Introducere în Reinforcement Learning

Cursul #6

Ștefan Iordache, Cătălina Iordache, Ciprian Păduraru

Cuprins

On-Policy vs Off-Policy

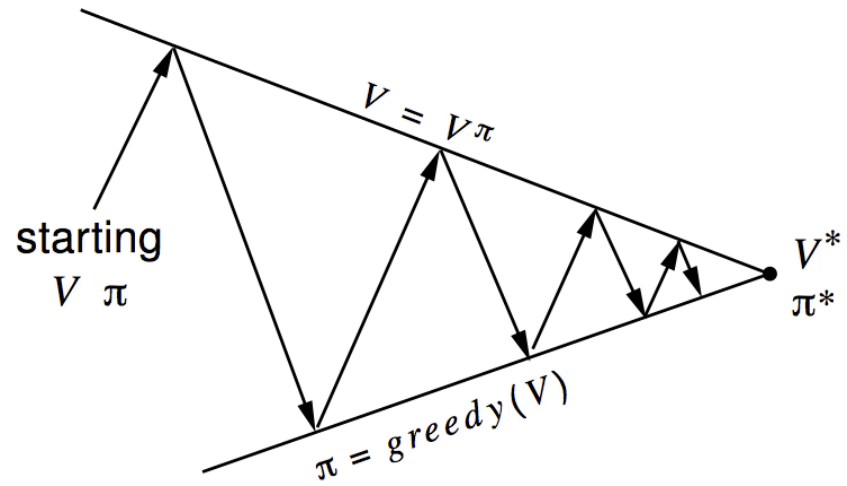
- **On-Policy**
  - ”Calificare la locul de muncă”
  - Învățăm politica π din experiențe extrase cu ajutorul aceleiași politici.
- **Off-Policy**
  - ”Învățăm uitându-ne la altcineva”
  - Învățăm politica π din experiențe extrase cu altă politică (μ).

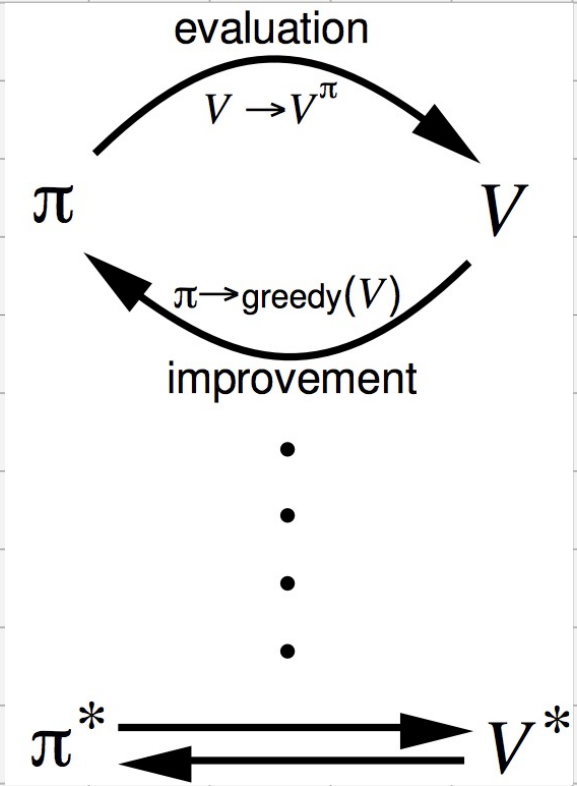
01

On-Policy MC

Generalizare!

- **Evaluarea politicii:** Estimarea V_π
- **Îmbunătățire politicii:** Generarea $\pi' \geq \pi$





Îmbunătățiri: V(s) vs. Q(s, a)

Îmbunătățirea politicii cu ajutorul V(s)

- **Necesită existența modelului din spatele MDP-ului (procesul decizional Markov)**

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \, \mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s')$$

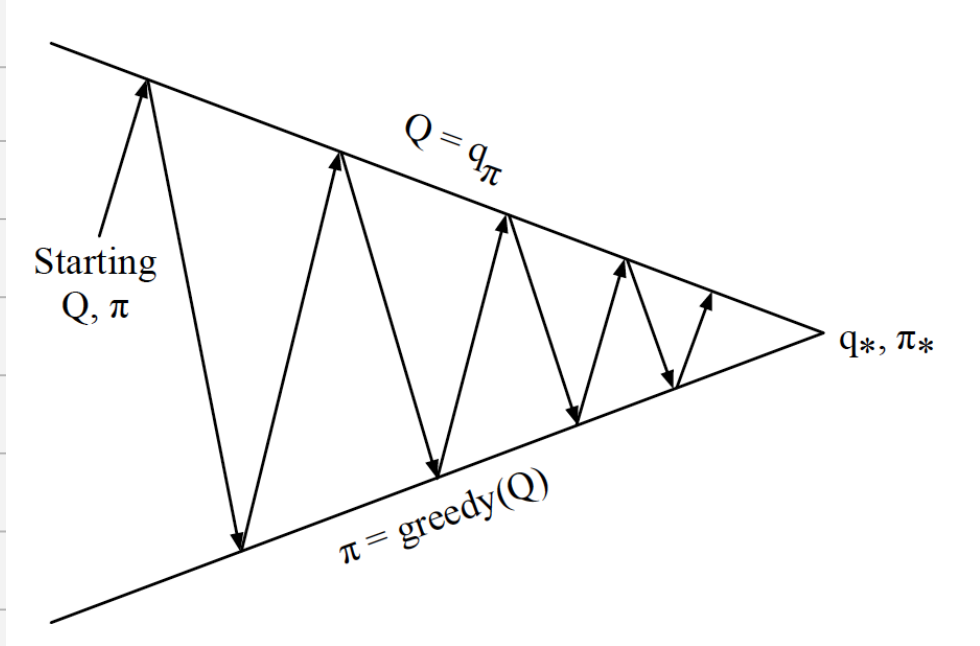
Îmbunătățirea politicii cu ajutorul Q(s, a)

- **Este model-free!**

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \, Q(s, a)$$

Alegem ”Action-Value Function”

- **Evaluarea politicii:** **Estimarea Q = q_π**



Dar cum alegem acțiunile?

GREEDY!!!!

Exemplu: Avem două cutii (stânga și dreapta) din care extragem obiecte. Se execută următoarele acțiuni conform strategiei greedy:

- Deschidem cutia din dreapta => reward 0 = V(dreapta) = 0
- Deschidem cutia din stânga => reward +1 = V(stânga) = +1
- Deschidem cutia din stânga => reward +3 = V(stânga) = +3
- Deschidem cutia din stânga => reward +2 = V(stânga) = +2

Suntem siguri că alegem cea mai bună cutie???

Soluția pentru Greedy? ϵ-Greedy

ϵ-Greedy -> Explorare!!!

- Toate cele m acțiuni sunt încercate cu probabilitate diferită de zero.
- Alegem cu probabilitate **1 - ϵ** acțiunea greedy.
- Cu probabilitate **ϵ** alegem o acțiune aleatorie.

$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a) \\ \epsilon/m & \text{otherwise} \end{cases}$$

Teoremă!

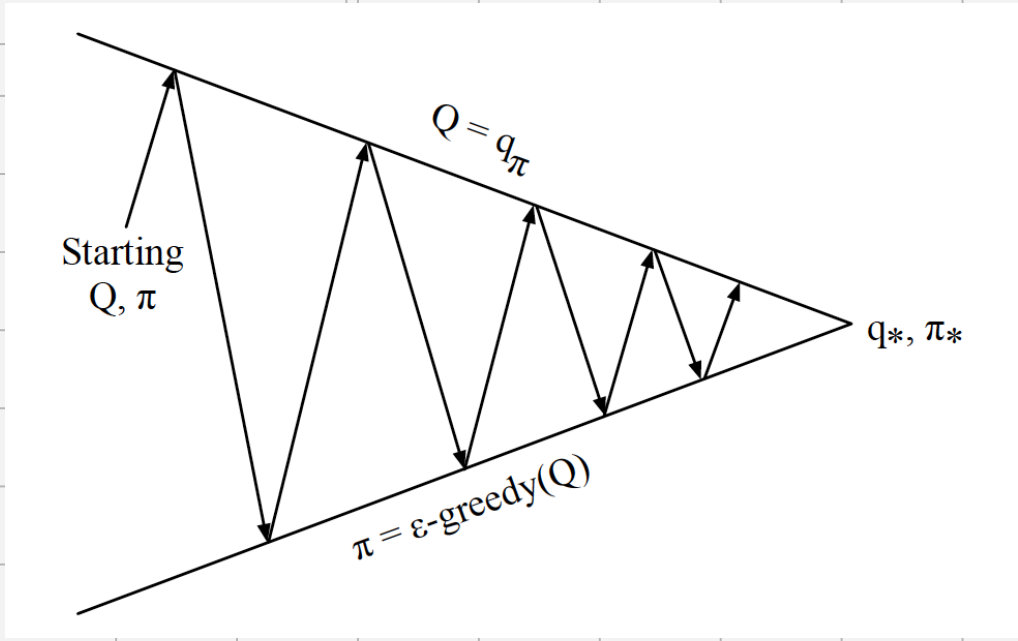
Pentru orice politică ε-greedy π, politica π’ obținută cu ajutorul q_π este o îmbunătățire față de politica anterioară, $v_{\pi'}(s) \geq v_\pi(s)$.

Mai detaliat...

$$
\begin{aligned}
q_\pi(s, \pi'(s)) &= \sum_{a \in \mathcal{A}} \pi'(a|s) q_\pi(s, a) \\
&= \epsilon/m \sum_{a \in \mathcal{A}} q_\pi(s, a) + (1 - \epsilon) \max_{a \in \mathcal{A}} q_\pi(s, a) \\
&\geq \epsilon/m \sum_{a \in \mathcal{A}} q_\pi(s, a) + (1 - \epsilon) \sum_{a \in \mathcal{A}} \frac{\pi(a|s) - \epsilon/m}{1 - \epsilon} q_\pi(s, a) \\
&= \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s, a) = v_\pi(s)
\end{aligned}
$$

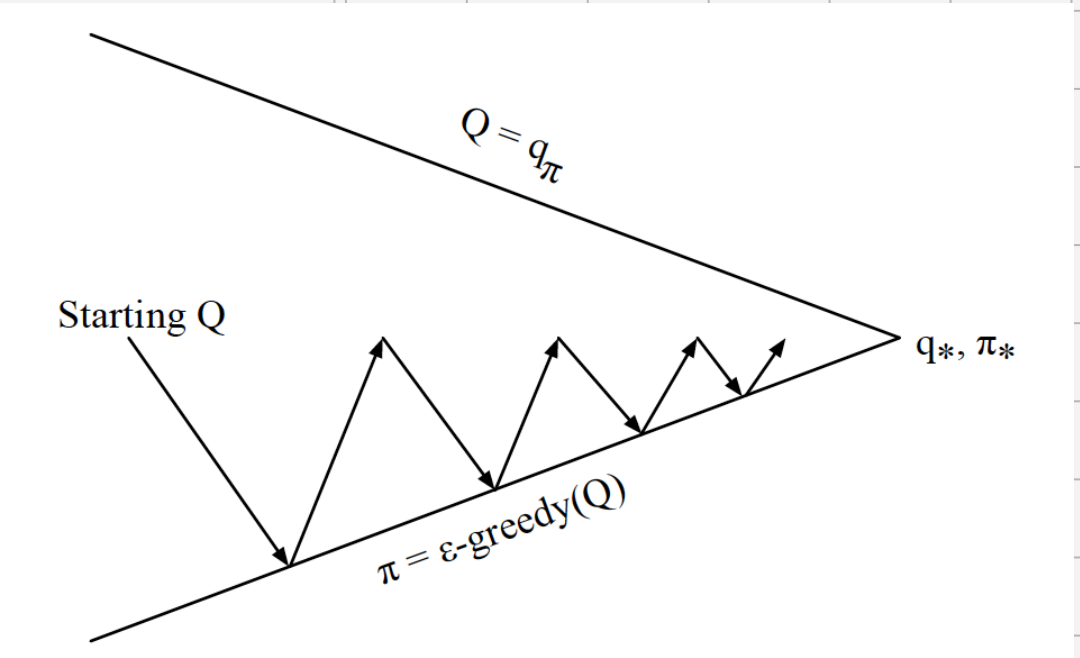
Un portret final!

- **Evaluarea politicii:** Estimarea $Q = q_\pi$
- **Îmbunătățirea politicii:** ε-greedy



Dar putem simplifica!

- **Evaluarea politicii:** **Estimarea Q $\approx q_\pi$**
- **Îmbunătățirea politicii:** **ε-greedy**



GLIE (Greedy in the Limit with Infinite Exploration)

- **Toate perechile stare-acțiune sunt explorate “la infinit”.**

$$\lim_{k \to \infty} N_k(s, a) = \infty$$

- **Policita converge către una de tip greedy.**

$$\lim_{k \to \infty} \pi_k(a|s) = \mathbf{1}(a = \operatorname*{argmax}_{a' \in \mathcal{A}} Q_k(s, a'))$$

GLIE Monte-Carlo

- Extragem episodul cu indicele k, folosind π: $\{\boldsymbol{S_1}, \boldsymbol{A_1}, \boldsymbol{R_1}, \ldots, \boldsymbol{S_T}\} \sim \boldsymbol{\pi}$
- Pentru fiecare stare S_t și acțiune A_t din episod:

$$N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)} (G_t - Q(S_t, A_t))$$

- Îmbunătățim politica folosind valorile noi pentru “action-value function”:

$$\epsilon \leftarrow 1/k$$

$$\pi \leftarrow \epsilon\text{-greedy}(Q)$$

Teoremă!

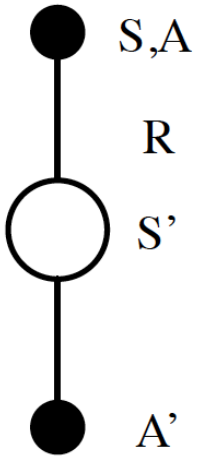
GLIE Monte-Carlo converge către zona optimă a funcției valoare-acțiune, $Q(s,a) \rightarrow q_*(s,a)$.

02

On-Policy TD

MC vs. TD

- **Ne reamintim avantajele TD:**
  - Varianță mai mică!
  - Online!
  - Învață din secvențe incomplete!
- **Ce putem face în continuare?**
  - **Aplicăm TD pentru Q(s, a) cu ε-greedy, la fiecare pas de timp t**



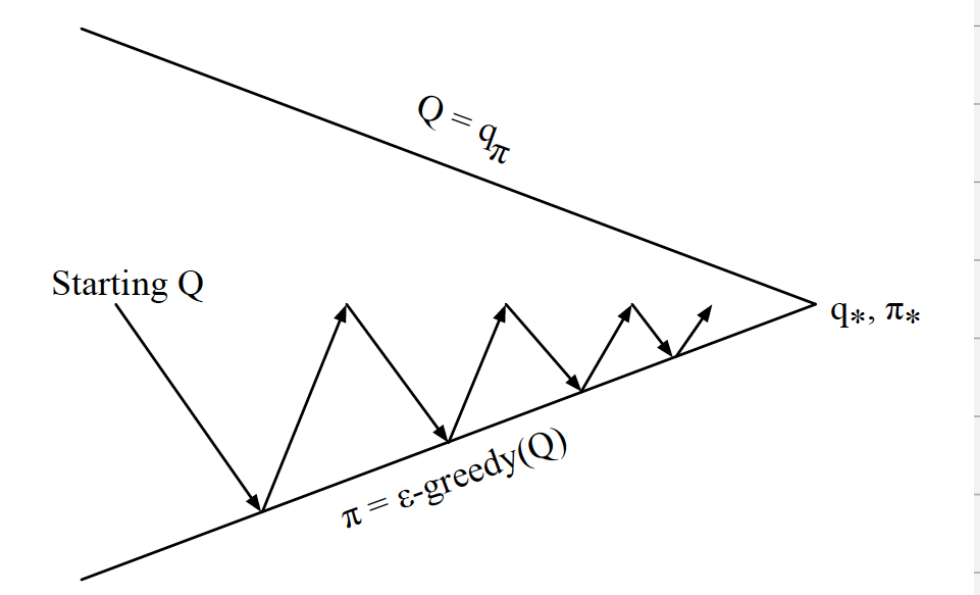
SARSA(λ)

$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma Q(S',A') - Q(S,A) \right)$$

Cum funcționează SARSA?

La fiecare pas de timp!!!

- **Evaluarea politicii:** **Estimarea Q** $\approx q_\pi$
- **Îmbunătățirea politicii:** **ε-greedy**



Algoritm - SARSA On-Policy Control

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Repeat (for each step of episode):
        Take action A, observe R, S'
        Choose A' from S' using policy derived from Q (e.g., ε-greedy)
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma Q(S',A') - Q(S,A)\big]$
        $S \leftarrow S'$; $A \leftarrow A'$;
    until S is terminal

n-Step SARSA

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

$$
\begin{aligned}
n = 1 \quad (Sarsa) \quad & q_t^{(1)} = R_{t+1} + \gamma Q(S_{t+1}) \\
n = 2 \quad & q_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(S_{t+2}) \\
\vdots \quad & \vdots \\
n = \infty \quad (MC) \quad & q_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T
\end{aligned}
$$

n-Step SARSA

- **n-Step Q-Return**

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

- **n-Step SARSA update**

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(q_t^{(n)} - Q(S_t, A_t) \right)$$

03

Off-Policy Learning

Off-Policy Learning - Introducere

- Evaluăm politica $\pi(a|s)$ pentru a calcula $v_\pi(s)$ sau $q_\pi(s,a)$.
- Dar! Urmăm comportamentul politicii $\mu(a|s)$.

- De ce este important acest tip de învățare în domeniu?
  - Este o practică bună!
  - Putem reutiliza experiențele din politici mai vechi.
  - Putem învăța *politica optimă* folosind o *politică exploratorie*.

Importance Sampling

Putem estima totul folosind distribuția unei alte politici.

$$\begin{aligned}\mathbb{E}_{X\sim P}[f(X)] &= \sum P(X)f(X) \\ &= \sum Q(X)\frac{P(X)}{Q(X)}f(X) \\ &= \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)}f(X)\right]\end{aligned}$$

Importance Sampling - MC

- **Folosim return-urile generate de politica μ pentru a evalua politica π.**

- **Va trebui să ajustăm G^t și $V(S_t)$.**

- **Atenție!!! Putem introduce varianță foarte mare!**

$$G_t^{\pi/\mu} = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \frac{\pi(A_{t+1}|S_{t+1})}{\mu(A_{t+1}|S_{t+1})} \cdots \frac{\pi(A_T|S_T)}{\mu(A_T|S_T)} G_t$$

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

Q-Learning

Importance Sampling - TD

- **Folosim target-urile generate de politica μ pentru a evalua politica π.**
- **Va trebui să ajustăm $V(S_t)$.**
- **Varianță mai mică față de versiunea cu Monte Carlo.**

$$V(S_t) \leftarrow V(S_t) + \alpha \left(\frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} (R_{t+1} + \gamma V(S_{t+1})) - V(S_t) \right)$$

Ce este Q-Learning?

- Aplicăm tehnica off-policy learning pentru Q(s, a).
- Nu este necesar să folosim importance sampling!
- Următoarea acțiune este aleasă folosind comportamentul politicii:

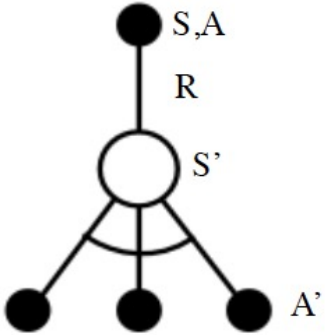
$$A_{t+1} \sim \mu(\cdot \mid S_t)$$

  - Considerăm posibilitatea unei acțiuni alternative:

$$A' \sim \pi(\cdot \mid S_t)$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma Q(S_{t+1}, A') - Q(S_t, A_t) \right)$$

Ce este Q-Learning?



$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma \max_{a'} Q(S',a') - Q(S,A) \right)$$

Thanks!

Este timpul pentru întrebări!!!

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

+40 7.. ... ...