

Machine Learning

Apprentissage par renforcement

Introduction

Marc Métivier

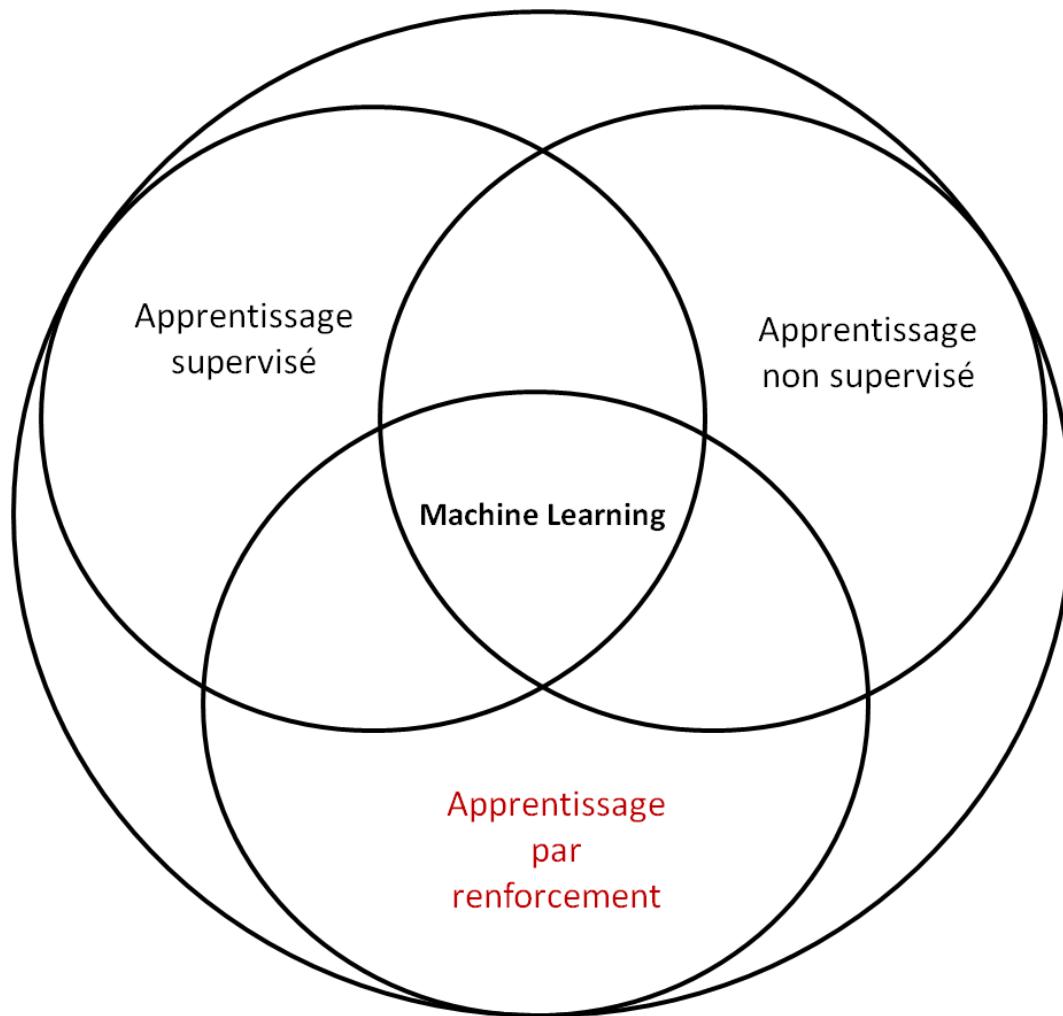
marc.metivier@u-paris.fr



Livres

- **An Introduction to Reinforcement Learning**, Sutton and Barto (Edition 2)
 - MIT Press, 2018
 - Disponible gratuitement en ligne :
 - <http://incompleteideas.net/book/the-book.html>
- **Algorithms for Reinforcement Learning**, Szepesvari
 - Morgan and Claypool, 2010
 - Disponible gratuitement en ligne :
 - <http://www.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf>

Branches de l'apprentissage automatique



L'apprentissage par renforcement

"Reinforcement Learning" en anglais

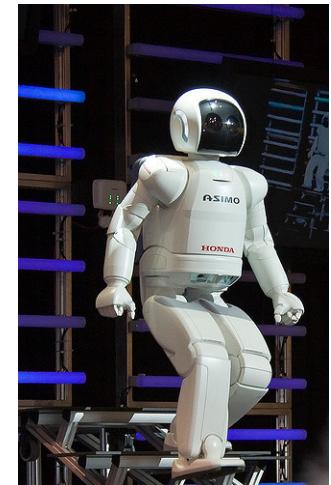
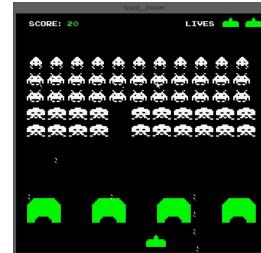
Qu'est-ce qui le rend différent des autres paradigmes d'apprentissage ?

- Il n'y a **pas de superviseur**, juste un signal de **récompense**
- Les **retours sont retardés**, et non immédiats
- Le temps a une réelle importance : **données séquentielles non iid**
- Les **actions** de l'agent **affectent les données suivantes** qu'il reçoit

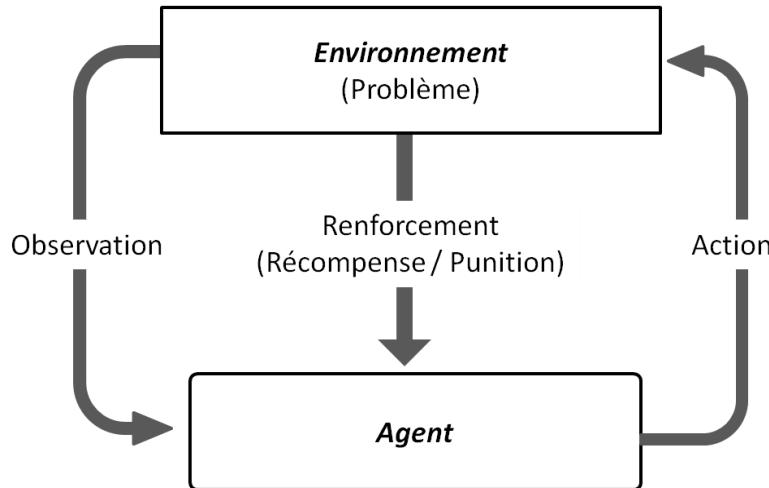
La notion d'agent est indissociable de l'apprentissage par renforcement

Exemples d'apprentissage par renforcement

- Faire des manœuvres de vol avec un hélicoptère
- Gérer un portefeuille d'investissement
- Contrôler une centrale électrique
- Faire marcher un robot humanoïde
- Jouer au Backgammon
- Jouer au jeu de Go
- Jouer à différents jeux Atari

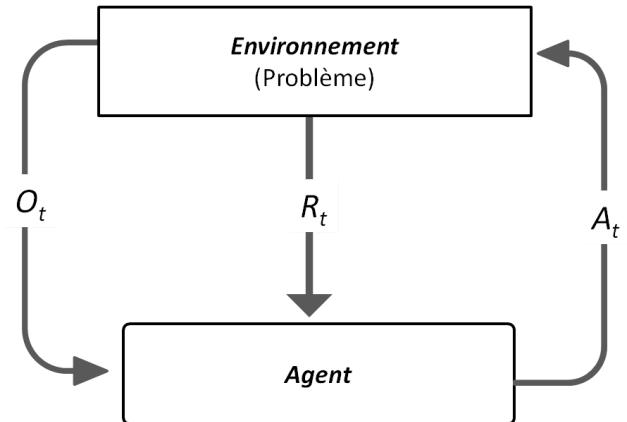


Agent et environnement



- **Renforcement** : un signal scalaire que reçoit l'**agent** dans certaines situations
 - Une valeur positive est une **récompense**
 - Une valeur négative est une **punition**
- L'**objectif** de l'agent est de **maximiser le renforcement cumulé**

Temps



A chaque cycle t :

- L'agent reçoit une observation O_t et un renforcement R_t
- L'agent exécute une action A_t

Histoire (ou trace)

- Une séquence d'observations, d'actions et de renforcements

$$H_t = O_1, R_1, A_1, O_2, R_2, A_2, \dots, A_{t-1}, O_t, R_t$$

- i.e. toutes les variables observables jusqu'à l'instant t

Etats & Observations

Etat de l'environnement

- L'**état de l'environnement** est sa **représentation interne**
- A chaque instant, l'environnement est dans un certain **état**
- Généralement, cet **état** est non-visible par l'agent
- L'agent ne perçoit qu'une **observation** de cet état

Observabilité **complète**

- Un environnement est **complètement observable** si les états et les observations peuvent être confondus

$$O_t = S_t \quad \forall t$$

- L'agent perçoit directement les états de l'environnement

Processus décisionnel de Markov (MDP)

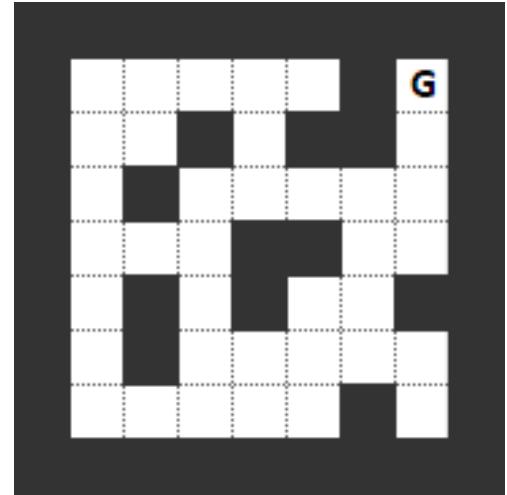
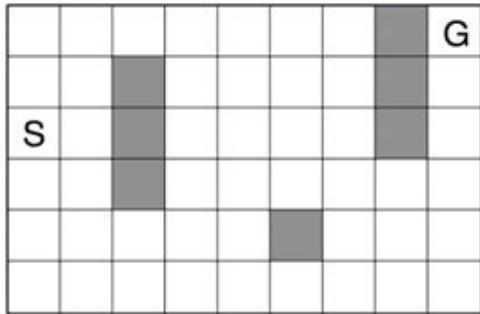
Définition

Un **Processus décisionnel de Markov** est un tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$:

- \mathcal{S} est un **ensemble fini d'états**
- \mathcal{A} est un **ensemble fini d'actions**
- \mathcal{P} est la **distribution des transitions entre états**
 - $\mathcal{P}(s'|s, a)$ dénote la probabilité d'atteindre l'état s' si l'agent exécute l'action a dans l'état s
- $\mathcal{R} : S \times A \times S \rightarrow \mathbb{R}$ est la **fonction de renforcement**
 - $\mathcal{R}(s, a, s')$ est le renforcement perçu lorsque l'agent atteint l'état s' après avoir exécuté l'action a dans l'état s
- $\gamma \in [0, 1]$ est le **facteur d'amortissement** (discount factor)

Processus décisionnel de Markov (MDP)

Exemple : des labyrinthes



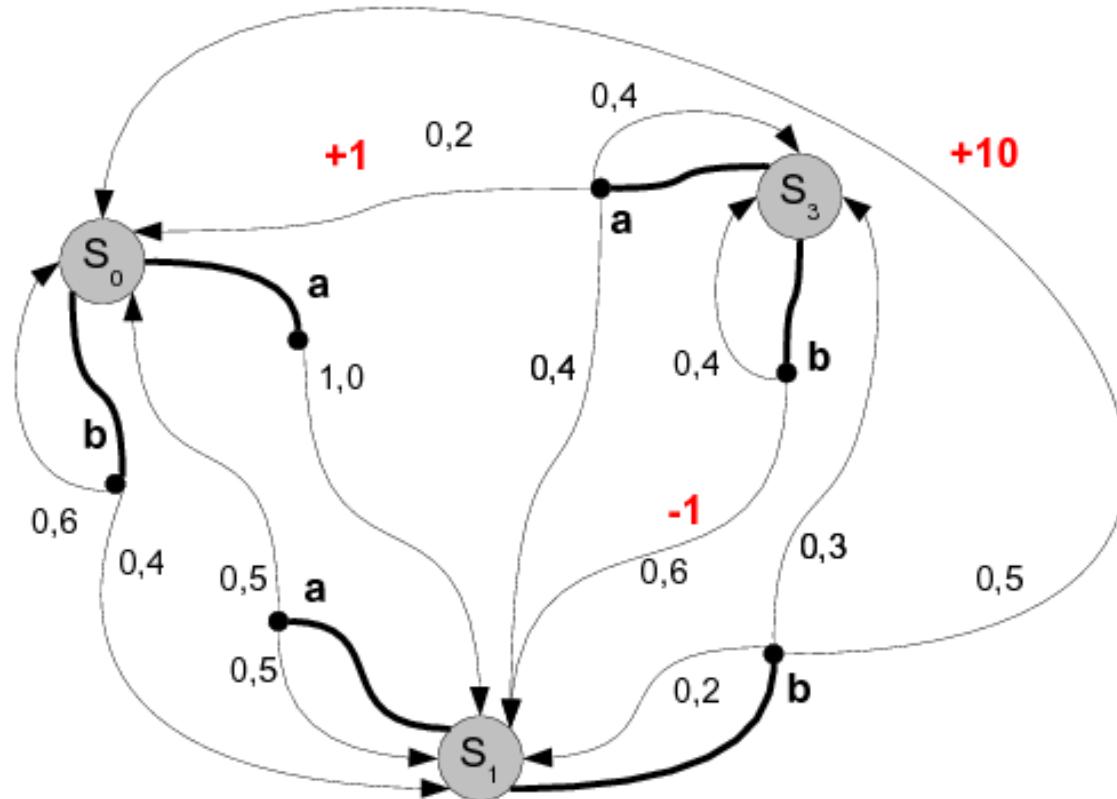
Actions : les 4 déplacements Nord / Sud / Est / Ouest

Renforcements : +1 lorsque l'action permet d'entrer dans la case "G", 0 sinon

- Version déterministe : chaque action n'a qu'une seule conséquence possible
- Version non-déterministe : les actions sont bruitées
 - Par exemple : 1 chance sur 10 de faire une autre action au hasard

Processus décisionnel de Markov (MDP)

Exemple plus général



Ce MDP contient 3 états (S_0 , S_1 , S_2) et 2 actions (a et b).

Quelle est la meilleure stratégie ?

Processus décisionnel de Markov (MDP)

La propriété de Markov :

Le futur ne dépend que du présent (et pas du passé)

Etats markoviens

Définition

Un état S_t est **markovien** (ou "de Markov") si et seulement si :

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \dots, S_t]$$

- L'état courant capture toutes les informations essentielles de l'histoire passée
- Une fois l'état connu, on peut oublier le passé

Dans un MDP tous les états sont markoviens

Processus décisionnel de Markov (MDP)

Les MDPs décrivent formellement un environnement pour l'AR



qui est complètement observable et satisfait la propriété de Markov

Malgré ces restrictions, c'est un cadre très riche !

- Presque tous les problèmes d'apprentissage par renforcement peuvent être formalisés avec des MDPs
- Tout dépend de la définition de ce qu'est un **état** et une **action**
 - Le contrôle optimal utilise des MDP continus
 - L'observabilité partielle peut être convertie en MDPs

Politique

Définition

Une **politique** π est une distribution sur les actions en fonction l'état

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

Une **politique** permet de décrire le comportement de l'agent

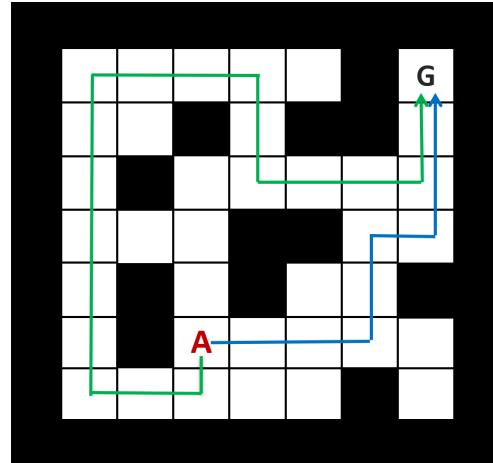
- Elle spécifie quelle action l'agent exécutera selon l'état courant

Objectif : trouver une **politique** qui **maximise les récompenses à long-terme**

Maximiser les récompenses à long-terme ?

Trouver la politique π^* qui maximise $\sum_{t=1}^{\infty} R_t$?

Prenons un exemple :



Si le seul retour non nul est lorsqu'on atteint G, **les deux politiques auront le même total de gains.**

Pourtant la politique bleue semble préférable...

Critère de performance

Qu'est-ce que l'on cherche à maximiser exactement ?

$$R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4 + \cdots + \gamma^t R_{t+1} + \cdots = \sum_{t=1}^{\infty} \gamma^{t-1} R_t$$

avec γ un **facteur d'amortissement (discount factor)** $\gamma \in [0, 1]$

La somme γ -pondérée des renforcements

- Renforcements pondérées par leur distance à l'état d'origine
 - Le renforcement reçu après $t + 1$ cycles est multiplié par γ^t
 - Plus un renforcement est loin dans le futur, plus il est amorti
- Si $\gamma = 1$, on retrouve la somme des renforcements $\sum_{t=1}^{\infty} R_t$
- D'autres critères peuvent être envisagés dans des cas spécifiques

Retour à long-terme

Plus généralement, on définit une notion de **retour** au temps t

Définition

Le **retour** G_t est le total des **renforcements amortis** reçus à partir du temps t .

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k}$$

avec $\gamma \in [0, 1]$

A chaque état S_t , son retour G_t

- Le retour = le renforcement immédiat et une partie des renforcements futurs
- Si $\gamma = 0$, seuls comptent les renforcements immédiats (l'agent est myope)

Fonctions de valeurs d'une politique

Définition

La **fonction valeur d'état** $V^\pi(s)$ est l'espérance du **retour**, lorsqu'on **est dans l'état s et que l'on suit la politique π** .

$$V^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

Définition

La **fonction valeur d'action** $Q^\pi(s, a)$ est l'espérance du **retour**, lorsqu'on **exécute l'action a dans l'état s et que l'on suit ensuite la politique π** .

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

Équations de Bellman

Valeur d'un état = le gain immédiat + la valeur amortie de l'état suivant

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right] \end{aligned}$$

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

De même pour la valeur d'une action

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]]$$

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

Équations de Bellman

Détails des formules

Calcul de V^π

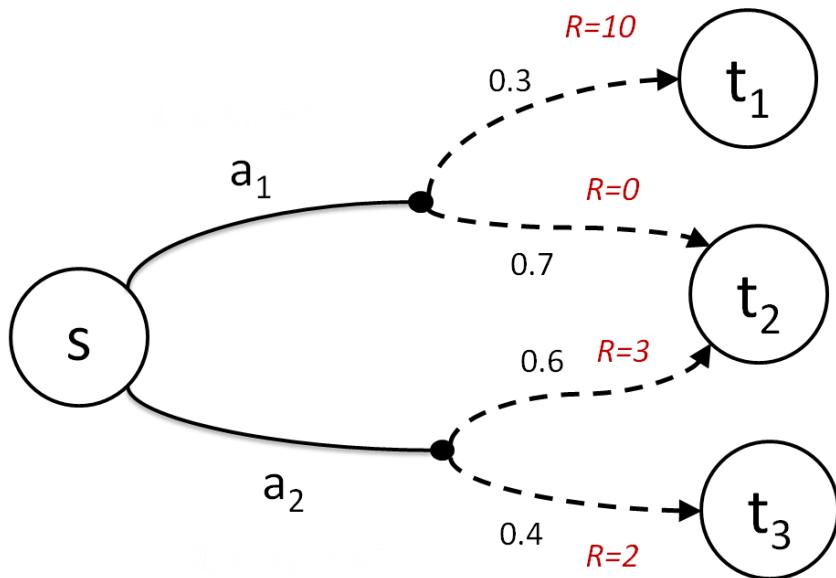
$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi[R_{t+1} + \gamma V^\pi(S_{t+1}) \mid S_t = s] \\ &= \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a) \left[\mathcal{R}(s, a, s') + \gamma V^\pi(s') \right] \end{aligned}$$

Relations entre V^π et Q^π

$$Q^\pi(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a) \left[\mathcal{R}(s, a, s') + \gamma V^\pi(s') \right]$$

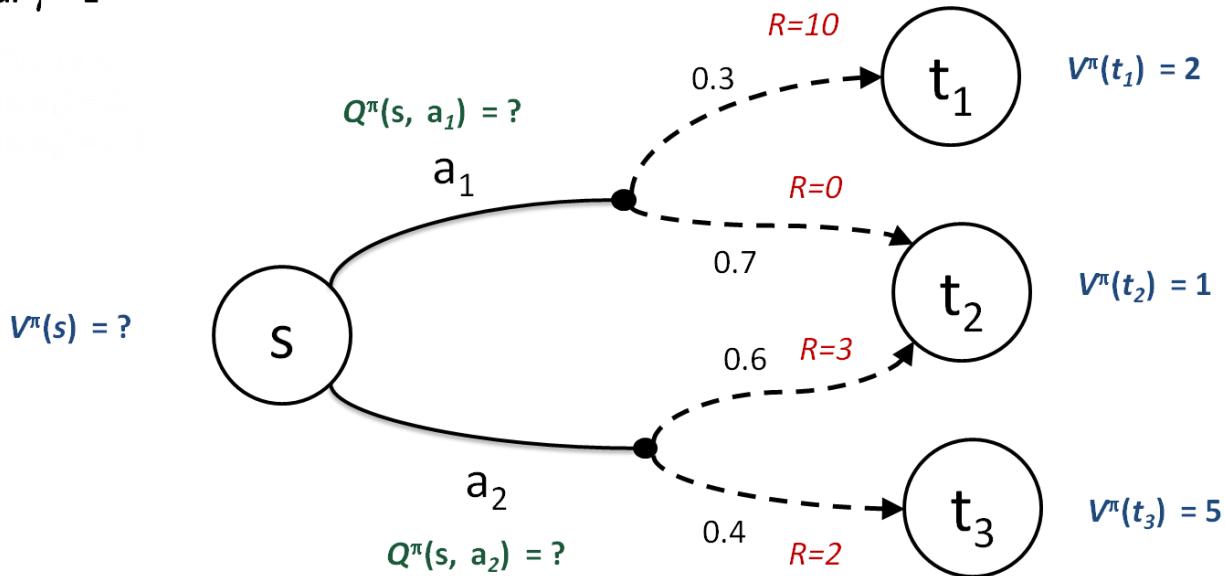
$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q^\pi(s, a)$$

Exemple



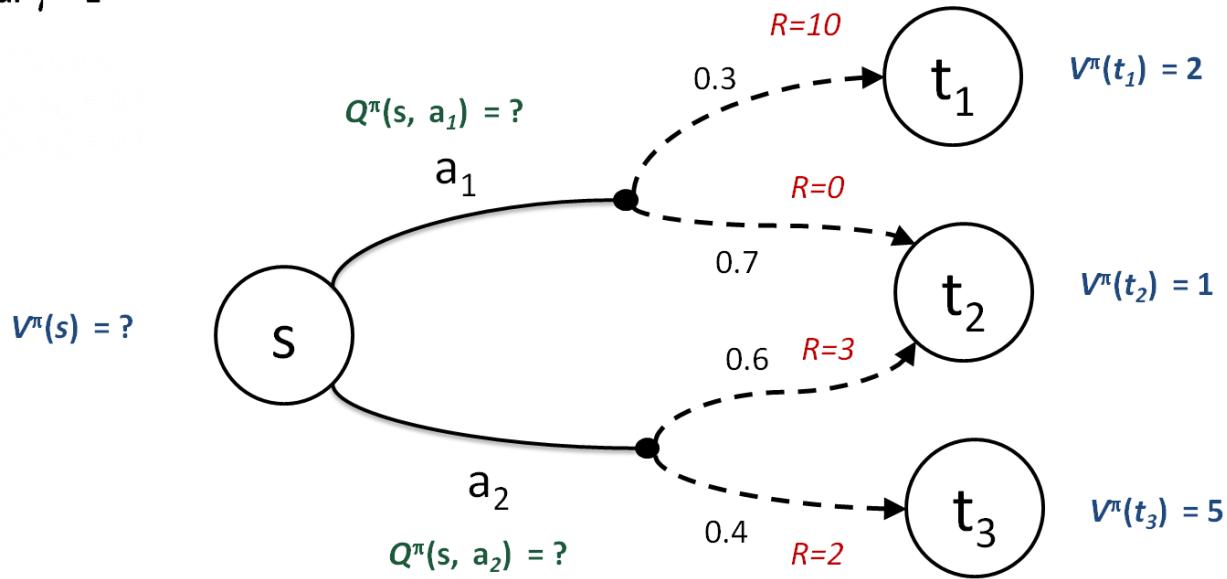
Exemple

Pour $\gamma = 1$



Exemple

Pour $\gamma = 1$



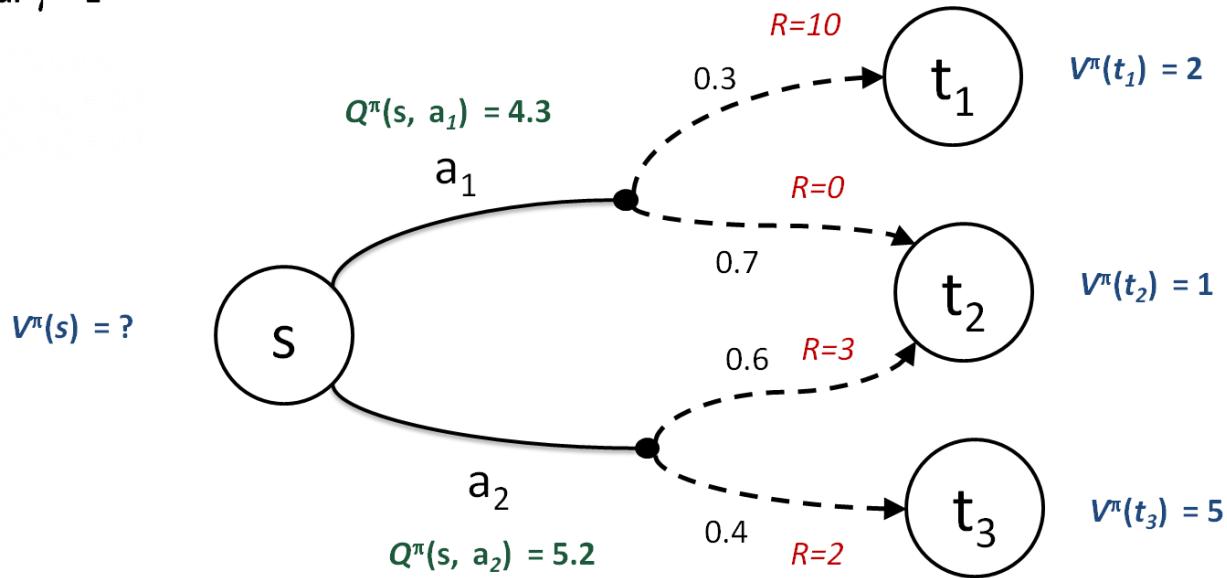
Utilisation de : $Q^\pi(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) [\mathcal{R}(s, a, s') + \gamma V^\pi(s')]$

$$Q^\pi(s, a_1) = 0.3 \times (10 + \gamma V^\pi(t_1)) + 0.7 \times (0 + \gamma V^\pi(t_2))$$

$$Q^\pi(s, a_2) = 0.6 \times (3 + \gamma V^\pi(t_2)) + 0.4 \times (2 + \gamma V^\pi(t_3))$$

Exemple

Pour $\gamma = 1$



$$\text{Utilisation de : } Q^\pi(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V^\pi(s') \right]$$

$$Q^\pi(s, a_1) = 0.3 \times (10 + 1 \times 2) + 0.7 \times (0 + 1 \times 1) = 4.3$$

$$Q^\pi(s, a_2) = 0.6 \times (3 + 1 \times 1) + 0.4 \times (2 + 1 \times 5) = 5.2$$

Exemple

Pour $\gamma = 1$

Politique π :

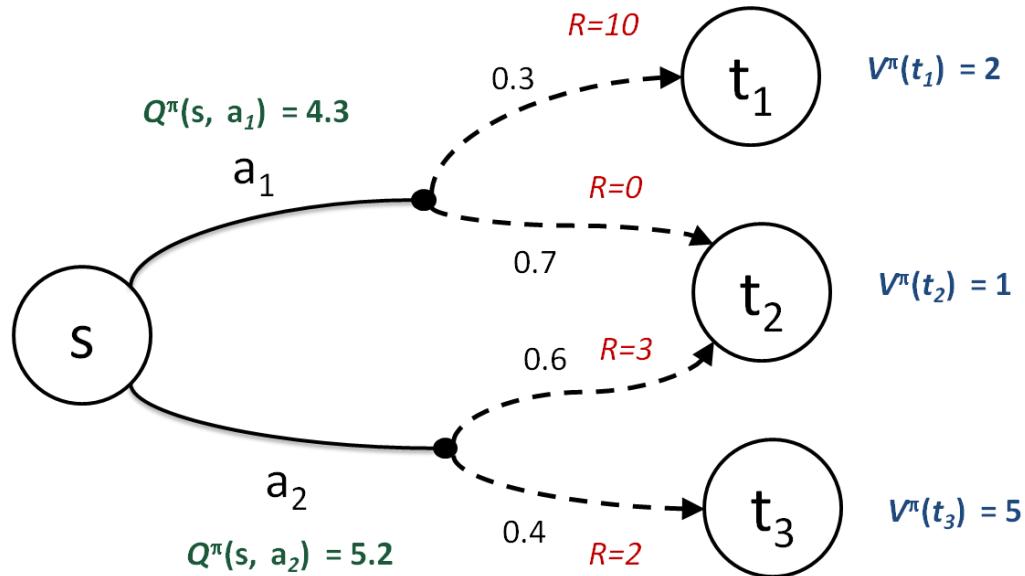
$$\pi(s, a_1) = 0.4$$

$$\pi(s, a_2) = 0.6$$

$$V^\pi(s) = 4.84$$

$$Q^\pi(s, a_1) = 4.3$$

$$Q^\pi(s, a_2) = 5.2$$



Utilisation de : $V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q^\pi(s, a)$

$$\begin{aligned} V^\pi(s) &= \pi(a_1 \mid s) \times Q^\pi(S, a_1) + \pi(a_2 \mid s) \times Q^\pi(S, a_2) \\ &= 0.4 \times 4.3 + 0.6 \times 5.2 \\ &= 4.84 \end{aligned}$$

Exemple

Pour $\gamma = 0.9$

Politique π :

$$\pi(s, a_1) = 0.4$$

$$\pi(s, a_2) = 0.6$$

$$V^\pi(s) = 4.632$$

$$Q^\pi(s, a_1) = 4.17$$

$$Q^\pi(s, a_2) = 4.94$$

$$R=10$$

$$0.3$$

$$R=0$$

$$0.7$$

$$R=3$$

$$0.6$$

$$R=2$$

$$0.4$$

$$t_1$$

$$V^\pi(t_1) = 2$$

$$t_2$$

$$V^\pi(t_2) = 1$$

$$t_3$$

$$V^\pi(t_3) = 5$$

$$Q^\pi(s, a_1) = 0.3 \times (10 + 0.9 \times 2) + 0.7 \times (0 + 0.9 \times 1) = 4.17$$

$$Q^\pi(s, a_2) = 0.6 \times (3 + 0.9 \times 1) + 0.4 \times (2 + 0.9 \times 5) = 4.94$$

$$V^\pi(s) = 0.4 \times 4.17 + 0.6 \times 4.94 = 4.632$$

Exemple

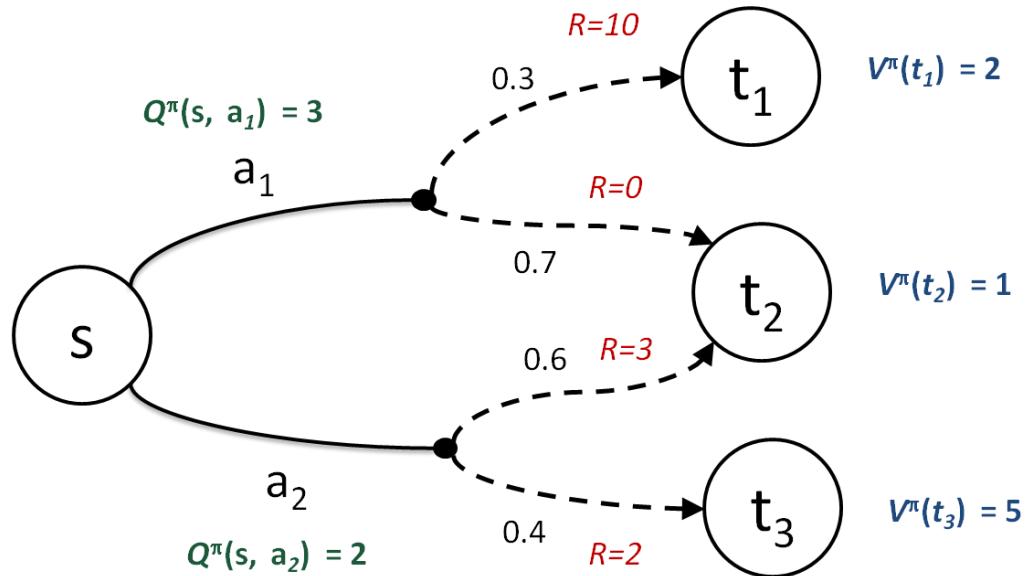
Pour $\gamma = 0$

Politique π :

$$\pi(s, a_1) = 0.4$$

$$\pi(s, a_2) = 0.6$$

$$V^\pi(s) = 2.4$$



$$Q^\pi(s, a_1) = 0.3 \times (10 + 0 \times 2) + 0.7 \times (0 + 0 \times 1) = 3$$

$$Q^\pi(s, a_2) = 0.6 \times (3 + 0 \times 1) + 0.4 \times (2 + 0 \times 5) = 2$$

$$V^\pi(s) = 0.4 \times 3 + 0.6 \times 2 = 2.4$$

Fonctions de valeurs optimales

Définition

La **fondction valeur d'état optimale** $V^*(s)$ est la fonction valeur d'état **maximale** sur toutes les politiques.

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

Définition

La **fondction valeur d'action optimale** $Q^*(s, a)$ est la fonction valeur d'action **maximale** sur toutes les politiques.

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

- Un MDP est "résolu" lorsque l'on connaît les fonctions valeurs optimales

Politique optimale

On définit un ordre partiel sur les politiques :

$$\pi \geq \pi' \text{ si } V^\pi(s) \geq V^{\pi'}(s), \forall s \in \mathcal{S}$$

Théorème

Pour tout *processus décisionnel de Markov* :

- Il existe une politique optimale π^* qui est supérieure ou égale à toutes les autres politiques, $\pi^* \geq \pi, \forall \pi$
- Toute politique optimale réalise la fonction valeur d'état optimale

$$V^{\pi^*}(s) = V^*(s), \forall s$$

- Toute politique optimale réalise la fonction valeur d'action optimale

$$Q^{\pi^*}(s, a) = Q^*(s, a), \forall s \forall a$$

Trouver une politique optimale

Une politique optimale peut être trouvée par maximisation de Q^*

$$\pi^*(a \mid s) = \begin{cases} 1 & \text{si } a = \arg \max_{a \in \mathcal{A}} Q^*(s, a) \\ 0 & \text{sinon.} \end{cases}$$

- Remarque : ce type de sélection est appelée "*greedy*"

$$\pi^* = \text{greedy}(Q^*)$$

Dans tout MDP, il existe une politique optimale déterministe

- Si l'on connaît Q^* , on a immédiatement la politique optimale

Equation d'optimalité de Bellman

Relation entre les fonctions de valeurs optimales

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a)$$

$$Q^*(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V^*(s') \right]$$

Equation d'optimalité de Bellman pour V^*

$$V^*(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V^*(s') \right]$$

Equation d'optimalité de Bellman pour Q^*

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

Comment calculer les valeurs optimales ?

Il faut résoudre l'équation d'optimalité de Bellman

- L'équation d'optimalité de Bellman est non-linéaire (présence du max)
- En général, il n'existe pas de solution de forme close

III → On utilise des **solutions itératives**

- La programmation dynamique
- Monte-Carlo RL
- Q-learning
- Sarsa
- Dyna
- ...

La programmation dynamique (DP)

En anglais : ***Dynamic Programming*** (DP)

"Programmation dynamique" ?

- **Dynamique** : le problème a une composante **séquentielle** ou **temporelle**
- **Programmation** : **optimiser** un "programme", c'est-à-dire **une politique**
 - Même élément de langage que pour la "programmation linéaire"

Une méthode pour résoudre des problèmes complexes

- **Décomposer** le problème en **sous-problèmes**
- **Résoudre** les **sous-problèmes**
- **Combiner** les solutions des **sous-problèmes**

La programmation dynamique (DP)

Elle s'applique à un **MDP connu complètement**

- Les fonctions de transitions et de renforcement sont connus
- C'est une forme de **planification** pour les problèmes stockastiques

Deux usages :

- La **prédiction**
 - Entrées : un MDP et une politique π
 - Sortie : la fonction valeur V^π
- Le **contrôle**
 - Entrées : un MDP
 - Sortie : la fonction valeur optimale V^* et une politique optimale π^*

Evaluation itérative d'une politique

Objectif : évaluer une politique donnée π (calculer V^π)

Principe : mise à jour itérative des valeurs par l'équation de Bellman

- $V_1 \rightarrow V_2 \rightarrow \dots \rightarrow V_\infty = V^\pi$

Algorithme Iterative Policy Evaluation (IPE)

Initialiser les valeurs : $V_1(s) = 0 \quad \forall s \in \mathcal{S}$

Pour $k = 1, 2, \dots$:

Pour chaque $s \in \mathcal{S}$:

$$V_{k+1}(s) \leftarrow \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V_k(s') \right]$$

- L'algorithme s'exécute jusqu'à convergence de la fonction
- La convergence vers V^π est prouvée

Amélioration de politique

Comment améliorer une politique à partir de sa fonction valeur ?

- Choisir la politique **gloutonne** ("greedy") sur la fonction de valeur V^π

$$\pi' = \text{greedy}(V^\pi)$$

- donc :

$$\pi'(a \mid s) = \begin{cases} 1 & \text{si } a = \arg \max_{a \in \mathcal{A}} Q^\pi(s, a) \\ 0 & \text{sinon.} \end{cases}$$

- avec (rappel) :

$$Q^\pi(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a) \left[\mathcal{R}(s, a, s') + \gamma V^\pi(s') \right]$$

Algorithme des politiques itérées

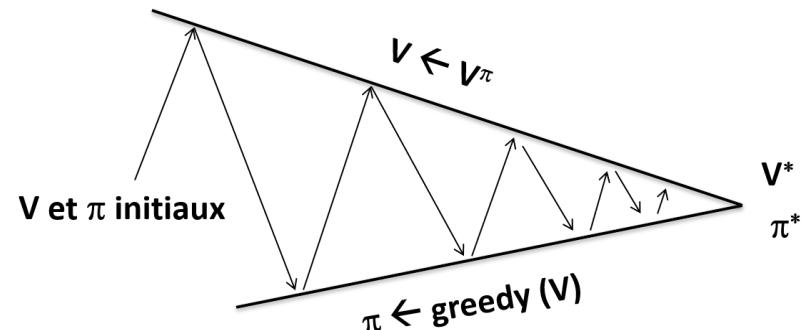
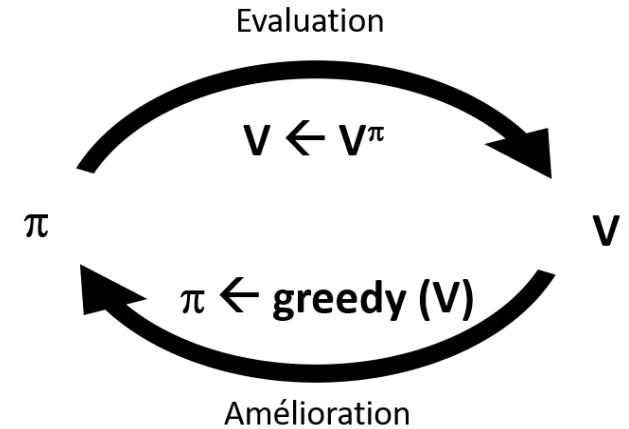
Objectif : trouver une politique optimale π^*

Principe :

Répéter :

- **Evaluation** de la politique π
 - Calcul de V^π par **IPE**
- **Amélioration** de la politique π
 - $\pi \leftarrow \text{greedy}(V^\pi)$

jusqu'à convergence de π et V^π ...



Algorithme des politiques itérées

Amélioration

Est-ce que l'évaluation de la politique doit converger vers V_π ?

On peut **introduire une condition d'arrêt**

Exemples :

- Arrêter lorsqu'il y a ϵ -convergence
- Arrêter après k itérations, k étant un paramètre fixe

Faut-il partir de valeurs nulles au début de l'évaluation ?

On peut conserver la fonction valeurs de la politique précédente

→ On obtient alors l'algorithme **modifié des politiques itérées** (MPI)

Algorithme des valeurs itérées

Intuition

Si on connaît les valeurs optimales de certains états ($V^*(s')$) alors les valeurs de leurs états connectés ($V^*(s)$) peuvent être calculées en un seul cycle de mise à jour :

$$V^*(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V^*(s') \right]$$

Or, les états terminaux ont toujours une valeur nulle...

- III → Partir des états terminaux puis appliquer cette mise à jour itérativement
- III → On obtient directement la fonction optimale
- III → Pas besoin de politique explicite

Algorithme des valeurs itérées

Objectif : calculer la fonction valeur optimale V^*

- Mise à jour itérative en appliquant l'équation d'**optimalité** de Bellman
 - $V_1 \rightarrow V_2 \rightarrow \dots \rightarrow V_\infty = V^*$

Algorithme des Valeurs itérées (VI)

Initialiser les valeurs : $V_1(s) = 0 \quad \forall s \in \mathcal{S}$

Pour $k = 1, 2, \dots$:

Pour chaque $s \in \mathcal{S}$:

$$V_{k+1}(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) [\mathcal{R}(s, a, s') + \gamma V_k(s')]$$

- Les fonctions V_k convergeront vers la fonction optimale V^*
- La politique optimale s'obtient par **greedy** : $\pi^* \leftarrow \text{greedy}(V^*)$

Algorithme des valeurs itérées

Amélioration

L'algorithme de base fait des mises à jours synchrones

- les valeurs de *tous les états sont mises à jour en parallèle*

Amélioration : utiliser des mises à jour asynchrones

- Utiliser une seule fonction V :

$$V(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V(s') \right]$$

- Mettre à jour en priorité les états ayant une forte **erreur de Bellman**

$$\left| \max_{a \in \mathcal{A}} \left(\sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \left[\mathcal{R}(s, a, s') + \gamma V(s') \right] \right) - V(s) \right|$$

Convergence

Quelques notations

Les équations de Bellman peuvent être exprimées sous forme matricielle :

$$V^\pi = \mathcal{P}^\pi \mathcal{R}^\pi + \gamma \mathcal{P}^\pi V^\pi = T^\pi(V^\pi)$$

- T^π est l'**opérateur de Bellman**
- Si on considère la norme ∞ :

$$\| \mathbf{u} - \mathbf{v} \|_\infty = \max_{s \in \mathcal{S}} | u(s) - v(s) |$$

- Alors T^π est une γ -contraction :

$$\begin{aligned} \|T^\pi(\mathbf{u}) - T^\pi(\mathbf{v})\|_\infty &= \|\mathcal{P}^\pi \mathcal{R}^\pi + \gamma \mathcal{P}^\pi \mathbf{u} - \mathcal{P}^\pi \mathcal{R}^\pi + \gamma \mathcal{P}^\pi \mathbf{v}\|_\infty \\ &= \|\gamma \mathcal{P}^\pi(\mathbf{u} - \mathbf{v})\|_\infty \\ &\leq \|\gamma \mathcal{P}^\pi\| \|\mathbf{u} - \mathbf{v}\|_\infty \\ &\leq \|\mathbf{u} - \mathbf{v}\|_\infty \end{aligned}$$

Convergence

L'opérateur de Bellman T^π est une γ -contraction

- T^π converge vers un point fixe unique (théorème du point fixe de Banach)
- V^π est un point fixe de T^π

De mène l'**opérateur d'optimalité de Bellman** T^* est une γ -contraction

- V^* est un point fixe de T^*

III → En conséquence :

- L'évaluation itérée d'une politique π converge vers V^π
- L'algorithme des politiques itérées converge vers V^* et π^*
- L'algorithme des valeurs itérées converge vers V^* et π^*

Extension des MDPs

POMDP : Partially Observable MDP

- L'agent perçoit des ***observations*** et non l'état réel de l'environnement
- La fonction d'état de croyance :
 - Associer une probabilité à chaque état selon l'observation

DEC-MDP : Decentralized MDP

- Plusieurs agents agissent en même temps

Stochastic games

- Un jeu matriciel dans chaque état

FMDP : Factored MDP

- Les états ont une représentation factorisé

RMDP : Relational MDP

- Les états ont une représentation en logique du 1er ordre