

# Apprentissage par renforcement

Cours 8  
ARF Master DAC

Nicolas Baskiotis

`nicolas.baskiotis@lip6.fr`

`http://webia.lip6.fr/~baskiotis`

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

S2 (2017-2018)

# Plan

## 1 Introduction

## 2 Formalisation et outils

# Introduction

Qui a-t-il de commun entre :

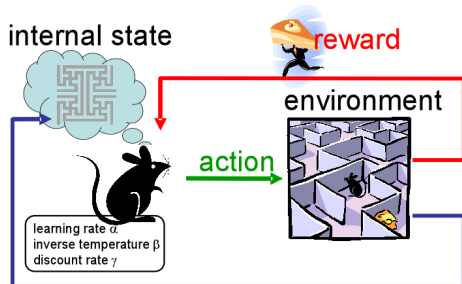
- Chien de Pavlov
- Boite de Skinner
- Un drone/véhicule autonome
- jeu d'échecs

# 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

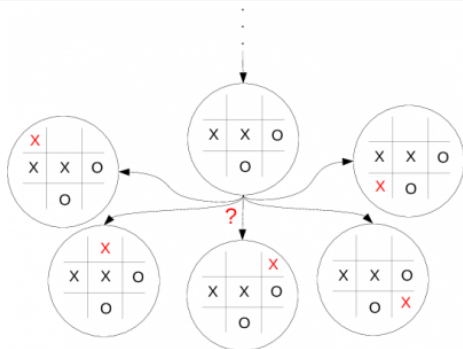


# Objectif : adaptation du système à son environnement

## 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 ?

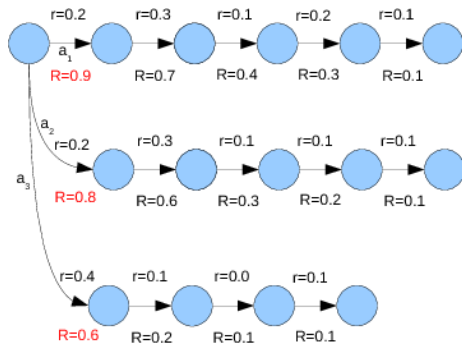


# Problématique

## Comment choisir une action ?

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

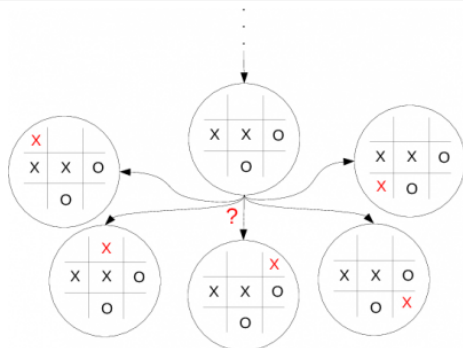
⇒ **fonction de valeur** d'états (ou d'état/action) : indication sur le long terme des récompenses attendues ( $\neq$  récompenses immédiates)



# 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 → exploitation
  - ▶ non → exploration des autres actions possibles
- L'exploration peut-être dangereuse
- A-t-on assez exploré ? (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 → exploitation
  - ▶ non → exploration des autres actions possibles
- L'exploration peut-être dangereuse
- A-t-on assez exploré ? (récompenses relatives ...)

## Algorithmes

- greedy (glouton),
- $\epsilon$ -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 scénarios *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 scénarios déjà joués.

# Plan

1 Introduction

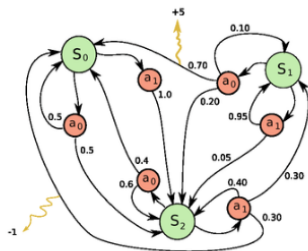
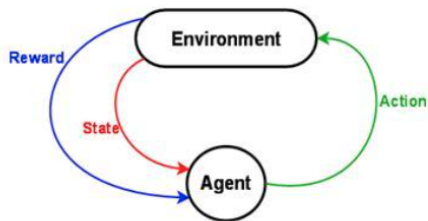
2 Formalisation et outils

# Markov Decision Process (MDP)

## Définition du modèle

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

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



# Formalisation

## 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) :  
 $[(s_1, a_1, r_1), (s_2, a_2, r_2), \dots, (s_n, a_n, r_n)], (s_i, a_i, r_i) \in \mathcal{S} \times \mathcal{A} \times \mathbb{R}$

## Fonctions valeur

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

# Equation de Bellman

## Quelques relations

- $\pi$  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  $\pi^*$
- $V^*(s) = \max_\pi V^\pi(s) = \max_a r(s, a) + \gamma \sum_{s'} p(s'|s, a) V^*(s')$
- $Q^*(s, a) = \max_\pi Q^\pi(s, a) = \mathbb{E}(r_{t+1} + \gamma V^*(s_{t+1}|s_t = s, a_t = a))$   
 $= r(s, a) + \gamma \max_a \sum_{s'} p(s'|s, a) Q^*(s', a')$

# 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  $\pi$  : model-free
    - ▶ on estime le MDP puis on en déduit  $\pi$  : 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 ?
- ⇒ approximation par apprentissage.

# Résolution exacte : Programmation dynamique

- Opérateur de Bellman :  $T^\pi$ , 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
- Value iteration : trouver  $V^*$  par itération sur  $T^*$  :  $V_{i+1} \leftarrow T^* V_i$
- Policy iteration :
  - ▶  $V_{i+1}^\pi \leftarrow T^\pi V_i^\pi$
  - ▶  $\forall s \pi'(s) \leftarrow \operatorname{argmax}_a \sum_{s'} p(s'|s, a) [r(s, a) + \gamma V^\pi(s')]$