9.11.2022.

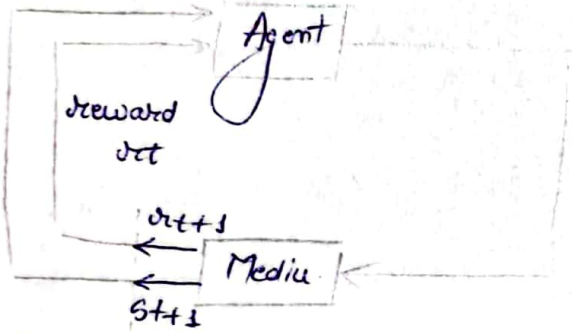
- noțiuni / concepte
- clasificare

Inteligență artificială
Curs 7
Reinforcement Learning

Agentul trebuie să învețe un comportament fără vreo formă de instruire:

- → va avea niște stări
- → face o serie de acțiuni
- → va primi un feedback sub forma unor recompense (pozitive / negative)

În final, agentul trebuie să găsească politica pentru a trece dintr-o stare în alta.



Traiectorie / ieșiri
(Secv. de stări și acțiuni + recompensele asociate):
$(s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$

Scopul este să maximizăm recompensele, deci vrem să determinăm secv. de stări care ne duce spre această stare finală pe baza datelor de antrenare (S, A, R)

Acea Reinforcement Learning - Aplicații

- → Alpha Go (programul a învățat să joace câteva 2600) → știrea cu cinpat o imagine cu un acar.
- → 2016 → GO
- → 2017 → Alpha Go Zero a învățat să joace fără informații din jocuri ale oamenilor.
- → 2017 → Alpha Zero a învățat doar pe baza experienței proprii, în 8 ore (timp relativ mic)

Procese de decizie Markov

Mediul { **determinist** (cunoaștem probabilitatea de a ajunge într-o stare) — starea în care ajungem în urma unei decizii
stochastic (nu întotdeauna ajungem în starea dorită, ci cu o anumită probabilitate pe baza unui model de tranziții $P(S'|S,a)$

Mediul stochastic îl vom folosi mai departe.

Presupunerea Markov constă în cât de mult ne uităm înapoi.

ex. Pp Markov de ordinul 1 ne uităm doar la starea anterioară

$P(S_t|S_{t-1}..S_0) = P(S_t|S_{t-1})$ (starea curentă depinde de un nr. finit de stări anterioare)

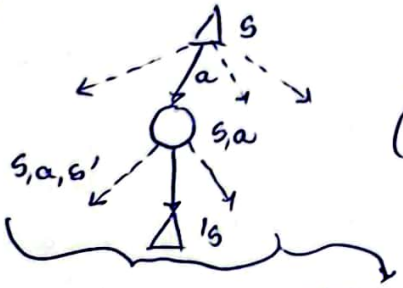
MDP = problemă pt un mediu stochastic în care se aplică modelul de tranziție a lui Markov

Este format din { **stări** $s \in S$
modelul de tranziții $P(S'|S,a)$ (probabilitatea de a ajunge într-o stare aplicând o acțiune pe starea curentă)
funcția de recompense $R(S)$

Rezolvarea MDP presupune găsirea unei soluții / politici. Aceștia sunt alg de căutare nedeterministă

Alg. sunt de tip programare dinamică (doar pt dimensiuni reduse ale problemei)

ex. identificarea, pentru fiecare stare, care este acțiunea cea mai potrivită.



S = stare
(S,a,S') = tranziție

Pentru această pb. se poate asocia un graf / arbore de căutare, unde fiecărei stări i se poate asocia un arbore de căutare.

Într-o problemă de căutare identificăm starea inițială și finală.

Într-un proces de decizie Markov vrem să identificăm o politică optimă π (strategie):

$\pi : S \to A$, $\pi(s)$ = acțiunea recomandată în starea s.

Pentru a vedea cât de optimă este o politică, vom calcula **utilitatea**.

Utilitatea = tot câștigul obținut (printr-o sumă ponderată a stărilor mai apropiate)

Câștigul / Recompensa = doar de la o stare la alta.

Orizont finit: $U_k([s_0, s_1, \ldots, s_{N+k}]) = U_k([s_0, s_1, \ldots, s_N])$, $\forall k > 0$.

Politica nu este staționară într-un orizont, adică se poate schimba în timp.

Orizont infinit → poate avea oricâte stări.
politica optimă este staționară

Vom avea 2 tipuri de recompense pt un orizont infinit

recompensă aditivă:

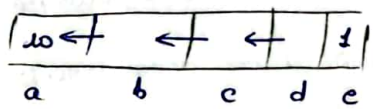
$U_k([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + \ldots$

recompense actualizate (pe asta o vom folosi)

$\gamma \in [0,1]$ = discountul (factorul de discount)

$U_k([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$

stările mai apropiate de starea inițială au discount mai mare (contează mai mult)

ex. 

(recompensele inițiale $R(b) = R(c) = R(d) = 0$), $\gamma = 1$

Vom calcula politicile: U_b, U_c, U_d.

$U_b = R(b) + \gamma R(a)$ (la stânga) $= +1$
$= R(b) + \gamma R(c) + \gamma^2 R(d) + \gamma^3 R(e)$ (la dreapta) $= 1$ } ⇒ politica spune să mergem la stânga.

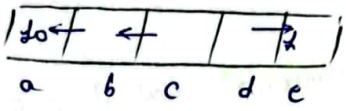
$U_c = R(c) + \gamma R(b) + \gamma^2 R(a)$ (la stânga)
$R(c) + \gamma R(d) + \gamma^2 R(e)$

Dacă luăm $\gamma = 0,1$

$$U(b) = \begin{cases} \overset{\circ}{R(b)} + \gamma R(a) = 0,1 \cdot 10 = 1 \text{ (stânga)} \\ \gamma^3 R(c) + \gamma^2 R(d) + \gamma R(c) + R(b) = 0,001 \text{ (dreapta)} \end{cases}$$

$$U(c) = \begin{cases} \gamma^2 R(a) = 0,01 \cdot 10 = 0,1 \text{ stânga} \\ \gamma^2 R(e) = 0,01 \text{ dreapta} \end{cases}$$

$$U(d) = \begin{cases} \gamma^3 R(a) = 0,001 \cdot 10 = 0,01 \text{ (stânga)} \\ \gamma R(e) = 0,1 \end{cases}$$



Dacă recompensele sunt mărginite și $\gamma < 1 : U_h([s_0, s_1, s_2, \dots]) =$

$$= \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{max} = \frac{R_{max}}{1-\gamma}$$

Pentru a identifica o politică optimă, vom folosi noțiunea de **utilitate așteptată**:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$ (dacă suntem într-o stare, avem mai multe secv. de stări în care putem ajunge $\Rightarrow$ considerăm media)

Politica optimă (π^*) = maximizează valoarea utilității așteptate: $\pi_s^* = \text{argmax}_{\pi} \underbrace{U^{\pi}(s)}_{\text{utilitatea adevărată}}$

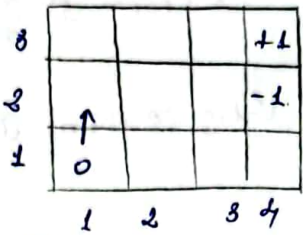
$U^{\pi^*}(s)$ **utilitatea adevărată** este suma discounturilor pe care un agent o are dacă execută politica optimă

$$\pi^*(s) = \text{argmax}_{a \in A(s)} \sum_{s'} P(s'/s,a)\, U(s')$$

Ecuația Bellman: utilitatea unei stări este recompensa imediată pt acea stare ($R(s)$), plus utilitatea maximă a stării următoare.

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'/s,a)\, U(s')$$

Utilitatea stării (1,1).



Mediul este stochastic ⇒ probabilitatea nu este mereu 1.

$U(1,1) = -0,04 + \gamma \max [\, 0,8\,U(1,2) + 0,1\,U(1,2) + 0,1\,U(2,1) +$

$+ 0,1\,U(1,1)$, (stânga sus)

$0,9\,U(1,1) + 0,1\,U(1,2)$, (stânga)

$0,9\,U(1,1) + 0,1\,U(2,1)$, (jos)

$0,8\,U(2,1) + 0,1\,U(1,2) + 0,1\,U(1,1)\,]$ dreapta

Acțiunea cea mai bună este dus.

Metode de rezolvare ale proceselor de decizie Markov (prin programare dinamică)

Metoda T.: iterarea valorilor până se stabilizează

Pasul 1) Inițializează utilitățile cu val. arbitrare

Pasul 2) Actualizează utilitatea fiecărei stări din utilitățile vecinilor:

$$U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'/s,a)\, U_i(s')$$

Pasul 3) Se repetă pt. fiecare s simultan, până când se stabilizează.

ex (de pe ayt): $U_{i+1}(s) = \max_a \sum P(s'/s,a)\,[R(s) + \gamma \cdot U(s')]$.

↓ Se va da la examen.

$U_1(\text{cool}) = \max \{ P(\text{Cool}/\text{Cool}, a=\text{Slow})\,[\overbrace{R(\text{Cool})}^{1} + \gamma \overbrace{U(\text{Cool})}^{0}] = 1,\ a=\text{Slow}$

$P(\text{Warm}/\text{Cool}, a=\text{Fast})\,[\overbrace{R(\text{Cool})}^{2} + \gamma \overbrace{U(\text{Warm})}^{0}] +$

$P(\text{Cool}/\text{Cool}, a=\text{Fast})\,[\overbrace{R(\text{Cool})}^{2} + \gamma \overbrace{U(\text{Cool})}^{0}] = 2,\ a=\text{Fast} \}$

$U_1(\text{Warm}) = \begin{cases} 0,5\,[1] + 0,5\,t = 1 & a = \text{Slow} \\ 1 \cdot [-10] = -10 & a = \text{Fast} \end{cases}$

Metoda II Iterarea politicilor (până politica converge)

Pasul 1) evaluarea politicii : acțiunea e fixată de politică.

$U_i(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$ → pt. actualizarea val. utilităților.

sistem de ecuații liniare cu n necunoscute.
→ se poate calcula ușor.

Vom aplica alg. Value iteration: $U_{i+1}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$

Pasul 2) îmbunătățirea politicii: $\pi_i^*(s) = \max_a \sum_s P(s'|s,a) \cdot U(s')$

Dacă acțiunea estimată este diferită de politica curentă, atunci se actualizează politica.

Extensii ale MDP - variantă POMDP (Partial Observability) → vom avea observații parțiale / cu zgomot, adică nu știm în ce stări vom putea ajunge, ci se bazează pe observațiile făcute de agent, estimate cu probab $P(o|s)$