

Chapitre 5 . Q-Learning

- Q-Learning = principal algorithme de RL
- permet d'apprendre la politique optimale d'une MDP sans connaître ni ρ , ni γ .
- l'apprentissage est réalisé par interaction avec l'environnement.

ce qu'on cherche à apprendre :
les paires (état, action) de la politique optimale.

Deux ingrédients :

- Equation de Bellman pour la fonction q^*
- Différence Temporelle

Eq. d'optimalité de Bellman

pour Q :

$$Q^*(s, a) = \sum_{s' \in \mathcal{S}} P(s'|s, a) (r(s'|s, a) + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a'))$$

Comme pour l'algo. TD, on considère une trajectoire (indexée par t) et avec une estimation \hat{Q} de Q^* , le terme $r_t + \max_{a'} \hat{Q}(s_{t+1}, a') - \hat{Q}(s_t, a_t)$ est une différence temporelle que l'on peut utiliser comme correction pour $\hat{Q}(s_t, a_t)$.

L'algorithme peut alors être décrit par la formule:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t) \left[r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right]$$

Algorithme (Q-Learning):

Input : S, A, γ

$$\hat{q}(s, a) \leftarrow 0 \quad \forall s, a \in S \times A$$

$$n(s, a) \leftarrow 0 \quad \forall s, a \in S \times A$$

Répéter

initialiser l'état initial s_0

$$t \leftarrow 0$$

tant que épisode non terminé faire

selectiviser l'action a_t et l'émettre

observer r et s_{t+1}

$$\hat{q}(s_t, a_t) \leftarrow \hat{q}(s_{t+1}, a_t) + \alpha(r_t + \gamma \max_{a'} \hat{q}(s_{t+1}, a') - \hat{q}(s_t, a_t))$$

$$n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$$

$$t \leftarrow t + 1$$

fin tant que
jusqu'à critère d'arrêt vérifié

Un exemple : Retour au Grid World.

On reprend la classe Game.

→ mais nous fermet cette fois-ci de récupérer la récompense.

SARSA: c'est autre algorithme
 d'apprentissage mais on-policy
 (l'action utilisée dans la mise à
 jour de q est celle qui est exécutée)

$$q(s_t, A_t) = q(s_t, A_t) + \gamma [R_{t+1} + \gamma q(s_{t+1}, a_{t+1}) - q(s_t, a_t)]$$

SARSA

The diagram illustrates the SARSA update rule. It shows the current state-action pair (s_t, A_t) on the left, which is updated to a new value based on the current reward R_{t+1} and the value of the next state-action pair (s_{t+1}, a_{t+1}) . The update is performed by adding a term $\gamma [R_{t+1} + \gamma q(s_{t+1}, a_{t+1}) - q(s_t, a_t)]$ to the current value $q(s_t, A_t)$. The label 'SARSA' is written below the equation, with arrows pointing to the state-action pairs in the update term.

