum sollicitudin lacus cursus lacinia ultricies. Cras urna eros, venenatis ut enim ut, lacinia sagittis tortor. Phasellus facilisis viverra finibus. Vestibulum tincidunt fringilla purus ac condimentum. Sed sed egestas eros, ac mattis purus. Curabitur fermentum tempor odio eget eleifend.

Glava 3

Uvodni pojmovi

Definicija 3.1. Ovo je definicija.

Definicija 3.2. Ovo je *druga* **definicija**.



SlikaProba slike, sve je c1irilicom

3.1 Topološki prostori

Definicija 3.3. (1) Ovako se piše takozvana *inline* formula, $X \in \mathcal{O}$, koja se uklapa u tekst.

(2) A ovako se piše formula koja je u zasebnom redu

$$ax + by = c, \sum_{n=1}^{\infty} \frac{1}{n(n+1)} = \dots$$

(3) Ukoliko želite da jedna formula ima više redova, jedan od načina da to postignete je

$$\begin{aligned} S &= punoformula, punoformula \\ &= 2. \end{aligned}$$

3.2 Puno ime

Definicija 3.4. *Simpleks*

Teorema 3.1. Za svako $n \geqslant 0$, sledec1a tvrđenja su ekvivalentna i tačna:

3.3 Borsukov graf

Teorema 3.1. Za pozitivan realan broj $\alpha < 2$, neka je $B(n+1, \alpha)$ (beskonačan)

Dokaz. Pretpostavimo da važi teorema Borsuk-Ulama i □

Literatura

Campbell, M., Hoane Jr, A. J., & Hsu, F.-h. (2002). Deep blue. *Artificial intelligence*, *134*(1-2), 57–83.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, *550*(7676), 354–359.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354.