Apprentissage par renforcement

Cours 9
ARF Master DAC

Nicolas Baskiotis
nicolas.baskiotis@lip6.fr
http://webia.lip6.fr/~baskiotisn

équipe MLIA, Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie (UPMC)

S2 (2016-2017)

Plan

Introduction

Qui a-t-il de commun entre :

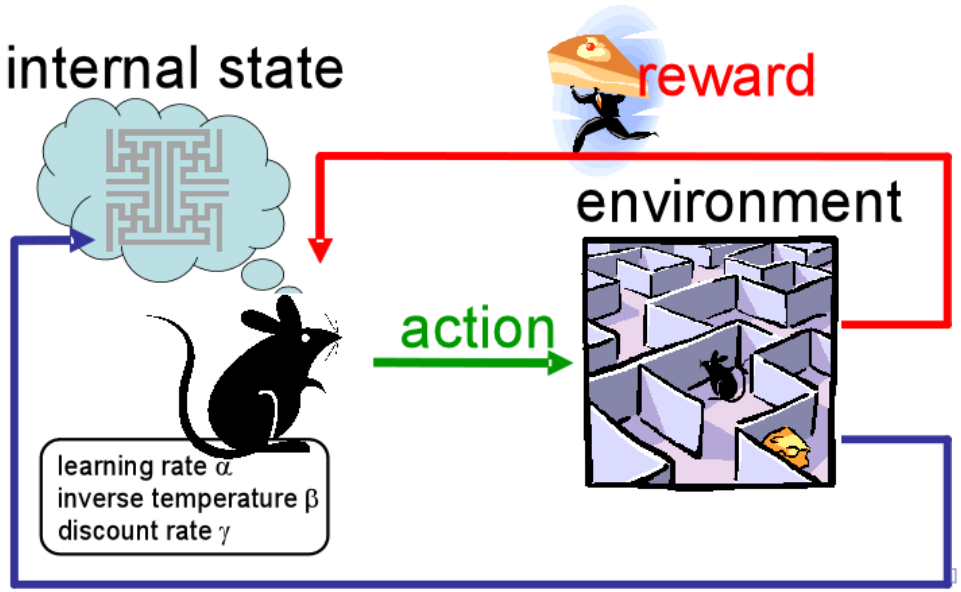
- Chien de Pavlolv
- Boite de Skinner
- Un drone/véhicule autonome
- jeu d'echecs

Principe

Lexique

- Agent, Environnement
- Etat (*state*) : ce que perçoit l'agent
- Action : une interaction de l'agent avec l'environnement
- Récompense (*reward*) : une quantité perçue après chaque action
- Politique (*policy*) : une fonction de sélection de l'action selon l'état

Objectif : trouver une politique qui permet de maximiser l'ensemble des récompenses reçues

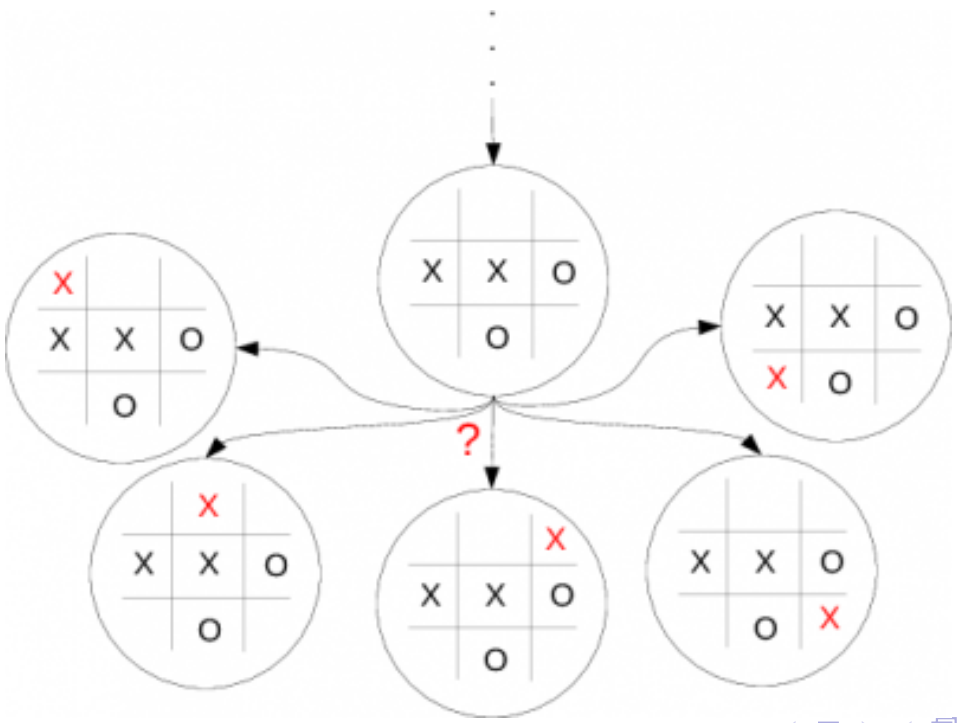


Objectif : adaptation du système à son environnement

Reproduction artificielle du comportement du “conditionnement”

Comment :

- enseigner un comportement à l’aide de récompenses ?
- réagir à une situation donnée ?
- agir de manière à maximiser les récompenses ?



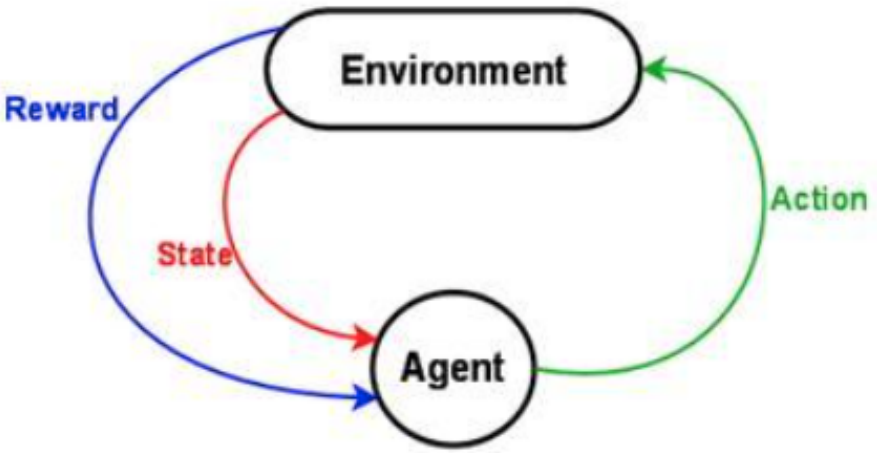
Plan

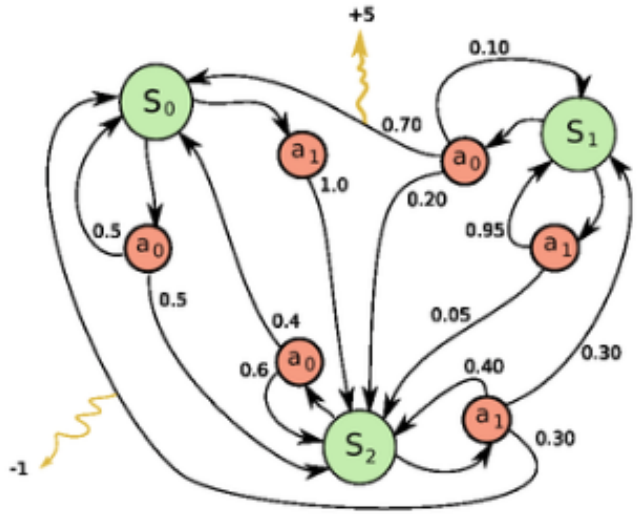
Markov Decision Process (MDP)

Définition du modèle

- $\mathcal{S}$: un espace d'états
- $\mathcal{A}$: un espace d'actions
- $T : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$: fonction de transition
- $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$: récompense

Hypothèse markovienne : la récompense et la fonction de transition ne dépendent que de l'état (et action) en cours, pas de l'historique.



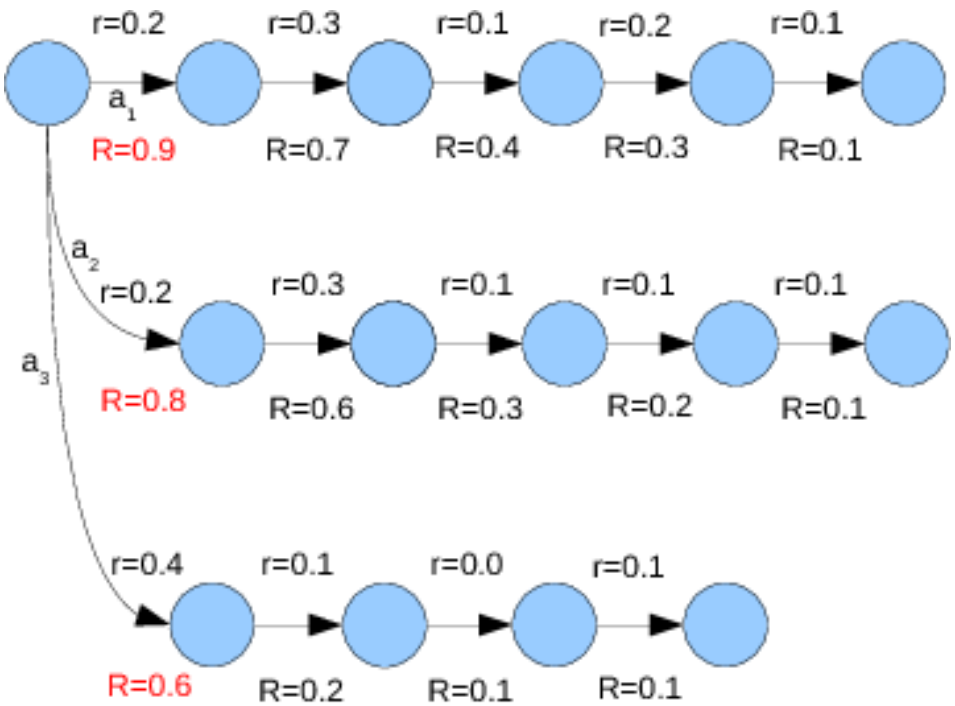


Problèmatique

Comment choisir une action ?

- regarder la récompense liée à chaque action
- Mais aussi les récompenses futurs !

$\Rightarrow$ **fonction de valeur** d'états (ou d'état/action) : indication sur le long terme des récompenses attendues ($\neq$ récompenses immédiates)



Formalisation

Fonctions valeur

- Reflète la récompense à moyen terme $\rightarrow$ aggrégation des récompenses
- Sur une séquence : $R_{t_0} = \sum_{t=t_0}^{t_T} \gamma^t r_t$
- $V^\pi(s) = \mathbb{E}_\pi(R_{t_0} | s_{t_0} = s) = \mathbb{E}_\pi \left[\sum_{t=t_0}^{t_T} \gamma^t r(s_t, \pi(s_t), s_{t+1}) \middle| s_{t_0} = s \right]$
- $Q^\pi(s,a) = \mathbb{E}_\pi(R_{t_0} | s_{t_0} = s, a_{t_0} = a)$

Définitions

- Une politique $\pi : \mathcal{S} \rightarrow \mathcal{A}$ (ou dans le cas probabiliste dans $\Pi(\mathcal{A})$
- Une fonction de valeur d'états : $V^\pi : \mathcal{S} \rightarrow \mathbb{R}$
- Une fonction de valeur d'actions : $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- Une séquence (un scénario) :
  $[(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_n, a_n, r_n)], \quad (s_i, a_i, r_i) \in \mathcal{S} \times \mathcal{A} \times \mathbb{R}$

Equation de Bellman

Quelques relations

- π et MDP déterministe : $V^{\pi}(s) = r(s, \pi(s)) + \gamma V^{\pi}(s')$
- MDP non déterministe : $V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s'|s, \pi(s)) V^{\pi}(s')$
- non déterministe : $V^{\pi}(s) = \sum_{a} \pi(s, a)[r(s, a) + \gamma \sum_{s'} p(s'|s, a) V^{\pi}(s')]$
- même relation pour $Q(s, a)$

Comparaison de politiques

- $\pi \geq \pi' \iff \forall s \in \mathcal{S} V^{\pi}(s) \geq V^{\pi'}(s)$
- Il existe une politique optimale, noté π^*
- $V^*(s) = max_{\pi} V^{\pi}(s) = max_{a} \; r(s, a) + \gamma \sum_{s'} p(s'|s, a) V^*(s')$
- $Q^*(s, a) = max_{\pi} Q^{\pi}(s, a) = \mathbb{E}(r_{t+1} + \gamma V^*(s_{t+1}|s_t = s, a_t = a))$
  $= r(s, a) + \gamma max_{a} \; \sum_{s'} p(s'|s, a) Q^*(s', a')$

Différentes approches

- Le MDP est donné : problème d'optimisation et de planification (programmation dynamique).
- Apprentissage par renforcement : le MDP est inconnu (seules les actions sont connues).
  - on cherche uniquement à trouver une politique π : model-free
  - on estime le MDP puis on en déduit π : model-based

- Value iteration : itère uniquement sur la fonction de valeur d'actions, puis en déduit une politique
- Policy iteration : mise-à-jour simultanée de la politique et la fonction valeur
- Et pour les états continus ? actions continus ?

$\Rightarrow$ approximation par apprentissage.

Plan

Résolution exacte : Programmation dynamique

MDP connu, déterministe ou non

Il est possible de résoudre exactement le problème pour un MDF fini.

- Opérateur de Bellman : T^π, opérateur de mise-à-jour de V sur un pas : $V(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s'|s, \pi(s)) V^\pi(s')$
- Opérateur d'optimalité de Bellman : T^* : $V(s) = max_a [r(s, a) + \gamma \sum_{s'} p(s'|s, a) V(s')]$
- Opérateur contractant $\rightarrow$ existence d'un point fixe

Les questions :

- Comment évaluer une politique ?
- Comment améliorer une politique ?

Evaluation de politique

Evaluation itérative

- La valeur d'une politique est entièrement déterminer par $V^{\pi}(s)$
- Comment évaluer V^{π} ?

$\Rightarrow$ Utilisation itérative de l'opérateur T^{π} :
$V(x)^{t+1} = \gamma \sum_{s'} p(s'|s, \pi(s))(V^{t}(s') + r(s, \pi(s), s'))$

Amélioration de la politique

- Utilisation de l'opérateur d'optimalité de Bellman
- Les nouvelles valeurs de V^{π} guide vers une politique meilleure.

Amélioration de politique

Policy iteration

Alternativement :

- évaluer la politique π^t en calculant V^{π^t}
- calculer la nouvelle politique π^{+1}

Value Iteration

- Ne faire qu'un pas d'évaluation lors du calcul de V^{π^t}
- sortir uniquement à la fin la politique optimale

Plan

Algorithme de Monte-Carlo

Principe :

- estimation de $V_\pi(s)$ par moyenne empirique de la récompense sur un grand nombre d'échantillons
- Estimateur non biaisé si les échantillons sont indépendants.
- Model free : aucun besoin du modèle ou d'approximation.
- Apprend à partir d'épisodes joués selon la politique

$\Rightarrow$ limite : les épisodes doivent se terminer.

Moyenne incrémentale

Pour $x_1, \ldots, x_k$ échantillons :

- $\mu_k = \frac{1}{k}\sum_{j=1}^{k} x_j = \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$
- Dans le cas non stationnaire : $\mu_k = \mu_{k-1} + \alpha(x_k - \mu_{k-1})$

Algorithme de Monte-Carlo

Evaluation de politique : estimation de $V_\pi(s)$

- Répéter
  1. Générer un épisode $(s_1, a_1, r_1) \ldots (s_T, a_T, r_T)$ avec la politique π
  2. Fixer $R = 0$
  3. Pour $t = T - 1$ à $t = 1$:
     - $R = \gamma R + r_t$
     - $V(s_t) = V(s_t) + \alpha(R - V(s_t))$

Le même algorithme peut être utilisé pour $Q_\pi(s, a)$

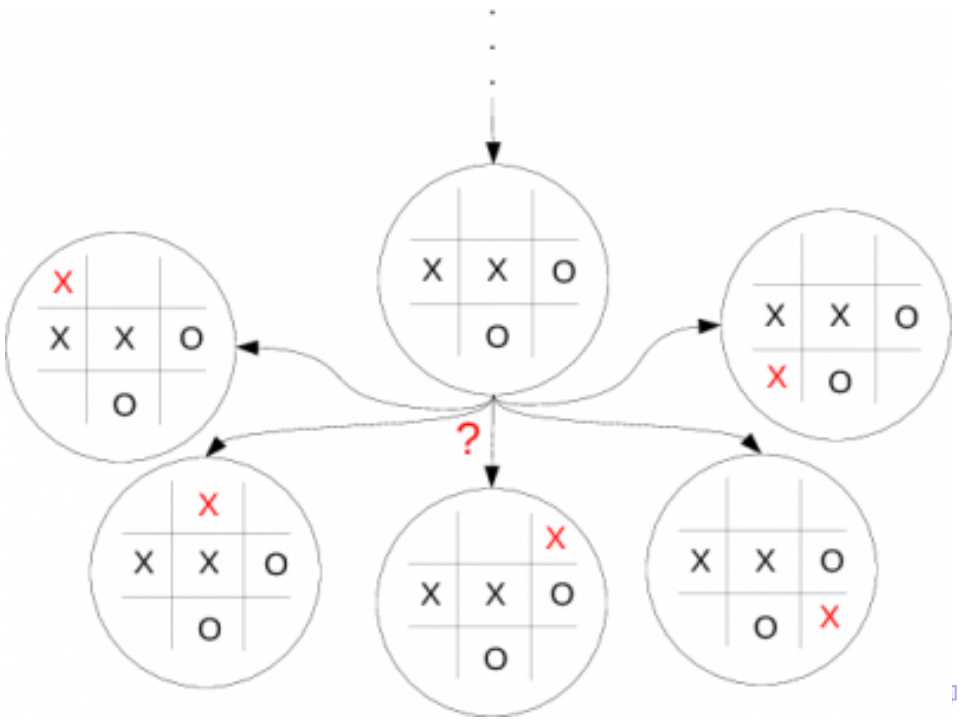
Différentes variantes

- First visit : un état n'est mis à jour qu'une seule fois par scénario, la première fois qu'il est rencontré.
- Every visit : tel que ci-dessus.
- Nécessite dans tous les cas la fin du scénario pour propager la récompense.

Dilemme Exploration/Exploitation

A chaque état :

- Décision de l'action sur l'*estimation* de la fonction de valeurs
- A-t-on confiance ou non en cette estimation ?
  - oui $\rightarrow$ exploitation
  - non $\rightarrow$ exploration des autres actions possibles
- L'exploration peut-être dangereuse
- A-t-on assez explorer ? (récompenses relatives . . .)



Dilemme Exploration/Exploitation

A chaque état :

- Décision de l'action sur l'*estimation* de la fonction de valeurs
- A-t-on confiance ou non en cette estimation ?
  - oui $\rightarrow$ exploitation
  - non $\rightarrow$ exploration des autres actions possibles
- L'exploration peut-être dangereuse
- A-t-on assez explorer ? (récompenses relatives . . .)

Algorithmes

- `greedy` (glouton),
- ϵ-`greedy` : glouton avec une probabilité de $1 - \epsilon$ au hasard sinon
- La famille UCB.

En résumé

Apprentissage par renforcement :

- Représenter le système comme un ensemble d'états, d'actions et de récompenses
- Apprendre à évaluer un état/une action en fonction des récompenses
- Trouver la meilleure séquence d'actions
- Explorer quand peu d'informations disponibles
- Exploiter pour rester dans des sénarios *viables*

On peut également chercher :

- à modéliser ou non l'environnement (model-free, model-based),
- apprendre online (en interagissant directement avec l'environnement)
- ou offline à partir de sénarios déjà joués.