Introducere în Reinforcement Learning

Cursul #5

Ștefan Iordache, Cătălina Iordache, Ciprian Păduraru

Cuprins

01

Recapitulare

Repetiția: mama învățăturii!

Recapitulare

- Politici MDP
- Control MDP
- În căutarea politicii

02

Monte Carlo

Monte-Carlo în Reinforcement Learning

- Metodele MC învață **direct din experiențe (episodice).**
- MC este **model-free:** nu cunoaște tranzițiile sau recompensele procesului decizional Markov (MDP).
- MC invață din **episoade complete.**

- MC folosește *cea mai simplă idee*: **mean return**
- MC poate fi aplicat pe **MDP-urile episodice** → ***pentru toate episoadele există final***

Evaluarea policitii – Monte Carlo

- **Scop**: învățarea v_π din episoade sub o politică π.
- **Return**
- **Value function**
- **Policy evaluation**: se folosește <u>*empirical mean return*</u> în loc de *expected return*.

- $S_1, A_1, R_2, \ldots, S_k \sim \pi$
- $G_t = R_{t+1} + \gamma * R_{t+2} + \ldots + \gamma^{T-1} * R_T$
- $v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$

First-Visit Monte Carlo

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Every-visit Monte-Carlo Policy Evaluation

<u>Pentru fiecare moment de timp t în care starea s este vizitată într-un episod:</u>

- Counter: $N(s) \leftarrow N(s) + 1$

- Incrementăm return-ul total, sub formă de sumă:
  $S(s) \leftarrow S(s) + G_t$

- Valoarea este estimată în funcție de “***răspunsul mediu***” *(mean return)*:

$$V(s) = S(s) / N(s)$$

<u>$V(s) \rightarrow v_\pi(s)$, **deoarece** $N(s) \rightarrow \infty$</u>

Bias, varianță & MSE (Mean Squared Error)

- Se consideră un model statistic care este parametrizat de $\boldsymbol{\theta}$ și determină o probabilitate de distribuție peste datele observate P(x|$\boldsymbol{\theta}$)
- Se consideră statistica $\widehat{\boldsymbol{\theta}}$, care oferă o estimare a lui $\boldsymbol{\theta}$ și este o funcție a datelor observate.

Bias, Varianță și MSE

Bias-ul unui estimator $\hat{\theta}$:
$Bias_\theta(\hat{\theta}) = \mathbb{E}_{x|\theta}[\hat{\theta}] - \theta$

Varianța unui estimator $\hat{\theta}$:
$\mathrm{Var}(\hat{\theta}) = \mathbb{E}_{x|\theta}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$

MSE pentru un estimator $\hat{\theta}$:
$MSE(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + Bias_\theta(\hat{\theta})^2$

First-Visit Monte Carlo on Policy Evaluation

- Inițializare N(s) = 0; G(s) = 0; oricare ar fi s ∈ S
- Pentru:
  - Extragem un episod i = $s_{i,1}$, $a_{i,1}$, $r_{i,1}$, $s_{i,2}$, $a_{i,2}$, $r_{i,2}$, ……. $s_{i,T}$
  - Definim $G_{i,t} = r_{i,t} + \gamma * r_{i,t+1} + \gamma^2 * r_{i,t+2} + \ldots + \gamma^{T_i - 1} * r_{i,T_i}$ ca return începând cu pasul t, în cadrul episodului selectat i.
  - Pentru fiecare stare s vizitată într-un episod i:
    - Pentru **primul** timp t în care o stare s este vizitată într-un episod i:
      - N(s) = N(s) +1 (incrementare counter pentru toate vizitele)
      - G(s) = G(s) + $G_{i,t}$ (incrementare return total)
      - V^{π}(s) = G(s) / N(s) (actualizăm estimarea)

First-Visit Monte Carlo on Policy Evaluation

Proprietăți

I

Estimatorul V^{π} **este un estimator 'unbiased'**
$\mathbb{E}_{\pi}[G_t \mid s_t = s]$

II

Din teorema numerelor mari, N(s) -> ∞
V^{π} -> $\mathbb{E}_{\pi}[G_t \mid s_t = s]$

Every-Visit Monte Carlo on Policy Evaluation

- Inițializare N(s) = 0; G(s) = 0; oricare ar fi s ∈ S
- Pentru:
  - Extragem un episod i = $s_{i,1}$, $a_{i,1}$, $r_{i,1}$, $s_{i,2}$, $a_{i,2}$, $r_{i,2}$, ……. $s_{i,T}$
  - Definim $G_{i,t} = r_{i,t} + \gamma * r_{i,t+1} + \gamma^2 * r_{i,t+2} + \ldots + \gamma^{T_i - 1} * r_{i,T_i}$ ca return începând cu pasul t, în cadrul episodului selectat i.
  - Pentru fiecare stare s vizitata într-un episod i:
    - Pentru **fiecare** timp t în care o stare s este vizitata intr-un episod i:
      - N(s) = N(s) +1 (incrementare counter pentru toate vizitele)
      - G(s) = G(s) + $G_{i,t}$ (incrementare return total)
      - V^{π}(s) = G(s) / N(s) (actualizăm estimarea)

Every-Visit Monte Carlo on Policy Evaluation

Proprietăți

I

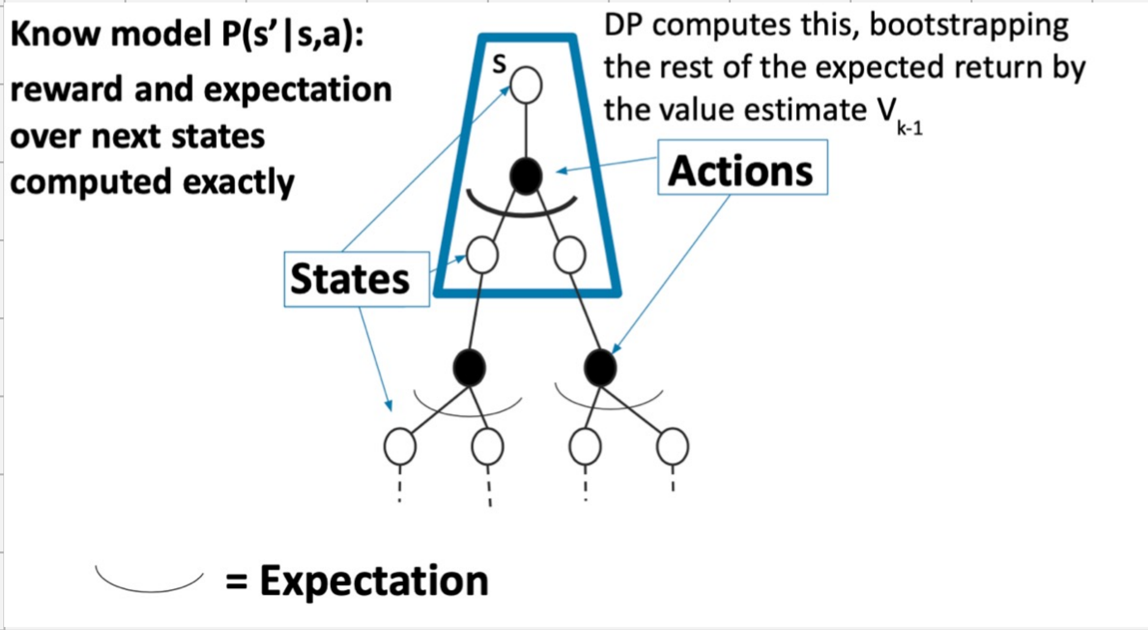
Estimatorul V^{π} **“every-visit” MC este un estimator ‘biased’ al lui** V^{π}

II

Dar!!! În practică este consistent și obține des o valoare MSE mai bună!

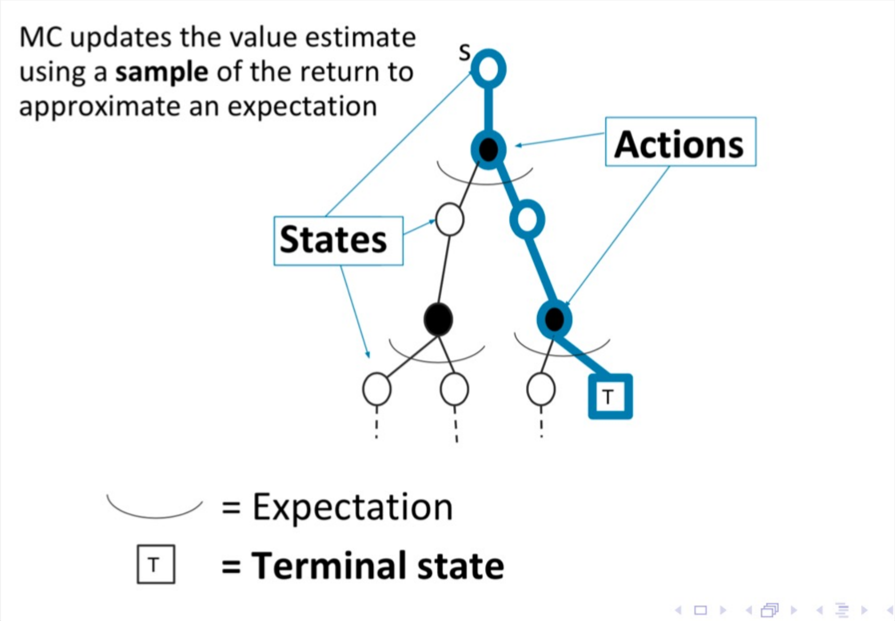
Evaluarea politicii cu programare dinamică

$$V^{\pi}(s) \leftarrow \mathbb{E}_{\pi}[r_t + \gamma * V_{k-1} | s_t = s]$$



Bootstrapping: Update-ul pentru V folosește o estimare.

Evaluarea politicii Monte Carlo



- Metoda MC actualizează valoarea estimată folosind o probă (sample) a recompenselor pentru a aproxima ce se va întâmpla.

Media incrementala

Media $\mu_1, \mu_2, \ldots$ *a unei secvențe* $x_1, x_2, \ldots$ poate fi calculată în mod incremental:

$$\begin{aligned}
\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\
&= \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\
&= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) \\
&= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)
\end{aligned}$$

Actualizări incrementale pentru Monte-Carlo

- Considerăm episodul: S_1, A_1, R_2, ..., S_T. Actualizăm incremental V(s) după acest episod.
- Pentru fiecare stare S_t cu return-ul G_t:
  - $N(St) \leftarrow N(S_t) + 1$
  - $V(St) \leftarrow V(St) + (G_t - V(St)) * \frac{1}{N(St)}$
- În problemele non-staționare, poate fi util să urmăm o **medie ajustabilă** (**running**), adică să uităm de vechile episoade:
  - $V(S_t) < - V(S_t) + \propto (G_t - V(St))$

03

Temporal Difference

Ce este Temporal Difference

TD are la bază strategia de învățare din ***episoade incomplete prin bootstrapping.***

MC vs. TD

Monte Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$$

Temporal Difference

$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$

- $R_{t+1} + \gamma V(S_{t+1})$ poartă denumirea de "temporal difference target".
- $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ reprezintă eroarea TD.

Algoritm TD

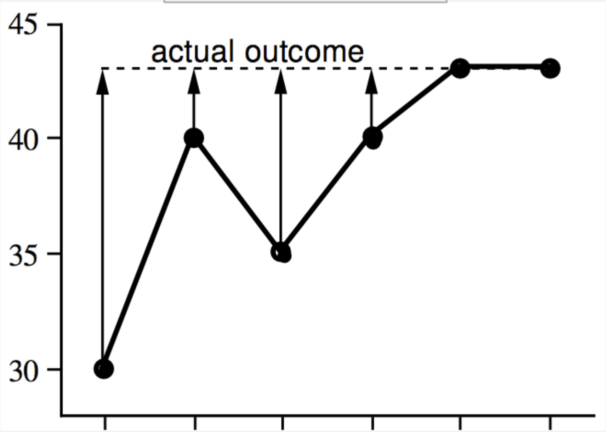
- Input: politica π ce urmează să fie evaluată
- Parametri: mărimea pasului $\alpha \in (0,1]$
- Inițializăm V(s) arbitrat, pentru toate stările, exceptând V(terminal) = 0
- Iterăm pentru fiecare episod:
  - Inițializăm S
  - Iterăm pentru fiecare pas din episod, până la starea terminală:
    - $A \leftarrow$ *acțiunea dată de politica* π *pentru* S
    - Executăm acțiunea A, observăm R, S'
    - $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
    - $S \leftarrow S'$

Exemplu TD

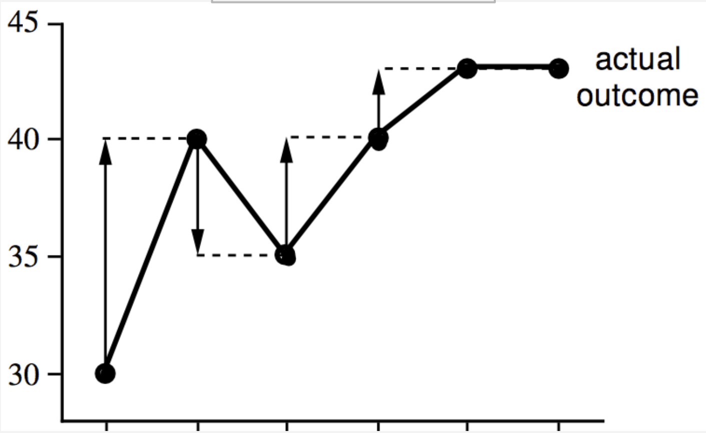
Stare	Timp	Timp prezis (pentru plecare)	Timp total prezis
Plecare din birou	0	30	30
Ajungem la mașină, plouă	5	35	40
Ieșire autostradă	20	15	35
Vehicul încet în față	30	10	40
Stradă rezidențială	40	3	43
Am ajuns acasă	43	0	43

Exemplu TD vs. MC

MC



TD



Avantaje & Dezavantaje TD vs. MC

TD învață înainte de a ști rezultatul final! Nu are nevoie de return/outcome.

- Învățare "online" – TD.
- MC așteaptă până la finalul episodului pentru a afla return-ul.

- TD învață din secvențe parțiale, iar MC doar din episoade complete.
- TD funcționează în medii continue, fără stări terminale. MC nu poate ajunge la această performanță.

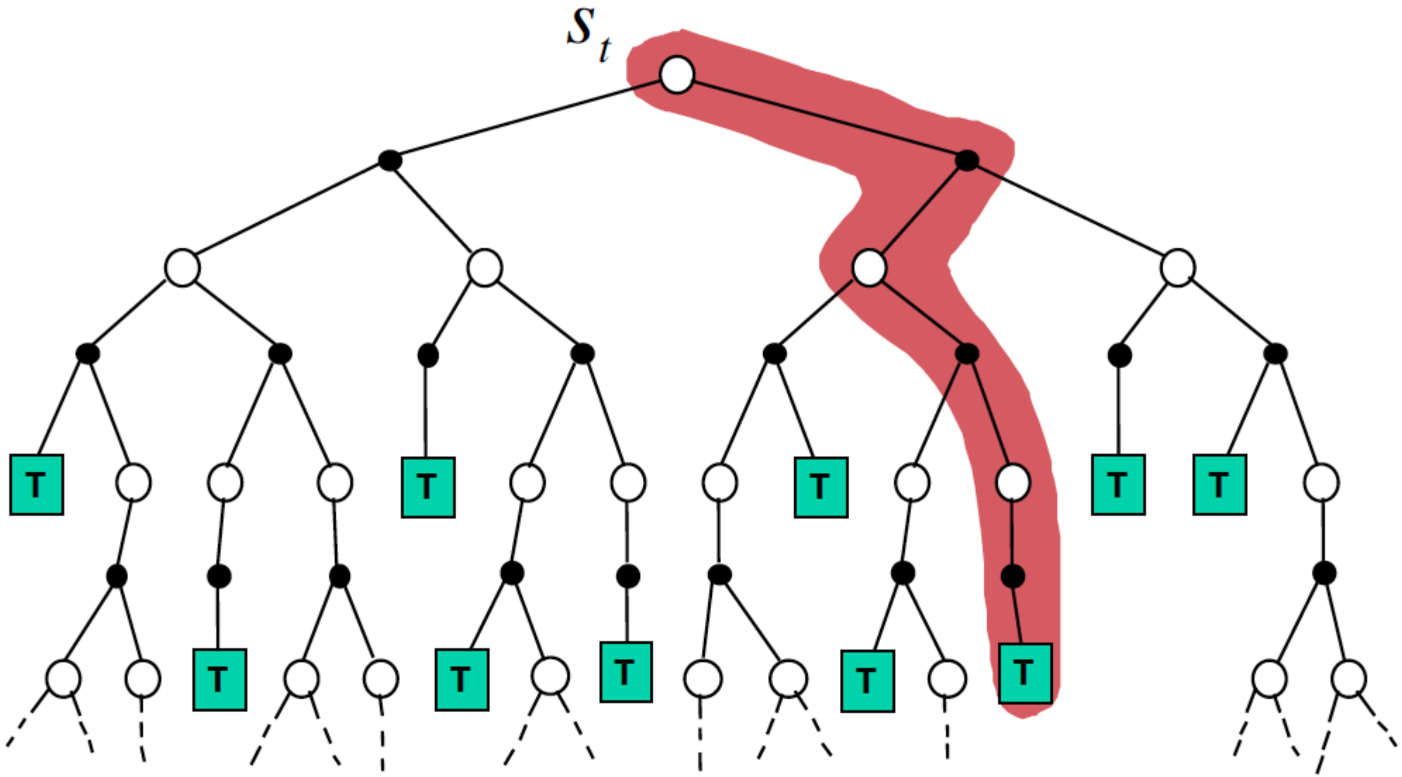
MC

- Varianță mare, bias zero!
- Convergență bună!
- Simplu de înțeles și aplicat!

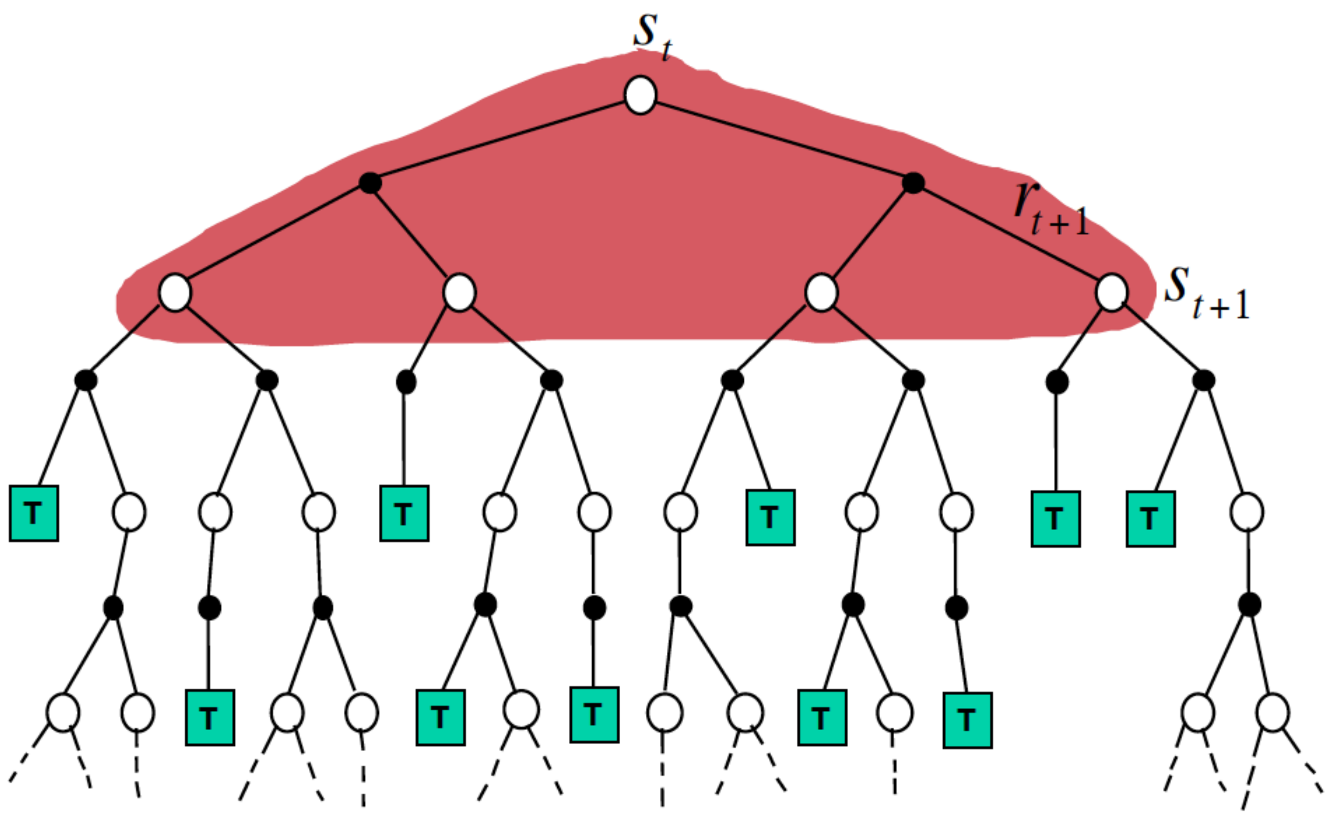
TD

- Mai eficient fată de MC!
- Sensibil la punctul de plecare (valoare inițială)!
- Convergență pentru TD(0)!

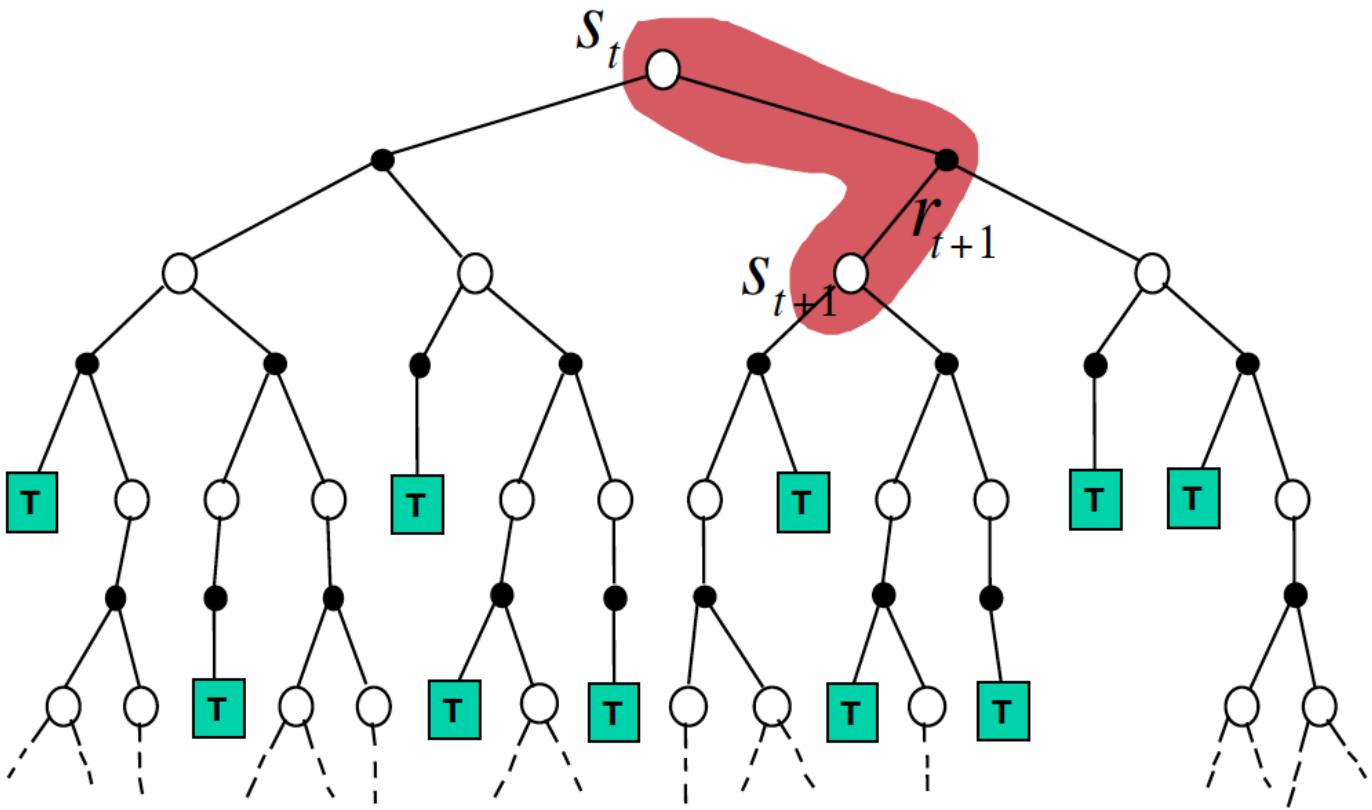
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$



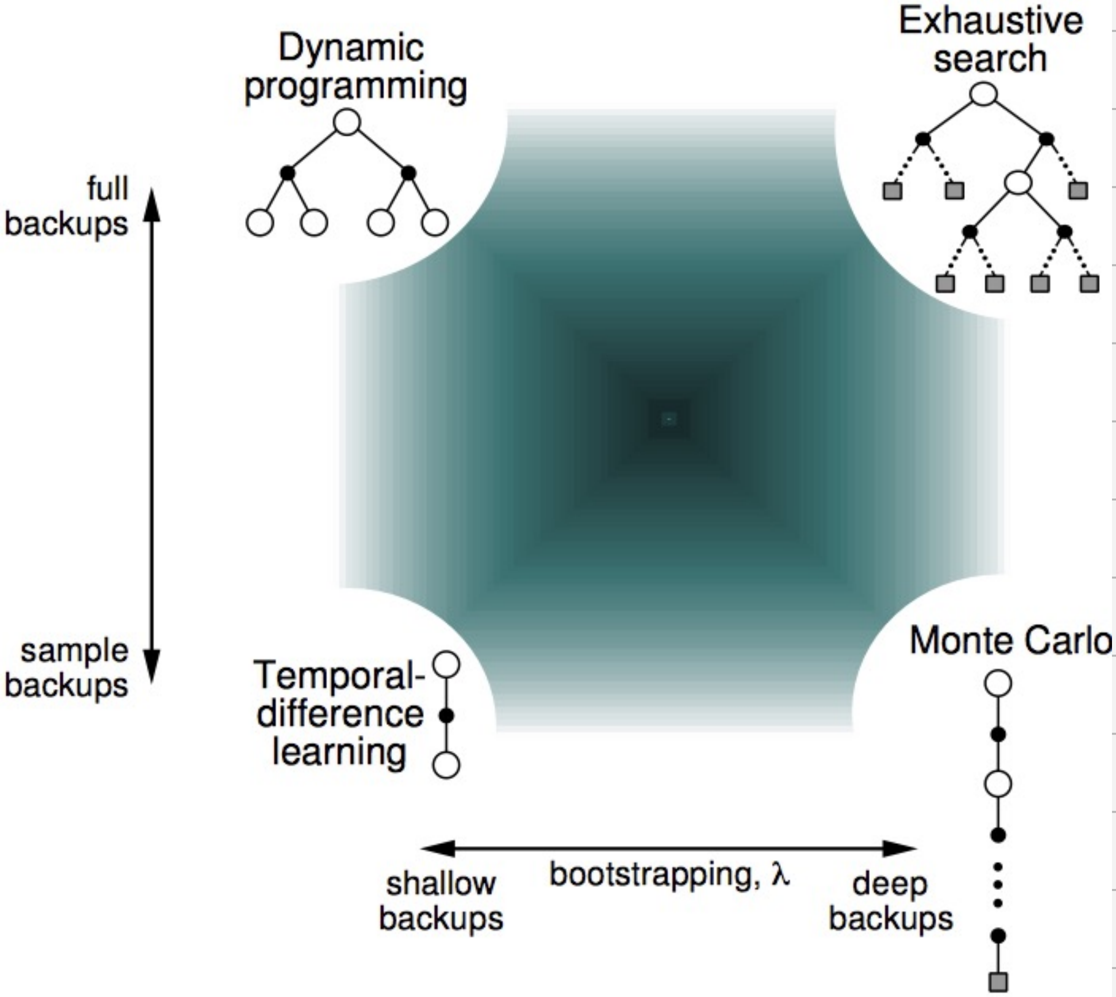
$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1})\right]$$



$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



Marea pictogramă a RL-ului!

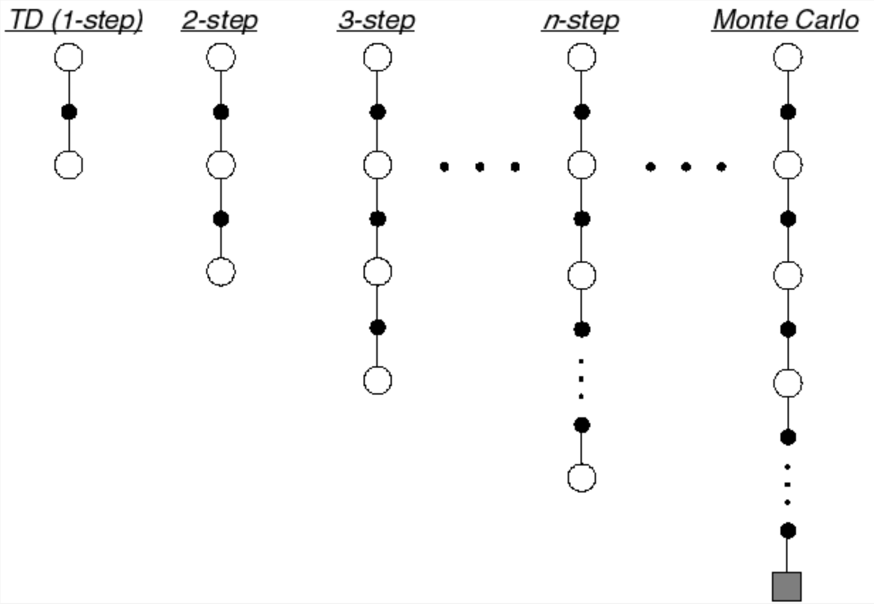

Dynamic programming
Exhaustive search
full backups
sample backups
Temporal-difference learning
Monte Carlo
shallow backups
bootstrapping, λ
deep backups

04

TD(λ)

”n-Step Prediction” – $TD(\lambda)$

- **Considerăm posibilitatea de boostrapping pentru n pași în viitor, aplicație pentru algoritmul TD.**



”n-Step Return” – $TD(\lambda)$

- “n-Step Return” pentru n = 1, 2, ...

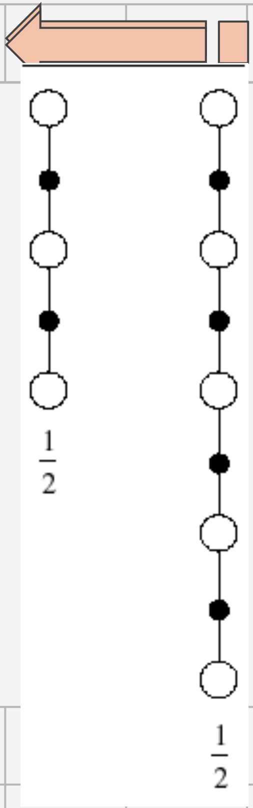
$$
\begin{array}{lll}
n = 1 & (TD) & G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\
n = 2 & & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\
\vdots & & \vdots \\
n = \infty & (MC) & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T
\end{array}
$$

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

”n-Step Temporal Difference Learning” – $TD(\lambda)$

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^{(n)} - V(S_t) \right)$$

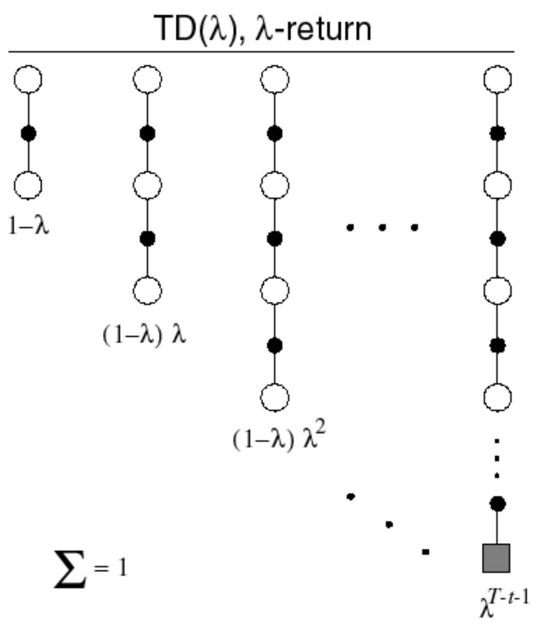
”n-Step Returns” – aplicarea mediei



- **Da, putem realiza o medie între return-uri, pentru valori diferite ale lui n!**

- **Exemplu:** $\frac{1}{2}G^{(2)} + \frac{1}{2}G^{(4)}$

λ-return



- **λ-return** este definit sub forma G_t^{λ} și constă în combinarea n-step return-urilor $G_t^{(n)}$.

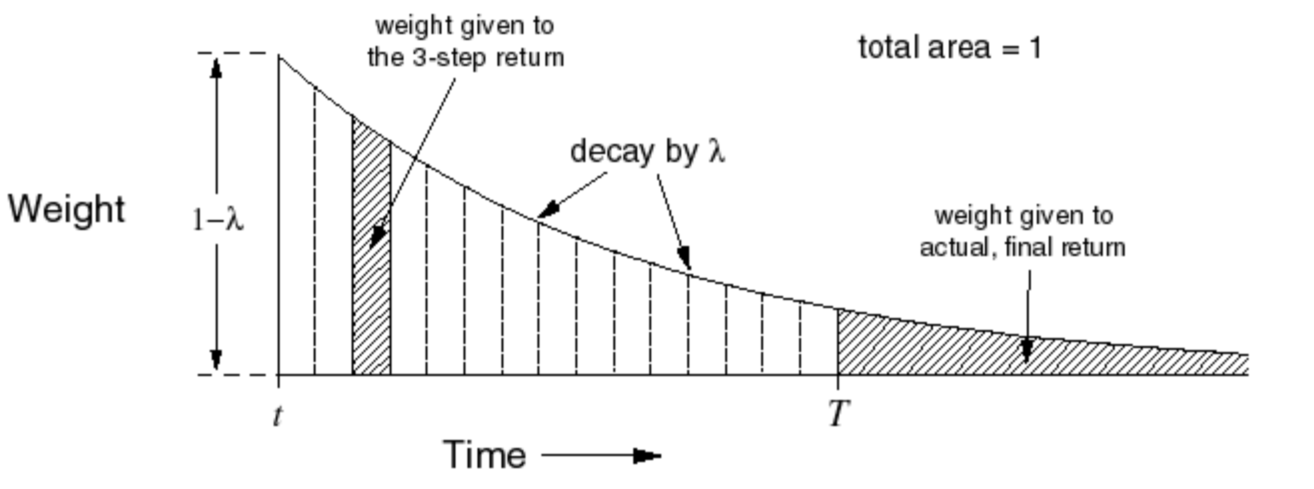
- Folosim pentru calcul $(1-\lambda)\lambda^{n-1}$, astfel:

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

- Mai departe, definim $forward - view\ TD(\lambda)$:

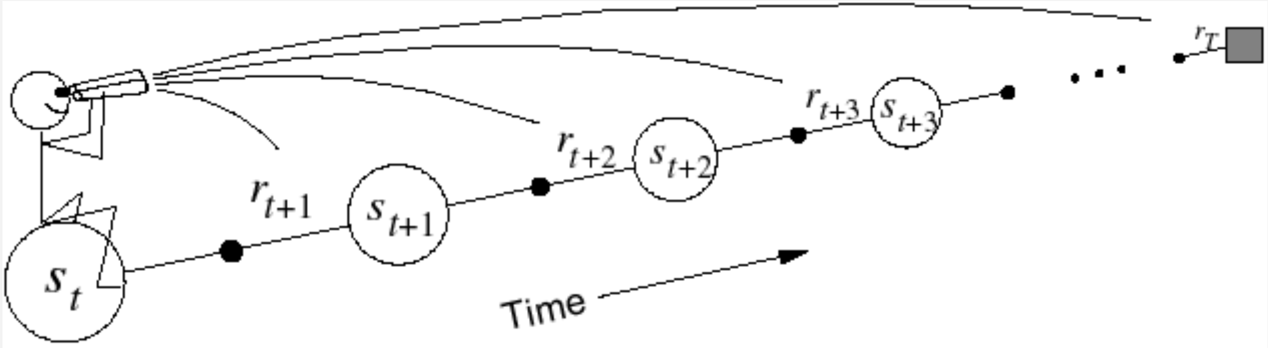
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^{\lambda} - V(S_t)\right)$$

λ-return



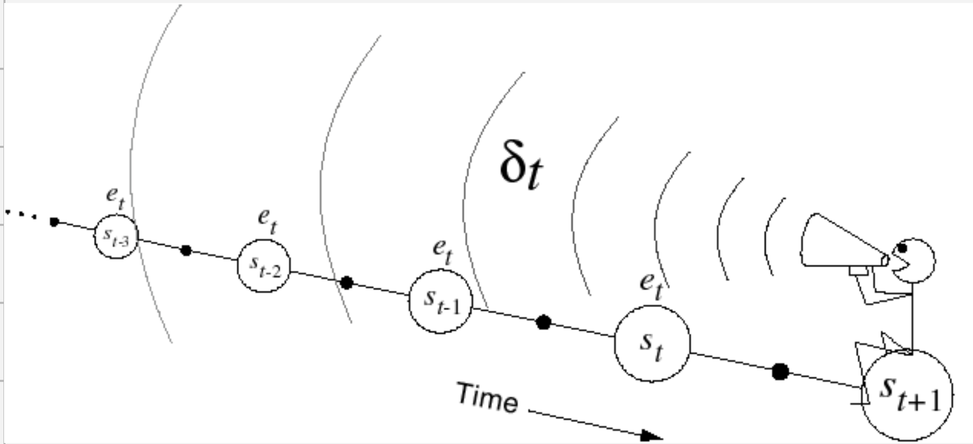
$$G_t^{\lambda} = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

Forward View – $TD(\lambda)$



- ESTE O FORMĂ TEORETICĂ A CEEA CE SE VA ÎNTÂMPLA!
- Se aseamănă cu Monte Carlo, datorită calculării folosind episoade complete!

Backward View – $TD(\lambda)$



- REPREZINTĂ MECANISMUL NECESAR ÎNVĂȚĂRII!
- **Eligibility trace!!!** $E_t(s)$

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$
$$V(s) \leftarrow V(s) + \alpha \delta_t E_t(s)$$

$TD(\lambda) - TD(0)$ – TD(1)

- Dacă $\lambda = 0$, atunci doar starea curentă primește actualizări!
- Echivalent??? TD(0)

$$E_t(s) = \mathbf{1}(S_t = s)$$
$$V(s) \leftarrow V(s) + \alpha\delta_t E_t(s)$$

$$V(S_t) \leftarrow V(S_t) + \alpha\delta_t$$

- Dacă $\lambda = 1$, recompensele vor fi migrate către finalul episodului! Numărul de actualizări în acest caz este egal cu cel realizat de MC (every-visit).

Teoremă!

Suma actualizărilor offline este identică pentru forward-view și back-ward $TD(\lambda)$.

$$\sum_{t=1}^{T} \alpha \delta_t E_t(s) = \sum_{t=1}^{T} \alpha \left(G_t^{\lambda} - V(S_t) \right) \mathbf{1}(S_t = s)$$

λ diferit de 1?

$$
\begin{aligned}
G_t^\lambda - V(S_t) = -V(S_t) \quad & + \quad (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma V(S_{t+1})\right) \\
& + \quad (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})\right) \\
& + \quad (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})\right) \\
& + \quad ... \\
= -V(S_t) \quad & + \quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma V(S_{t+1}) - \gamma\lambda V(S_{t+1})\right) \\
& + \quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma V(S_{t+2}) - \gamma\lambda V(S_{t+2})\right) \\
& + \quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma V(S_{t+3}) - \gamma\lambda V(S_{t+3})\right) \\
& + \quad ... \\
= \quad & \quad\quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right) \\
& + \quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma V(S_{t+2}) - V(S_{t+1})\right) \\
& + \quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma V(S_{t+3}) - V(S_{t+2})\right) \\
& + \quad ... \\
= \delta_t + \gamma\lambda\delta_{t+1} & + (\gamma\lambda)^2\delta_{t+2} + ...
\end{aligned}
$$

λ egal cu 1? → $Every - Visit\ Monte\ Carlo$

$$
\begin{aligned}
& \delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} + \dots + \gamma^{T-1-t}\delta_{T-1} \\
= & \ R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \\
+ & \ \gamma R_{t+2} + \gamma^2 V(S_{t+2}) - \gamma V(S_{t+1}) \\
+ & \ \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3}) - \gamma^2 V(S_{t+2}) \\
& \quad \vdots \\
+ & \ \gamma^{T-1-t} R_T + \gamma^{T-t} V(S_T) - \gamma^{T-1-t} V(S_{T-1}) \\
= & \ R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots + \gamma^{T-1-t} R_T - V(S_t) \\
= & \ G_t - V(S_t)
\end{aligned}
$$

Thanks!

Este timpul pentru întrebări!!!

stefan.iordache10@s.unibuc.ro
catalina.patilea@s.unibuc.ro
ciprian.paduraru@fmi.unibuc.ro

+40 7.. … …