SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR 1

Podržano učenje

Ines Kovač

Zagreb, siječanj 2023.

Sadržaj

1. Uvod

Podržano učenje je jedna od najpopularnijih istraživačkih tema u području moderne umjetne inteligencije te njena popularnost samo raste. Zasniva se na ideji učenja iz interakcija s okolinom koja nas okružuje. Ostvarivanjem veze s okolinom proizvodi se mnoštvo informacija o vezi uzrok - posljedica, o posljedicama akcija i o tome što učiniti kako bi se postigli ciljevi. Tijekom našeg života ovakve interakcije su nedvojbeno veliki izvor znanja o našoj okolini, ali I o nama samima.

Podržano učenje je jedno od područja strojnog učenja koje je usredotočeno na inteligentnog agenta I kako on poduzima akcije u svojem okruženju s ciljem maksimiziranja kumulativne nagrade. Podržano učenje jedno je od tri osnovne paradigme strojnog učenja uz nadzirano učenje I nenadzirano učenje.

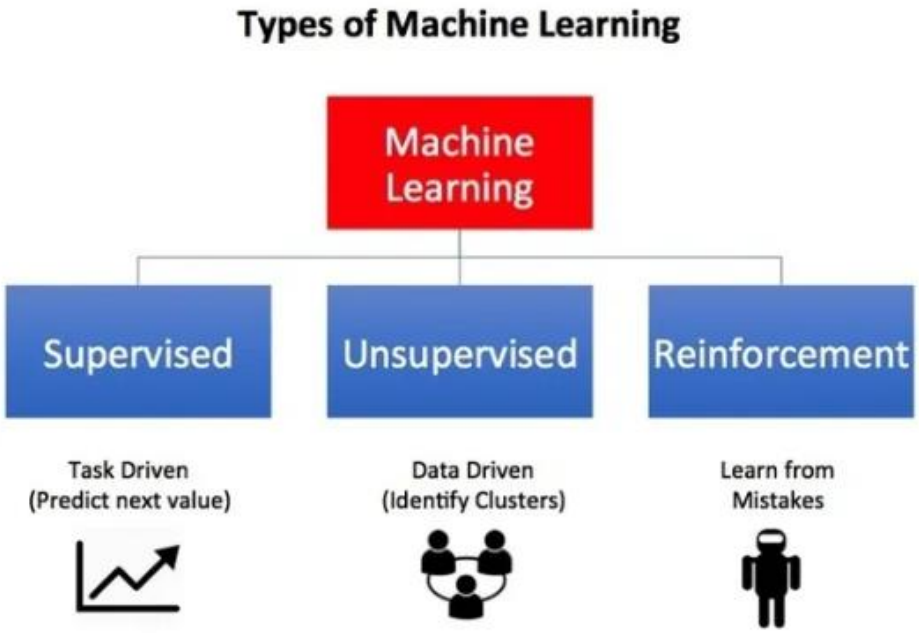

Figure 1 Tipovi strojnog učenja

Podržano učenje razlikuje se od nadziranog učenja po tome što za ovaj oblik učenja nisu potrebni označeni ulazno/izlazni parovi podataka I po tome što ne zahtjeva eksplicitno ispravljanje podakcija koje nisu optimalne. Umjesto toga, fokus je na pronalaženju ravnoteže između *istraživanja* (nepoznatog teritorija) I *iskorištavanja* (trenutačnog znanja).

U usporedbi s nenadziranim učenjem, podržano učenje razlikuje se u pojmu ciljeva. Dok je cilj nenadziranog učenja pronaći sličnosti i razlike između ulaznih podataka, u slučaju podržanog učenja, cilj je pronaći odgovarajući akcijski model koji bi maksimizirao ukupnu kumulativnu

nagradu agenta. Slika u nastavku ilustrira petlju povratne sprege akcija - nagrada generičkog RL modela.

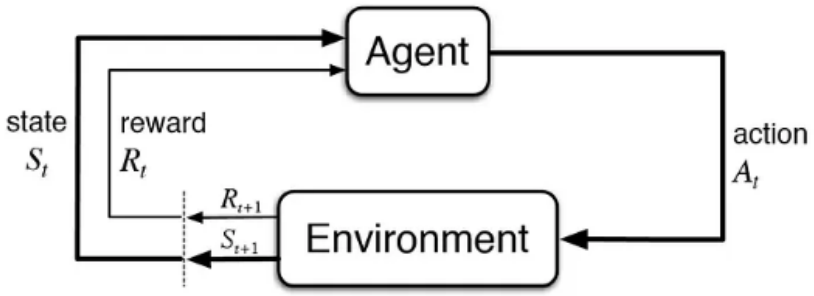

Figure 2 Tok akcija, stanja i nagrada

Podržano učenje je učenje što učiniti, kako mapirati situacije I akcije kako bi se maksimizirala numerička vrijednost signala nagrade. Agentu, "učeniku", se ne govori koje akcije treba poduzeti, već mora otkriti koje radnje donose najveću nagradu tako što će ih isprobati.

1.2. Elementi Podržanog učenja

Uz agenta I njegovo okruženje postoje jos četiri glavna podelementa sustava podržanog učenja, to su: *politika*, *signal nagrade, funkcija vrijednosti* I, opcionalno, *model* okoliša.

Politika definira način ponašanja agenta u određenom trenutku. Grubo govoreći, politika je preslikavanje percipiranih stanja okoliša na akcije koje treba poduzeti u tim stanjima. Odgovara onome što bi se u psihologiji nazvalo skupom pravila ili asocijacija stimulus – odgovor. Ponekad je politika jednostavna funkcija ili *lookup tabica,* dok u drugim slučajevima može uključivati opsežna izračunavanja kao što je proces pretraživanja. Politika je srž agenta u smislu da je sama dovoljna za određivanje ponašanja agenta. Općenito, politike mogu biti stohastičke, određujući vjerojatnosti za svaku akciju.

Definicija 1. Politika je funkcija koja definira ponašanje agenta, $\pi : S \rightarrow A$

- *Deterministička politika $\pi(s) = a$*
- *Stohastička politika $\pi(s|a) = P(A_t = a \mid S_t = s)$*

Signal nagrade definira cilj problema podržanog učenja. Na svakom vremenskom koraku, okruženje šalje agentu jedan *broj* koji se naziva nagrada. Agentov jedini cilj je maksimalno povećati ukupnu nagradu koju dugoročno ostvaruje. Signal nagrade tako definira koji su dobri i loši događaji za agenta. U biološkom sustavu, nagrade bismo mogli smatrati analognima doživljajima zadovoljstva ili boli. Oni su neposredna i definirajuća obilježja problema s kojim se agent suočava. Signal nagrade primarna je osnova za promjenu politike; ako je akcija odabrana

politikom praćena niskom nagradom, tada se politika može promijeniti da odabere neku drugu akciju u toj situaciji u budućnosti. Općenito, signali nagrade mogu biti stohastičke funkcije stanja okoliša i poduzetih akcija.

Funkcija vrijednosti određuje što je dobar odabir dugoročno, za razliku od *signala nagrade* koji pokazuje što je dobro u neposrednom smislu. Grubo govoreći, *vrijednost stanja* je ukupan iznos nagrade koju agent može očekivati da će akumulirati u budućnosti, počevši od tog stanja. Dok *nagrade* određuju neposrednu, intrinzičnu poželjnost stanja okoliša, *vrijednosti* ukazuju na dugoročnu poželjnost stanja nakon uzimanja u obzir stanja koja će vjerojatno uslijediti i nagrada dostupnih u tim stanjima. Na primjer, stanje može uvijek donijeti nisku trenutnu *nagradu*, ali još uvijek ima visoku *vrijednost* jer ga redovito slijede druga stanja koja donose visoke *nagrade*, ili obrnuto. Analogno, nagrade su nešto poput zadovoljstva (ako je veliko) i boli (ako je nisko), dok vrijednosti odgovaraju profinjenijoj i dalekovidnijoj prosudbi o tome koliko smo zadovoljni ili nezadovoljni stanjem našeg okruženja.

Dakle, nagrade su primarne, dok su vrijednosti, kao predviđanja nagrada, sekundarne. Bez nagrada ne bi moglo biti vrijednosti, a jedina svrha procjenjivanja vrijednosti je postizanje veće nagrade. Ipak, vrijednosti su ono što nas najviše zanima kada donosimo i procjenjujemo odluke. Izbori akcija donose se na temelju vrijednosnih prosudbi. Tražimo akcije koje donose stanja najveće vrijednosti, a ne najveće nagrade, jer te akcije dugoročno donose najveću količinu nagrade. Nažalost, puno je teže odrediti vrijednosti nego nagrade. Nagrade nam u osnovi okruženje daje izravno, a vrijednosti se moraju procijeniti i ponovno procijeniti iz sekvenci opažanja koje agent čini tijekom čitavog životnog vijeka. Zapravo, najvažnija komponenta gotovo svih algoritama podržanog učenja je metoda za učinkovitu procjenu vrijednosti. Središnja uloga procjene vrijednosti vjerojatno je najvažnija stvar koja je naučena o podržanom učenju tijekom posljednjih nekoliko desetljeća.

2. Konačni Markovljevi lanci odlučivanja

Markovljevi lanci odlućivanja, kraće MDP, su klasična formalizacija sekvencijalnog donošenja odluka, gdje radnje utječu ne samo na trenutne nagrade, već I na sljedeće situacije ili stanja, a preko njih I na buduće nagrade. Stoga MDP-ovi uključuju odgođenu nagradu I potrebu balansiranja trenutne I odgođene nagrade. Kod MDP-a procjenjujemo vrijednost *q(s, a)* svake akcije *a* u svakom stanju *s*. Ove komponente ovisne o stanju su esencijalne za dodjeljivanje nagrada za dugoročne posjedice pojednačnim odabranim akcijama.

Markovljevi lanci odlučivanja su idealzirane matematičke forme paradigme podržanog učenja za koje se mogu napraviti precizni teorijski izrazi. Ključni elementi matematičke strukture problema su *eng. returns*, *funkcije vrijednosti (value function) I Bellmanove jednadžbe.* Postoji veliki raspon primjena koje se mogu formulirati kao MDP-ovi te kao I u cijeloj umjetnoj inteligenciji, postoji konflikt između širine primjenjivosti I matematičke traktabilnosti.

U konačnom MDP-u svi skupovi stanja *S*, akcija *A* I nagrada *R* imaju konačan broj elemenata. Za određene vrijednosti tih slučajnih varijabli, $s' \in S$ i $r \in R$, postoji vjerojatnost da se te vrijednosti pojave u trenutku t u ovisnosti na određene vrijednosti prethodnog stanja I akcije:

$$p(s', r \mid s, a) = Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$$

za sve $s', s \in S, r \in R$ i $a \in A(s)$. Funkcija *p* definira dinamiku MDP-a i ona je obična deterministička funkcija od četiri elementa. Vjerojatnost svake moguće vrijednostii S_t i R_t ovise samo o neposredno prethodnom stanju S_{t-1} i radnji R_{t-1} I ne ovise o ranijim stanjima i akcijama. Ovo je najbolje shvatiti kao restrikciju stanja, a ne proseca odlučivanja. Ako stanje uključuje informacije o svim aspektima prijašnjih interakcija agenta I okoline koje utječu na budućnost, kažemo da to stanje ima *Markovljevo svojstvo*.

2.2. Hipoteza nagrade

Sve što podrazumijevamo pod ciljevima I svrhama možemo shvatiti kao maksimiziranje očekvane vrijednosti kumulativnog zbroja prikupljenog skalarnog signala nagrade.

Korištenje signala nagrade za formaliziranje ideje cilja jedno je od najuzrazitijih obilježja podržanog učenja. Agent uvijek uči maksimizirati svoju nagradu. Ako želimo da učini nešto za nas, moramo mu osigurati nagrade na takav način da će njihovim maksimiziranjem agent također postići naše ciljeve. Stoga je ključno da nagrade koje postavimo uistinu pokazuju što želimo postići. Konkretno, signal nagrade nije mjesto za prenošenje prethodnog znanja agentu o tome *kako* postići ono što želimo. Umjesto toga, signal nagrade je način komuniciranja s agenom *što* želimo da postigne.

3. Dinamičko programiranje

Dinamičko programiranje (DP) odnosi se na skup algoritama koji s emogu koristiti za izračunavanje optimalnih politika uz savršeni model okoline kao MDP. Klasični algoritmi dinamičkog programiranja imaju ograničenu korist u podržanom učenju zbog pretpostavke savršenog modela I zbog velike računalne zahtjevnosti, ali su teorijski važni.

U večini slučajeva pretpostavlja se da je model konačni MDP, tj., da stanja, akcije I nagrade dolaze iz konačnih skupova. Iako ideje DP-a mogu biti primjenjene na problemima kontinuirane prirode, tj., problemima kod kojih stanja, akcije I nagrade dolaze iz kontinuiranih skupova, egzaktna rješenja moguča su samo u posebnim slučajevima.

Ključna ideja dinamičkog programiranj, I podržanog učenja općenito, je korištenje funkcije vrijednosti za organiziranje I strukturiranje pretrage za dobrom politikom. Mmožemo lako dobiti optimalne politike nakonn što smo pronašli funkcije optimalnih vrijednostu *v** ili *q** koje zadovoljavaju Bellmanove jednadžbe optimalnosti:

$$\begin{aligned} v_*(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_*(s')\Big], \text{ or} \\ q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \;\Big|\; S_t = s, A_t = a\Big] \\ &= \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma \max_{a'} q_*(s', a')\Big], \end{aligned}$$

DP algoritmi se dobivaju pretvaranjem Bellmanovih jednadžbi poput gore navedenih u pravila ažuriranja za poboljšanje aproksimacija funkcija željene vrijednosti.

3.2. Osnovne komponente

Evaluacija politike odnosi se na iterativnoizračunavanje funkcija vrijednosti za danu politiku.

Poboljšanje politike odnosi se na izračun poboljšane politike s obzirom na funkciju vrijednosti za tu politiku.

Kombinirajući ove dvije komponente dobivamo *iteraciju politike* I *iteraciju vrijednosti*, dvije anjpopularnije metode dinamičkog učenja.

$$
\begin{aligned}
v_{k+1}(s) &\doteq \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_k(s')\Big],
\end{aligned}
$$

Figure 3 Funkcija iteracije vrijednosti

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))\big[r + \gamma V(s')\big]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Figure 4 Pseudokod iteracije politike

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

Figure 5 Pseudokod iteracije vijednosti

Sve ove metode mogu se koristiti za određivanje optimalne politike I funkcije vrijednosti za konačne MDP-ove ukoliko imamo sve potrebne informacije o MDP-u.

LITERATURA

[1] Richard S. Sutton, Andrew G. Barto (2018.), Reinforcement Learning, A Bradford Book.

[2] Prezentacije predmeta Podržano učenje, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu

[3] https://www.gymlibrary.dev/environments/classic_control/cart_pole/