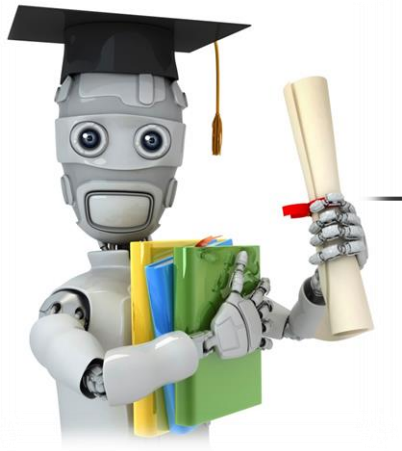
CHAPITRE 4 (Suite)

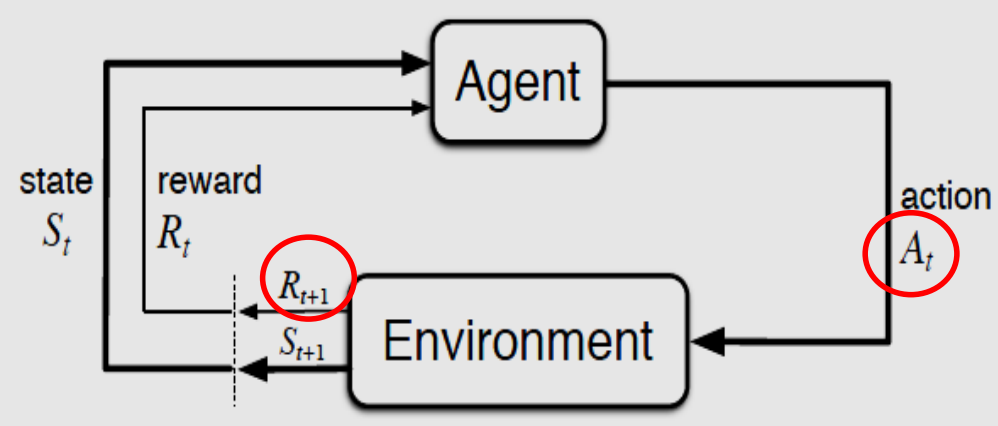


Apprentissage par Renforcement

A Goal Directed Learning from Interaction

Rappel

Recherche par un **agent autonome** d'une **politique** de décision **optimale**. Ceci via des **interactions** de genre « **essai-erreur** » avec son **environnement** suite desquelles il est **récompensé** ou **pénalisé**. L'agent apprend la meilleure politique, qui est la séquence des actions qui **maximisent** sa **récompense totale**



- $\pi: S \times A \longrightarrow [0,1]$

  $(s,a) \longrightarrow \pi(s,a) = \wp[A_t = a | S_t = s]$

- $p(s'|s,a) = \Pr\{S_{t+1} = s' | S_t = s, A_t = a\}$

- $r(s,a,s') = E[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s']$

- $V_\pi : S \to \mathbb{R}$ (un vecteur avec une entrée par état)

$$V_\pi(s) = E_\pi\ [G_t | S_t = s] = E_\pi[\ \sum_{k=0}^{\infty} \gamma^k\, R_{t+k+1}\ | S_t = s]$$

$$V_\pi(s) = E_\pi\ [G_t | S_t = s] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k\, R_{t+k+1} \middle| S_t = s \right]$$

- $q_\pi : S \times A \to \mathbb{R}$ (une matrice avec une entrée par état/action)

$$q_\pi(s,a) = E_\pi\ [G_t | S_t = s, A_t = a] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k\, R_{t+k+1} \middle| S_t = s, A_t = a \right]$$

- Équations de Bellman

$$V_\pi(\mathrm{s}) = \sum_a \pi(a|s)\ \ \sum_{\boldsymbol{s',r}} \boldsymbol{p(s',r|s,a)}[R_{t+1} + \boldsymbol{\gamma V_\pi(s')}]$$

- À savoir :

$$V_\pi(\mathrm{s}) = \sum_a \pi(a|s)\ \boldsymbol{q_\pi(s,a)}$$

Exemple

- Objectif: atteindre le coin supérieur à gauche ou le coin inférieur à droite.
- L'état de l'agent correspond à sa position dans la grille.
- $A = \{\boldsymbol{h}aut, \boldsymbol{b}as, \boldsymbol{g}auche, \boldsymbol{d}roite\}$
- Politique de l'agent: $\pi(a|s) = 0.25\ \forall s\ \forall a$
- La fonction de transition est déterministe. Si l'action fait sortir l'agent de la grille alors considérer que sa position reste inchangée.
- $r = -1$ pour chaque transition
- $\gamma = 1$

0.0	**A**	-20.	-22.
-14.	-18.	-20.	**B**
C	-20.	-18.	-14.
-22.	-20.	**D**	0.0

Fonction de valeur V_π(s)

Travail demandé

Calculer : $V_\pi(\mathrm{A}) = ?$

$q_\pi(B, bas) = ?$

Algorithmes D'apprentissage Par Renforcement

- ***Objectif?*** approximer la politique de décision optimale
- Si on connait le ***modèle de l'environnement*** *(fonction de transition & de récompense)* on peut appliquer la méthode de la **programmation dynamique**. *(model-based)*
- Pratiquement, ce modèle est souvent inconnu *(model-free)* :
  - Méthodes de Monte-Carlo (Sampling)
    - Every-Visit MC
    - First-Visit MC
  - Temporel Difference Learning (Bootstraping)
    - Q-learning
    - SARSA

3. MÉTHODES DE MONTE-CARLO

- Fonction de valeur:

$$V_\pi(s) = E_\pi[G_t | s_t = s]$$

$$V_\pi(\mathrm{s}) = \sum_a \pi(a|s) \sum_{s',r} \boldsymbol{p(s', r|s, a)}[R_{t+1} + \boldsymbol{\gamma V_\pi(s')}]$$

- On veut **Apprendre** (ou Estimer) la fonction de valeur V_π d'une politique π.
- Comment ?
  - ✓ On dispose d'un grand nombre trajectoires issues de chaque état s (sorte d'échantillon de transitions) de type « état , action, renforcement, … » obtenues par ***interaction directe*** ou par ***Simulation***:

$$S_1, A_1, R_2, S_2, \ldots\ldots, S_k$$

- ✓Estimer V (s) en **moyennant** les retours observées sur chacune de ces trajectoires:
  - $Somme(s) \leftarrow Somme(s) + G_t$
  - $N(s) \leftarrow N(s) + 1$
  - $V(s) \leftarrow Somme(s)/N(s)$

Loi des grands nombres

La valeur estimée (moyenne empirique) associée à chaque état converge vers la valeur exacte (espérance) de l'état pour la politique qu'il suit:

$$V(s) \rightarrow V_\pi(s) \text{ lorsque } N(s) \rightarrow \infty$$

Exemple

$$Somme(s) \leftarrow Somme(s) + G_t$$

$$N(s) \leftarrow N(s) + 1$$

$$V(s) \leftarrow Somme(s)/N(s)$$

- Soit un MDP avec deux états P et Q. On donne les Deux trajectoires suivantes:
  - $P, +3, P, +2, Q, -4, \quad P, \quad +4, Q, -3$
  - $Q, -2, \quad P, \quad +3, Q, -3$
- Comment procéder ?
  1. Mise à jour de la valeur d'un état **plusieurs fois** le long d'une même trajectoire. ***(Estimation Biaisée)***
  2. **Solution:** Ne mettre à jour la valeur d'un état que lors de sa **première rencontre** le long de la trajectoire observée.
- Calculer:
  - $V(P) = ? \quad V(Q) = ?$

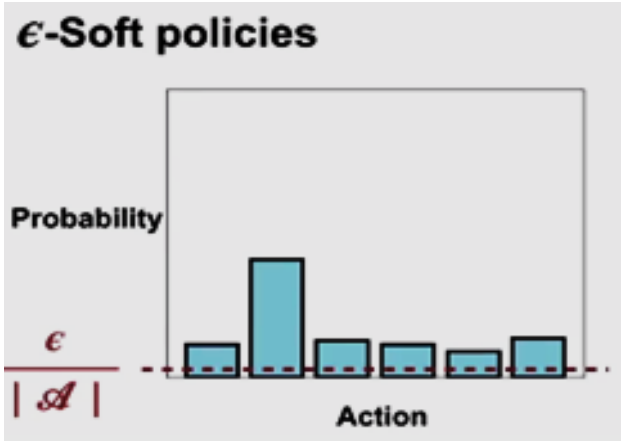
Estimation de la fonctions d'utilité liée aux actions $q_\pi(s,a)$

- Quand la ***fonction de transition est inconnu***, il est plus utile d'estimer $q_\pi(s,a)$ que d'estimer $V_\pi(s)$.
- Nous allons garder les mêmes idées de :
  - *First-Visit MC* : considérer seulement la première occurrence du **couple (s,a)**
  - *Every-Visit MC* : considérer toutes les occurrences…
- Afin **d'évaluer toutes les actions pour chaque état** , il faut ***maintenir l'exploration*** : utiliser plutôt des ***politiques stochastiques*** avec une probabilité non nulle de choisir chacune des actions :

$$\pi(a|s) > 0 \; \forall \, s \in S \; \forall \, a \in A(s)$$

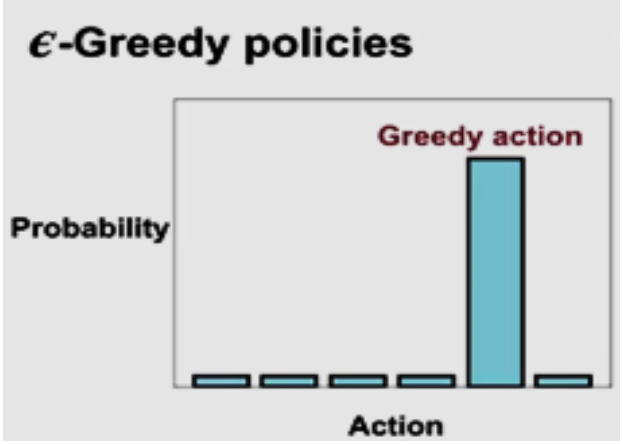
Politiques stochastiques

- **Politique ε −soft :** toute politique dont la probabilité du choix de chaque action **est au moins** $\frac{\varepsilon}{|\mathcal{A}|}$



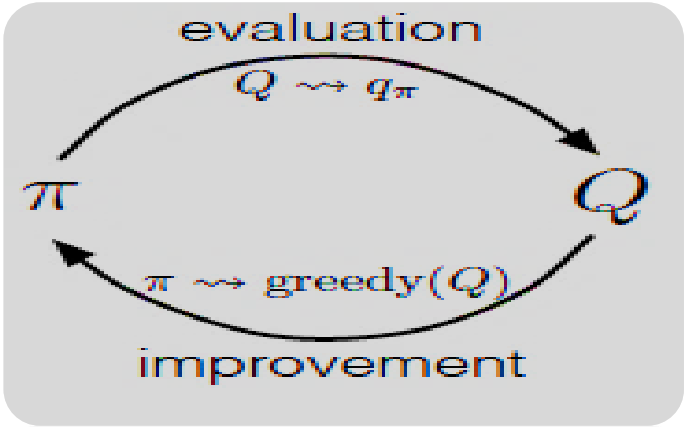
- **Uniform Random policy:** chaque action $\frac{\varepsilon}{|\mathcal{A}|}$
- **Politique ε – greedy**: souvent choisit l'action gloutonne et occasionnellement une des actions non-gloutonnes:

$$\pi(s,a) \leftarrow \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|A(s)|} & \text{if } a = a^* \\ \frac{\varepsilon}{|A(s)|} & \text{if } a \neq a^* \end{cases}$$



De l'estimation au contrôle

- On cherche à trouver la politique optimale.
- On va faire alterner , jusqu'à convergence: Evaluation et Amélioration .



- commencer par une politique arbitraire π_0
- Evaluer cette politique c'est-à-dire estimer $q_{\pi_0}(s, a)$ comme déjà vu
- Améliorer π_0 c'est-à-dire construire une politique π_1 comme la politique gloutonne respectivement à q_{π_0}
- Ce cycle est répété jusqu'à convergence vers la politique optimale:

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{A} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{A} \dots \xrightarrow{A} \pi_* \xrightarrow{E} q_{\pi_*}$$

Remarque

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots$$

$$G_t = R_{t+1} + \gamma(\boldsymbol{R_{t+2} + \gamma^1 R_{t+3} + \gamma^2 R_{t+4} + \ldots})$$

$$G_t = R_{t+1} + \gamma G_{t+1}$$

Exemple

Considérer $\gamma = 0.5\ et\ T = 5$. La séquence des récompenses suivantes est reçues: $R_1 = -1\ R_2 = 2\ R_3 = 6\ \ R_4 = 3\ \ R_5 = 2.$

Trouver: $G_5, G_4, \ldots, G_0$.

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s,a) \leftarrow$ arbitrary

$Returns(s,a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ε-soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s,a)$

$Q(s,a) \leftarrow \text{average}(Returns(s,a))$

(c) For each s in the episode:

$A^* \leftarrow \arg\max_a Q(s,a)$

For all $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1-\varepsilon+\varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

Prédiction incrémentale Monte Carlo

- On dispose d'un ensemble d'observations: x_1, x_2,...x_k et on veut calculer la moyenne:

$$\mu_k = \frac{1}{k}\sum_{j=1}^{k} x_j$$

- Mise à jour incrémentale :

$$\mu_k \leftarrow \mu_k + \frac{1}{k}(x_k - \mu_k)$$

- Prédiction de la fonction de valeur dans MC

$$V(s_t) \leftarrow V(s_t) + \frac{\mathbf{1}}{\boldsymbol{N(s_t)}}(G_t - V(s_t))$$

- Définition plus générale ($0 < \alpha < 1$):

$$V(s_t) \leftarrow V(s_t) + \boldsymbol{\alpha}(G_t - V(s_t))$$

4. Méthodes De Différence Temporelle

Prédiction TD(0) (one-step TD)

$$V(s_t) \leftarrow V(s_t) + \boldsymbol{\alpha}(G_t - V(s_t))$$

Idée des méthodes TD: Ne pas attendre la fin de l'épisode mais faire la mise à jour à chaque étape t :

1. ***Sampling*** : apprendre à base de l'expérience **"$\mathbf{R_{t+1}}$"**
2. ***Bootstrapping*** *:* MAJ des estimations basées sur d'autres estimations déjà acquises **"$\boldsymbol{V(s_{t+1})}$ "**

$$V(s_t) \leftarrow V(s_t) + \alpha(\boldsymbol{R_{t+1} + \gamma V(s_{t+1})} - V(s_t))$$

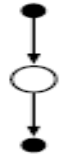
- Erreur TD converge itérativement à zéro:

$$\boldsymbol{\delta_t = R_{t+1} + \gamma V(s_{t+1}) - V(s_t)}$$

Tabular TD(0) for estimating v_π

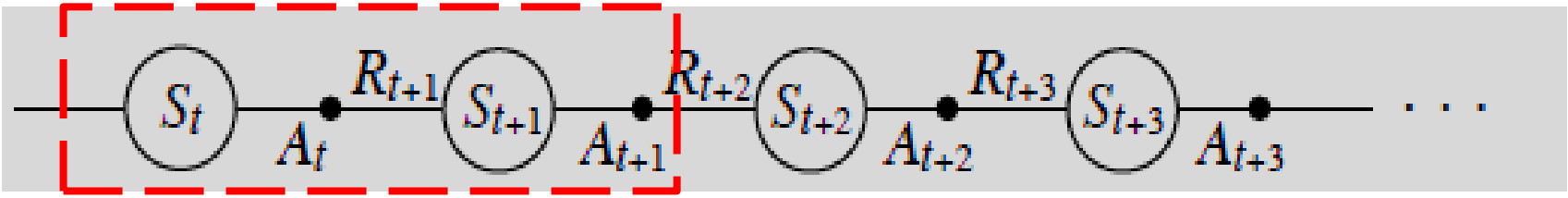
Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\mathit{terminal}) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until S is terminal



SARSA : Contrôle TD sur politique

- SARSA: le nom provient des 4 éléments utilisés dans la règle de Mise-jour :



- SARSA comme c'est un algorithme de contrôle cherche à estimer Q_π . Alors

$$V(s_t) \leftarrow V(s_t) + \alpha(\boldsymbol{R_{t+1} + \gamma V(s_{t+1})} - V(s_t))$$

Devient:

$$\boldsymbol{Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))}$$

- SARSA est Sur-Politique : estimer Q_π et π c'est la politique de comportement.
- GPI (evaulation & amélioration)
- Exploration (politique $\varepsilon - greedy$)

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$

Initialize $Q(s,a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal,\cdot) = 0$

Loop for each episode:

Initialize S

Choose A from S using policy derived from Q (e.g., ε-greedy)

Loop for each step of episode:

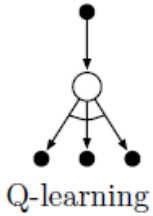
Take action A, observe R, S'

Choose A' from S' using policy derived from Q (e.g., ε-greedy)

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

$S \leftarrow S'$; $A \leftarrow A'$;

until S is terminal



Q-Learning : Contrôle TD hors politique

- Dans SARSA:

$$\boldsymbol{Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))}$$

- Dans Q-Learning:

$$\boldsymbol{Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(R_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t)\right)}$$

L'action a utilisée dans la mise à jour n'est pas forcément celle qui sera prise dans l'état suivant S_{t+1}.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\mathit{terminal}, \cdot) = 0$

Loop for each episode:

Initialize S

Loop for each step of episode:

Choose A from S using policy derived from Q (e.g., ε-greedy)

Take action A, observe R, S'

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

$S \leftarrow S'$

until S is terminal