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

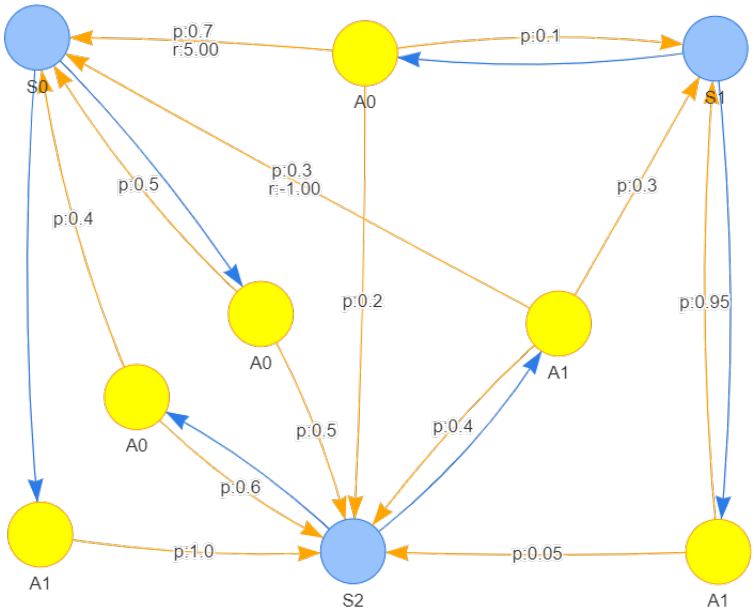
U narednom delu će biti razmatrane varijante Markovljevog procesa odlučivanja i rešenja koja su moguća za te varijante, pa je potrebno definisati Markovljev proces odlučivanja i uvesti potrebnu notaciju. Markovljevi procesi odlučivanja su uopštenje Markovljevih lanaca sa dodatim akcijama, za definiciju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

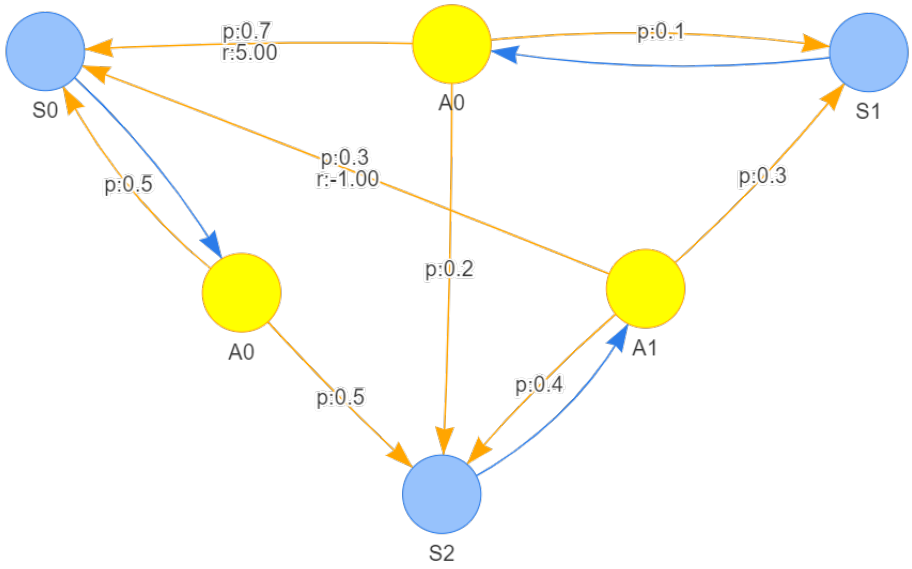
Rešenje Markovljevog procesa odlučivanja je optimalna funkcija π^*:

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

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



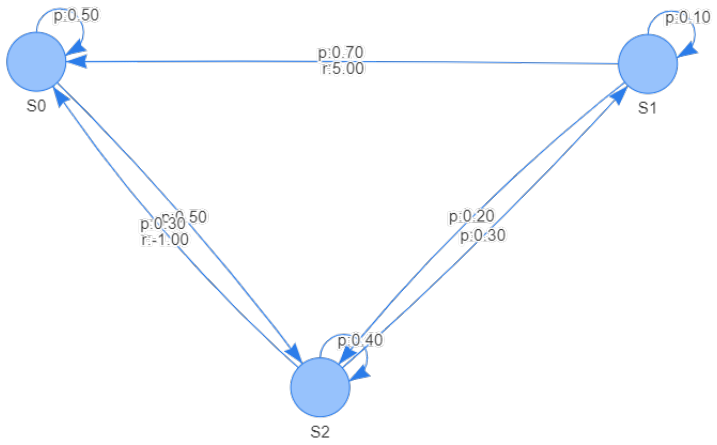
Slika 1: Primer Markovljevog procesa odlučivanja



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 $\lim_{t\to\infty} E_r(t) = \vec{c} \cdot \left(\vec{L_0} \cdot (I-A)^{-1}\right)$.

Korisno je napomenuti da postoji još jedan značajan način da se pronađe očekivana nagrada koju neka politika ostvarujue, a to je pomoću takozvanog *belmanovog operatora očekivanja*. Može se definisati očekivana vrednost za politiku π iz stanja s kao $V^\pi(s)$. Funkcija V se naziva funkcija vrednosti stanja. Funkcije stanja mogu se posmatrati kao vektori u banahovom prostoru $\mathbb{R}^{|S|}$ gde je S skup svih mogućih stanja. Na ovom prostoru se definiše i norma, na primer maksimum norma $|v|_\infty$. Pa se zatim može uvesti belmanov operator očekivanja za stacionarnu determinističku politiku π:

$$B^\pi(V)_s = \sum_{s'} T(S, \pi(S), S')(R(S, \pi(S), S') + \gamma V(S'))$$

ili slično i za nedeterminističke politike. Pa se može pokazati (Puterman, 2014) da je belmanov operator očekivanja kontrakcija na datom prostoru. Pa postoji stacionarna tačka $B^\pi(x) = x$. Dokazuje se (Puterman, 2014) da je tačka x upravo V^π. Sličan dokaz će biti izložen za algoritam optimizacije vrednosti.

2.3 Optimizacija politike (policy iteration)

Može se pokazati (Puterman, 2014) da je optimalna politika π^* stacionarna (ne zavisi od vremena) i deterministička, pa će samo takve politike biti razmatrane nadalje.
da je broj mogućih stacionarnih, determinističkih politika $|A|^{|S|}$ gde je A skup mogućih akcija, a S skup mogućih stanja. Pa je jedan način za pronalaženje optimalne politike evaluacija svake moguće politike pomoću prethodno izvedene metode. Međutim, moguće je pronaći optimalnu politiku efikasnije algoritmom optimizacije funkcije vrednosti (tj. value iteration).

2.4 Optimizacija vrednosti (value iteration)

Uvedimo notaciju za očekivanu vrednost koja se može ostvariti iz stanja (funkciju vrednosti stanja) s ako se prati neka stacionarna, deterministička politika π:

$$V^\pi(s) = \sum_{s'} T(s, \pi(s), s')R(s, \pi(s), s') + \gamma V^\pi(s')$$

Primetimo da sa obzirom da je funkcija V diskretna, a broj stanja konačan možemo reći da se sve funkcije vrednosti stanja nalaze u banahovom prostoru $R^{|S|}$ gde je S skup svih mogućih stanja. Dodatno u ovom prostoru uzećemo beskonačno normu $||v||_\infty := \max(|v_1|, |v_2|, ..., |v_n|)$. Algoritam optimizacije funkcije vrednosti stanja nalazi funkciju stanja $V^*(s)$ koja predstavlja maksimalnu moguću očekivanu vrednost koja se može ostvariti is stanja s ako se i u narednim stanjima izvršavaju optimalne akcije. Kasnije će biti pokazano da V^* odgovara stacionarnoj, determinističkoj, optimalnoj politici π^*. U svrhu nalaženja V^* uvedimo takozvani *belmanov optimizacioni operator* B^* koji deluje na vektore iz datog banahovog prostora:

$$B^*(V)_s = \max_a \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V_{s'})$$

Algoritam optimizacije funkcije vrednosti stanja uključuje iterativnu primenu belmanovog operatora na početnu funkciju vrednosti stanja (tj. na početni vektor). Kako bi pokazali da algoritam konvergira i da konvergira upravo na vrednost V^* biće dokazano da je operator B^* kontrakcija na datom banahovom prostoru.
Prema banahovoj teoremi fiksne tačke ukoliko je B^* kontrakcija na prostoru $R^{|S|}$ onda postoji vektor x^* takav da važi $B^*(x^*) = x^*$ i dodatno za sve ostale vektore u tom prostoru važi da ukoliko se definiše red $x_n = B^*(x_{n-1})$ onda je $\lim_{n\to\infty} x_n = x^*$. Na kraju potrebno je još pokazati da je $x^* = V^*$, odnosno da algoritam konvergira na optimalnu vrednost.

Teorema 2.1. $(\forall U \in R^{|S|}, \forall V \in R^{|S|}, U \neq V)||B^*(V) - B^*(U)||_\infty \leq k||V - U||_\infty$ gde $k \in [0, 1)$
Po definiciji je:

$$B^*(V)_s = \max_a \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V_{s'})$$

Pošto je skup akcija konačan, postoji akcija a_1 koja maksimizuje datu sumu, pa se može zapisati:

$$B^*(V)_s = \sum_{s'} T(s, a_1, s')(R(s, a_1, s') + \gamma V_{s'})$$

Analogno za U_s

$$B^*(U)_s = \sum_{s'} T(s, a_2, s')(R(s, a_2, s') + \gamma U_{s'})$$

Pošto akcija a_2 maksimizuje prethodnu sumu važi sledeća nejednakost:

$$B^*(U)_s \geq \sum_{s'} T(s, a_1, s')(R(s, a_1, s') + \gamma U_{s'})$$

Pa za razliku $B^*(V)_s - B^*(U)_s$ važi nejednakost:

$$B^*(V)_s - B^*(U)_s \leq \sum_{s'} T(s, a_1, s')(R(s, a_1, s') + \gamma V_{s'}) - \sum_{s'} T(s, a_1, s')(R(s, a_1, s') + \gamma U_{s'})$$

$$B^*(V)_s - B^*(U)_s \leq \sum_{s'} (T(s, a_1, s')(R(s, a_1, s') + \gamma V_{s'} - R(s, a_1, s') - \gamma U_{s'}))$$

$$B^*(V)_s - B^*(U)_s \leq \gamma \sum_{s'} (T(s, a_1, s')(V_{s'} - U_{s'}))$$

Pošto je $|x| \geq x$, važi:

$$B^*(V)_s - B^*(U)_s \leq \gamma \sum_{s'} T(s, a_1, s')|V_{s'} - U_{s'}|(a)$$

Primetimo da ukoliko razmenimo vrednosti U i V dobija se sledeća nejednakost:

$$B^*(U)_s - B^*(V)_s \leq \gamma \sum_{s'} T(s, a_1, s')|V_{s'} - U_{s'}|(b)$$

Iz a, b sledi:

$$|B^*(V)_s - B^*(U)_s| \leq \gamma \sum_{s'} T(s, a_1, s')|V_{s'} - U_{s'}|$$

Ukoliko se $|V_{s'} - U_{s'}|$ zameni sa $\max_{s''}|V_{s''} - U_{s''}|$ desna strana nejednakosti se može samo povećati pa važi nejednakost:

$$|B^*(V)_s - B^*(U)_s| \leq \gamma \sum_{s'} T(s, a_1, s')\max_{s''}|V_{s''} - U_{s''}|$$

$$|B^*(V)_s - B^*(U)_s| \leq \gamma \max_{s''}|V_{s''} - U_{s''}|$$

$$|B^*(V)_s - B^*(U)_s| \leq \gamma ||V - U||_\infty$$

Primetimo da nejednakost $|B^*(V)_s - B^*(U)_s| \leq \gamma ||V - U||_\infty$ važi za svako U, V i svako stanje s, što je dovoljan uslov za:

$$||B^*(V) - B^*(U)||_\infty \leq \gamma ||V - U||_\infty$$

A kako je $\gamma < 1$ teorema je dokazana.

Lema 2.4. $(\forall s)(\forall V \in \mathbb{R}^{|S|})B^*(V)_s \geq B^\pi(V)_s$
Za stacionarne, determinističke politike je dokaz jednostavan, pa je izložen dokaz za stacionarne nedeterminističke politike koje su nadskup stacionarnih, determinističkih politika. Po definiciji operatora B^* važi:

$$B^*(V)_s = \max_a \gamma \sum_{s'} T(s, a, s')(R(s, a, s') + V_{s'}) \geq \gamma \sum_{s'} T(s, a, s')(R(s, a, s') + V_{s'})(\forall a)$$

Ako se obe strane nejednakosti pomnože sa $\pi(a|s)$ i sumiraju po svakom a dobija se sledeća nejednakost:

$$\sum_a \pi(a|s)B^*(V)_s \geq \gamma \sum_a \pi(a|s) \sum_{s'} T(s,a,s')(R(s,a,s') + V_{s'})(\forall a)$$
$$\sum_a \pi(a|s)B^*(V)_s \geq \gamma \sum_a \sum_{s'} \pi(a|s)T(s,a,s')(R(s,a,s') + V_{s'})(\forall a)$$

Pošto je definicija $B^\pi(V)_s = \gamma \sum_a \sum_{s'} \pi(a|s)T(s,a,s')(R(s,a,s') + V_{s'})$ i kako je $\sum_a \pi(a|s)B^*(V)_s = B^*(V)_s$. Dokazano je da $B^*(V)_s \geq B^\pi(V)_s$.

Lema 2.5. Operator B^* optimizuje funkciju vrednosti stanja, odnosno nakon primene B^* funkcija vrednosti stanja se ne može pogoršati:
$(\forall V)(\forall s)B^*(V)_s \geq V_s$
Uzmimo politiku π tako da je V stacionarna tačka B^π. Treba dokazati:

$$B^*(V)_s \geq V_s$$

A pošto je $(\forall s)B^\pi(V)_s = V_s$ nejednakost je ekvivalentna:

$$B^*(V)_s \geq B^\pi(V)_s$$

Što važi na osnovu leme 2.4 čime je dokaz kompletiran.

Lema 2.6. Stacionarna tačka V^* operatora B^* je optimalna funkcija vrednosti, odnosno: $(\forall V)(\forall s)V^*_s \geq V_s$
Pretpostavimo suprotno: $(\exists v)(\exists s)V^*_s < V_s$
Na osnovu monotonosti operatora B^* može se primeniti operator B^* na desnu stranu nejednakosti proizvoljan broj puta:

$$V^*_s < B^*(V)_s$$
$$V^*_s < \lim_{k\to\infty} (B^*)^k(V)_s$$

Pošto je B^* kontrakcija, a V^* stacionarna tačka za B^* važi:

$$\lim_{k\to\infty} (B^*)^k(V)_s = V^*_s$$

Pa je nejednakost ekvivalentna:

$$V^*_s < V^*_s$$

Što je kontradikcija, čime je dokazana početna lema.

Ovim je pokazano da je moguće pronaći optimalnu funkciju vrednosti stanja počevši od bilo koje funkcije vrednosti stanja i iterativnom primenom belmanovog optimizacionog operatora. Još preostaje da se dokaže da postoji stacionarna, deterministička politika koja dostiže očekivanu vrednost optimalne funkcije stanja.

Teorema 2.2. U proizvoljnom markovljevom procesu odlučivanja politika:

$$\pi^*(s) = \text{argmax}_a \left(\sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V^*_{s'}) \right)$$

je stacionarna, deterministička politika čija je očekivana nagrada $(\forall s) V_s^{\pi^*} = V_s^*$, politika π^* je optimalna.
Kako je V^* stacionarna tačka B^* važi $(\forall s) B^*(V^*)_s = V^*_s$.

$$V^*_s = B^*(V^*)_s$$
$$V^*_s = \max_a \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V^*_{s'})$$
$$V^*_s = \sum_{s'} T(s,\pi^*(s),s')(R(s,\pi^*(s),s') + \gamma V^*_{s'})$$

Kako je $V_s^{\pi^*} = \sum_{s'} T(s,\pi^*(s),s')(R(s,\pi^*(s),s') + \gamma V_{s'}^{\pi^*})$, a iz definicije π^* je $V_{s'}^{\pi^*} = V^*_{s'}$ dokazano je da:

$$(\forall s) V^*_s = V_s^{\pi^*}$$

Zajedno sa metodom za pronalaženje optimalne funkcije stanja V^* ova teorema daje efikasan način da se pronađe optimalna politika odlučivanja π^*, što upotpunjuje algoritam optimizacije vrednosti.

2.5 Stohastička aproksimacija funkcije vrednosti stanja

U kontekstu primene markovljevih procesa odlučivanja potrebno je razmatrati i procese u kojima verovatnoće nisu unapred poznate. Na primer, za primenu belmanovog optimizacionog operatora B^* potrebno je znati vrednosti $T(s,a,s')$ i $R(s,a,s')$ za svako s, a i s'. U praksi ove vrednosti najčešće nisu poznate, a kasnije će se razmatrati i procesi gde je broj stanja previše velik da bi se te vrednosti uopšte čuvale ili sračunale. Pošto nije moguće znati prave vrednosti, neophodno je aproksimirati verovatnoću. Pošto je belmanov operator očekivanja:

$$B^\pi(V)_s = \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V_{s'})$$

Primetimo da se data suma može razdvojiti:

$$B^\pi(V)_s = \sum_{s'} T(s,a,s')R(s,a,s') + \sum_{s'} T(s,a,s')\gamma V_{s'}$$

Dodatno, primetimo da je $\sum_{s'} T(s,a,s')R(s,a,s')$ očekivana vrednost nagrade iz stanja s, kada se izvrši akcija a, odnosno: $\sum_{s'} T(s,a,s')R(s,a,s') = \mathbb{E}(R|s,a)$. Slično, suma $\sum_{s'} T(s,a,s')V_{s'}$ je očekivana vrednost buduće nagrade, odnosno: $\sum_{s'} T(s,a,s')\gamma V_{s'} = \gamma\mathbb{E}(V|s,a)$. Pa se belmanov operator može ekvivalentno zapisati kao:

$$B^{\pi}(V)_s = \max_a \left(\mathbb{E}(R|s,a) + \gamma\mathbb{E}(V|s,a)\right)$$

U cilju aproksimacije potrebnih očekivanih vrednosti biće dokazane sledeće leme:

Lema 2.7. Za red $E_n = \frac{1}{n+1}(nE_{n-1} + x_n), E_0 = x_0$ važi $E_n = \frac{1}{n+1}\sum_{i=0}^{n} x_i$
Lema će biti dokazana indukcijom. Induktivna hipoteza je: $E_n = \frac{1}{n+1}\sum_{i=0}^{n} x_i$. Baza je $E_0 = \frac{1}{1}\sum_{i=0}^{0} x_i = x_0$, a po definiciji reda je $E_0 = x_0$ čime je baza indukcije dokazana.
Uzmimo da važi tvrdnja za $n-1$: $E_{n-1} = \frac{1}{n}\sum_{i=0}^{n-1} x_i$. Po definiciji je $E_n = \frac{1}{n+1}(nE_{n-1} + x_n)$. Ako se zameni vrednost za E_{n-1}, dobija se jednakost:

$$E_n = \frac{1}{n+1}(n\frac{1}{n}\sum_{i=0}^{n-1} x_i + x_n)$$
$$E_n = \frac{1}{n+1}(\sum_{i=0}^{n-1} x_i + x_n)$$
$$E_n = \frac{1}{n+1}\sum_{i=0}^{n} x_i$$

Čime je dokazana lema.

Lema 2.8. Neka je definisan red: $E_n = E_{n-1} + \frac{1}{n+1}(x_n - E_{n-1}), E_0 = x_0$, onda je $\lim_{n\to\infty} E_n = \mathbb{E}(X)$
Prvo je dokazana ekvivalencija sa jednostavnijim redom:

$$E_n = E_{n-1} + \frac{1}{n+1}(x_n - E_{n-1})$$
$$E_n = E_{n-1} + \frac{x_n}{n+1} - \frac{E_{n-1}}{n+1}$$
$$E_n = \frac{n}{n+1}E_{n-1} + \frac{x_n}{n+1}$$
$$E_n = \frac{1}{n+1}(nE_{n-1} + x_n)$$

Pa zajedno sa prethodnom lemom sledi:

$$E_n = \frac{1}{n+1}\sum_{i=0}^{n} x_i$$

Primetimo da je izraz $\frac{1}{n+1}\sum_{i=0}^{n} x_i$ srednja vrednost uzorka $x_0, x_1, ..., x_n$. Odnosno:

$$E_n = \frac{1}{n+1}\sum_{i=0}^{n} x_i = \overline{X_n}$$

Pa se početno tvrđenje svodi na:

$$\lim_{n\to\infty} \overline{X_n} = \mathbb{E}(X)$$

Što je tvrdnja zakona velikih brojeva.

Algoritam određivanja funkcije vrednosti stanja pod politikom π se može formulisati na sledeći način:

$$V_s \leftarrow \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V_{s'})$$

Ako je data epizoda generisana politikom π: $s_0, a_0, r_0, s_1, a_1, r_1, ..., s_n, a_n, r_n$. Na osnovu date epizode, motivisani prethodnim lemama možemo formulisati stohastičku verziju prethodnog algoritma:

$$V_{s_t} \leftarrow V_{s_t} + \alpha_t(R_t + \gamma V_{s_{t+1}} - V_{s_t})$$

Formalnije je dokazano (Sutton and Barto, 2018) da i stohastička verzija konvergira na pravu vrednost V^π.

2.6 Učenje Q vrednosti

Slično kao funkcija vrednosti stanja, može se definisati funkcija očekivane nagrade $Q^\pi(s,a)$ koja predstavlja očekivanu nagradu ako se u stanju s izvrši akcija a i nadalje deluje po politici π. Veoma slično kao za funkciju vrednosti stanja V^π, za Q funkciju se pokazuje da važi:

$$Q^\pi(s,a) = \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma Q(s', \pi(s)))$$

Postoji optimalna Q funkcija $Q^*(s,a)$ koja predstavlja maksimalnu očekivanu nagradu koja se može dobiti ako se u stanju s izvrši akcija a i nadalje deluje optimalno.

$$Q^*(s,a) = \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma \max_{a'} Q^*(s',a'))$$

Ako je optimalna politika π^* važi:

$$V^* = V^{\pi^*}(s) = \max_a \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V^{\pi^*}(s'))$$

Sa obzirom da su Q funkcija i V funkcija povezane sledećom jednakosti:

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s,a)$$

Za politiku π^* važi:

$$V^{\pi^*}(s) = \max_a Q^{\pi^*}(s,a)$$

Pa se jednačina za Q^* može zapisati ekvivalentno na sledeći način:

$$Q^*(s,a) = \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V^{\pi^*}(s'))$$

Stoga je jasno da ako su date vrednosti optimalne funkcije V^* odmah su date i vrednosti optimalne Q funkcije Q^*. Što opravdava uvođenje algoritma optimizacije Q vrednosti koji je analogan prethodnom algoritmu učenja optimalne vrednosti stanja V^*.

$$Q(s,a) \leftarrow \sum_{s'} T(s,a,s')(R(s,a,s') + \gamma \max_{a'} Q(s',a'))$$

Pa zatim i stohastičku verziju tog algoritma:
Ako je data epizoda: $E : s_0, a_0, r_0, s_1, a_1, r_1, ..., s_n, a_n, r_n$
Onda je u iteraciji n, za svako s i svako a:

$$Q_n(s,a) = \begin{cases} (1-\alpha_n) Q_{n-1}(s,a) + \alpha_n (r_n + \gamma \max_{a'} Q_{n-1}(s',a')), & \text{ako } s_n = s, a_n = a \\ Q_{n-1}(s,a), & \text{u suprotnom} \end{cases}$$

Kada $n \to \infty$ dokazano je da $Q_n(s,a) \to Q^*(s,a)$ (Watkins and Dayan, 1992) (Sutton and Barto, 2018) pod uslovima:

- Kada $n \to \infty$ svako stanje i svaka akcija će se pojaviti u sekvenci E beskonačno puta, odnosno: $(\forall s)(\forall a) p(s_n = s, a_n = a) > 0$.
- $(\forall s)(\forall a) \sum_i^\infty \alpha_{n_i(s,a)} = \infty$, ali $\sum_i^\infty \left(\alpha_{n_i(s,a)}\right)^2 < \infty$ gde je $n_i(s,a)$ pozicija u sekvenci E kada je akcija a izvršena u stanju s.
- Nagrade su konačne

Literatura

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3), 279–292.

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

Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

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.

Sutton, R., & Barto, A. (2018). *Reinforcement learning, second edition: An introduction*. MIT Press. https://books.google.rs/books?id=uWV0DwAAQBAJ

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.