Introduction à l'apprentissage par renforcement

Cours 1 : Introduction

Zhi YAN

ENSTA

12 décembre 2024

Qu'est-ce que le RL ?

Définition :

- L'apprentissage par renforcement (en anglais, *Reinforcement Learning (RL)*) est une méthode d'apprentissage automatique qui consiste à apprendre par essais et erreurs en interagissant avec un environnement.

Concepts :

- **Agent :** Celui qui apprend (par ex., un robot, un programme informatique).
- **Environnement :** Le monde dans lequel l'agent agit.
- **État :** La situation actuelle de l'environnement.
- **Action :** Les choix que l'agent peut faire.
- **Récompense :** La récompense obtenue par l'agent après avoir effectué une action.

ENSTA
IP PARIS

Qu'est-ce que le RL ?

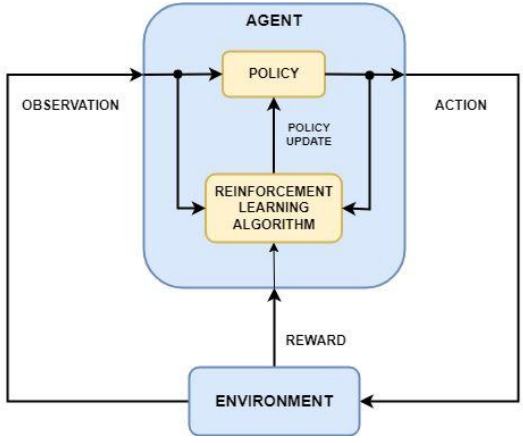


Figure: Schéma d'apprentissage par renforcement.

Source de l'image : `www.mathworks.com`

Qu'est-ce que le RL ?

- Essais et erreurs : Un agent apprend en interagissant avec son environnement et en ajustant ses actions en fonction des résultats obtenus.
- Récompenses différées : Les récompenses ne sont pas toujours immédiates, elles peuvent être obtenues après plusieurs actions.

RL vs Les autres méthodes d'apprentissage automatique

Apprentissage supervisé :

- Données étiquetées (entrée/sortie).
- L'algorithme apprend une fonction de mapping entre les entrées et les sorties.

RL :

- Pas besoin de grandes quantités de données étiquetées.
- L'agent apprend par interaction avec l'environnement.

ENSTA
IP PARIS

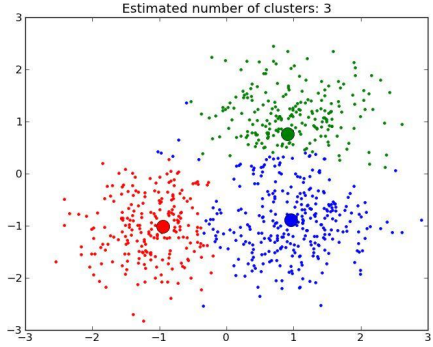
RL vs Les autres méthodes d'apprentissage automatique

Apprentissage non supervisé :

- Données non étiquetées.
- L'algorithme cherche à trouver des structures cachées dans les données.

RL :

- L'objectif est de prendre des décisions, pas de trouver des structures.



ENSTA
IP PARIS

RL vs Les autres méthodes d'apprentissage automatique

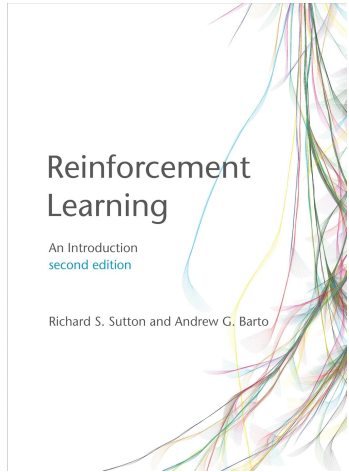
Caractéristique	Apprentissage supervisé	Apprentissage non supervisé	RL
Données	Étiquetées	Non étiquetées	Interactions
Objectif	Apprendre une fonction de mapping	Trouver des structures cachées	Maximiser une récompense
Méthode	Entraînement sur des exemples	Clustering, réduction de dimension	Essais et erreurs

Scénarios d'application

- **Jeu :** Atari, AlphaGo, etc.
- **Contrôle des robots :** contrôle du bras robotisé, conduite autonome, etc.
- **Système de recommandation :** recommandations personnalisées, etc.
- **Investissement financier :** négoce d'actions, trading quantitatif, etc.
- **Soins de santé :** recherche et développement de médicaments, diagnostic médical, etc.

Textbook

Richard S. Sutton and Andrew G. Barto.
Reinforcement learning: An introduction. *MIT press*, 2018.



Processus de décision de Markov (MDP)

Qu'est-ce qu'un MDP ?

- Un cadre mathématique utilisé pour modéliser les problèmes de prise de décision où les résultats sont en partie aléatoires et en partie contrôlables.
- Un cadre qui peut résoudre la plupart des problèmes de RL.

La propriété de Markov :

- la probabilité de) L'état futur dépend uniquement de l'état actuel.
- Formulation : $\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, S_2, \ldots, S_t]$

ENSTA
IP PARIS

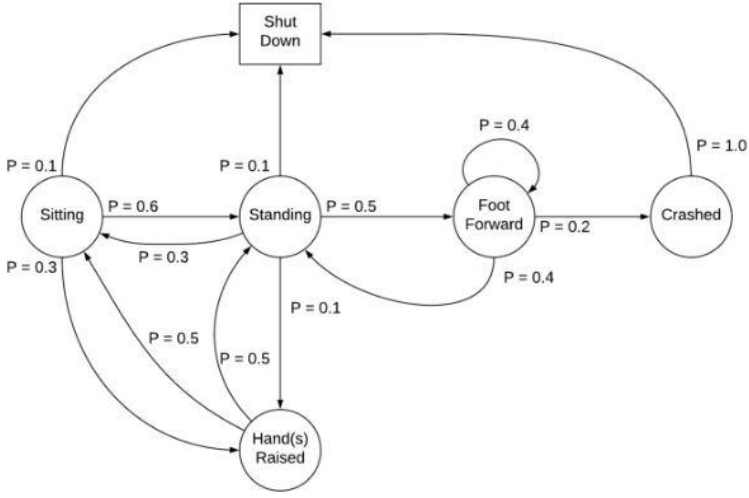
Processus de décision de Markov (MDP)

Processus de Markov (i.e. chaîne de Markov)

- Défini par un tuple $< S, P >$
  - S : Un ensemble (fini) d'états, qui obéissent à la propriété de Markov.
  - P : Une matrice de probabilité de transition d'état.
- Formulation : $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$

Processus de décision de Markov (MDP)

Un exemple de chaîne de Markov pour l'exemple du robot :



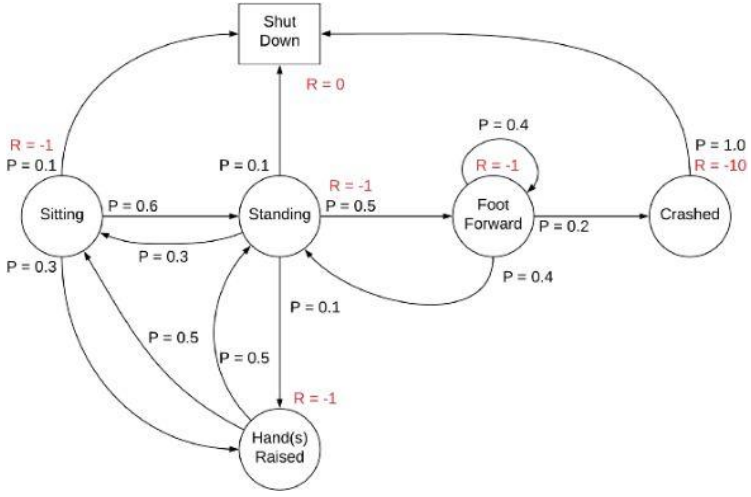
Processus de décision de Markov (MDP)

Processus de récompense de Markov (MRP)

- Prise de décision : Qu'est-ce qui est bon et qu'est-ce qui est mauvais ?
- Défini par un tuple $< S, P, R, \gamma >$
  - S : Un ensemble (fini) d'états.
  - P : Une matrice de probabilité de transition d'état.
  - R : Une fonction de récompense, $\mathcal{R}_s = \mathbb{E}[R_{t+1}|S_t = s]$.
  - γ : Un facteur d'actualisation (hyperparamètre), $\gamma \in [0, 1]$.
- *Petite question : Pourquoi R_{t+1} mais pas R_t ?*

Processus de décision de Markov (MDP)

Un exemple de processus de récompense de Markov pour l'exemple du robot :



Processus de décision de Markov (MDP)

Assemblage final :

- $< P, R >$: Aléatoire ou contrôlable ? $\implies$ Action !
- Défini par un tuple $< S, A, P, R, \gamma >$
  - S : Un ensemble (fini) d'états.
  - A : Un ensemble (fini) d'actions.
  - P : Une matrice de probabilité de transition d'état, $\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$.
  - R : Une fonction de récompense, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$.
  - γ : Un facteur d'actualisation (hyperparamètre), $\gamma \in [0, 1]$.

ENSTA
IP PARIS

Processus de décision de Markov (MDP)

Retour

- L'objectif d'un problème de RL ? $\Longrightarrow$ Défini par un signal de récompense.
- L'objectif de l'agent : Maximiser la récompense totale (qu'il reçoit sur le long terme).
  - Les récompenses sont **temporaires**.
  - Une récompense temporairement importante **ne signifie pas** une récompense totale plus importante.
- La récompense totale, c.-à-d. le retour :
  $G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
- *Petite question : Pourquoi γ ?*

ENSTA
IP PARIS

Processus de décision de Markov (MDP)

Facteur d'actualisation

- Motivation 1 : Les récompenses futures sont incertaines.
- Motivation 2 : Pour la commodité mathématique.

En pratique, une valeur plus faible encourage la réflexion à court terme, tandis qu'une valeur plus élevée met l'accent sur les résultats à long terme (plus prévoyant).

ENSTA
IP PARIS

Processus de décision de Markov (MDP)

Politique

- Que veut apprendre le RL ? $\implies$ Une politique (notée π, un modèle de résolution de problèmes).
- Une politique définit la réflexion qui sous-tend la prise d'une décision (choisir une action) et définit donc le comportement d'un agent RL.
- Formellement, une politique est une distribution de probabilités sur l'ensemble des actions a, étant donné l'état actuel s : $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$
- Les actions très gratifiantes auront une probabilité élevée et vice versa.

Processus de décision de Markov (MDP)

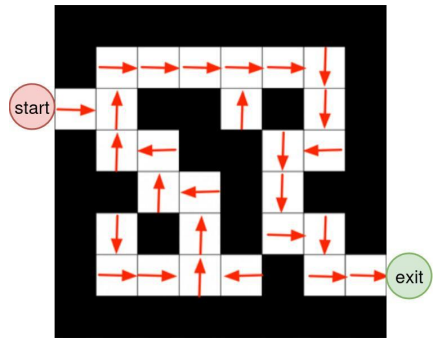
Fonction de valeur (d'état)

- Que cherche à optimiser le RL ? $\Longrightarrow$ La valeur à long terme d'un état ou d'une action.
- La fonction de valeur d'état s est le retour attendu à partir de l'état s : $v(s) = \mathbb{E}[G_t \mid S_t = s]$
- $v(s)$ estime à quel point il est bon pour l'agent d'être dans un état donné.
- $v(s)$ définie par rapport à π.

Processus de décision de Markov (MDP)

Équation de Bellman

- Une représentation standard des fonctions de valeur.
- La base de nombreux algorithmes de RL.
- Décrire la **relation récursive** entre la fonction de valeur d'état et la fonction de valeur d'action.



Processus de décision de Markov (MDP)

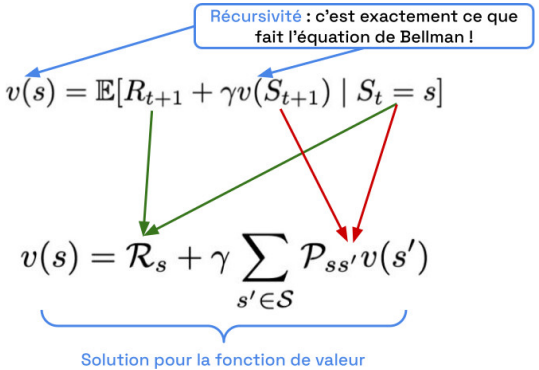
Équation de Bellman

- Elle décompose la fonction de valeur en deux composantes :
  - La récompense immédiate R_{t+1}.
  - La valeur actualisée de l'état futur $\gamma v(S_{t+1})$.

$$
\begin{aligned}
v(s) &= \mathbb{E}[G_t \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3}) + \cdots \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]
\end{aligned}
$$

Processus de décision de Markov (MDP)

Équation de Bellman



Processus de décision de Markov (MDP)

Équation de Bellman

- MRP $\rightarrow$ MDP :
  - **Fonction de valeur d'état pour MDP** : Le retour attendu à partir de l'état s, puis en suivant la politique π :
    $v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$
  - **Fonction de valeur d'action pour MDP** : Le retour attendu à partir de l'état s, en effectuant l'action a, puis en suivant la politique π :
    $q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$
- Bellman pour MDP :

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

$$q_\pi(s, a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

ENSTA
IP PARIS

Processus de décision de Markov (MDP)

Équation de Bellman

- Les solutions pour la fonction de valeur d'état et la fonction de valeur d'action :

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) q_\pi(s, a)$$

$$q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

- **Récursivité :** Remplacer la fonction de valeur d'action dans la fonction de valeur d'état et vice versa :

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

$$q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a' \mid s') q_\pi(s', a')$$

Processus de décision de Markov (MDP)

Fonctions de valeur optimale MDP

- On peut choisir la politique avec la valeur la plus élevée pour les états et les actions :

$$v_*(s) = \max_\pi v_\pi(s) \quad q_*(s,a) = \max_\pi q_\pi(s,a)$$

$$\pi_*(a \mid s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in \mathcal{A}} q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Équation d'optimalité de Bellman

$$v_*(s) = \max_a q_*(s,a)$$

$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

ENSTA IP PARIS

Q-learning

1. Initialiser $Q(s, a)$ pour tous les états s et actions a (par exemple, à 0)
2. Définir le taux d'apprentissage α ($0 < \alpha \leq 1$)
3. Définir le facteur d'actualisation γ ($0 < \gamma \leq 1$)
4. **Répéter** (pour chaque épisode) :
5. Initialiser l'état s
6. **Répéter** (pour chaque étape de l'épisode) :
7. Choisir l'action a parmi s en utilisant une politique ε-greedy
8. Prendre l'action a, observer la récompense r et l'état suivant s'
9. Mettre à jour $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
10. $s \leftarrow s'$
11. **Jusqu'à ce que** s soit terminal
12. **Jusqu'à ce que** la condition de résiliation soit remplie

ENSTA
IP PARIS

Q-learning

- L'un des algorithmes de RL les plus représentatifs.
- Q signifie **"qualité"**, correspondant à l'obtention d'une estimation de la récompense $r(s, a)$.
- Créer une table et la maintenir (ligne 1 de l'algorithme), appelée **Q-table** :

Q-table	**a1**	**a2**
s1	$Q(s1, a1)$	$Q(s1, a2)$
s2	$Q(s2, a1)$	$Q(s2, a2)$

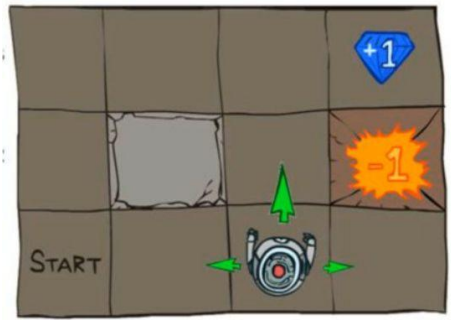
Q-learning

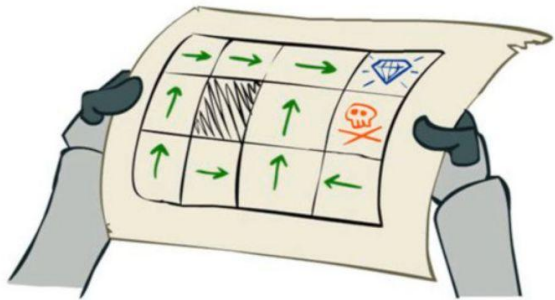
- α : Plus la valeur est grande, plus l'agent apprend.
- γ : Plus la valeur est grande, plus l'agent est prévoyant.
- Épisode : Le nombre de tours que l'agent va tenter.
- État : L'environnement dans lequel se trouve l'agent à un moment donné.
- Étape : La réalisation d'une action est considérée comme une étape.
- Action : Prise par l'agent qui affecte l'environnement.
- Politique ε-greedy :

$$\pi(a \mid s) = \begin{cases} \arg\max_a Q(s,a) & \text{probability } 1-\varepsilon \\ \text{a random action from } A(s) & \text{probability } \varepsilon \end{cases}$$

- r : Récompense.

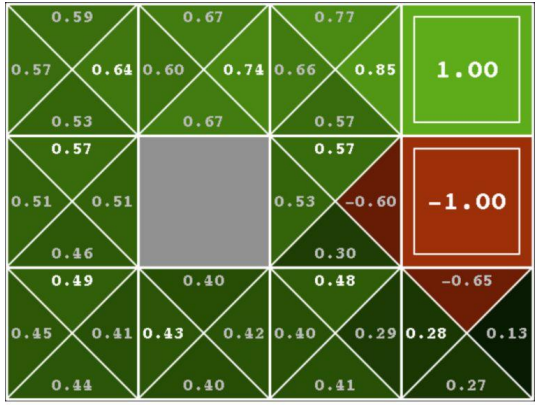
Q-learning





Source: P. Abbeel and D. Klein

Q-learning



Q-values after 100 iterations

Q-learning

- Zoom sur la ligne 9 :

$$Q(s,a) \leftarrow Q(s,a) + \alpha[\underbrace{r + \gamma \max_{a'} Q(s',a')}_{\text{estimated future rewards}} - \underbrace{Q(s,a)}_{\text{current Q-value}}]$$

$$\text{error}$$

- Tracer la source :

$$Q_\pi(s_t, a_t) = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid s_t, a_t]$$

- Objective :

$$\max_\pi \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t R(S_t, A_t, S_{t+1}) \mid \pi\right]$$

ENSTA
IP PARIS

Fin