



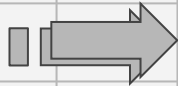
Introducere în Reinforcement Learning



Cursul #3



Ștefan Iordache, Cătălina Iordache, Ciprian Păduraru





Cuprins



Recapitulare

Repetiția: mama învățaturii!



Markov!

Da, insistăm!

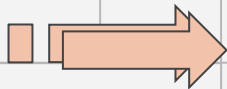
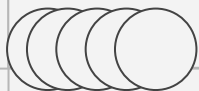
01

Recapitulare

Repetiția: mama
învățăturii!

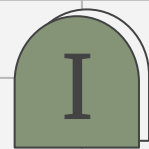


Întrebare rapidă



Discount Factor

Într-un *proces decizional Markov*, un **factor de discount γ mai mare** (apropiat de 1) implică o **importanță mai mare a recompenselor obținute pe termen scurt**, în comparație cu cele obținute pe termen lung?

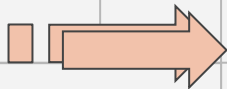
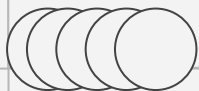


Variante posibile

- Adevărat
- Fals



Întrebare rapidă



Discount Factor

Într-un *proces decizional Markov*, un **factor de discount** γ mai mare (apropiat de 1) implică o **importanță mai mare a recompenselor obținute pe termen scurt**, în comparație cu cele obținute pe termen lung?



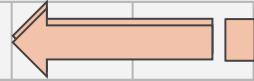
Răspuns

FALS! Doar valoarea 0 implică valorificarea exclusivă a recompensei imediate.

$$\begin{aligned} V^\pi(s_t = s) \\ = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} \\ + \dots | s_t = s] \end{aligned}$$

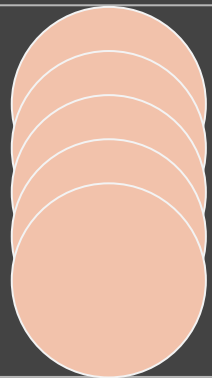


Ne aducem aminte

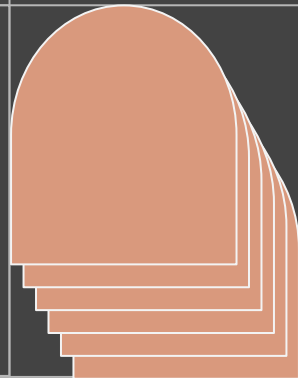


- **MODELUL**
 - Un procedeu matematic pentru exprimarea dinamicii mediului și a recompenselor.
- **POLITICA (POLICY)**
 - Funcție utilizată de agent pentru a realiza asocieri între stări și acțiuni.
- **VALUE FUNCTION**
 - Funcție ce oferă drept răspuns suma recompensele viitoare, folosind drept parametri starea sau/și acțiunea (curente), aplicată sub o anumită politică.

02



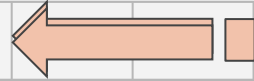
Markov!



Da, insistăm!



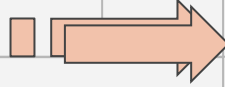
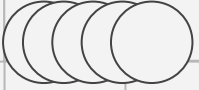
O proprietate de bază



- Starea = informație statistică a istoriei interacțiunilor cu mediul.
- Condiția unei stări s_{t+1} pentru a fi considerată de tip Markov:

$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t)$$

Lanțuri Markov



- **Proces aleatoriu** de tip **memoryless** (secvență de stări cu proprietatea Markov îndeplinită)
- Setul stărilor (S) este finit.
- P este modelul dinamic, ce explică tranzițiile

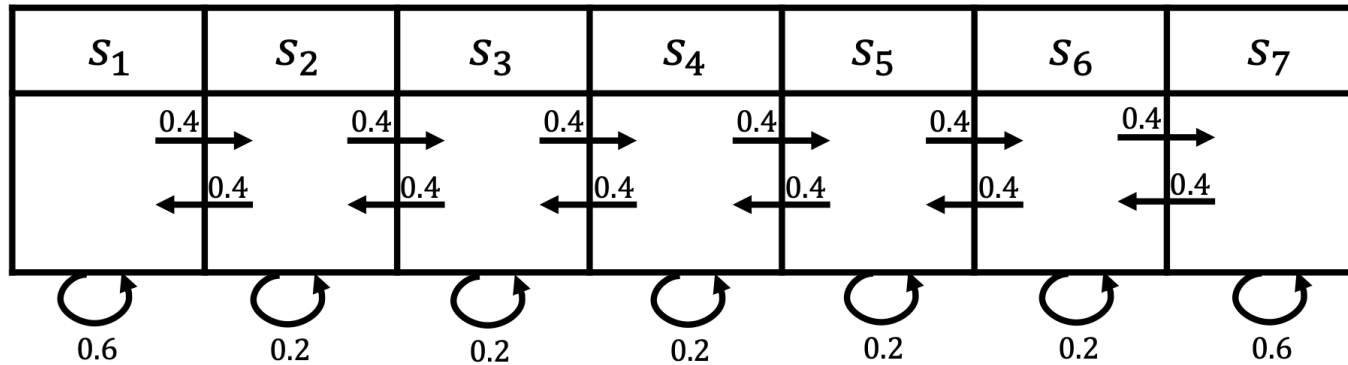
$$p(s_{t+1} = s' | s_t = s)$$

- **Atenție! Nu avem recompense sau acțiuni.**

$$P = \begin{bmatrix} P(s_1|s_1) & \cdots & P(s_n|s_1) \\ \vdots & \ddots & \vdots \\ P(s_1|s_n) & \cdots & P(s_n|s_n) \end{bmatrix}$$

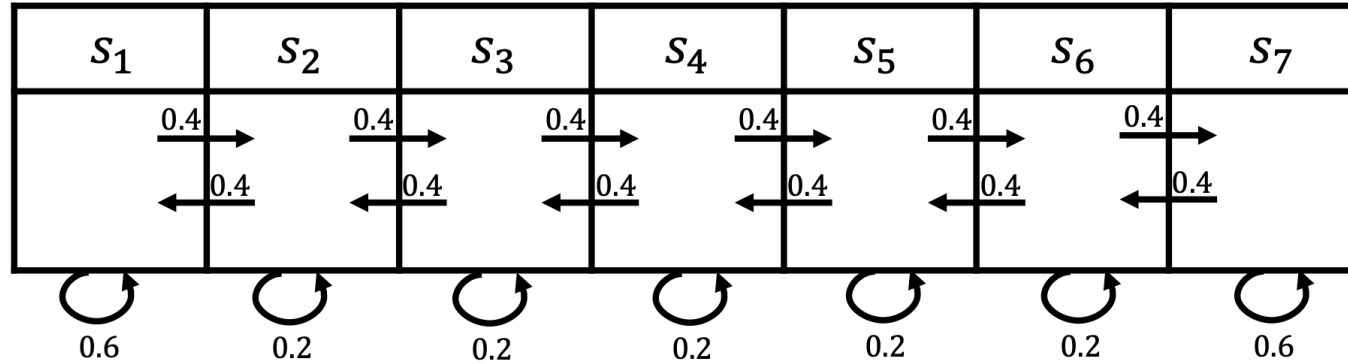


Mars Rover – matrice tranziții



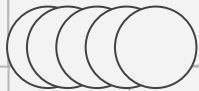
$$P = \begin{pmatrix} 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.2 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.2 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.2 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 \end{pmatrix}$$

Mars Rover – episoade

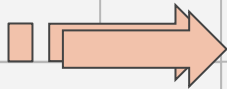


Exemple episoade (stare inițială s_4):

- $s_4, s_4, s_4, s_4, s_4, \dots$
- $s_4, s_5, s_6, s_5, s_4, s_3, s_2, s_1, s_2, \dots$
- $s_4, s_5, s_6, s_7, s_7, s_7, \dots$



Să adăugăm recompense: MRP



- **MRP (Markov Reward Process)**
= **Lanțuri Markov** +
Recompense

- R reprezintă funcția de acordare
a recompenselor:

$$R(s_t = s) = E[r_t | s_t = s]$$

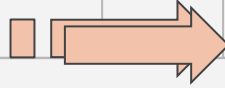
- **Atenție! Nu avem acțiuni.**

$$P = \begin{bmatrix} P(s_1|s_1) & \cdots & P(s_n|s_1) \\ \vdots & \ddots & \vdots \\ P(s_1|s_n) & \cdots & P(s_n|s_n) \end{bmatrix}$$

$$R = (r_1, r_2, \dots, r_n)$$



Return & State Value Function



- **ORIZONT (HORIZON)**
 1. Reprezintă numărul de momente de timp dintr-un episod.
 2. Poate fi *infinit* sau *finit* (*finite MRP*)
- **RETURN (G_t)** – Suma de recompense (cu discount), de la momentul t către orizont.
- **STATE VALUE FUNCTION ($V(s)$)** – Return-ul așteptat pornind din starea s .

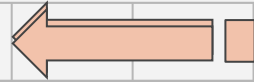
$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots$$

$$V(s) = E[G_t | s_t = s]$$

$$E[G_t | s_t = s] = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots | s_t = s]$$

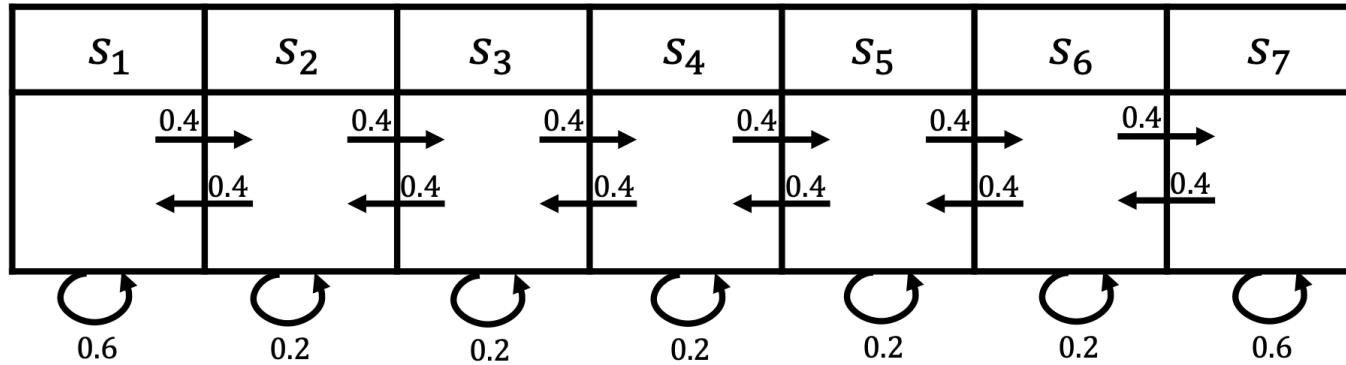


Observații



- **Tot ce am discutat generează un procedeu matematic convenabil, în condițiile în care evităm cazurile infinite.**
- **În mod natural, oamenii acționează sub un factor mereu mai mic decât 1.**

Mars Rover – Exemplu



Alocare recompense:

- +1 în starea s_1
- +10 în starea s_7
- 0 în orice altă stare

Exemplu (start s_4 , $\gamma = \frac{1}{2}$, 4 pași):

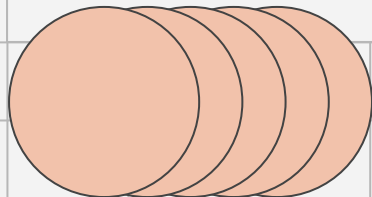
- $s_4, s_5, s_6, s_7 \Rightarrow 0 + \frac{1}{2} * 0 + \frac{1}{4} * 0 + \frac{1}{8} * 10 = 1.25$
- $s_4, s_4, s_5, s_4 \Rightarrow 0 + \frac{1}{2} * 0 + \frac{1}{4} * 0 + \frac{1}{8} * 0 = 0$
- $s_4, s_3, s_2, s_1 \Rightarrow 0 + \frac{1}{2} * 0 + \frac{1}{4} * 0 + \frac{1}{8} * 1 = 0.125$



Simulare!!!

Putem evalua algoritmi de tip MRP prin generarea
unui set suficient de mare de episoade, astfel
încât să satisfacem următoarea ecuație:

$$V(s) = \underbrace{R(s)}_{\text{Immediate reward}} + \underbrace{\gamma \sum_{s' \in S} P(s'|s) V(s')}_{\text{Discounted sum of future rewards}}$$



Formă matriceală – calculul V

$$\begin{pmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{pmatrix} = \begin{pmatrix} R(s_1) \\ \vdots \\ R(s_N) \end{pmatrix} + \gamma \begin{pmatrix} P(s_1|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \ddots & \vdots \\ P(s_1|s_N) & \cdots & P(s_N|s_N) \end{pmatrix} \begin{pmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{pmatrix}$$

$$V = R + \gamma PV$$

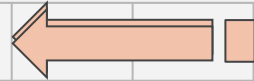
$$V - \gamma PV = R$$

$$(I - \gamma P)V = R$$

$$V = (I - \gamma P)^{-1}R$$

Complexitate $O(N^3)$ –
datorată calcului inversei

Un calcul mai rapid



- **PROGRAMARE DINAMICĂ!!!**

1. Inițializăm $V_0(s) = 0, \forall s$

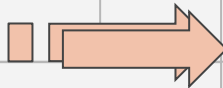
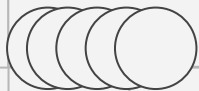
2. Pentru $k = 1$, până la convergență:

- a. Pentru fiecare s din S :

- $$V_k(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s) V_{k-1}(s')$$

- **Complexitate:** $O(|S|^2)$, pentru fiecare iterație ($|S| = N$).

Să adăugăm acțiuni: MDP




- **MDP (Markov Decision Process) = MRP + Acțiuni**
- A este un set finit de acțiuni.
- **Discount factor** $\gamma \in [0, 1]$.
- $R(s_t = s, a_t = a) = E[r_t | s_t = s, a_t = a]$
- $MDP = (S, A, P, R, \gamma)$

*Matricea P devine tri-
dimensională: $S \times S \times A$*

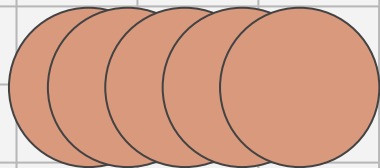


Mars Rover – Exemplu

MDP cu 2 acțiuni.

s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

$$P(s'|s, a_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad P(s'|s, a_2) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



Thanks!

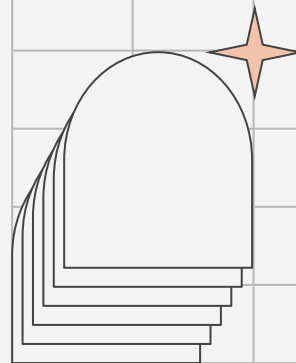
Este timpul pentru întrebări!!!

stefan.iordache10@s.unibuc.ro

catalina.patilea@s.unibuc.ro

ciprian.paduraru@fmi.unibuc.ro

+40 7.. ...



CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon** and infographics & images by **Freepik**

