Reinforcement Learning, yet another introduction.

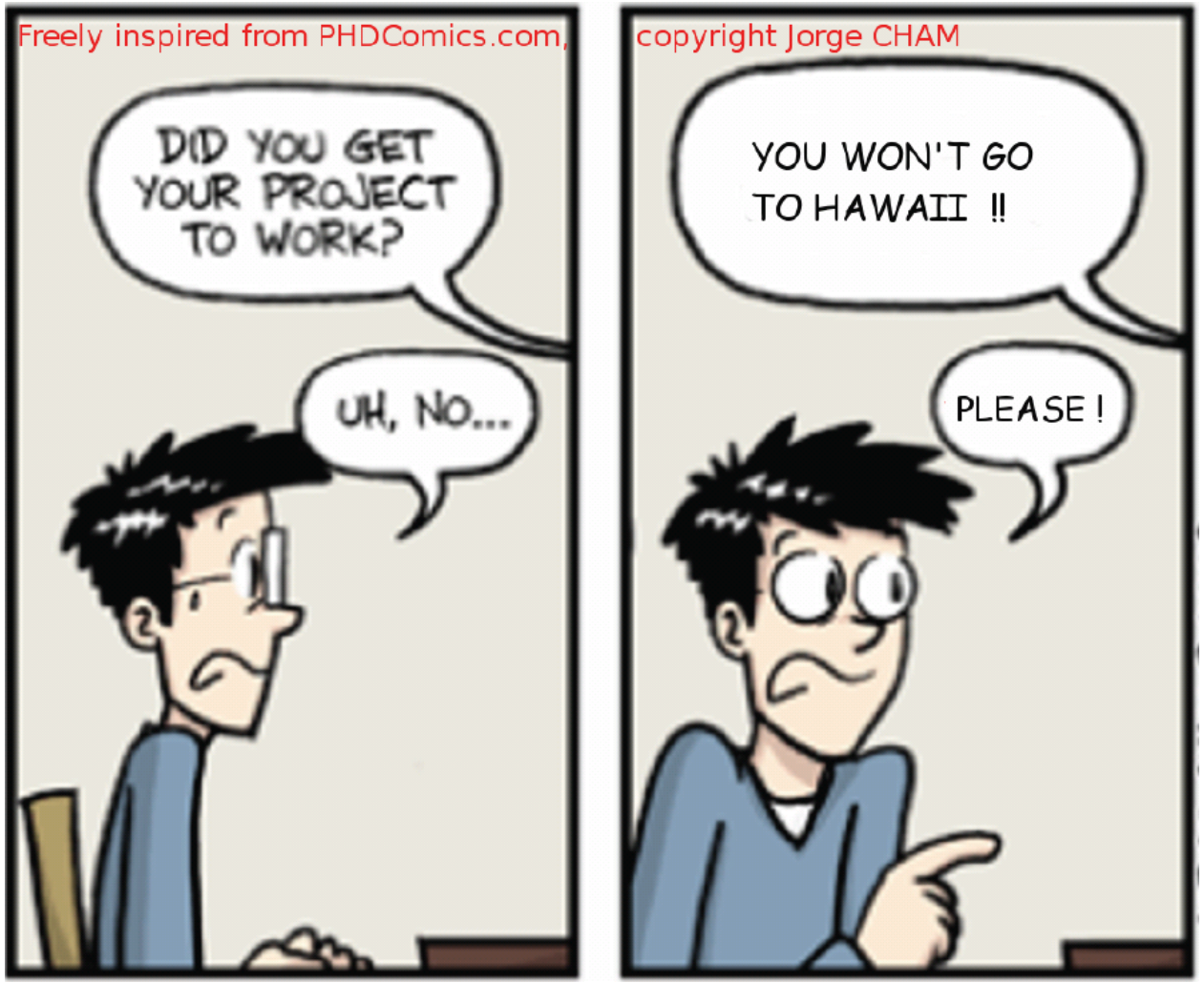
Emmanuel Rachelson

8 Décembre 2009

1. Intuition : là, toutes les décisions comptent !
2. Ecrire le problème
3. Planifier — Caractériser une politique optimale
4. Apprentissage par renforcement

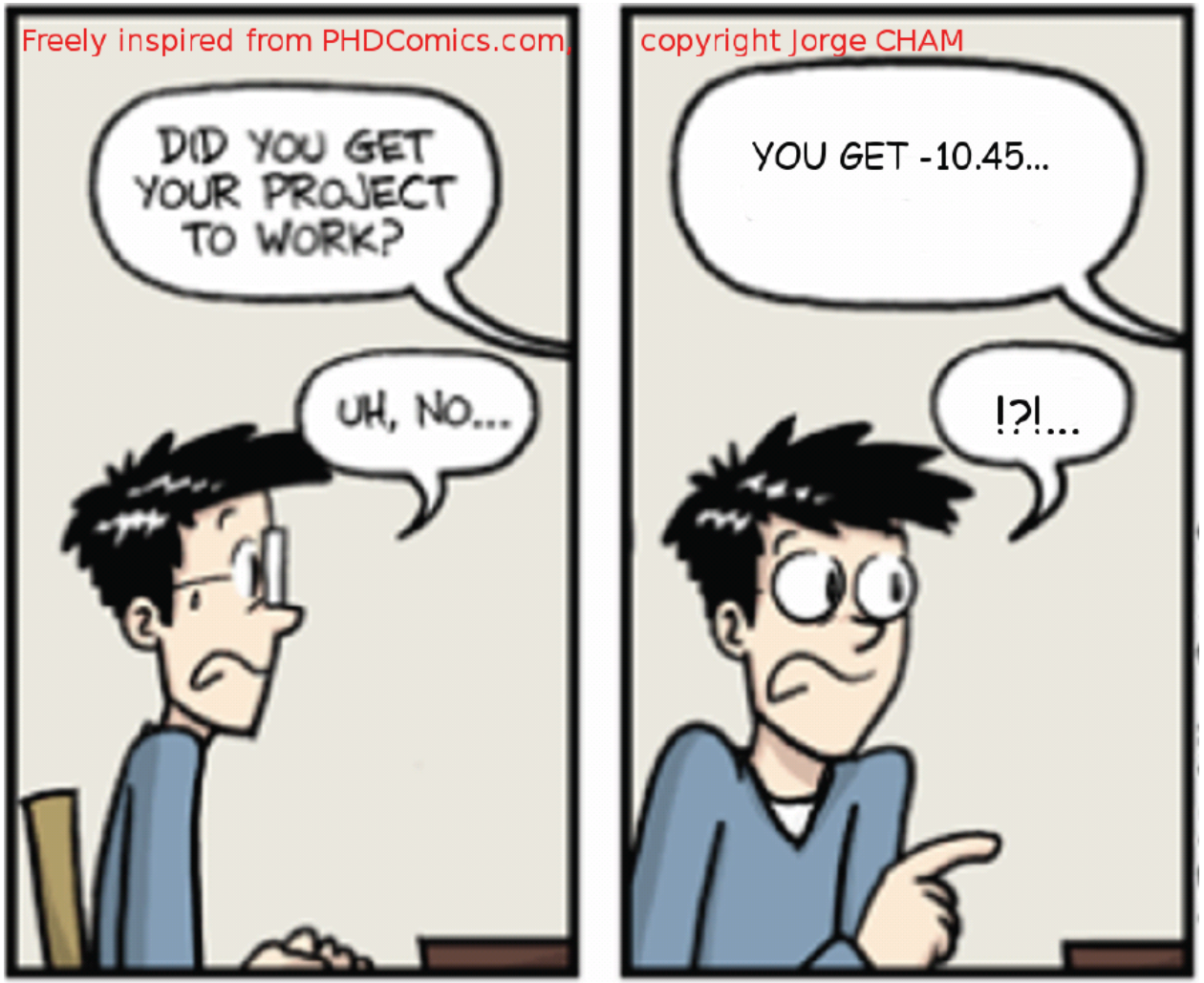
Décider sur l'avenir ...

... en se basant sur l'expérience actuelle (et ses implications).



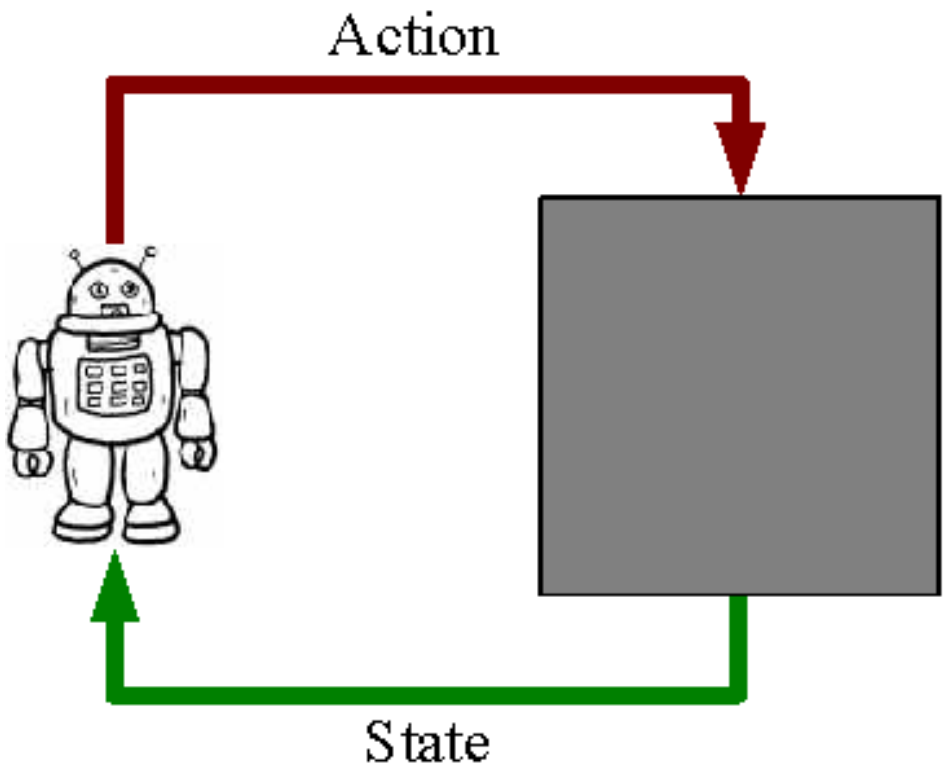
Décider sur l'avenir . . .

. . . et en l'évaluant.



Interaction agent-environnement en séquence

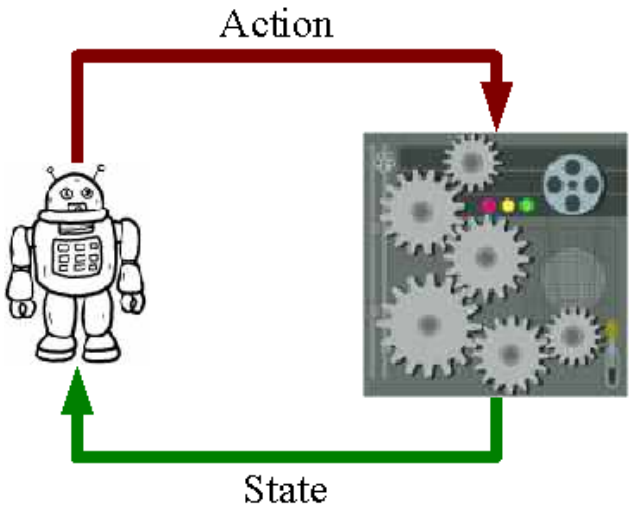
"Décision séquentielle"



- C'est la séquence de décisions qui importe.

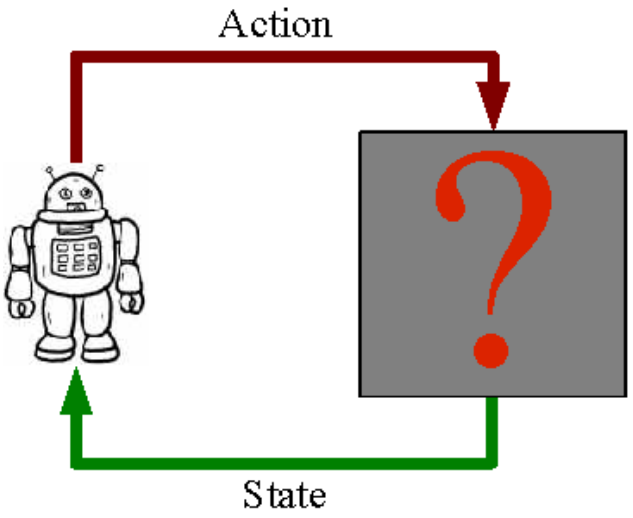
Interaction agent-environnement en séquence

Planifier ou apprendre ?



Modèle connu

Incertitude **dans** le modèle

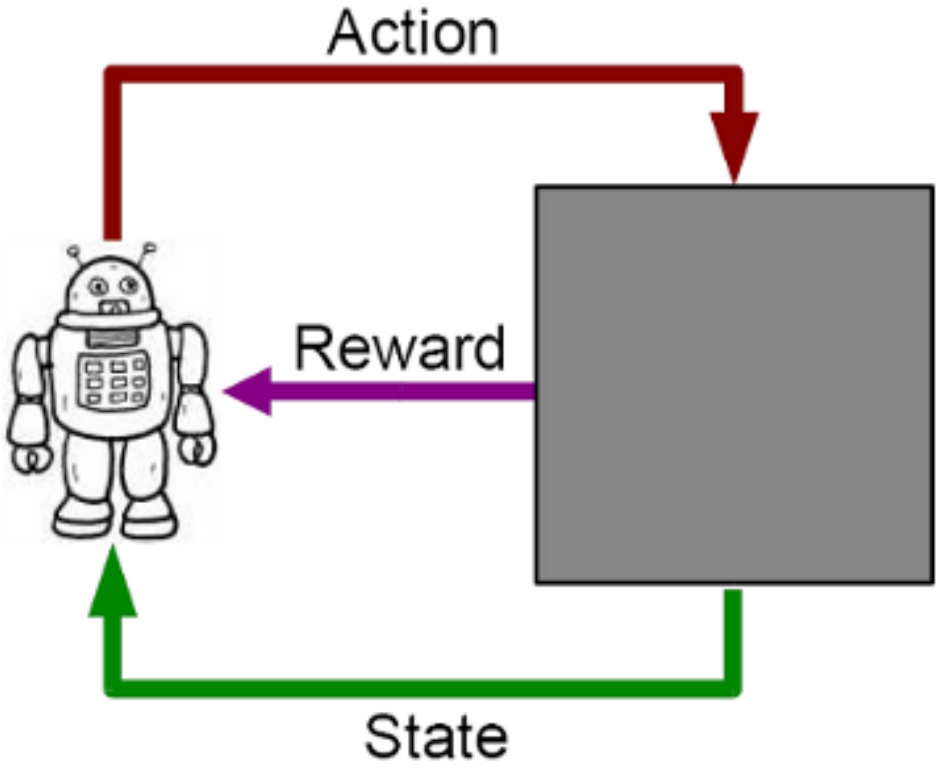


Modèle inconnu

Incertitude **dans** et **à propos** du modèle

Interaction agent-environnement en séquence

Décider dans le but d'optimiser une valeur **numérique**



Les questions

Quel cadre théorique ? Quelles hypothèses sur l'environnement ?

Comment caractérise-t-on une bonne stratégie ?

Doit-on nécessairement passer par un modèle pour trouver un bon comportement ?

Comment équilibrer l'acquisition d'informations et l'exploitation d'une bonne stratégie ?

Les questions

Quel cadre théorique ? Quelles hypothèses sur l'environnement ?

Comment caractérise-t-on une bonne stratégie ?

Doit-on nécessairement passer par un modèle pour trouver un bon comportement ?

Comment équilibrer l'acquisition d'informations et l'exploitation d'une bonne stratégie ?

Les questions

Quel cadre théorique ? Quelles hypothèses sur l'environnement ?

Comment caractérise-t-on une bonne stratégie ?

Doit-on nécessairement passer par un modèle pour trouver un bon comportement ?

Comment équilibrer l'acquisition d'informations et l'exploitation d'une bonne stratégie ?

Les questions

Quel cadre théorique ? Quelles hypothèses sur l'environnement ?

Comment caractérise-t-on une bonne stratégie ?

Doit-on nécessairement passer par un modèle pour trouver un bon comportement ?

Comment équilibrer l'acquisition d'informations et l'exploitation d'une bonne stratégie ?

1. Intuition : là, toutes les décisions comptent !
2. Ecrire le problème
3. Planifier — Caractériser une politique optimale
4. Apprentissage par renforcement

Les ingrédients

- Un ensemble de **pas de temps** T.
- L'**état** s du système ou de l'agent.
- L'ensemble des **actions** a qu'il peut effectuer.
- La **dynamique** du système $s' \leftarrow f(?)$.
- Les **récompenses** ou **signal de renforcement** à chaque pas $r(?)$.

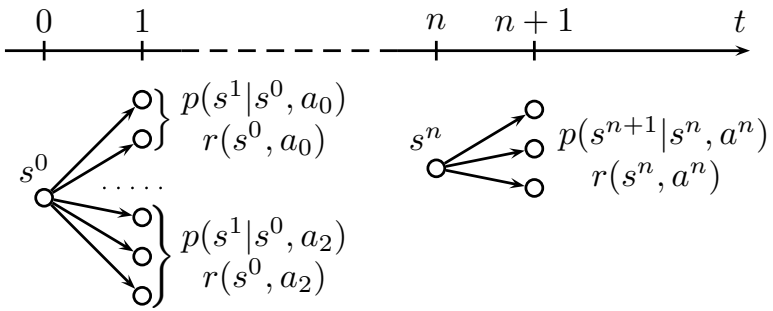
Processus Décisionnels de Markov

Décision séquentielle sous incertitude probabiliste:

Markov Decision Process (MDP)

Quintuplet $\langle S, A, p, r, T \rangle$
Modèle de transition Markovien $p(s'|s,a)$
Modèle de récompense $r(s,a)$
T ensemble d'instants de décision $\{0, 1, \ldots, H\}$

Horizon infini (ou non-borné) : $H \longrightarrow \infty$



Evaluer un comportement

Que me procure le comportement courant dans mon état actuel ?

Plusieurs critères possibles:

- Récompense moyenne $V(s) = E\left(\lim_{H\to\infty} \frac{1}{H} \sum_{\delta=0}^{H} r_\delta \middle| s_0 = s\right)$
- Récompense totale $V(s) = E\left(\lim_{H\to\infty} \sum_{\delta=0}^{H} r_\delta \middle| s_0 = s\right)$
- Récompense γ-pondérée $V(s) = E\left(\lim_{H\to\infty} \sum_{\delta=0}^{H} \gamma^\delta r_\delta \middle| s_0 = s\right)$

Qu'est-ce qu'un comportement ?

Politique : fonction décidant de l'action à effectuer.

Cas général : $\pi^n(s^0, a^0, s^1, a^1, \ldots, s^n)$ est une distribution sur a^n

Heureusement ...

Politique optimale

Pour un MDP avec critère γ-pondéré et horizon infini, parmi les politiques optimales, il en existe une qui soit **déterministe**, **Markovienne** et **stationnaire**

$$\pi^{n*}(h_n) \equiv d(A) \rightarrow \pi(s^n) = a^n$$

Fonction de valeur

"Ce qu'on peut espérer gagner à partir de s en appliquant π"

$$V^\pi(s) = E\left(\sum_{\delta=0}^{\infty} r(s_\delta, \pi(s_\delta)) \middle| s_0 = s\right)$$

Equation de Bellman

V^π est solution du système linéaire :

$$\forall s \in S, \quad V^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V^\pi(s')$$

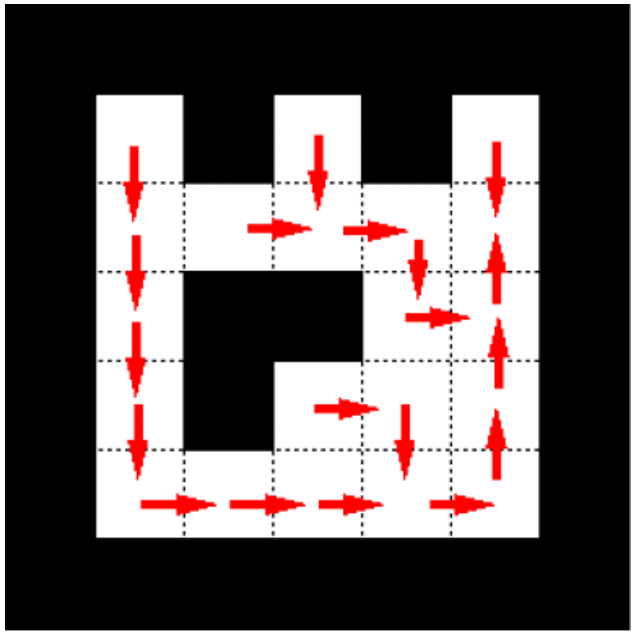
$$V^\pi = L^\pi V^\pi$$

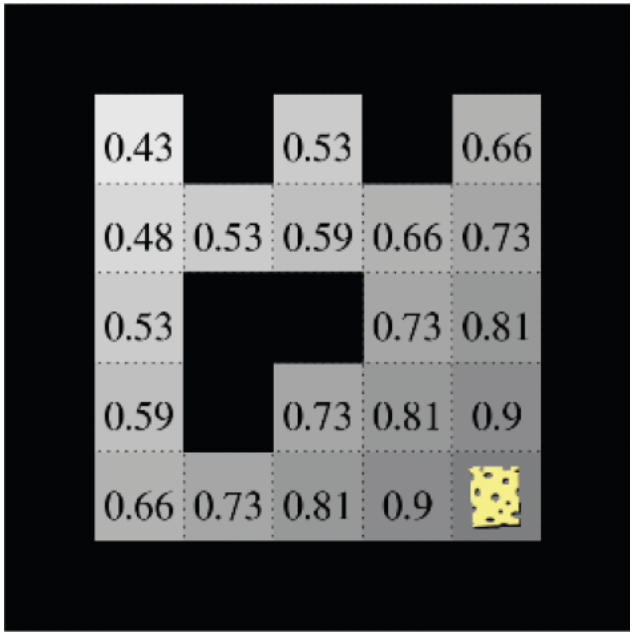
Pour mémoire, cas des politiques stochastiques:

$$\forall s \in S, \quad V^\pi(s) = \sum_{a \in A} \pi(s,a) \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^\pi(s') \right)$$

En bref

- Environnement $\leftrightarrow$ MDP.
- Comportement $\leftrightarrow$ politique $\pi : s \mapsto a$.
- Objectif $\leftrightarrow$ fonction de valeur $V^{\pi}(s)$.





- Planifier: p et r supposés connus.
- Apprendre: p et r inconnus.

$$\left.\right\} \quad \pi^* = \arg\max_{\pi} V^{\pi}(s)$$

Quelques éléments en plus

Q-fonctions ou fonctions de valeur d'actions.

$$Q^{\pi}(s,a) = r(s,a) + E\left(\sum_{\delta=1}^{\infty} \gamma^{\delta} r(s_{\delta}, \pi(s_{\delta}))\middle| s_1 \sim p(s'|s,\pi(s))\right).$$

Quelques éléments en plus

Q-fonctions ou fonctions de valeur d'actions.

$$Q^{\pi}(s,a) = r(s,a) + E\left(\sum_{\delta=1}^{\infty} \gamma^{\delta} r(s_{\delta}, \pi(s_{\delta}))\middle| s_1 \sim p(s'|s,\pi(s))\right).$$

Politique gloutonne.

$$\pi_1(s) = \arg\max_a Q(s,a)$$

$$\pi_1(s) = \arg\max_a r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s')$$

Quelques éléments en plus

Q-fonctions ou fonctions de valeur d'actions.

$$Q^{\pi}(s,a) = r(s,a) + E\left(\sum_{\delta=1}^{\infty} \gamma^{\delta} r(s_{\delta}, \pi(s_{\delta}))\middle| s_1 \sim p(s'|s,\pi(s))\right).$$

Politique gloutonne.

$$\pi_1(s) = \arg\max_a Q(s,a)$$

$$\pi_1(s) = \arg\max_a r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s')$$

Formulation matricielle.

$$V^{\pi} = r + \gamma P V^{\pi}$$

$$Q^{\pi} = r + \gamma P \Pi Q^{\pi}$$

1. Intuition : là, toutes les décisions comptent !
2. Ecrire le problème
3. Planifier — Caractériser une politique optimale
4. Apprentissage par renforcement

Caractériser une politique optimale

On cherche $\pi^* \in \mathscr{D}$ telle que $\pi^* = \arg\max_{\pi \in \mathscr{D}} V^\pi(s)$.

$$V^{\pi^*} = V^*, \; Q^{\pi^*} = Q^*$$

Or ...

Caractériser une politique optimale

On cherche $\pi^* \in \mathscr{D}$ telle que $\pi^* = \arg\max_{\pi \in \mathscr{D}} V^\pi(s)$.

$$V^{\pi^*} = V^*,\ Q^{\pi^*} = Q^*$$

Or ...

Equation d'optimalité de Bellman

La fonction de valeur optimale obéit au système d'équations :

$$V^*(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^*(s) \right\}$$

ou encore :

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

Trouver π^*

Trois approches “canoniques” :

- Programmation dynamique dans l’espace des valeurs
  Itération de la valeur
- Programmation dynamique dans l’espace des politiques
  Itération de la politique
- Programmation linéaire dans l’espace des valeurs

L’objectif de cette section n’est pas de donner une explication détaillée de chaque algorithme, ni de ses dérivés (cela prendrait très longtemps) mais d’illustrer trois manières fondamentalement différentes de rechercher une politique optimale.

Itération de la valeur

Idée clé :

$$V^*(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^*(s) \right\} = LV^*$$

Itération sur les valeurs

- L'opérateur $L : V \mapsto LV$ est contractant.
- L'espace des fonctions de valeur est un espace de Banach.

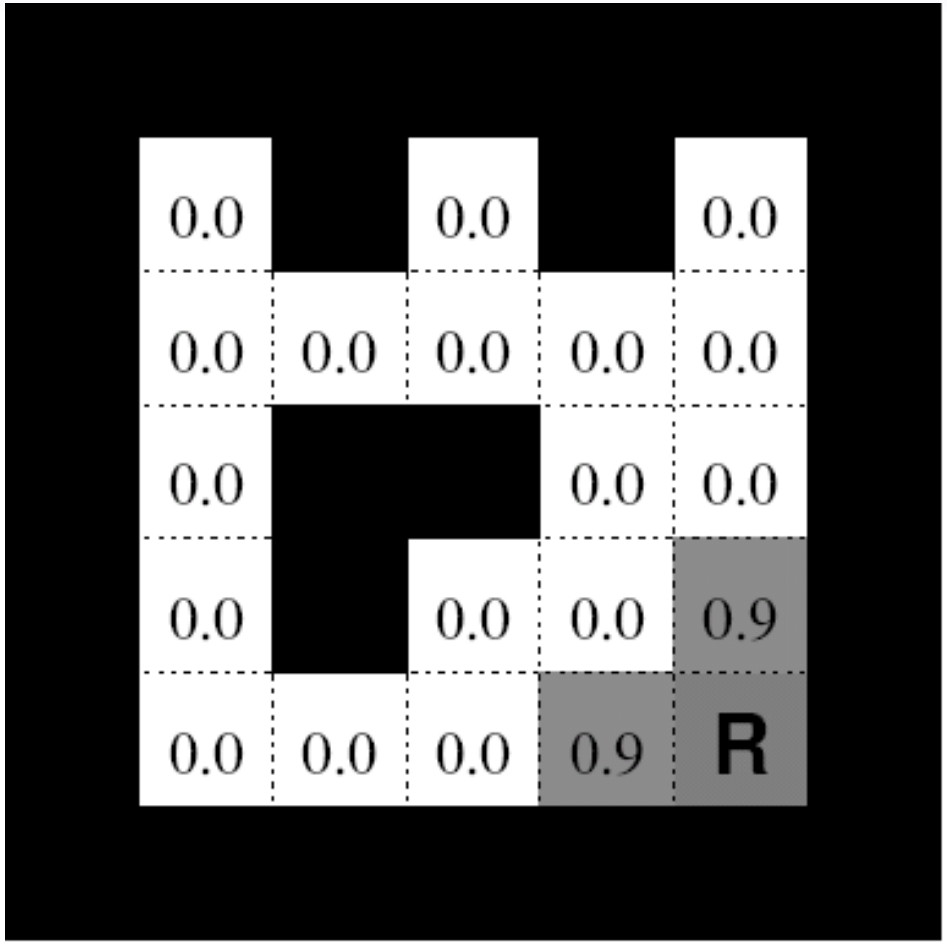
$\Rightarrow$ La suite $V_{n+1} = LV_n$ converge vers V^*.

π^* est la politique gloutonne par rapport à V^*.

Itération de la valeur

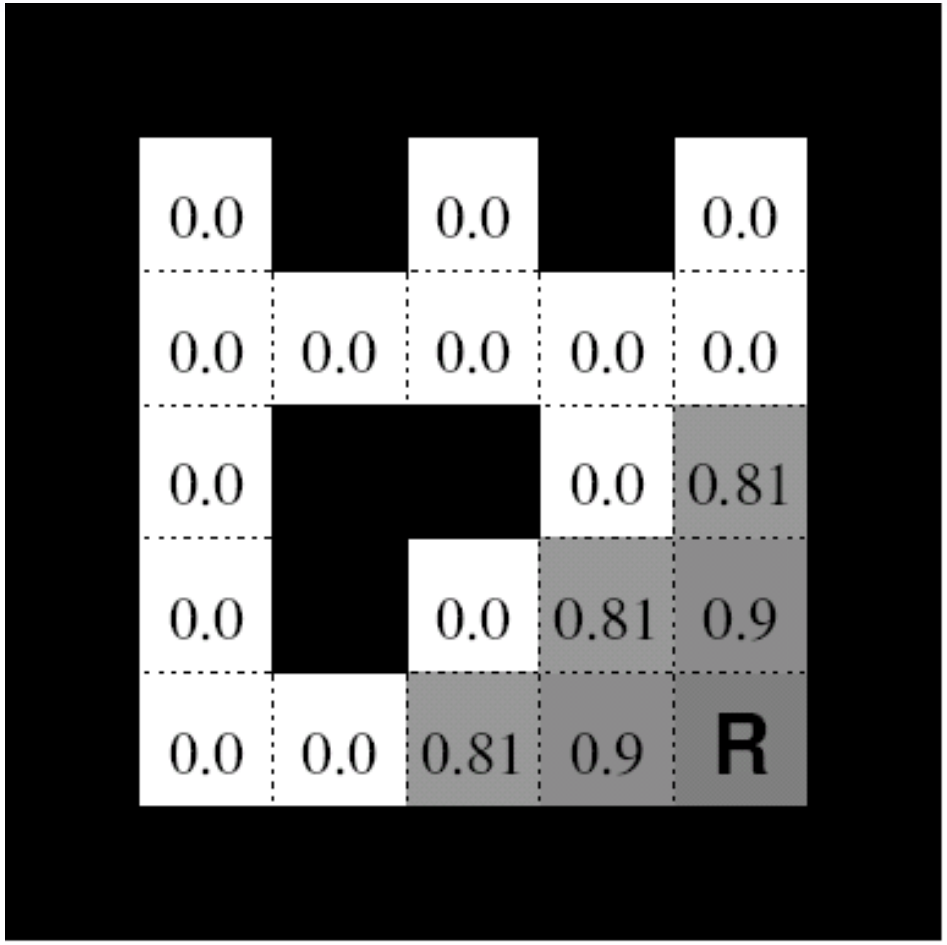
Init: $V' \leftarrow V_0$.
repeat
  $V = V'$
  for $s \in S$ **do**
    $$V'(s) \leftarrow \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V(s') \right\}$$
until $\|V' - V\| \leq \varepsilon$
return greedy policy w.r.t. V'

Illustration



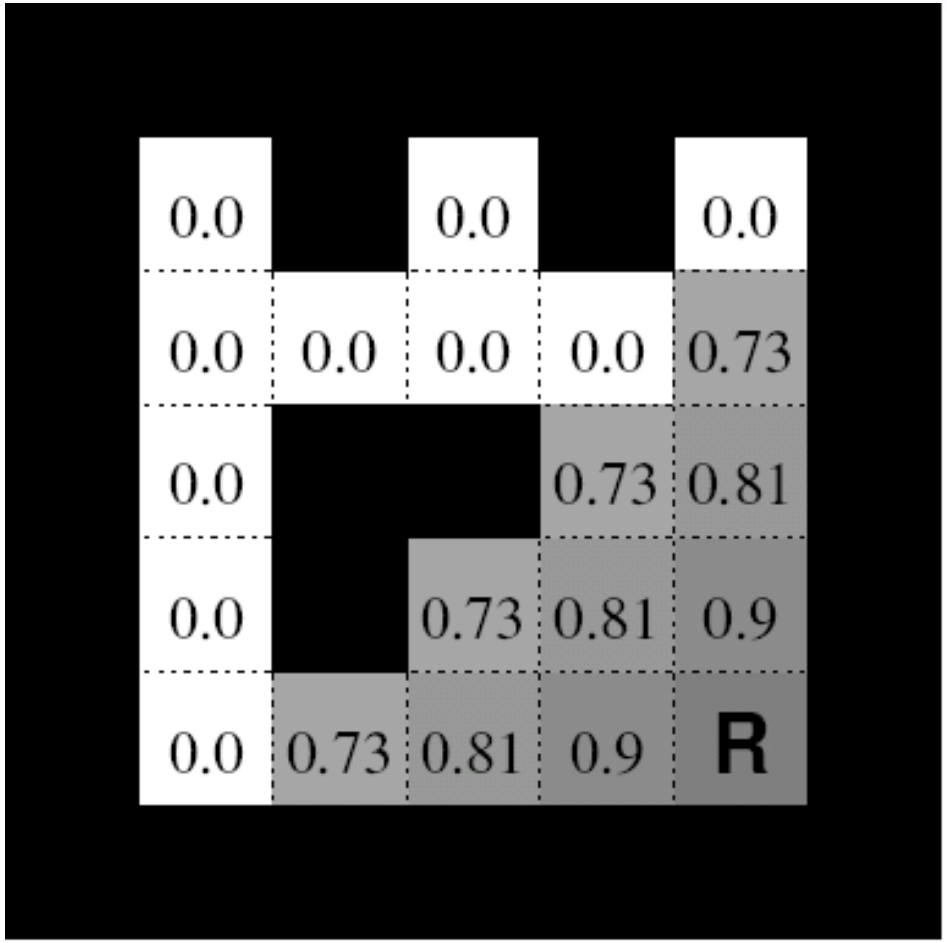
Itération 1

Illustration



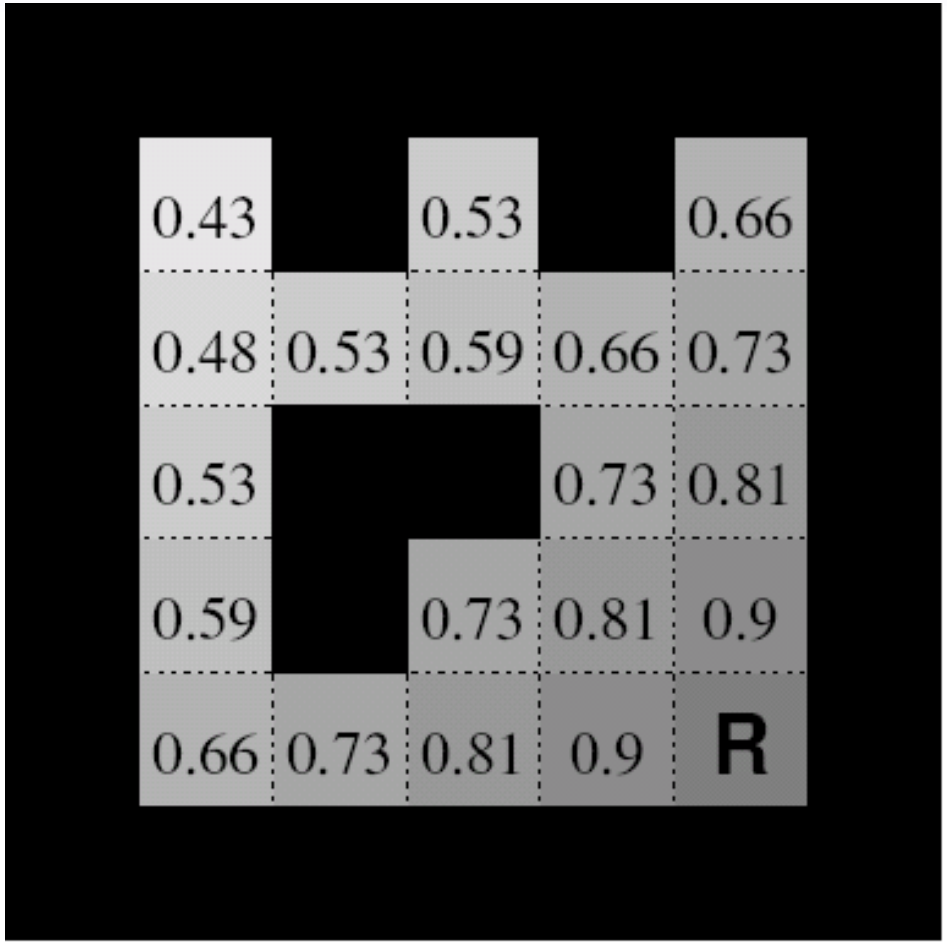
Itération 2

Illustration



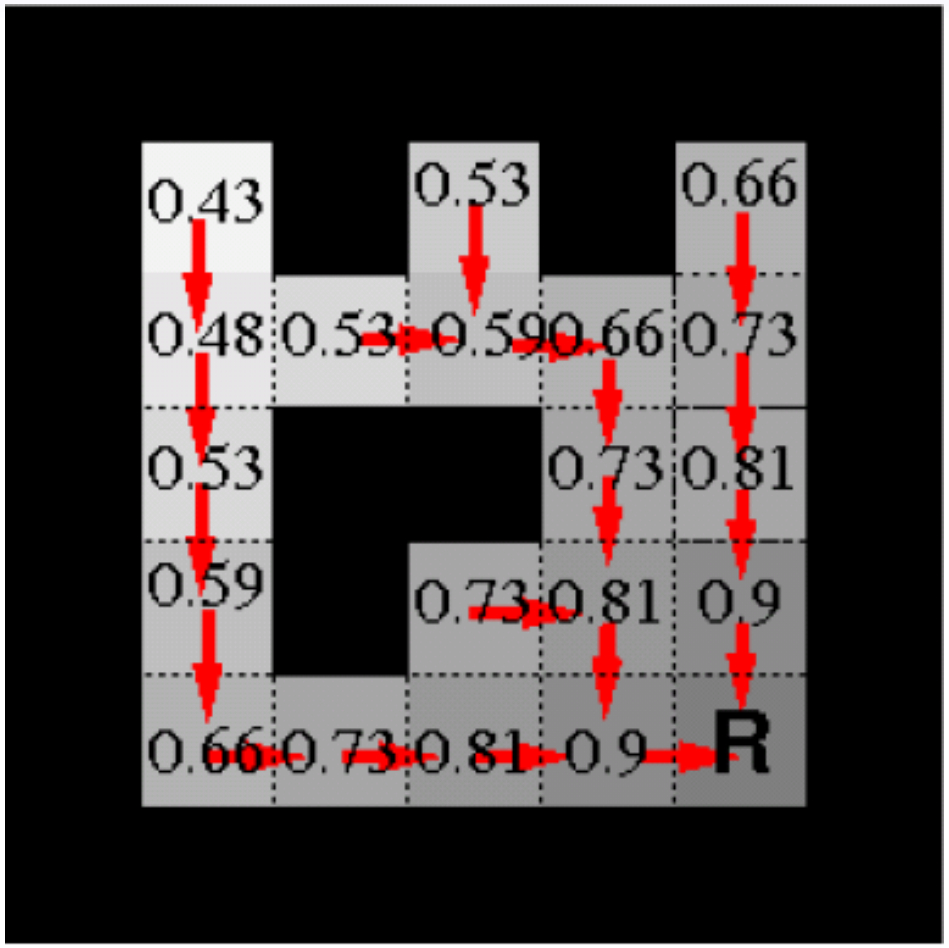
Itération 3

Illustration



Itération 4

Illustration



Politique finale

Itération de la politique

Idée clé :

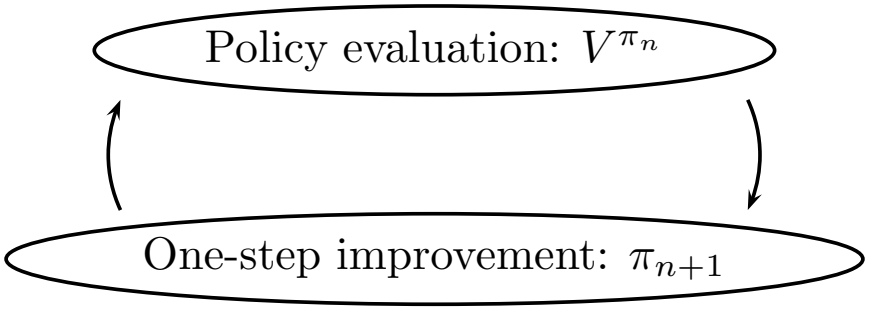
$$\pi^* = \arg\max_{\pi \in \mathscr{D}} V^\pi$$

$$V^\pi = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V^\pi(s') = L^\pi V^\pi$$

Itération sur les politiques

Plutôt que de chercher un chemin $V_0, V_1, \ldots$ parmi les valeurs (itération sur les valeurs), ne peut-on chercher un chemin $\pi_0, \pi_1, \ldots$ parmi les politiques ?
$\longrightarrow$ Améliorer itérativement une politique.

Itération de la politique



Itération de la politique

Init: $\pi' \leftarrow \pi_0$.

repeat

$\pi \leftarrow \pi'$

$V^\pi \leftarrow$ Solve $V = L^\pi V$

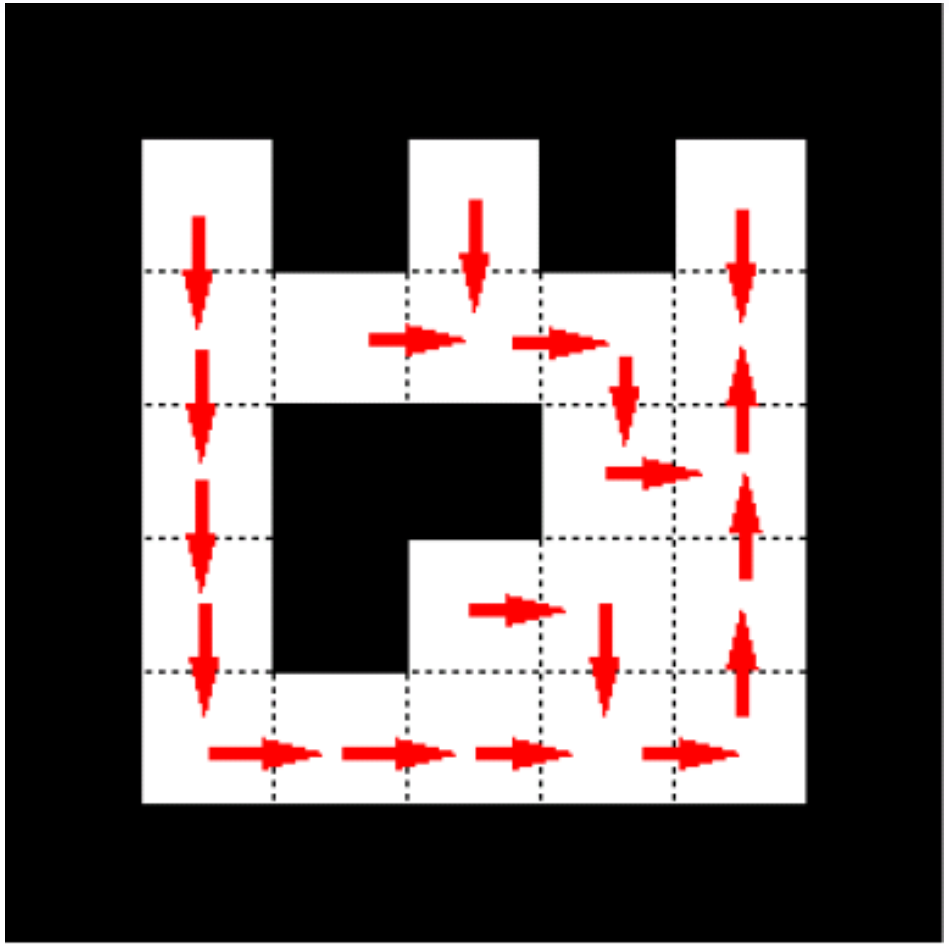
for $s \in S$ **do**

$$\pi'(s) \leftarrow \arg\max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^\pi(s') \right\}$$

until $\pi' = \pi$

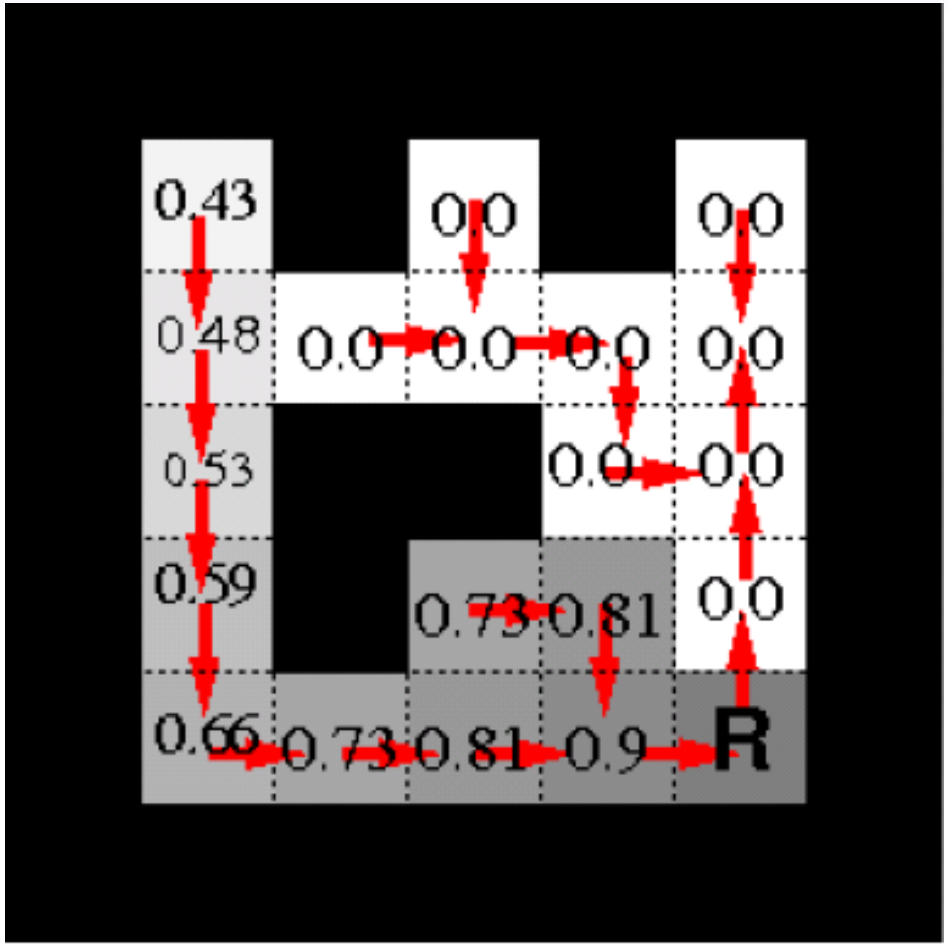
return π

Illustration



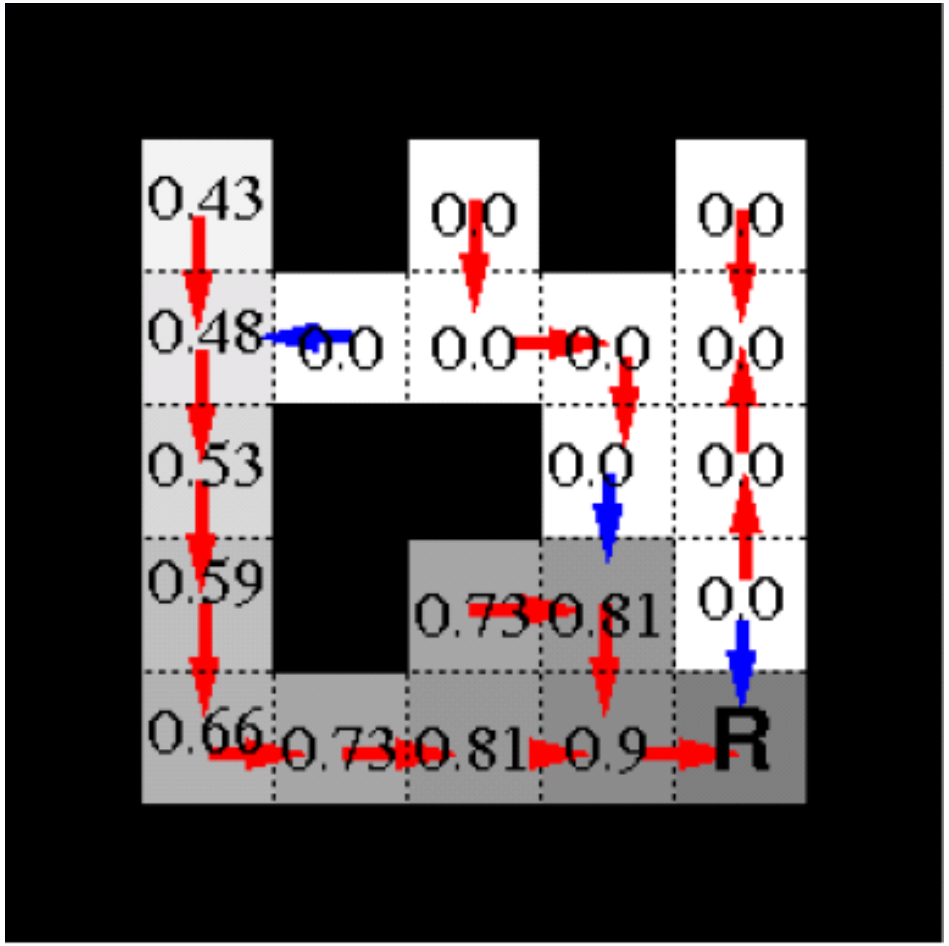
Initialisation

Illustration



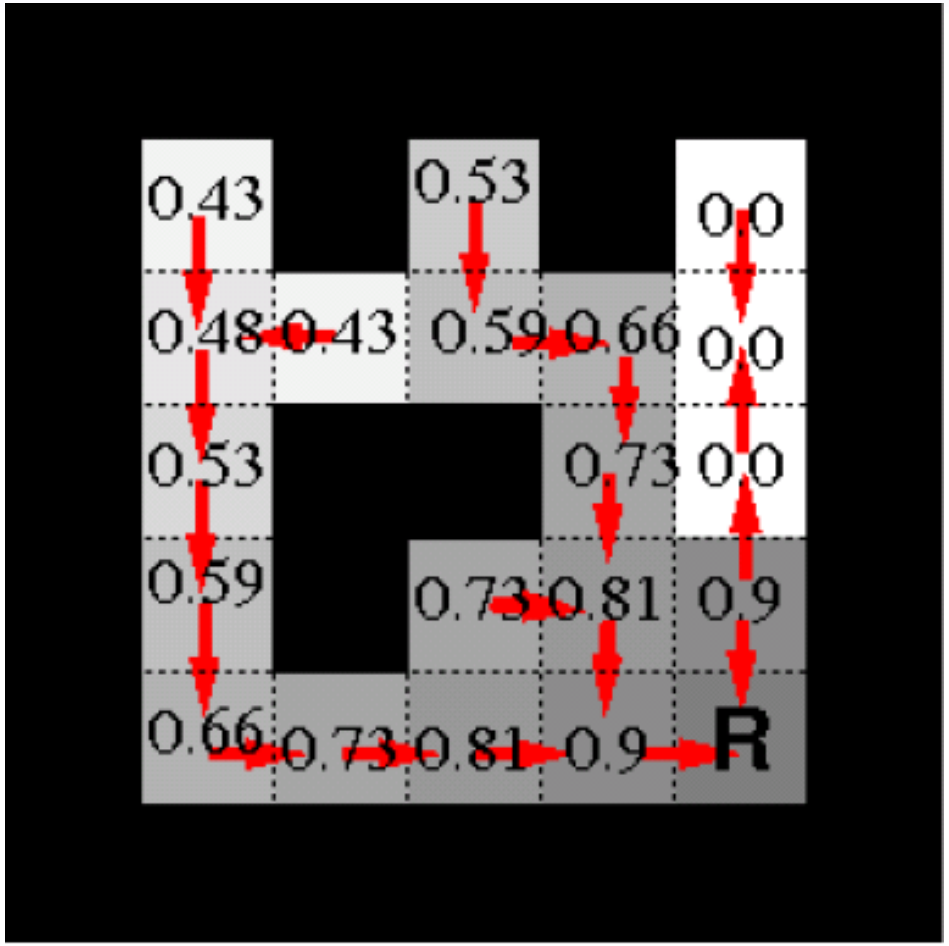
Itération 1 — Evaluation

Illustration



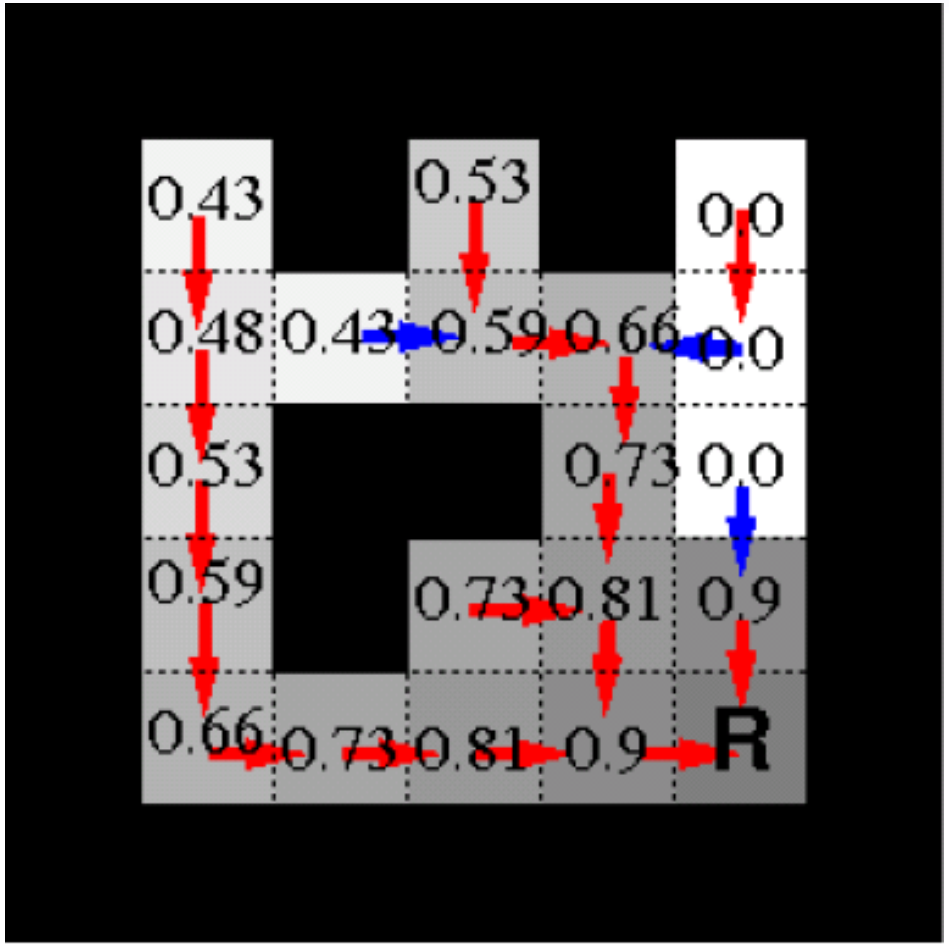
Itération 1 — Amélioration

Illustration



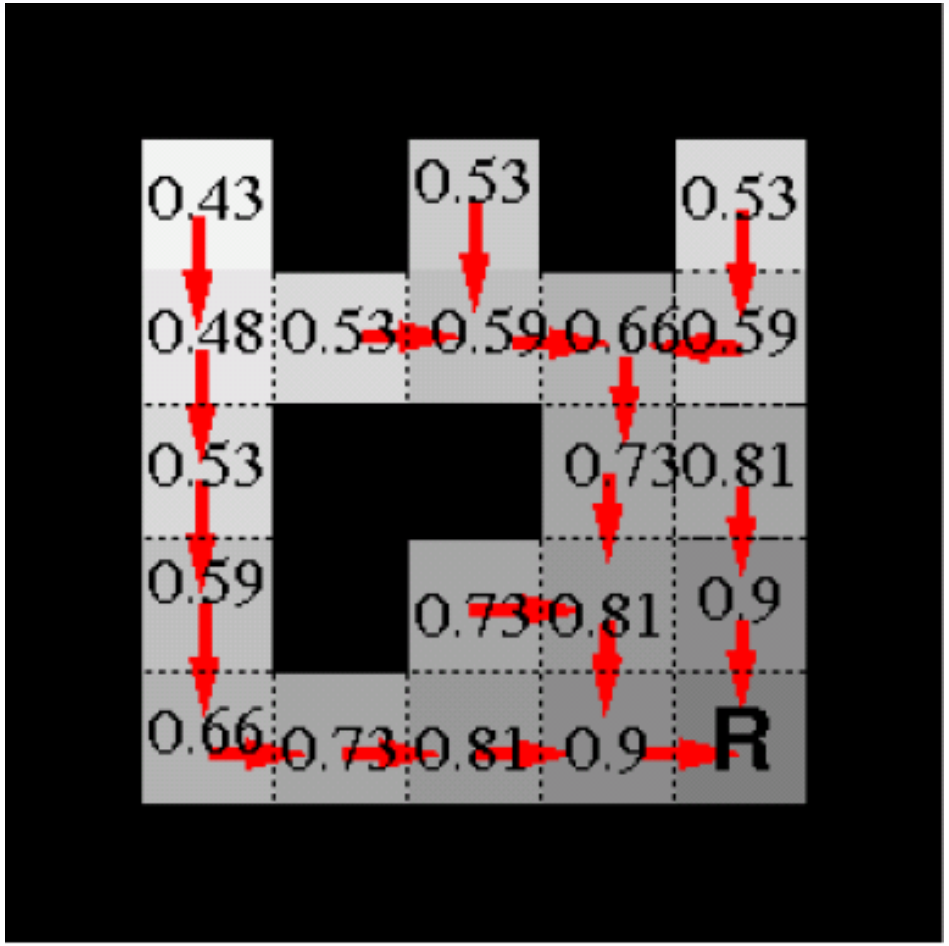
Itération 2 — Evaluation

Illustration



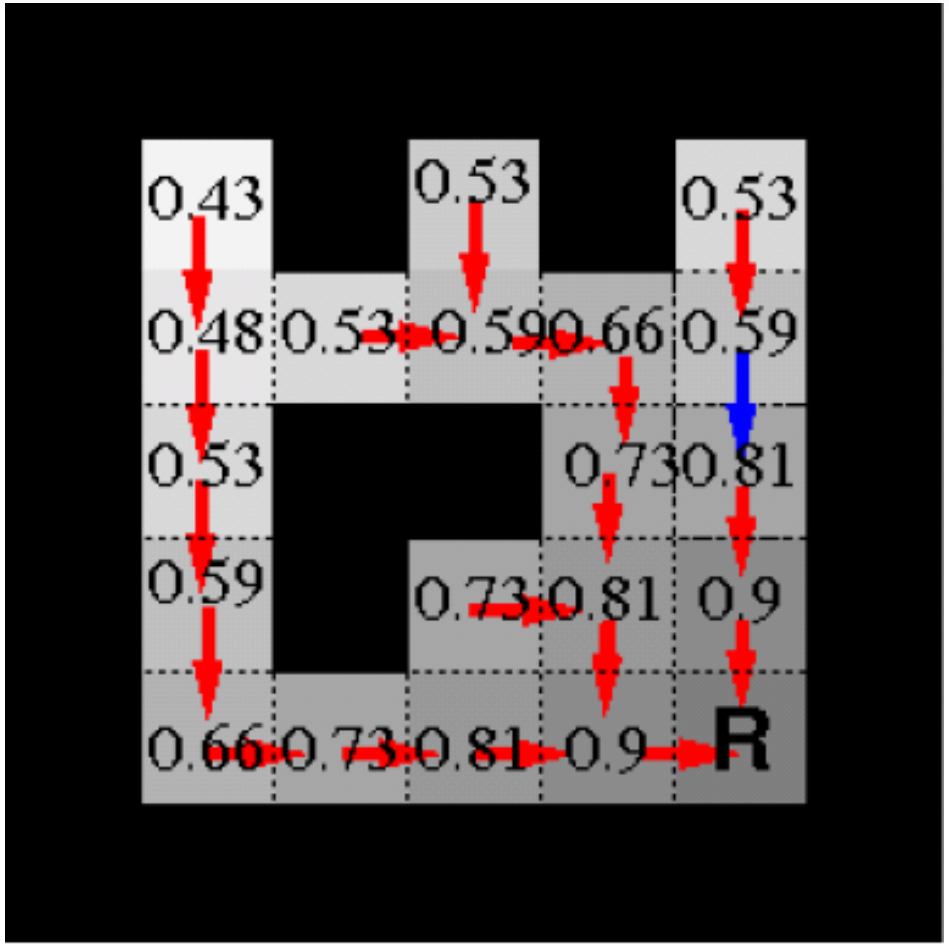
Itération 2 — Amélioration

Illustration



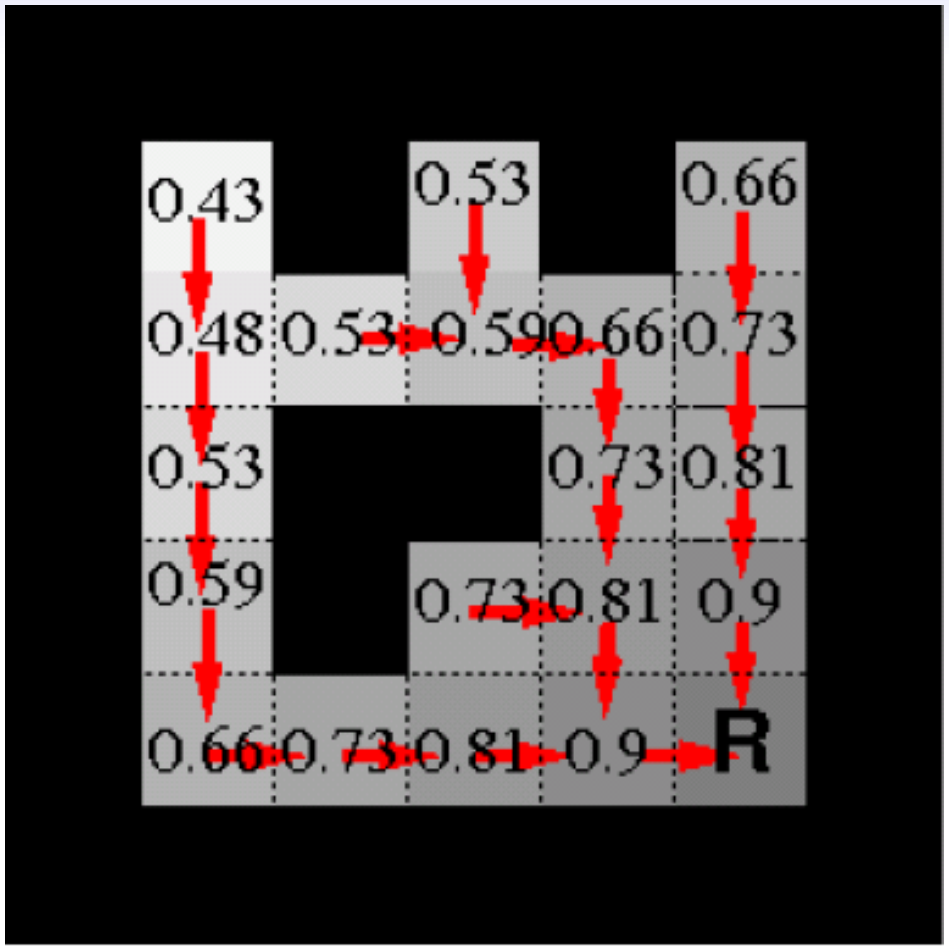
Itération 3 — Evaluation

Illustration



Itération 3 — Amélioration

Illustration



Itération 4 — Evaluation

La politique gloutonne est identique à la politique précédente $\Rightarrow$ stop.

Programmation linéaire

Idée clé : reformuler l'équation d'optimalité comme un problème linéaire.

$$\forall s \in S, V(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V(s') \right\}$$

Programmation linéaire

Idée clé : reformuler l'équation d'optimalité comme un problème linéaire.

$$\forall s \in S, V(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V(s') \right\}$$

$$\Leftrightarrow$$

$$\begin{cases} \min \sum_{s \in S} V(s) \\ s.c. V \geq LV \end{cases}$$

Programmation linéaire

Idée clé : reformuler l'équation d'optimalité comme un problème linéaire.

$$\forall s \in S, V(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V(s') \right\}$$

$$\Leftrightarrow$$

$$\begin{cases} \min \sum_{s \in S} V(s) \\ s.c. V \geq LV \end{cases}$$

$$\Leftrightarrow$$

$$\begin{cases} \min \sum_{s \in S} V(s) \\ \forall (s,a) \in S \times A, \quad V(s) - \gamma \sum_{s' \in S} p(s'|s,a) V(s') \geq r(s,a) \end{cases}$$

1. Intuition : là, toutes les décisions comptent !
2. Ecrire le problème
3. Planifier — Caractériser une politique optimale
4. Apprentissage par renforcement

Remettons tout en perspective

Jusque là, on a caractérisé et cherché des politiques optimales en s'appuyant sur les propriétés supposées de l'environnement.

Remettons tout en perspective

Jusque là, on a caractérisé et cherché des politiques optimales en s'appuyant sur les propriétés supposées de l'environnement.

A chaque fois on fait appel à la connaissance de p et r : on triche !
Où est l'apprentissage ?

RL within ML

ou “L’apprentissage par Renforcement au sein de l’Apprentissage Automatique”

RL within ML

Unsupervised Learning

- learning without a teacher
- information: none
- identify structure in the data
- clustering, self-organization
- k-means, Kohonen maps

Supervised Learning

- learning with a teacher
- information: correct examples
- generalize from examples
- classification, regression
- SVMs, neural networks, trees

Reinforcement Learning

- learning with a critic
- information: trial and error
- reinforce good choices
- value function, control policy
- TD-learning, Q-learning

RL within ML

Unsupervised Learning

- learning without a teacher
- information: none
- identify structure in the data
- clustering, self-organization
- k-means, Kohonen maps

Supervised Learning

- learning with a teacher
- information: correct examples
- generalize from examples
- classification, regression
- SVMs, neural networks, trees

Reinforcement Learning

- learning with a critic
- information: trial and error
- reinforce good choices
- value function, control policy
- TD-learning, Q-learning

RL within ML

Unsupervised Learning

- learning without a teacher
- information: none
- identify structure in the data
- clustering, self-organization
- k-means, Kohonen maps

Supervised Learning

- learning with a teacher
- information: correct examples
- generalize from examples
- classification, regression
- SVMs, neural networks, trees

Reinforcement Learning

- learning with a critic
- information: trial and error
- reinforce good choices
- value function, control policy
- TD-learning, Q-learning

RL within ML

Unsupervised Learning

- learning without a teacher
- information: none
- identify structure in the data
- clustering, self-organization
- k-means, Kohonen maps

Supervised Learning

- learning with a teacher
- information: correct examples
- generalize from examples
- classification, regression
- SVMs, neural networks, trees

Reinforcement Learning

- learning with a critic
- information: trial and error
- reinforce good choices
- value function, control policy
- TD-learning, Q-learning

RL within ML

Unsupervised Learning

- learning without a teacher
- information: none
- identify structure in the data
- clustering, self-organization
- k-means, Kohonen maps

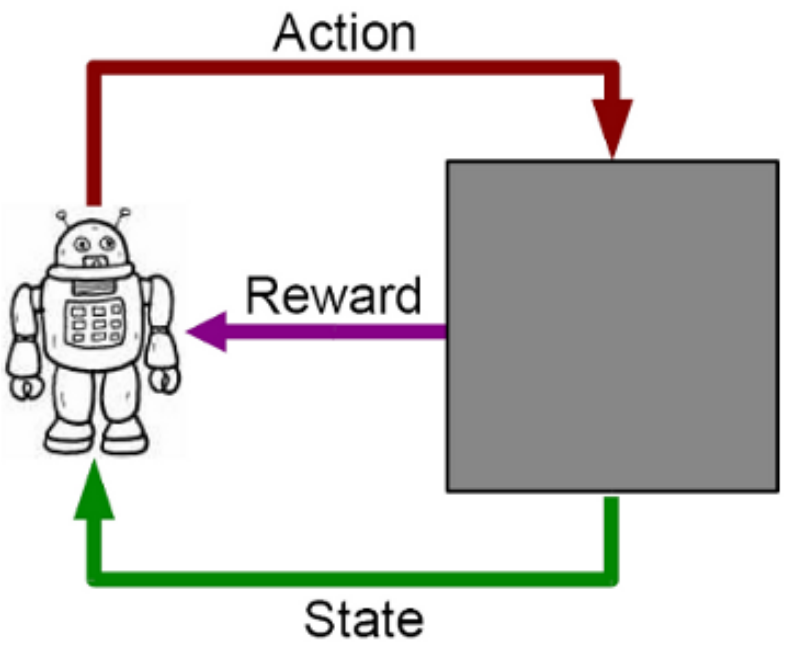
Supervised Learning

- learning with a teacher
- information: correct examples
- generalize from examples
- classification, regression
- SVMs, neural networks, trees

Reinforcement Learning

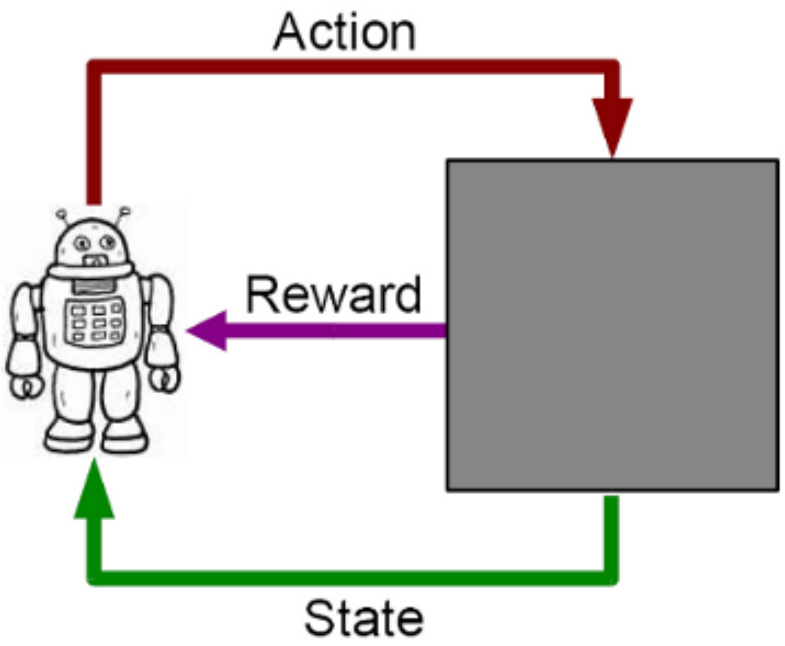
- learning with a critic
- information: trial and error
- reinforce good choices
- value function, control policy
- TD-learning, Q-learning

Apprentissage par renforcement



Evaluer et **améliorer** la politique sur la base de l'**expérience**.

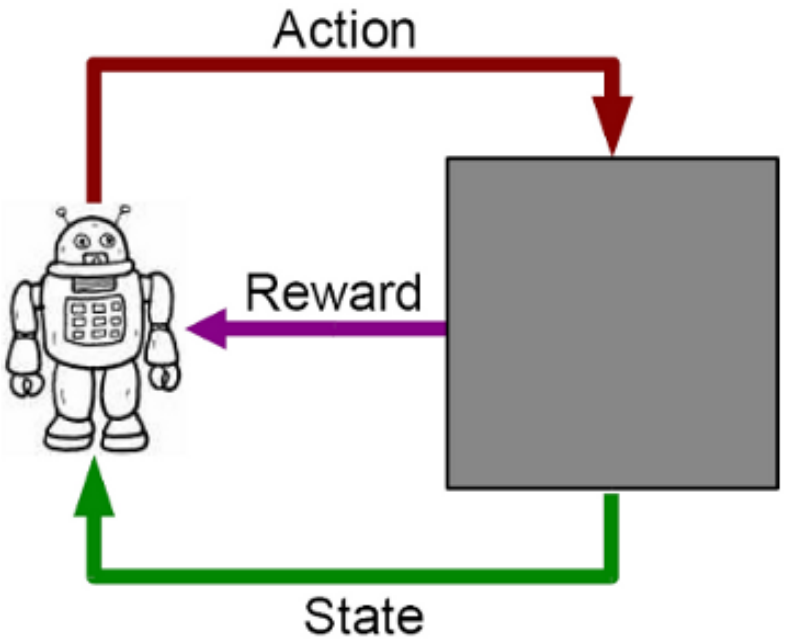
Apprentissage par renforcement



L'"**expérience**" ?

$\rightarrow (s, a, r, s')$

Apprentissage par renforcement



Les deux problèmes de l'AR.

- **Prédire** la valeur d'une politique.
- **Contrôler** le système.

Méthodes et approches

- Model-based vs. model-free.

$$\{(s,a,r,s')\} \quad \rightarrow \quad (p,r) \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

$$\{(s,a,r,s')\} \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

- Online vs. offline.

  Amélioration et prédiction simultanés. Apprentissage incrémental.
  Aspect "biologique", long terme.
  Amélioration et prédiction séparables. Apprentissage "batch" possible.
  Aspect "ingénierie".

- On-policy vs. off-policy.

  La politique suivie pendant l'apprentissage est la politique en cours d'apprentissage.
  La politique apprise et la politique appliquée sont indépendantes.

Méthodes et approches

- Model-based vs. model-free.

$$\{(s,a,r,s')\} \quad \rightarrow \quad (p,r) \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

$$\{(s,a,r,s')\} \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

- Online vs. offline.

Amélioration et prédiction simultanés. Apprentissage incrémental.
Aspect “biologique”, long terme.
Amélioration et prédiction séparables. Apprentissage “batch” possible.
Aspect “ingénierie”.

- On-policy vs. off-policy.

La politique suivie pendant l’apprentissage est la politique en cours d’apprentissage.
La politique apprise et la politique appliquée sont indépendantes.

Méthodes et approches

- Model-based vs. model-free.

$$\{(s,a,r,s')\} \quad \rightarrow \quad (p,r) \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$
$$\{(s,a,r,s')\} \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

- Online vs. offline.

  Amélioration et prédiction simultanés. Apprentissage incrémental.
  Aspect “biologique”, long terme.

  Amélioration et prédiction séparables. Apprentissage “batch” possible.
  Aspect “ingénierie”.

- On-policy vs. off-policy.

  La politique suivie pendant l’apprentissage est la politique en cours d’apprentissage.

  La politique apprise et la politique appliquée sont indépendantes.

Méthodes et approches

- Model-based vs. model-free.

$$\{(s,a,r,s')\} \quad \rightarrow \quad (p,r) \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$
$$\{(s,a,r,s')\} \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

- Online vs. offline.

Amélioration et prédiction simultanés. Apprentissage incrémental.
Aspect “biologique”, long terme.

Amélioration et prédiction séparables. Apprentissage “batch” possible.
Aspect “ingénierie”.

- On-policy vs. off-policy.

La politique suivie pendant l’apprentissage est la politique en cours d’apprentissage.

La politique apprise et la politique appliquée sont indépendantes.

Méthodes et approches

- Model-based vs. model-free.

$$\{(s,a,r,s')\} \quad \rightarrow \quad (p,r) \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$
$$\{(s,a,r,s')\} \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

- Online vs. offline.

Amélioration et prédiction simultanés. Apprentissage incrémental.
Aspect “biologique”, long terme.
Amélioration et prédiction séparables. Apprentissage “batch” possible.
Aspect “ingénierie”.

- On-policy vs. off-policy.

La politique suivie pendant l’apprentissage est la politique en cours d’apprentissage.
La politique apprise et la politique appliquée sont indépendantes.

Méthodes et approches

- Model-based vs. model-free.

$$\{(s,a,r,s')\} \quad \rightarrow \quad (p,r) \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$
$$\{(s,a,r,s')\} \quad \rightarrow \quad V^\pi \text{ ou } \pi^*$$

- Online vs. offline.

  Amélioration et prédiction simultanés. Apprentissage incrémental. Aspect “biologique”, long terme.

  Amélioration et prédiction séparables. Apprentissage “batch” possible. Aspect “ingénierie”.

- On-policy vs. off-policy.

  La politique suivie pendant l’apprentissage est la politique en cours d’apprentissage.

  La politique apprise et la politique appliquée sont indépendantes.

Prediction

ou

Passive Reinforcement Learning

Trois principales méthodes de prédiction

- Adaptive Dynamic Programming.

Modèle ?	Model-based
Online ?	NA
Learning policy ?	Off-policy

- Direct Utility Estimation.

Modèle ?	Model-free
Online ?	NA
Learning policy ?	On-policy

- Temporal Differences.

Modèle ?	Model-free
Online ?	NA
Learning policy ?	Off-policy

Adaptive Dynamic Programming

$$\{(s, a, r, s')\}$$
$$\downarrow$$
$$\begin{cases} \text{Comptage freq. ou param. adapt.} & \longrightarrow \quad p \\ \text{Moyenne} & \longrightarrow \quad r \end{cases}$$
$$\downarrow$$
$$\text{Résolution de } V = L^{\pi} V.$$

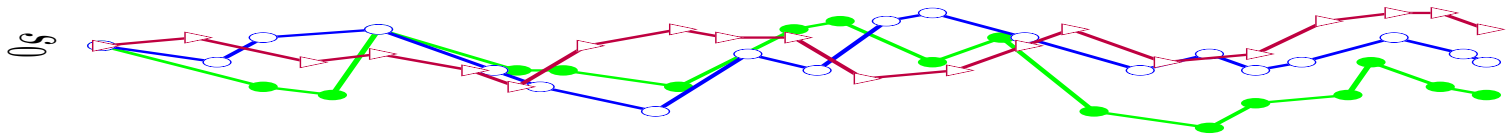
Propriétés :

- Converge à la limite vers p, r et V^{π} si échantillons i.i.d.
- Excellent usage des échantillons.
- ...mais énorme complexité en échantillons (sample complexity).

Direct Utility Estimation

$$V^{\pi}(s) = E\left(\sum_{\delta=0}^{\infty} \gamma^{\delta} r_{\delta} \middle| s_0 = s\right)$$

Estimation de Monte-Carlo :
1 traj. de longueur l en s = 1 échant. de $V^{\pi}(s) \pm \frac{\gamma^{l}(r_{max} - r_{min})}{1-\gamma}$.



Propriétés :

- Converge à la limite en nombre de trajectoires. Simple mais lent.
- Ignore les dépendances entre valeurs (équation de Bellman).
- Mauvais usage des échantillons (s, a, r, s').
- Très grosse complexité en échantillons.

Temporal Differences

Idée : apprendre localement V^π en respectant l'équation de Bellman.

Différence Temporelle

On considère une trajectoire $s_0, a_0, r_0, s_1, \ldots s_n$.

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

$\delta_t \leftrightarrow$ erreur sur $V(s_t)$ étant donnée l'équation de Bellman

TD learning : à chaque échantillon (s, a, r, s'),

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha\left(r + \gamma V^\pi(s') - V^\pi(s)\right)$$

- Décroissance du taux d'appr. α selon un "agenda" $\sigma_\alpha(\alpha_0, \alpha, n)$.
- Convergence rapide mais peut "osciller" plus que DUE.
- Bonne complexité en échantillons.

Temporal Differences

Idée : apprendre localement V^π en respectant l'équation de Bellman.

Différence Temporelle

On considère une trajectoire $s_0, a_0, r_0, s_1, \ldots s_n$.

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

$\delta_t \leftrightarrow$ erreur sur $V(s_t)$ étant donnée l'équation de Bellman

TD learning : à chaque échantillon (s, a, r, s'),

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \left(r + \gamma V^\pi(s') - V^\pi(s)\right)$$

Note : ci-dessus, version "online", mais il existe une version "batch" dont on parlera dans la deuxième partie.

Contrôle

ou

Active Reinforcement Learning

Remarques préliminaires

Pas de modèle ? Alors $Q(s, a)$ plutôt que $V(s)$.

Le dilemme exploration vs. exploitation :
exploiter la meilleure π trouvée pour maximiser le gain à court terme
ou
explorer pour découvrir de nouvelles parties de l'environnement ?

Politique gloutonne à la limite (d'une exploration infinie) (GLIE).

Remarques préliminaires

Pas de modèle ? Alors $Q(s, a)$ plutôt que $V(s)$.

Le dilemme exploration vs. exploitation :
exploiter la meilleure π trouvée pour maximiser le gain à court terme
ou
explorer pour découvrir de nouvelles parties de l'environnement ?

Politique gloutonne à la limite (d'une exploration infinie) (GLIE).

Remarques préliminaires

Pas de modèle ? Alors $Q(s, a)$ plutôt que $V(s)$.

Le dilemme exploration vs. exploitation :
exploiter la meilleure π trouvée pour maximiser le gain à court terme
ou
explorer pour découvrir de nouvelles parties de l'environnement ?

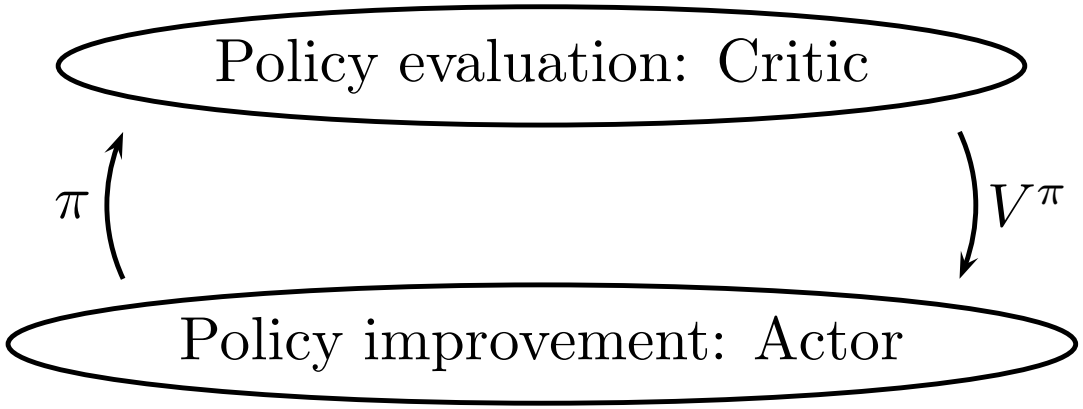
Politique gloutonne à la limite (d'une exploration infinie) (GLIE).

Architectures

- Model-based : architecture Dyna.
  Planifier incrémentalement avec un modèle appris incrémentalement.
  $\longrightarrow$ DYNA-PI, DYNA-Q, etc.
- Model-free : cadre Actor-Critic.
  Nombreux synonymes pour une idée commune :
  Asynchronous Policy Iteration, Optimistic Policy Iteration,
  Generalized Policy Iteration, Actor-Critic architecture, ...

Architecture Actor-Critic

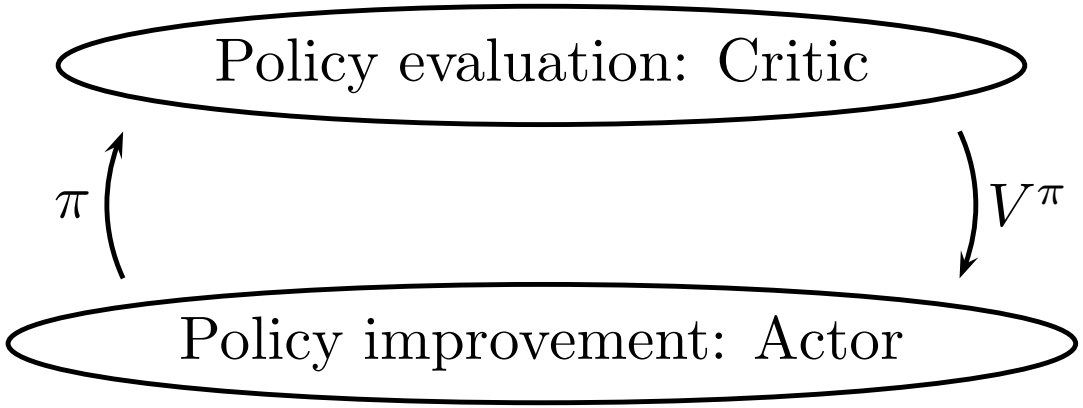
Généralisation de l'idée d'Itération de la politique.



L'itération de la valeur / de la politique entrent dans ce cadre !
L'itération de la politique modifiée également.
Ainsi que tous les algorithmes d'itération de la valeur/politique approchés.

Architecture Actor-Critic

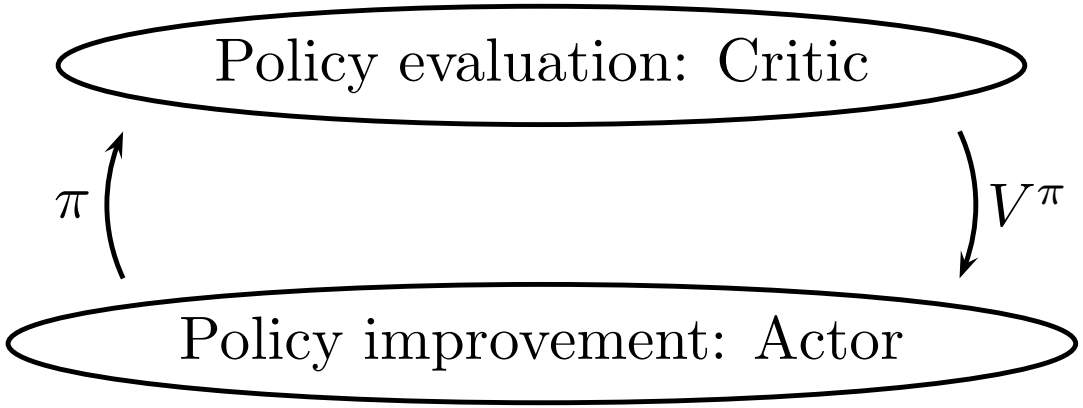
Généralisation de l'idée d'Itération de la politique.



Les notions et preuves de convergence et d'optimalité ne sont pas forcément toujours les mêmes dans les algos actor-critic.

Architecture Actor-Critic

Généralisation de l'idée d'Itération de la politique.



L'évaluation par méthodes TD permet de construire une famille d'algorithmes online.

SARSA

Idée

Utiliser TD sur les Q-valeurs pour chercher la fonction Q^* en faisant varier la politique d'exploration de l'aléatoire complet au glouton.

Etant donné (s, a, r, s', a') avec $a' = \pi(s')$:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma Q(s',a') - Q(s,a)\right)$$

π est un compromis entre π_0 et la politique gloutonne sur Q.

Propriétés :

- Méthode on-policy (l'exploration est contrainte par a').
- π_0 totalement aléatoire $\rightarrow \pi^*$ optimale.
- Converge lentement.

SARSA

Init: $D = \{(s, a, r, s')\}, Q \leftarrow Q_0, \pi \leftarrow \pi_0, \alpha \leftarrow \alpha_0, n \leftarrow 0.$
Init: σ_α - agenda d'α.
Init: σ_π - agenda de π.
for $(s, a, r, s') \in D$ **do**
  $Q(s, a) \leftarrow Q(s, a) + \alpha\,(r + \gamma Q(s', \pi(s')) - Q(s, a))$
  $\alpha \leftarrow \sigma_\alpha(\alpha_0, \alpha, n)$
  $\pi \leftarrow \sigma_\pi(\pi_0, \pi, n)$
  $n = n + 1$
return Q

σ_α et σ_π sont cruciaux pour la convergence, l'optimalité et surtout l'efficacité.

Q-learning

Idée : Incorporer l'étape de maximisation de la valeur dans la mise à jour TD.
Etant donné (s, a, r, s') :

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

Propriétés :

- Méthode off-policy.
- Converge lentement (*).

Q-learning

Init: $D = \{(s, a, r, s')\}, Q \leftarrow Q_0, \pi \leftarrow \pi_0, \alpha \leftarrow \alpha_0, n \leftarrow 0.$
Init: σ_α - agenda d'α.
for $(s, a, r, s') \in D$ **do**

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s,a) \right)$$

$$\alpha \leftarrow \sigma_\alpha(\alpha_0, \alpha, n)$$

$$n = n + 1$$

return Q

Vers l'infini et au delà ...

Au delà de ces algorithmes, plusieurs problèmes :

- Compromis exploration vs. exploitation optimal ?
  Algorithmes E^3 et R_{max}.
- Orienter la recherche de façon pertinente pour V^* ?
  Approches heuristiques (asynchrones) RTDP (& Co.), LAO* (& Co.).
- Rendre les approches DUE et la recherche dans les arbres efficaces ?
  Approches type "bandits" : RSPI, UCT.
- Traiter des très grands espaces d'états/actions ?
  Approches d'abstraction par approximation (Approximations linéaires, LSTD, LSPI), hiérarchisation (MAXQ, OORL), factorisation, ...
- ...

1. **Intuition : là, toutes les décisions comptent !**
2. **Ecrire le problème**
3. **Planifier — Caractériser une politique optimale**
4. **Apprentissage par renforcement**

Résumons ...

L'apprentissage par renforcement :

- Apprendre à prendre une **séquence** de décisions de façon **optimale**,
- sans **modèle** donné a priori.
- Cadre différent de l'apprentissage supervisé / non-supervisé.
- Vaste domaine avec ses sous-catégories d'algos et de problèmes : Model-based vs. model-free, online vs. offline, etc.
- Sujet de recherche très actif :
  Certaines grandes réussites (commande d'actionneurs en robotique, gestion de flotille d'ascenceurs, programme champion du monde de backgammon, meilleur programme de Go, stratégie de remplacement de flotte de satellites, etc.)
- ... et comportant encore de nombreux défis !

Résumons ...

L'apprentissage par renforcement :

- Apprendre à prendre une **séquence** de décisions de façon **optimale**,
- sans **modèle** donné a priori.
- Cadre différent de l'apprentissage supervisé / non-supervisé.
- Vaste domaine avec ses sous-catégories d'algos et de problèmes : Model-based vs. model-free, online vs. offline, etc.
- Sujet de recherche très actif :
  Certaines grandes réussites (commande d'actionneurs en robotique, gestion de flotille d'ascenceurs, programme champion du monde de backgammon, meilleur programme de Go, stratégie de remplacement de flotte de satellites, etc.)
- ... et comportant encore de nombreux défis !

Résumons . . .

L'apprentissage par renforcement :

- Apprendre à prendre une **séquence** de décisions de façon **optimale**,
- sans **modèle** donné a priori.
- Cadre différent de l'apprentissage supervisé / non-supervisé.
- Vaste domaine avec ses sous-catégories d'algos et de problèmes : Model-based vs. model-free, online vs. offline, etc.
- Sujet de recherche très actif :
  Certaines grandes réussites (commande d'actionneurs en robotique, gestion de flotille d'ascenceurs, programme champion du monde de backgammon, meilleur programme de Go, stratégie de remplacement de flotte de satellites, etc.)
- . . . et comportant encore de nombreux défis !

Résumons . . .

L'apprentissage par renforcement :

- Apprendre à prendre une **séquence** de décisions de façon **optimale**,
- sans **modèle** donné a priori.
- Cadre différent de l'apprentissage supervisé / non-supervisé.
- Vaste domaine avec ses sous-catégories d'algos et de problèmes : Model-based vs. model-free, online vs. offline, etc.
- Sujet de recherche très actif :
  Certaines grandes réussites (commande d'actionneurs en robotique, gestion de flotille d'ascenceurs, programme champion du monde de backgammon, meilleur programme de Go, stratégie de remplacement de flotte de satellites, etc.)
- . . . et comportant encore de nombreux défis !

Résumons . . .

L'apprentissage par renforcement :

- Apprendre à prendre une **séquence** de décisions de façon **optimale**,
- sans **modèle** donné a priori.
- Cadre différent de l'apprentissage supervisé / non-supervisé.
- Vaste domaine avec ses sous-catégories d'algos et de problèmes : Model-based vs. model-free, online vs. offline, etc.
- Sujet de recherche très actif :
  Certaines grandes réussites (commande d'actionneurs en robotique, gestion de flotille d'ascenceurs, programme champion du monde de backgammon, meilleur programme de Go, stratégie de remplacement de flotte de satellites, etc.)
- . . . et comportant encore de nombreux défis !

Résumons . . .

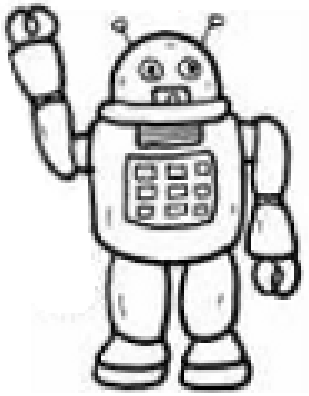
L'apprentissage par renforcement :

- Apprendre à prendre une **séquence** de décisions de façon **optimale**,
- sans **modèle** donné a priori.
- Cadre différent de l'apprentissage supervisé / non-supervisé.
- Vaste domaine avec ses sous-catégories d'algos et de problèmes : Model-based vs. model-free, online vs. offline, etc.
- Sujet de recherche très actif :
  Certaines grandes réussites (commande d'actionneurs en robotique, gestion de flotille d'ascenceurs, programme champion du monde de backgammon, meilleur programme de Go, stratégie de remplacement de flotte de satellites, etc.)
- . . . et comportant encore de nombreux défis !

Un peu de biblio

Sutton, R. S. and Barto, A. G. (1998).
Reinforcement Learning, an introduction.
The MIT Press, Cambridge, MA.

Bertsekas, D. P., and Tsitsiklis, J. N. (1996).
Neuro-Dynamic Programming.
Athena Scientific.

Puterman, M. L. (1994).
Markov Decision Processes.
John Wiley & Sons, Inc..

Ouf ! Questions ?