INF8175 - Intelligence artificielle

Méthodes et algorithmes

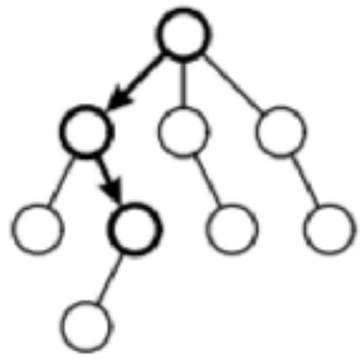
Module 9: Apprentissage par renforcement

POLYTECHNIQUE
MONTRÉAL

Quentin Cappart

Contenu du cours

Raisonnement par recherche (essais-erreurs avec de l'intuition)

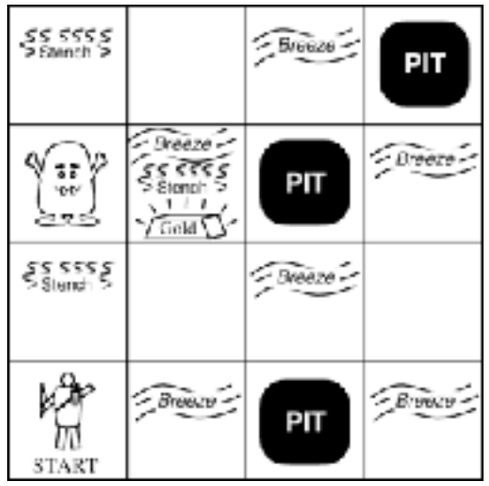


Module 1: Stratégies de recherche

Module 2: Recherche en présence d'adversaires

Module 3: Recherche locale

Raisonnement logique



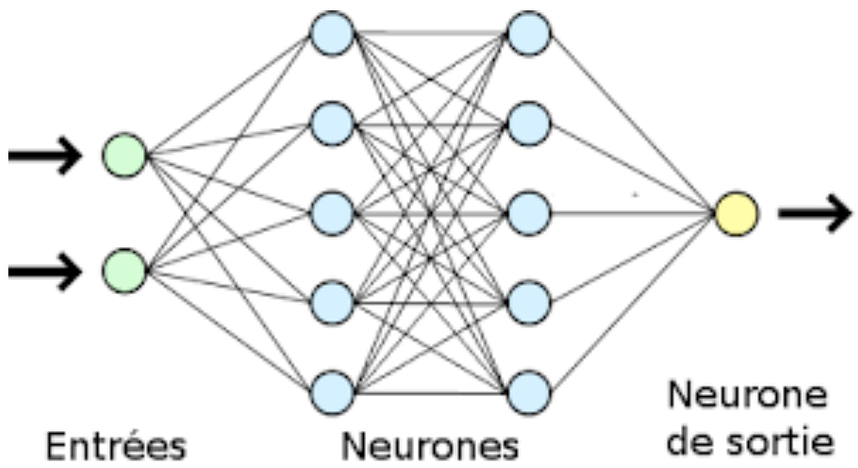
Module 4: Programmation par contraintes

Module 5: Agents logiques

Raisonnement par apprentissage

Module 6: Apprentissage supervisé

Module 7: Réseaux de neurones et apprentissage profond



Module 8: Apprentissage non-supervisé

Module 9: Apprentissage par renforcement

Considérations pratiques et sociétales

Module 10: Utilisation en industrie, éthique, et philosophie

Table des matières

Apprentissage par renforcement

Ce module ne fait pas partie de la matière d'examen

Caractéristiques de l'apprentissage par renforcement

Apprentissage supervisé

L'apprentissage se faisait à l'aide de données labellisées

$$\mathrm{D} : \left\{ \left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right) \right\}$$

L'objectif est d'arriver à apprendre sur base de données dont on connait la vraie valeur

Apprentissage non-supervisé

Apprentissage qui se fait uniquement sur base de données non-labellisées

$$\mathrm{D} : \left\{ x^{(1)}, x^{(2)}, \ldots, x^{(m)} \right\}$$

L'objectif est d'utiliser la similarité entre les données pour orienter l'apprentissage

Apprentissage par renforcement

Situations où un agent doit trouver la meilleure séquence de décisions pour remplir un objectif

L'apprentissage de l'agent se fait sur base d'un processus d'essais-erreurs

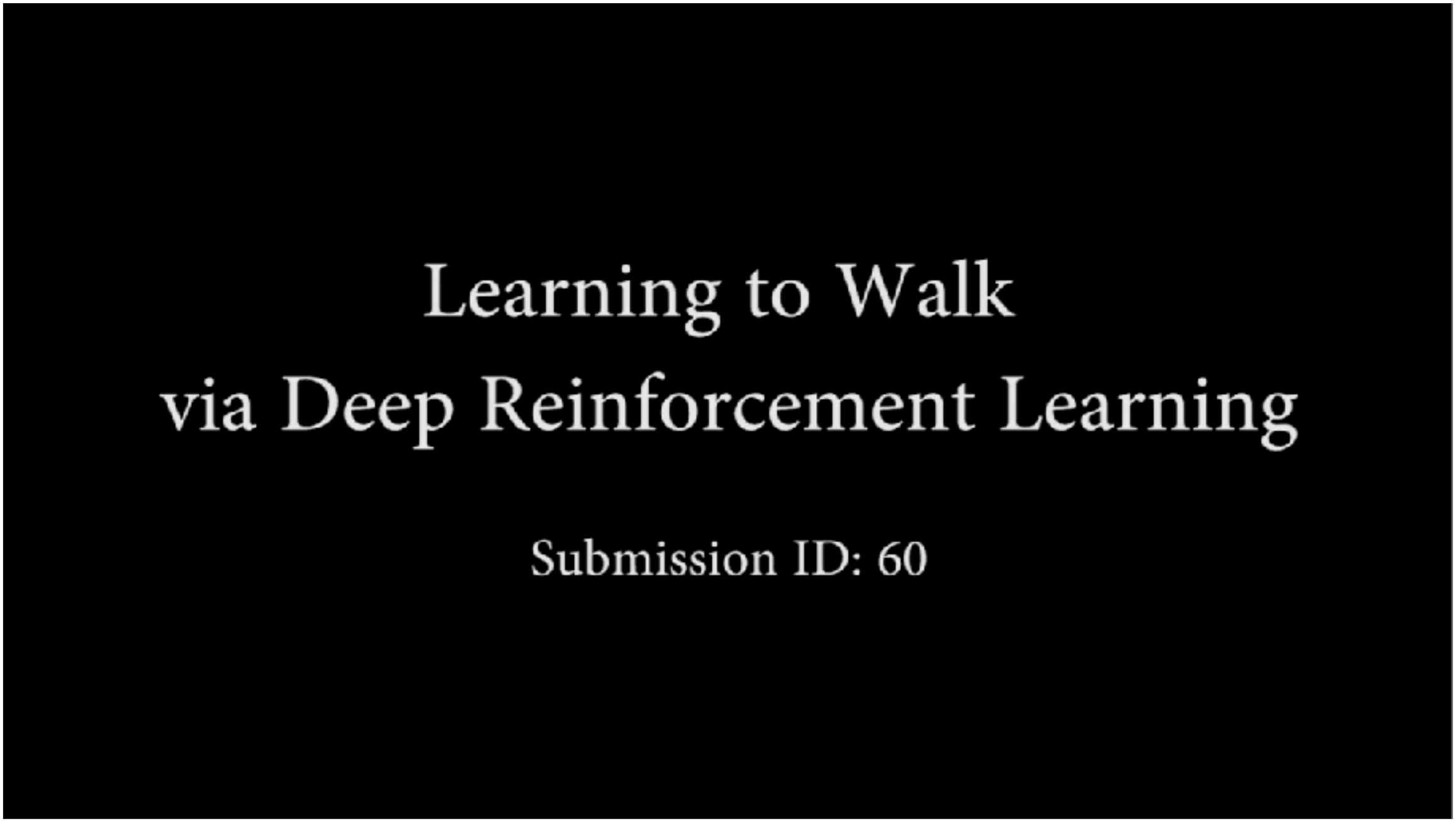
L'agent agit dans un environnement donné

Chaque action faite est susceptible de modifier l'état de l'environnement

L'agent va collecter de nouvelles données au fur et à mesure de ses essais

Requiert un moyen d'évaluer la qualité des essais que l'agent effectue

Application en robotique



https://www.youtube.com/watch?v=n2gE7n11h1Y

Learning to Walk via Deep Reinforcement Learning. Tuomas Haarnoja et al. Robotics: Science and Systems (RSS). 2019.

AlphaGo: Match contre Lee Sedol

https://www.youtube.com/watch?v=WXuK6gekU1Y

Notez la surprise générale lors du mouvement d'AlphaGo

Illustration d'un problème de décisions séquentielles

Processus de décisions séquentielles

Effectuer la meilleure séquence de décisions afin de réaliser un objectif

Exemple

Pacman (Première version du jeu date de 1980)

Objectif

Collecter le plus de points avant de se faire manger

Etat (*state*)

Toutes les informations concernant le plateau

Tout ce qui est pertinent pour la prise de décision

Position du Pacman, fantômes, bonus, etc.

Action

Décisions que le joueur peut faire

Une action va changer l'état actuel

Une action va engendrer un signal de récompense

Mouvement à gauche, droite, haut, bas ou rien faire

Récompense (*reward*)

Signal utilisé pour orienter la prise de décision

Valeur négative pour une mauvaise action

- 100 si on se déplace sur un fantôme

Valeur positive pour une bonne action

+ 10 si on prend un bonus

Vue globale d'un processus de décisions séquentielles

Agent

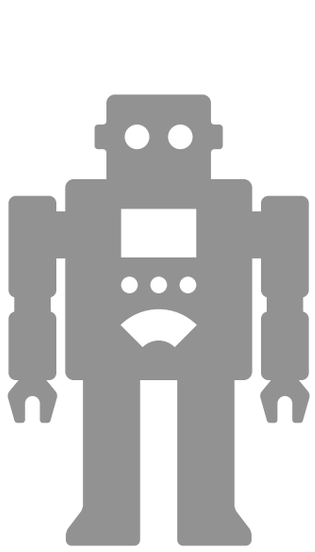
L'intelligence prenant les décisions (ou effectuant les actions) afin d'accomplir son objectif

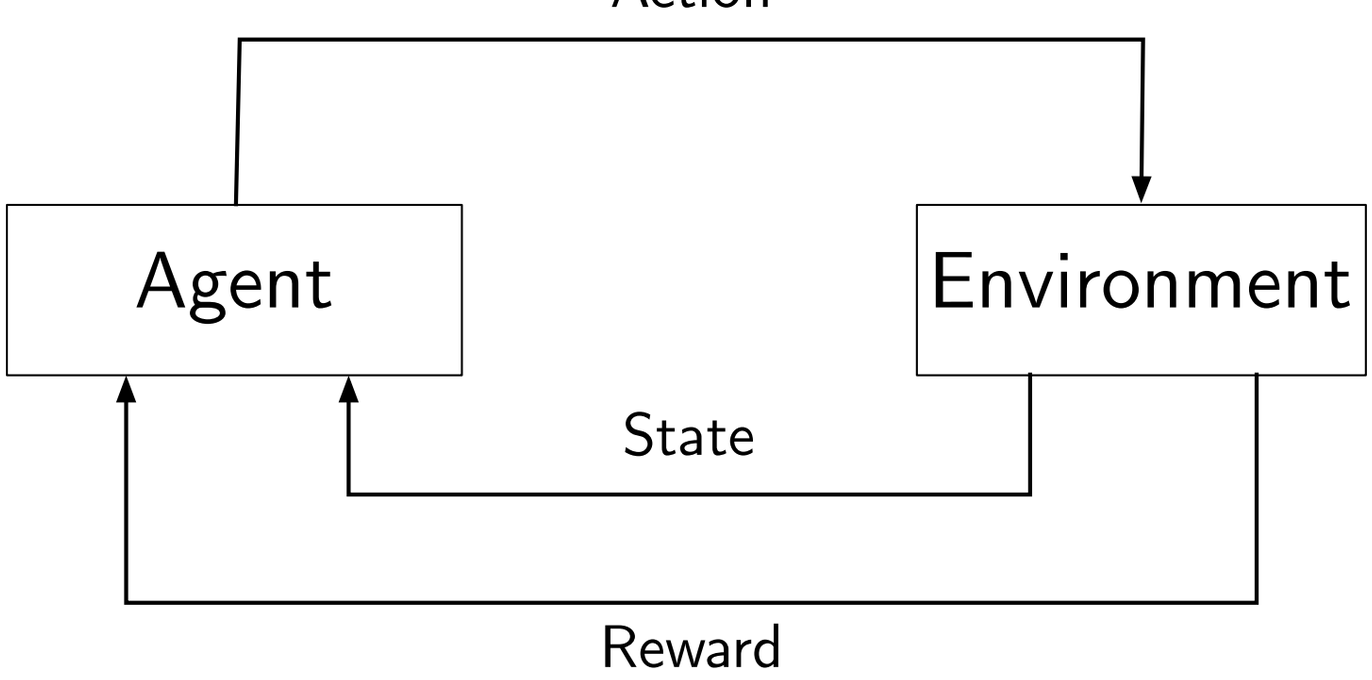
Environnement

Le monde dans lequel l'agent va interagir

Définit la dynamique du monde (états du système, les actions possibles, l'impact des actions, etc.)

Englobe toutes les informations utiles et pouvant être disponibles pour la prise de décisions





L'agent et l'environnement sont en continuelle interaction

L'agent modifie l'environnement par ses actions

L'environnement présente des situations différentes à l'agent

Récompense dans un processus de décisions séquentielles

Interprétation de la récompense

Signal numérique qui indique à quel point l'agent a bien agi pour un certain instant

Element fondamental qui est utilisé pour formaliser l'objectif à atteindre

Hypothèse de la récompense

N'importe quel objectif à atteindre peut être décrit comme une maximisation des **récompenses cumulées espérées** de toutes les actions effectuées.

Maximisation des récompenses

L'objectif correspond ainsi à maximiser le total des récompenses cumulées espérées

Hypothèse fondamentale en apprentissage par renforcement

? Que pensez vous de cette hypothèse ?

Peut sembler restrictif, mais est assez flexible et applicable dans beaucoup de situations

Cependant, il peut être difficile de définir certaines tâches en terme de récompenses...

Note

On parle de récompense *espérée*, car certaines actions peuvent donner différentes récompenses

Exemples de récompenses

Jeu du casse-briques

Le score du jeu (fonction du nombre de briques cassées)

Pousse l'agent à maximiser son score

Portefeuille d'investissement

+x pour chaque dollar gagné par vos placements

-x pour chaque dollar perdu par vos placements

Jeu d'échec

+1 en cas de victoire

0 en cas d'égalité

-1 en cas de défaite

? Que pensez vous de l'idée d'intégrer des connaissances du jeu (heuristiques) dans la fonction de récompense ?

Exemples de récompenses (mauvaise conception)

CoastRunner 7

- Jeu de course avec des bateaux
- Besoin de passer par différents checkpoints
- Plusieurs bonus peuvent être collectés (turbo, etc.)

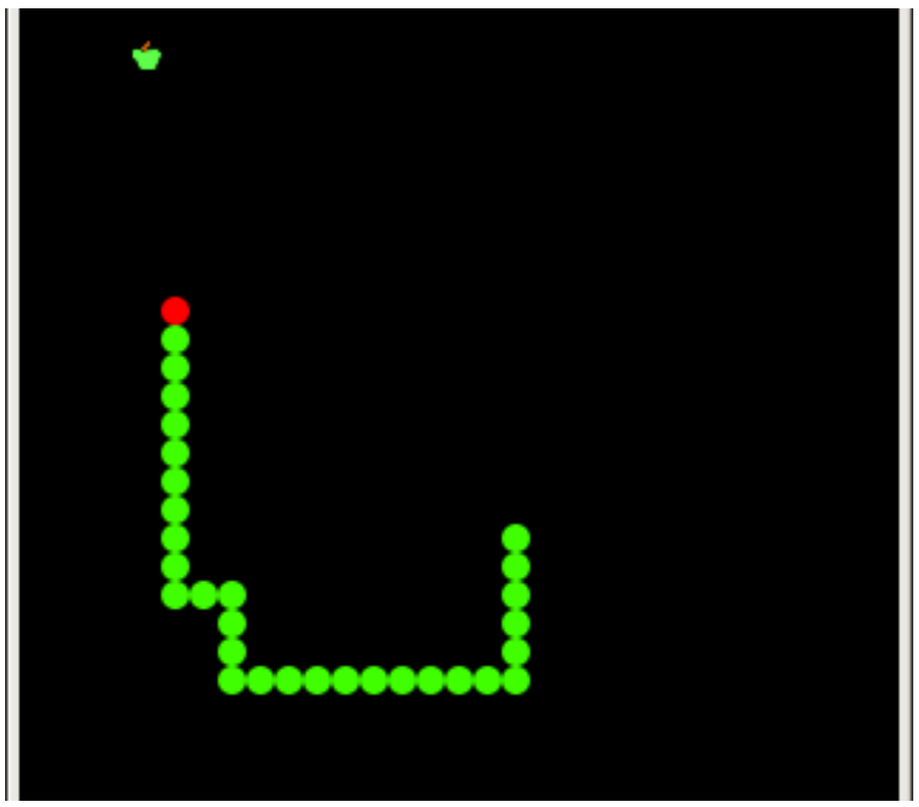
Signal de récompense

- Récompense positive si on passe par un checkpoint
- Récompense positive si on prend un bonus

? Que pensez-vous de cette conception ?

https://openai.com/blog/faulty-reward-functions/

L'agent préfère collecter les bonus plutôt que finir la course



Snake

- Collecter le maximum de points

Signal de récompense

- +1 si on collecte un fruit
- -1 pour chaque déplacement (pour aller sur le fruit au plus vite)

? Que pensez-vous de cette conception ?

L'agent va être pousser à se suicider pour limiter les pertes

On récompense des sous-objectifs basés sur notre connaissance du jeu, et non l'objectif final

Formalisation d'une fonction objectif

Objectif

Maximiser le total des récompenses cumulées

? Que pensez-vous de prendre, à chaque étape, l'action qui donne le plus de récompenses dans l'immédiat ?

Récompense à long terme

Maximiser chaque récompense immédiate ne revient pas à maximiser la récompense globale

Les actions faites peuvent avoir un impact à long terme

Les meilleures récompenses peuvent arriver très tard, et être conditionnelles à des actions effectués tôt

Souvent, il est préférable de sacrifier une récompense immédiate, pour une meilleure au long terme

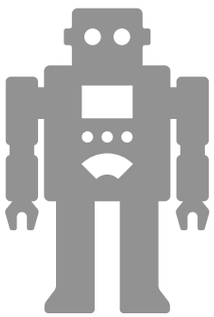
? Avez-vous des exemples concrets ?

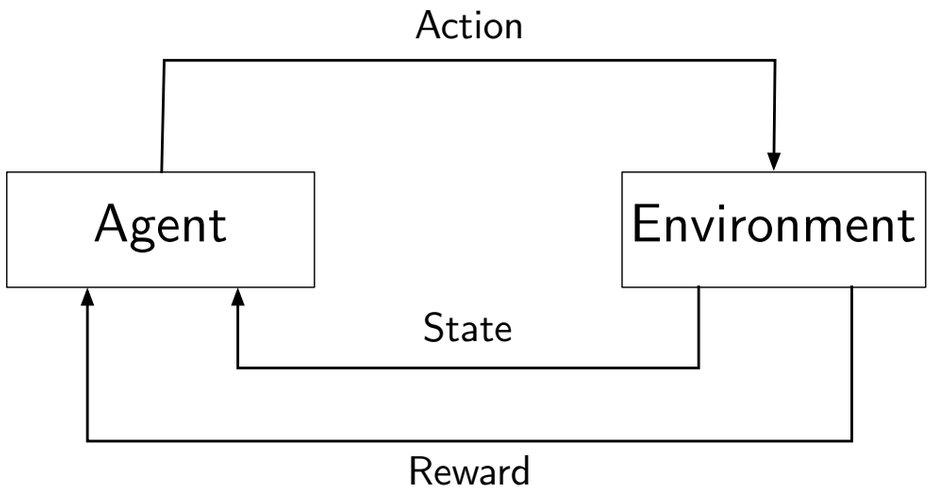
Exemples

Un investissement engendre des frais immédiats, mais peut prendre des années pour être profitable

Effectuer une maintenance a un coût, mais peut empêcher un dysfonctionnement dans les prochains jours

Processus de décision de Markov (MDP)





Ensembles essentiels

S : **l'ensemble de tous les états possibles**

A : **l'ensemble de toutes les actions possibles**

Formalisation d'un MDP

$t = 1,2,3,\ldots$: **instants temporels (time step)**

S_t : **l'état de l'environnement au temps** t

A_t : **l'action prise par l'agent au temps** t

$r : S \times A \to \mathbb{R}$: **la fonction de récompense, donnant une valeur à l'action faite à partir d'un certain état**

$p : S \times A \times S \to [0,1]$: **la fonction de probabilité d'aller d'un état à un autre en faisant une certaine action**

Hypothèse fondamentale d'un MDP

Chaque état est markovien

Etat markovien

Etat Markovien

Un état est markovien s'il contient toutes les informations utiles du passé

La probabilité d'aller à un autre état ne dépend que de notre état présent

$$\mathbb{P}(S_{t+1} \mid S_t) = \mathbb{P}(S_{t+1} \mid S_1, \ldots, S_t)$$

Intuitivement, l'état actuel contient toutes les informations utiles du passé

Une fois que l'on a notre état actuel, on peut jeter l'historique

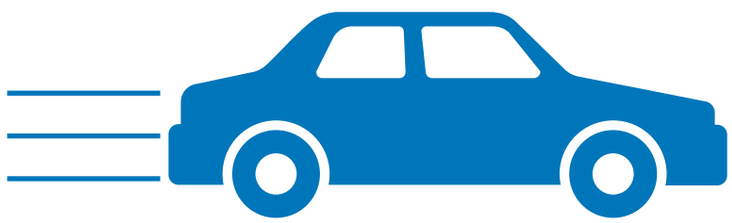
Hypothèse réaliste, et simplifiant énormément les algorithmes de décisions

Andrey Markov (1856-1922)

? Que faire en cas d'états non-markoviens ?

Augmenter l'état avec de nouvelles informations

Exemple: voiture en mouvement



Etat incomplet: position

Connaissant uniquement la position de la voiture, peut-on prédire la position ?

Etat amélioré: position, vitesse, accélération, etc.

Variantes des MDPs

Processus de décision de Markov stochastique

Chaque action comporte de l'aléatoire, et peut vous amener à différents états

Il en va de même pour la récompense

C'est pourquoi on parle de maximisation de la *récompense cumulée espérée*

Variante généralement considérée dans la formalisation standard

Processus de décision de Markov partiellement observable (POMDP)

L'agent n'a qu'une vision incomplète de l'environnement

L'agent ne reçoit qu'une observation de l'état de l'environnement

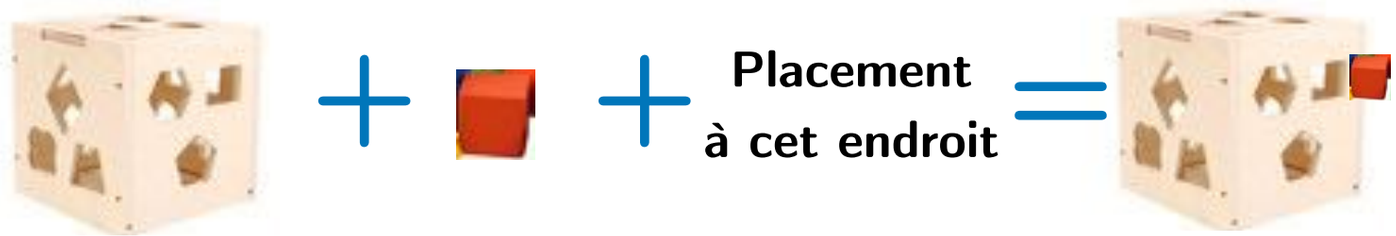
L'agent doit construire sa propre représentation de son état

Composantes d'un agent d'apprentissage par renforcement

https://cf.shopee.ph/file/673d6d6c4e658a40ea481efda94820e4_tn

Modèle de l'environnement (*model*)

La représentation que l'agent se fait de l'environnement



Fonction de valeur (*value function*)

La représentation que l'agent se fait de la qualité de chaque état et de chaque action

Politique de sélection d'actions (*policy*)

La décision de l'action à prendre étant donné notre état

...

Chaque agent de RL est basé sur une ou plusieurs de ces composantes

? Est-ce qu'un être humain peut être modélisé par ces trois composantes ?

Modèle de l'environnement (*model*)

Intuition

Représentation qu'a l'agent sur le fonctionnement de l'environnement

Reprend les informations sur l'effet des actions (modification de l'état, et récompenses collectées)

Fonction de transition

Probabilité d'aller sur un état suivant à partir d'une action faite à l'état actuel

$$p : S \times S \times A \to [0,1]$$

$$p(s' \,|\, s, a) = \mathbb{P}\left[S_t = s' \,|\, S_{t-1} = s, A_{t-1} = a\right]$$

Fonction de récompense

Espérance de la récompense immédiate si une action est faite à partir de l'état actuel

$$r : S \times A \to \mathbb{R}$$

$$r(s, a) = \mathbb{E}\left[R_t \,|\, S_t = s, A_t = a\right]$$

Commentaires

Notez que cette formalisation tient compte d'un processus de décisions stochastiques

En général, l'agent ne sait pas comment fonctionne l'environnement et doit le découvrir par essais/erreurs

Politique de sélection d'action (*policy*)

Intuition

Défini le comportement de l'agent

Etant donné notre état actuel, quelle action doit être prise

$$\pi : S \to A$$

Politique déterministe

Etant donné mon état, mon action sera toujours la même

$$a = \pi(s)$$

Politique stochastique

Etant donné mon état, le choix de mon action est sujet à de l'aléatoire contrôlé

$$\pi(a \mid s) : \mathbb{P}\left[A_t = a \mid S_t = s\right]$$

Notez la différence avec un environnement stochastique

Ici, c'est notre agent qui a un comportement stochastique

? En quelles situations cela peut-il être bénéfique pour l'agent ?

Parfois, la stratégie optimale implique d'agir de manière aléatoire

Fonction de valeur (*value function*)

Fonction de valeur d'un état (*state-value function*)

Représentation interne que l'agent se fait de la qualité de chaque état

Somme espérée des récompenses obtenues à partir d'un état, si on suit une certaine politique de sélection

$$V^{\pi}(s) = \mathbb{E}\left[R_t + R_{t+1} + R_{t+2} + R_{t+3} + \ldots \mid S_t = s\right]$$

Discounting factor

Valeur comprise entre 0 et 1 $\gamma \in [0,1]$

Permet de considérer le fait qu'il est préférable d'avoir une récompense tôt plutôt que tard

$$V^{\pi}(s) = \mathbb{E}\left[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \ldots \mid S_t = s\right]$$ **(formalisation générale)**

La valeur des récompenses futures décroit avec le temps

Valeur de 0: l'agent ne va considérer que la récompense immédiate

$$V^{\pi}(s) = \mathbb{E}\left[R_t \mid S_t = s\right]$$

Valeur de 1: l'agent va considérer toutes les récompenses avec la même importance

$$V^{\pi}(s) = \mathbb{E}\left[R_t + R_{t+1} + R_{t+2} + R_{t+3} + \ldots \mid S_t = s\right]$$

Fonction de valeur d'une action (action-*value function*)

On peut également construire une fonction de la qualité d'une action à partir d'un certain état

$$Q^{\pi}(s,a) = \mathbb{E}\left[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \ldots \mid S_t = s, A_t = a\right]$$ **(formalisation générale)**

Exemple: l'environnement du labyrinthe

Environnement du labyrinthe

S'échapper du labyrinthe en effectuant le moins de déplacements

Etats: la case où l'agent se trouve

$S =$ **ensemble des cases du labyrinthe**

Actions: déplacement d'une case

$A = \{$**left, right, top, bottom**$\}$

Fonctions de probabilité (ou de transition)

$p(s'|s, L) = 1$ **if** s' **is the left (L) cell of** s

$p(s'|s, L) = 0$ **otherwise**

$p(s'|s, R) = 1$ **if** s' **is the right (R) cell of** s

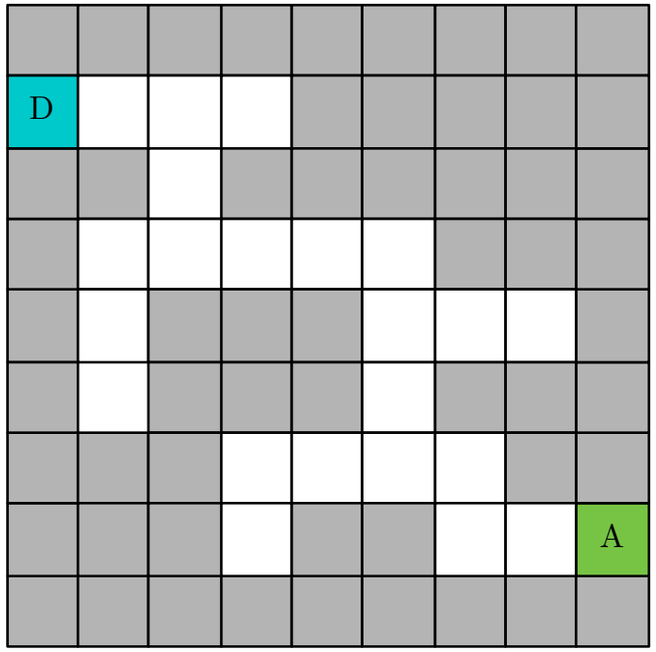
$p(s'|s, R) = 0$ **otherwise**

$p(s'|s, T) = 1$ **if** s' **is the top (T) cell of** s

$p(s'|s, T) = 0$ **otherwise**

$p(s'|s, B) = 1$ **if** s' **is the bottom (B) cell of** s

$p(s'|s, B) = 0$ **otherwise**



Fonction de récompense

$r(s, a) = -1 \quad \forall s \in S, \forall a \in A$

Exemple d'un agent d'apprentissage par renforcement

Modèle

L'agent a un accès au modèle complet de l'environnement

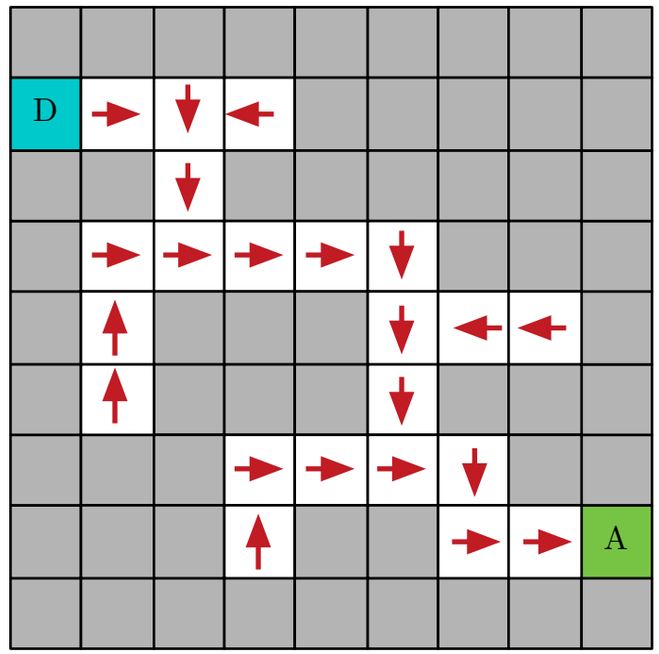
Il connait parfaitement ls fonctions de transition et de récompense

Politique de sélection

Pour chaque état, quelle action doit-être effectuée ?

Politique déterministe

$a = \pi(s)$ **(défini par les flèches)**



Fonction de valeur d'un état (*state-value function*)

Quelle est la qualité de chaque état ?

Valeur de -1 par cases pour atteindre l'arrivée depuis une certaine case

$\gamma = 1$ **(sans discount)**

$$V^{\pi}(s) = \mathbb{E}\left[R_t + R_{t+1} + R_{t+2} + R_{t+3} + \ldots \,|\, S_t = s\right]$$

D	-13	-12	-13					
		-11						
	-11	-10	-9	-8	-7			
	-12				-6	-7	-8	
	-13				-5			
			-6	-5	-4	-3		
			-7			-2	-1	A

Commentaires

Notez que l'agent considéré est déjà entraîné

On n'a pas encore abordé la manière de faire l'apprentissage

Catégorisation d'agents : avec ou sans modèle

? Comment créer un agent qui résout ce genre de problèmes ?

Plusieurs façons et choix de conception possibles

Un premier critère de distinction est l'utilisation/ou non d'un modèle complet de l'environnement

Agent utilisant un modèle (*model-based agent*)

Contient dans sa représentation un modèle complet de comment l'environnement fonctionne

Les fonctions de transition et de récompense sont entièrement connues

Donne la possibilité d'utiliser des algorithmes de recherche (A* par exemple)

Agent n'utilisant pas un modèle *(Model-free agent)*

Un modèle de l'environnement n'est pas disponible pour l'agent (ou pas utilisé)

L'agent ne peut apprendre le fonctionnement de l'environnement que par ses expériences

Comme vous pouvez vous y attendre, c'est une tâche très difficile

Rend difficile l'utilisation d'algorithmes de recherche

L'objectif est d'apprendre par des essais-erreurs une politique de sélection ou une fonction de valeur

Planification et apprentissage

Résolution d'un MDP

- Deux techniques fondamentales pour résoudre des problèmes de décision séquentiels

Apprentissage par renforcement

- Le modèle de l'environnement est inconnu
- L'agent interagit avec l'environnement afin d'améliorer sa politique de sélection
- La résolution est gouvernée par de l'apprentissage essais-erreurs

Planification (planning)

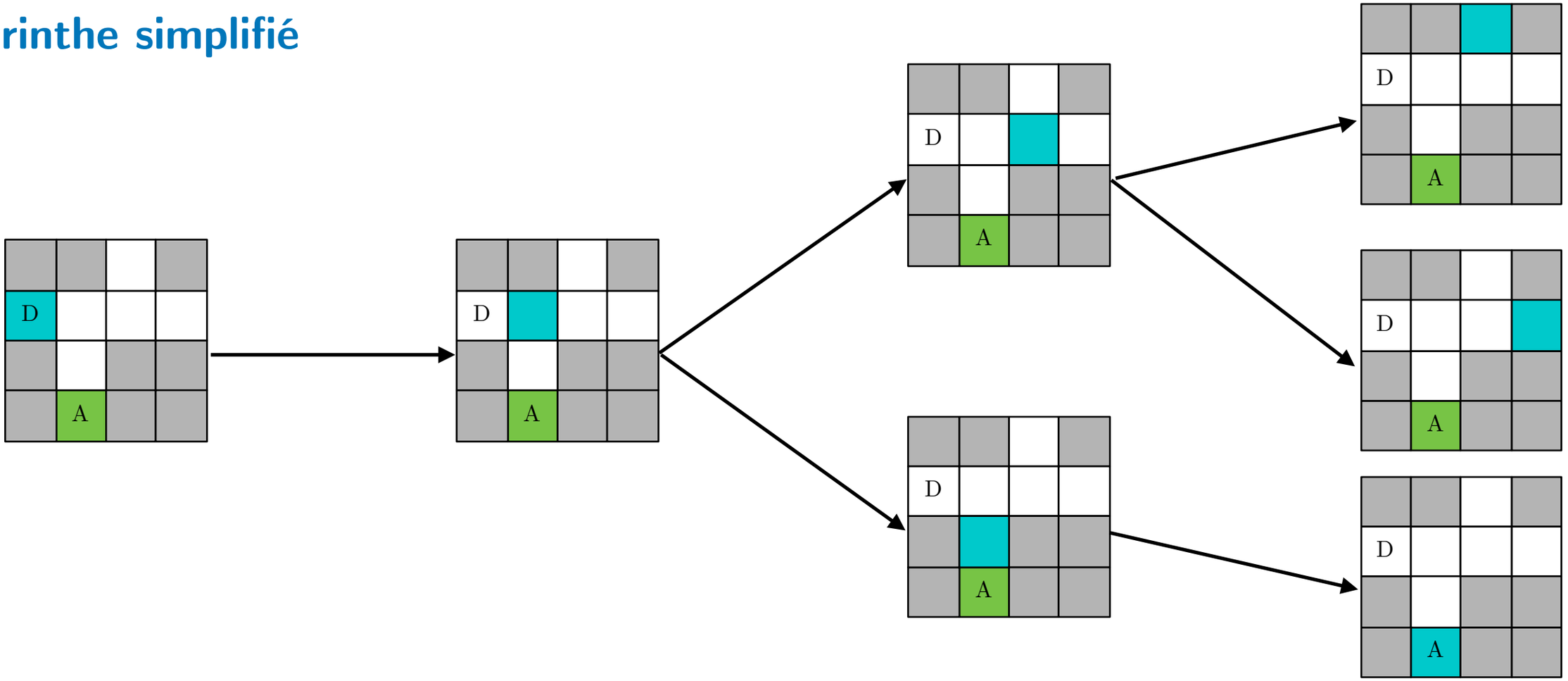
- Suppose que le modèle de l'environnement est connu
- L'agent utilise ce modèle pour améliorer sa politique de sélection
- Peut prendre par exemple la forme d'exploration dans un arbre d'états
- La résolution est gouvernée par un algorithme de recherche exploitant le mieux le modèle
- La difficulté est que le nombres de possibilités est exponentiellement large

Approches hybrides

- Utilisation de l'apprentissage par renforcement pour apprendre un modèle de l'environnement
- Résolution du problème en utilisant des algorithmes de planification

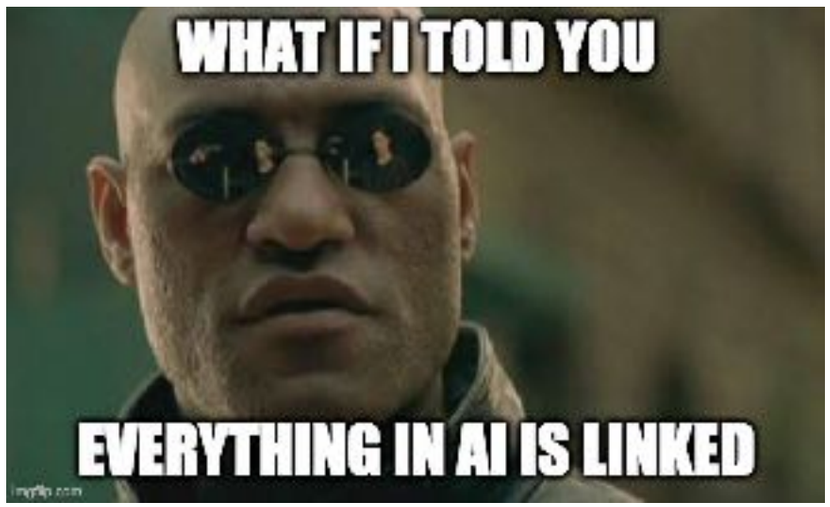
Résolution par planification

Labyrinthe simplifié



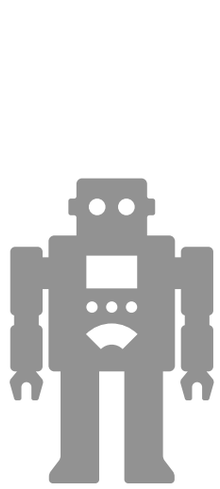
Algorithmes de résolution

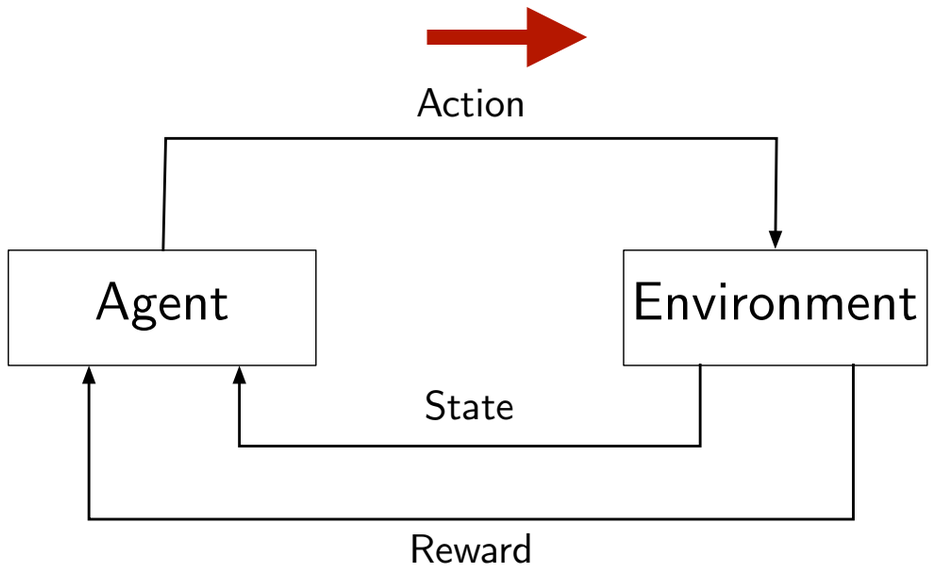
- Programmation dynamique (policy iteration, value iteration)
- Algorithmes de recherche (A*, minimax, monte carlo tree search)
- La difficulté vient du fait que le nombre d'états est en général très grand

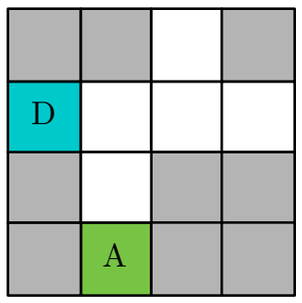


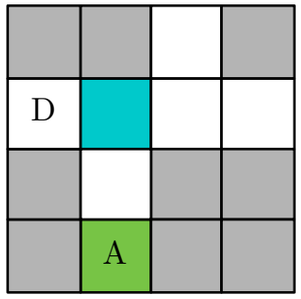
Résolution par apprentissage par renforcement

Labyrinthe simplifié









Récompense: -1

Le labyrinthe est inconnu: l'agent doit le découvrir par lui même

Algorithmes de résolution

Algorithmes basés sur l'apprentissage d'une fonction de valeur (Q-learning, SARSA, etc.)

Algorithmes basés sur l'apprentissage d'une politique de sélection (policy gradient, A3C, PPO, etc.)

Catégorisation d'agents : value-based et policy-based agents

Agent apprenant la fonction de valeur (*value-based agent*)

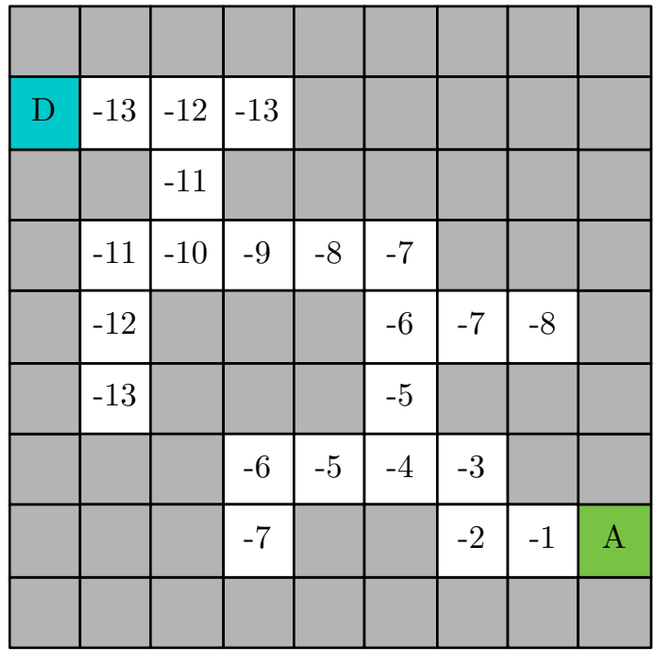
L'agent se borne à apprendre la meilleure fonction de valeur possible

L'objectif est d'avoir la meilleure estimation de la qualité de chaque état

Avec cette information, la policy devient implicite

? Quelle est cette fonction de sélection implicite ?

Chaque fois prendre l'action amenant à l'état qui a la meilleure qualité



$$\pi(a \mid s) = \textbf{argmax}_a \left(Q^\pi(s, a) \right) \ \forall s \in S$$

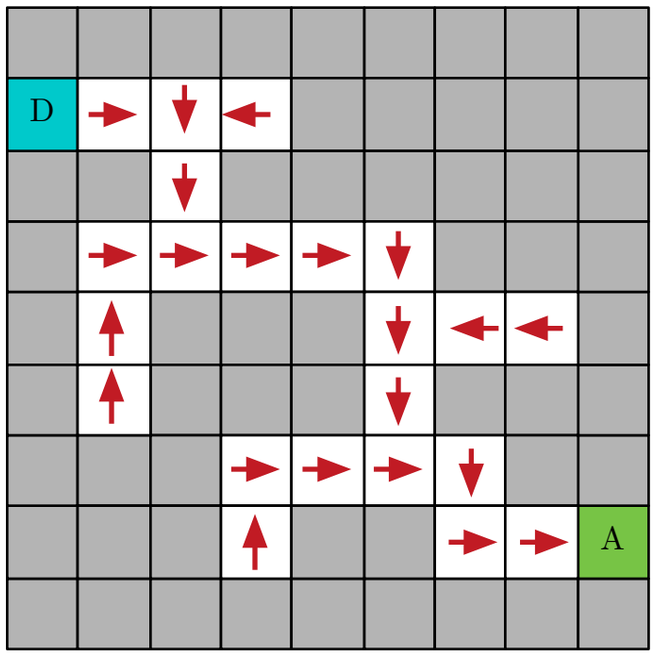
Agent apprenant la fonction de sélection (*policy-based agent*)

La fonction de valeur n'est qu'une étape intermédiaire

Ce qui nous intéresse, au final, c'est la politique de sélection

L'agent tente d'apprendre directement la meilleure politique de sélection

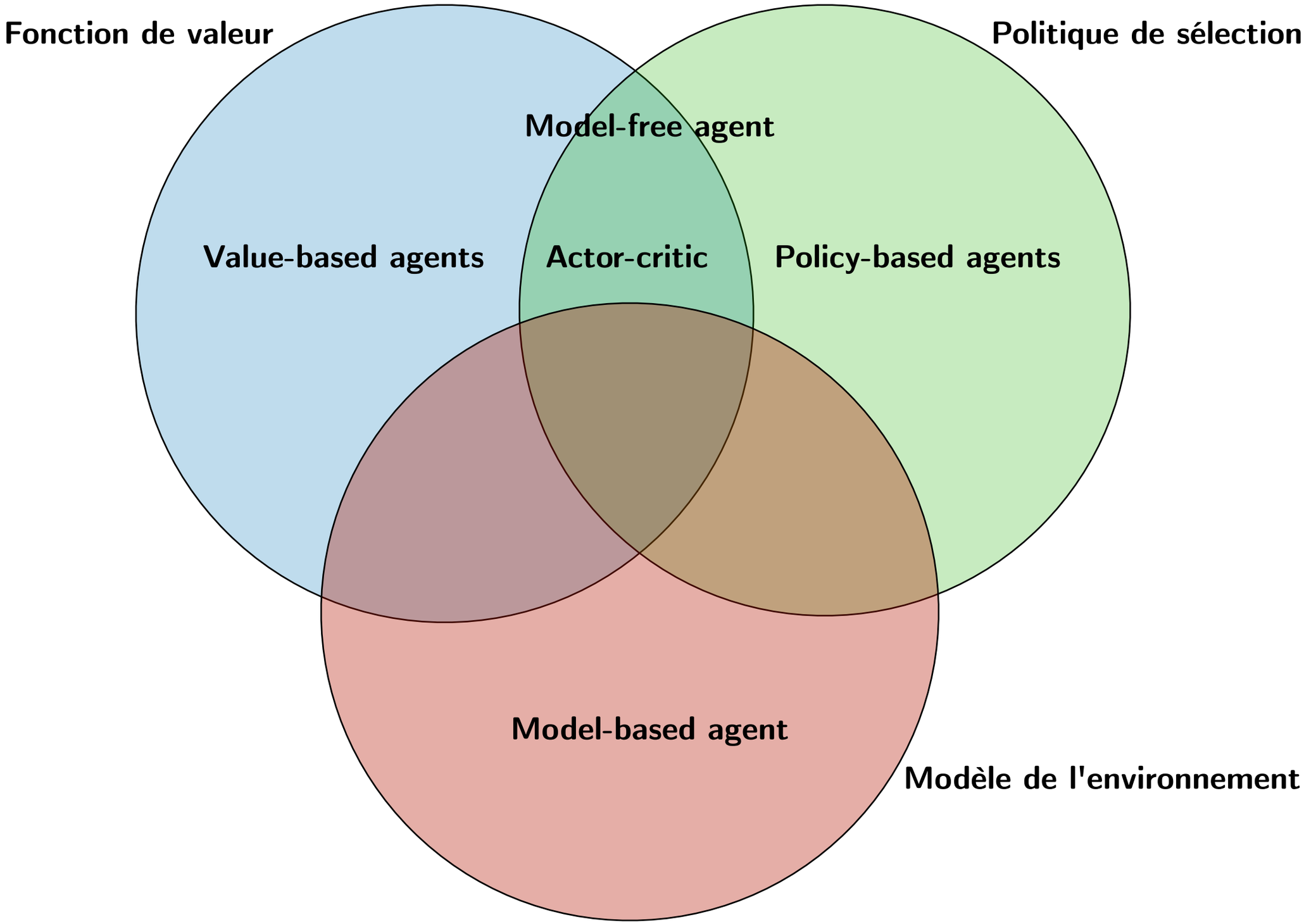
N'utilise pas de fonction de valeurs



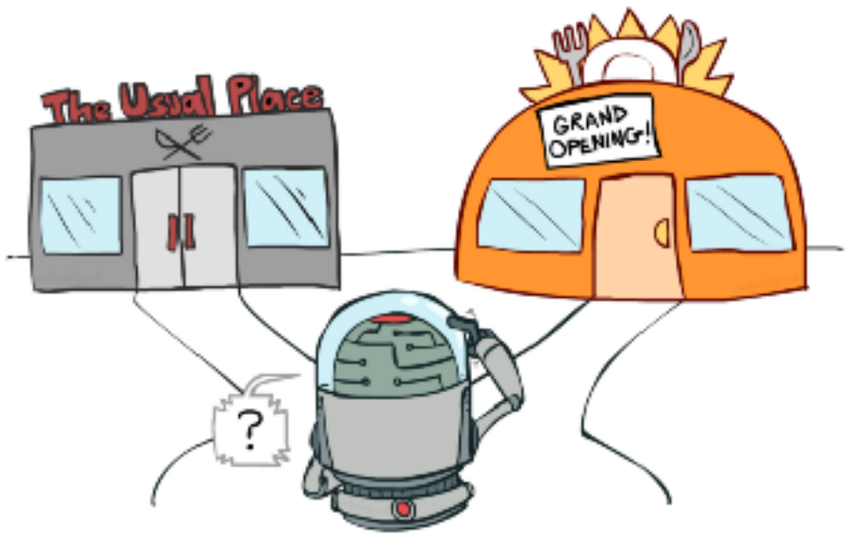
Agent hybride (*actor-critic*)

L'agent utilise à la fois la fonction de valeur et de sélection

Taxonomie des agents de RL



Compromis entre l'exploration et l'exploitation



http://ai.berkeley.edu/home.html

Intuition

L'apprentissage par renforcement se fait par essais-erreurs

L'agent doit découvrir une bonne politique à partir d'expériences

Nécessite d'effectuer des actions amenant à de bonnes récompenses

Mais aussi d'explorer des zones inconnues, pouvant être prometteuses

Exploration

L'objectif est d'acquérir de l'information sur l'environnement

Exploitation

L'objectif est d'exploiter l'information afin de maximiser la récompense

Compromis

Un bon agent intègre un compromis entre ces deux aspects

Exploitation pure: maximisation de la récompense sur base d'une vision incomplète de l'environnement

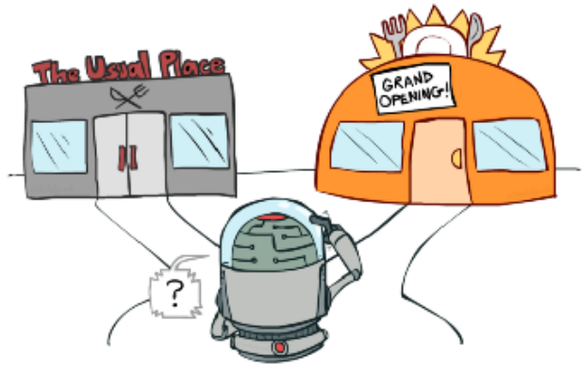
Exploration pure: aucune maximisation de la récompense n'est faite

Créer un agent ayant un bon compromis est un défi majeur en apprentissage par renforcement

Exemples de compromis

Choix d'un restaurant

- Aller continuellement au même restaurant que l'on sait bon
- Essayer un nouveau restaurant qui vient d'ouvrir



http://ai.berkeley.edu/home.html

Extraction de matières précieuses (cuivre, fer, or, etc.)

- Miner dans une zone où on a déjà trouvé des filons
- Miner dans une nouvelle zone
- Compétition de RL sur Minecraft: https://minerl.io/

Campagne de marketing

- Utiliser des publicités que l'on sait efficaces
- Essayer des nouveaux prototypes de publicités

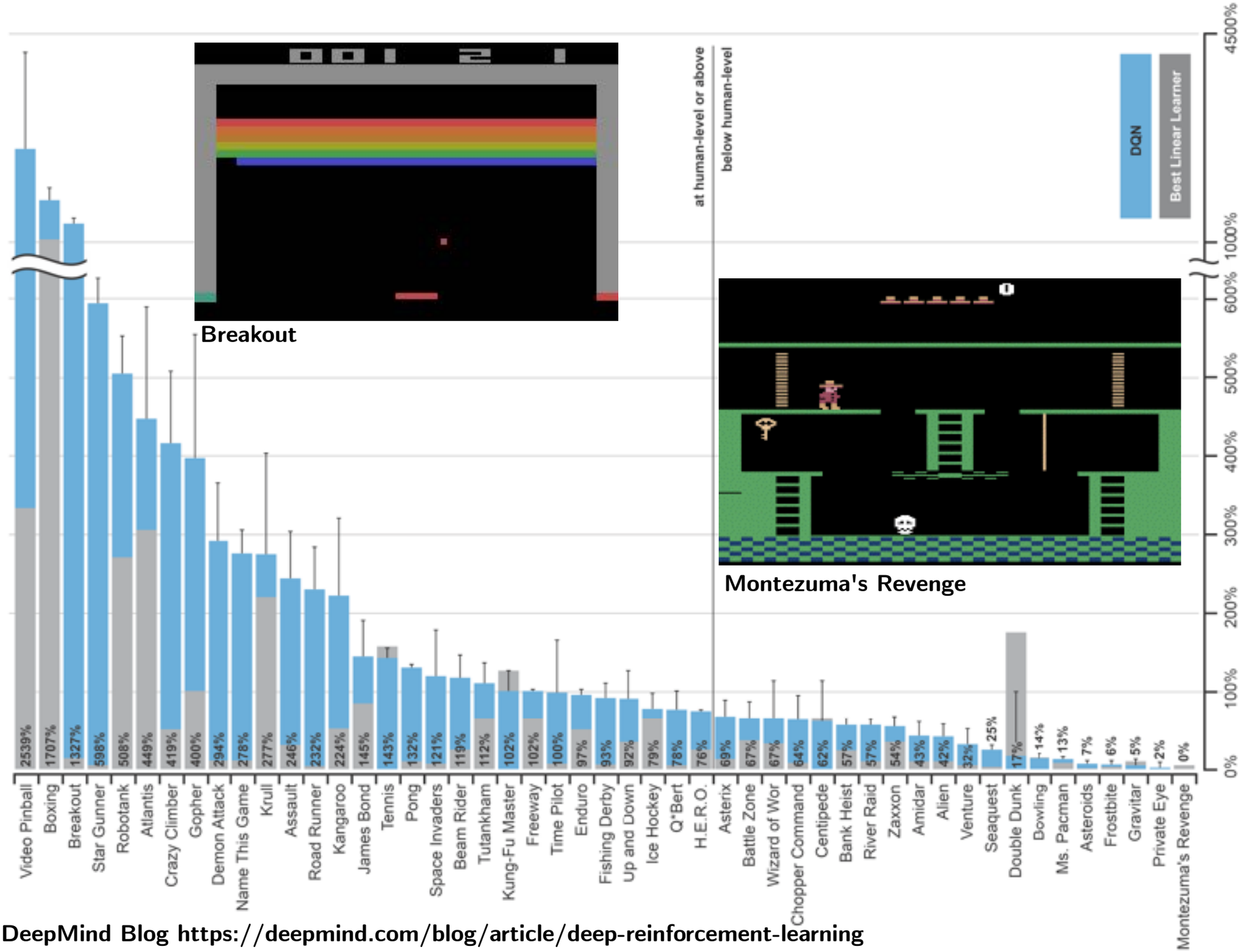
Stratégie pour des jeux

- Suivre une stratégie qui fonctionne bien
- Essayer une autre stratégie

Choix de films

- Regarder une grosse production hollywoodienne
- Tenter un film d'auteurs moins connus

Illustration de la difficulté pratique de l'exploration



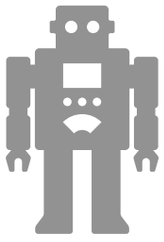
DeepMind Blog https://deepmind.com/blog/article/deep-reinforcement-learning

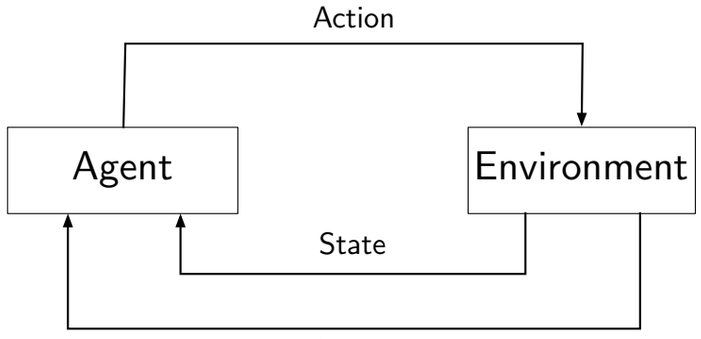
Montezuma's Revenge requiert une importante faculté exploration pour être résolu

Synthèse des notions vues

Apprentissage par renforcement

- Apprentissage qui se fait sur base d'un signal de récompense
- L'objectif est de résoudre un processus de décisions séquentielles
- Le cas standard est un processus de décision de Markov
  - Ensemble d'états
  - Ensemble d'actions
  - Fonction de transition
  - Fonction de récompense





Composantes d'un agent

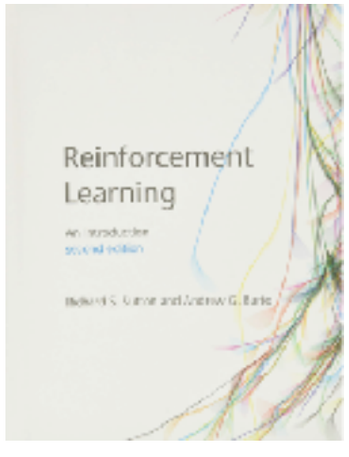
- Modèle de l'environnement: représentation que l'agent se fait de l'environnement
- Fonction de valeur: représentation que l'agent se fait sur la qualité de chaque état
- Politique de sélection: action que l'agent va effectuer, s'il se trouve dans un état spécifique

Cours donné à Polytechnique

- INF8953DE: Reinforcement learning (Sarath Chandar)

Autres ressources

- DeepMind: Cours en ligne
- Livre: Reinforcement learning: an introduction (Sutton et Barto, 2nd)



INF8175 - Intelligence artificielle

Méthodes et algorithmes

Apprentissage par renforcement: FIN

Quentin Cappart