Apprentissage par renforcement

Laëtitia Matignon

Laboratoire d'InfoRmatique en Image et Systèmes d'information
LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon
INSA de Lyon, bâtiment J. Verne
20, Avenue Albert Einstein - 69622 Villeurbanne cedex
http://liris.cnrs.fr

18 octobre 2013

Formes d'apprentissage

	Supervisé	Non supervisé	Par renforcement
On apprend	relations	structures	politique ou loi d'action
Information	sortie désirée	rien	récompense
Forme d'apprentissage	par instruction	par observation	par action/évaluation

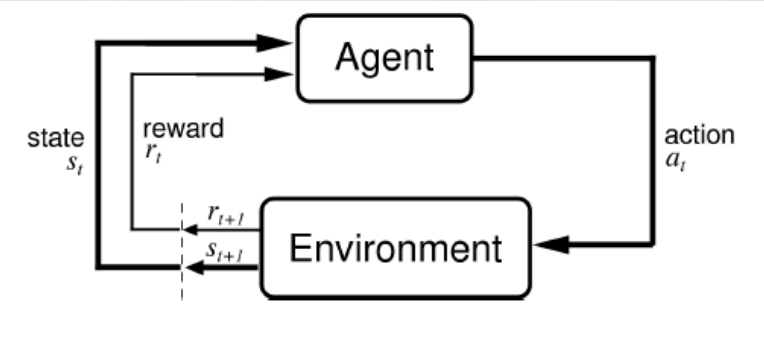
Apprentissage par renforcement

- Apprentissage basé sur l'interaction avec l'environnement
- Apprendre par l'expérience d'une stratégie comportementale (appelée **politique**) en fonction des échecs ou succès constatés (les **renforcements** ou **récompenses**).
- Apprentissage orienté-objectif
- Apprendre une politique - associer des actions à des situations pour maximiser la récompense.

Apprentissage par renforcement
Origines

- Psychologie expérimentale. Modèles de conditionnement animal (recherches initiées vers 1900 par Pavlov, Skinner et le courant béhavioriste) : manière dont une association entre des stimuli et des réflexes ou conséquences peut être renforcée, renforcement de comportement menant à une satisfaction
- Cadre mathématique adéquat : Programmation dynamique de Bellman (1950-60) en théorie du contrôle optimal, processus décisionnel markovien (MDP)

Le problème posé est celui d'un agent placé dans un **environnement stochastique** au sein duquel il doit atteindre un but.



MDP

- S ensemble fini d'états
- A ensemble fini d'actions
- $T : S \times A \times S \rightarrow [0; 1]$ fonction de transition
  $T(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$
- $R : S \times A \times S \rightarrow \Re$ fonction de renforcement

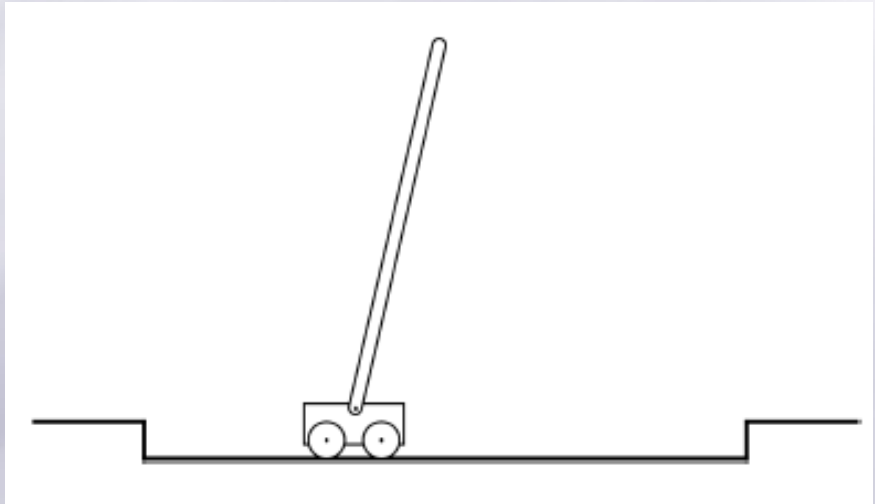
Résoudre un MDP consiste à calculer une politique.

Politique $\pi : S \rightarrow A$ ou $\pi : S \times A \rightarrow [0; 1]$

Choix de l'action a_t dans l'état s_t selon **politique** courante π_t.

Propriété de Markov

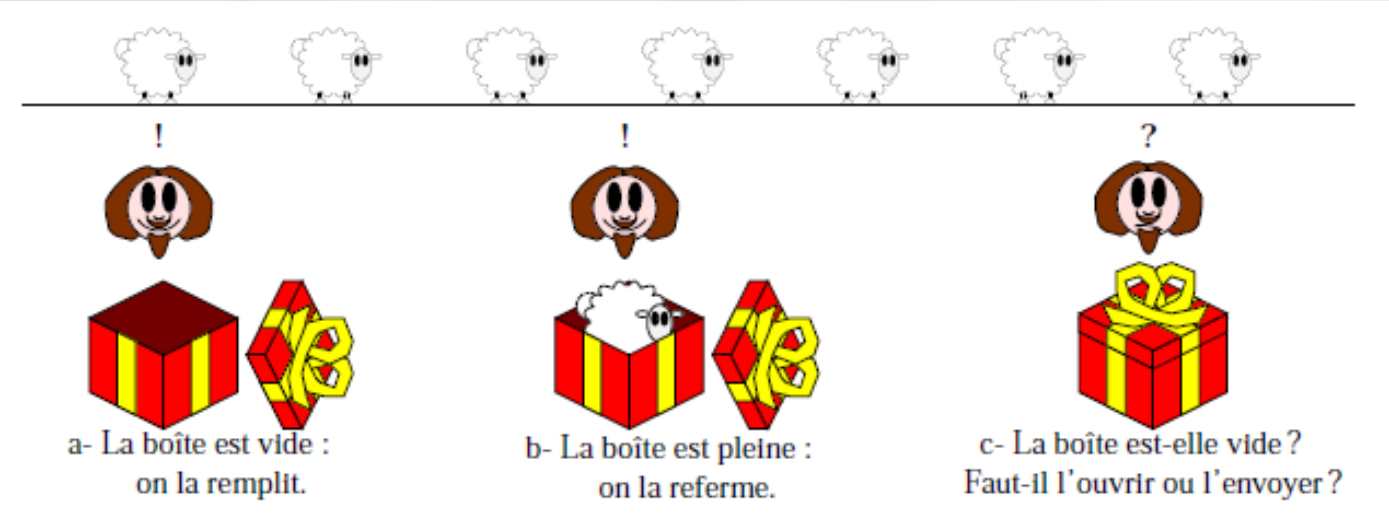
Les conséquences d'une action ne dépendent que de l'état actuel et non des états précédents : $P(s_{t+1}|h_t) = P(s_{t+1}|s_t, a_t)$



Info nécessaires pour que le problème soit markovien : angle du pendule, position du chariot, vitesse du pendule et du chariot, accélération

Propriété de Markov

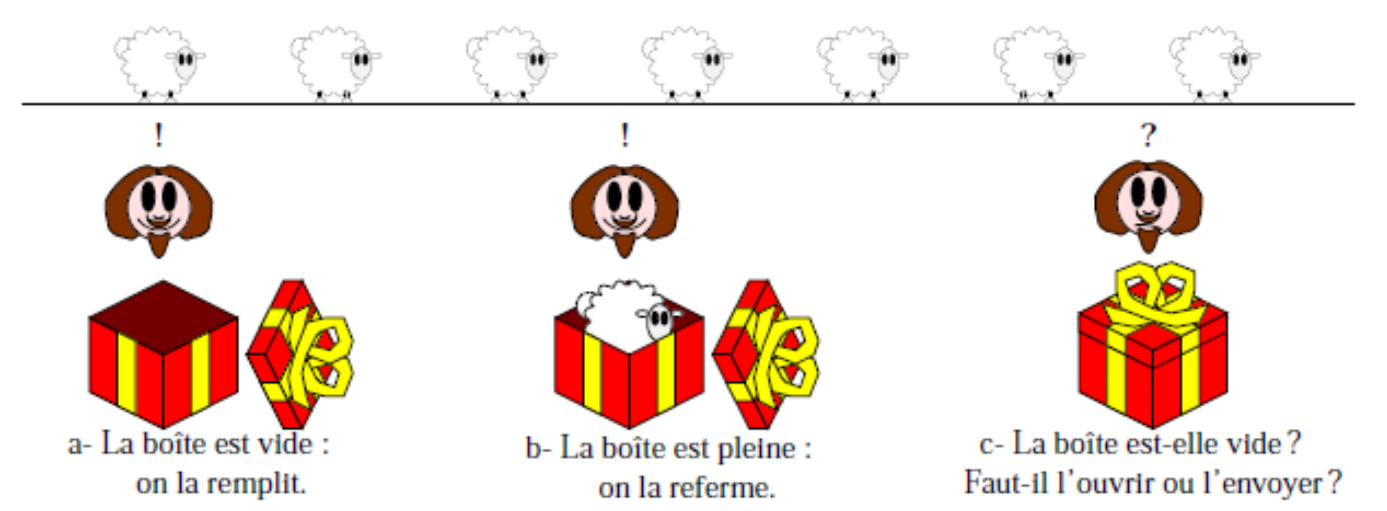
Les conséquences d'une action ne dépendent que de l'état actuel et non des états précédents : $P(s_{t+1}|h_t) = P(s_{t+1}|s_t, a_t)$



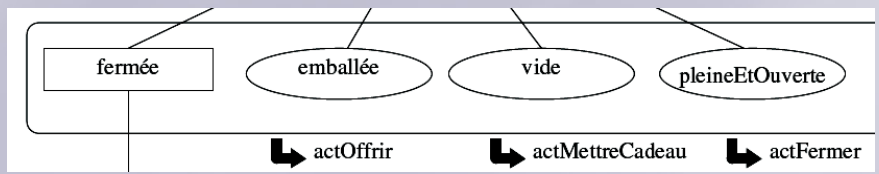
$S = \{fermee, videEtOuverte, pleineEtOuverte, emballee\}$
$A = \{ouvrirBoite, remplirBoite, fermerBoite, emballerBoite, envoyerBoite\}$

Propriété de Markov

Les conséquences d'une action ne dépendent que de l'état actuel et non des états précédents : $P(s_{t+1}|h_t) = P(s_{t+1}|s_t, a_t)$



$$S = \{fermee, videEtOuverte, pleineEtOuverte, emballee\}$$
$$A = \{ouvrirBoite, remplirBoite, fermerBoite, emballerBoite, envoyerBoite\}$$



Objectif de l'agent

Trouver une politique permettant de maximiser la récompense cumulée

- Compromis nécessaire entre récompenses immédiates et futures.

Calcul d'une récompense pondérée sur le long terme :

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

γ **facteur d'atténuation** permet de régler l'importance accordée aux récompenses futures vs. récompenses immédiates.

- $\gamma = 0$ agent cigale
- $\gamma = 1$ agent fourmi
- $\gamma \in [0, 1]$

Objectif de l'agent

Trouver une politique optimale permettant de maximiser la récompense cumulée

Fonction de valeur d'une politique

Espérance du critère lorsque part de l'état s et suit π
$V^{\pi}(s) = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\}$

Fonction action-valeur d'une politique

Espérance du critère lorsque l'on effectue l'action a dans l'état s puis on suit π $Q^{\pi}(s,a) = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\}$

Une politique optimale π^* vérifie $\forall \pi \forall s \quad V^{\pi^*}(s) \geq V^{\pi}(s)$
La (les) politique(s) optimale(s) ont pour valeur V^* qui est l'unique solution des équations de Bellman :

$$\forall (s,a) \in S \times A, Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s')$$

$$\forall s \in S, V^*(s) = max_{b \in A} Q^*(s,b)$$

Principe de Bellmann

Si l'on sait trouver la solution optimale (suite de décisions) à partir de l'étape $t+1$ quel que soit l'état s_{t+1}, alors on peut trouver la décision optimale à l'étape t pour tout état possible s_t (et donc travailler par récurrence).

Objectif de l'agent

Trouver une politique optimale permettant de maximiser la récompense cumulée

- Un MDP admet au moins une solution optimale qui se présente sous la forme d'une politique déterministe $\pi : S \rightarrow A$ où l'action à effectuer ne dépend que de l'état présent du processus [Puterman, 1994]
- Il existe un comportement optimal pour l'agent où ce dernier n'a besoin que de ses perceptions immédiates pour décider de la meilleure action à effectuer
- Quand on connaît Q^* on peut en déduire π^* à partir des actions gloutonnes :

$$\forall s \in S, \pi^*(s) = argmax_{a \in A} Q^*(s, a)$$

Value iteration [Bellman,1957]

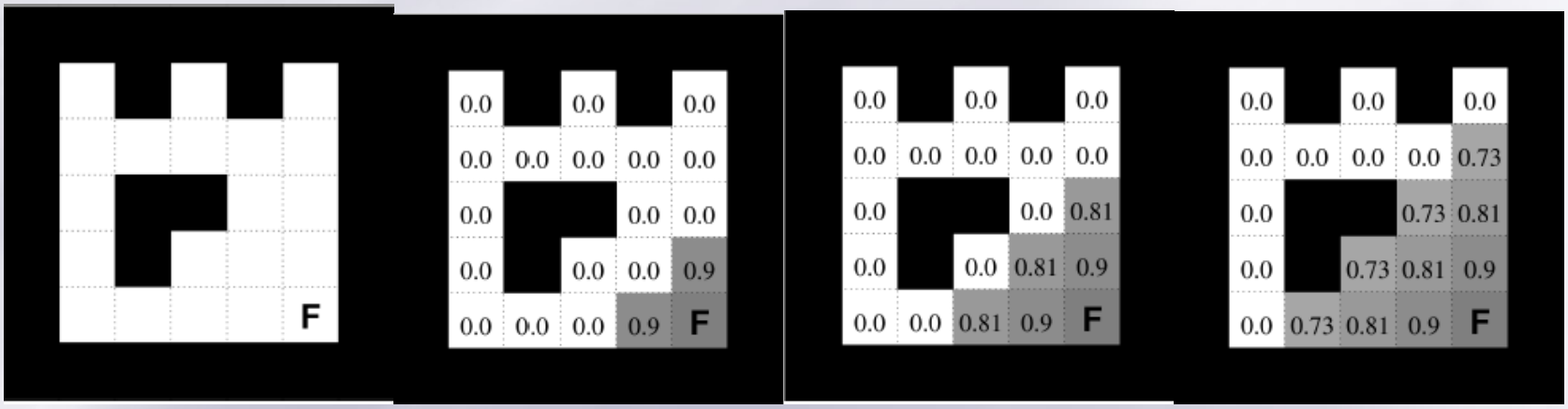
- modèle (T et R) connu
- évaluation itérative de Q^*, on en déduit π^*

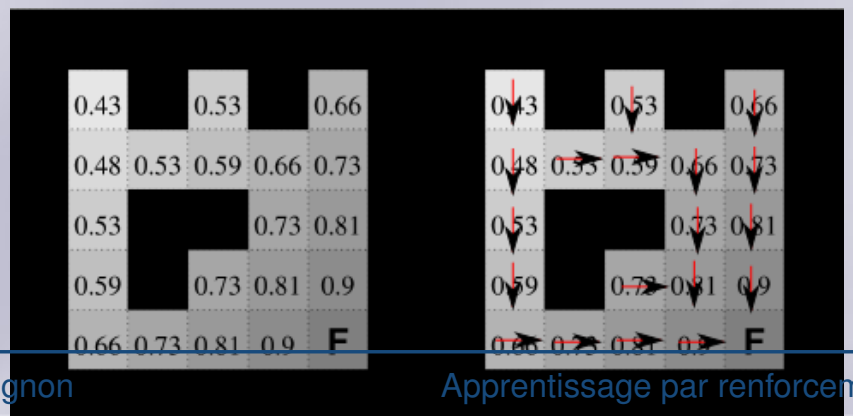
$$\forall (s,a), Q_{t+1}(s,a) \leftarrow R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') max_{b \in A} Q_t(s',b)$$

- théorème du point fixe de Banach : convergence de l'algorithme vers l'unique fonction solution de l'équation de Bellman
- critère d'arrêt $max_{s \in S} |V_t(s) - V_{t+1}(s)| < \epsilon$

Value iteration [Bellman,1957]

- $\forall (s,a), Q_{t+1}(s,a) \leftarrow R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') max_{b \in A} Q_t, (s',b)$
- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$





Que connaît l'agent ?

L'agent connaît toujours les états et actions du système.

- **avec modèle** : L'agent connaît T et R et utilise des algorithmes de planification
- **sans modèle** : L'agent estime T et R et utilise des algorithmes de planification : AR indirect
  - Dyna-Q, Priority Sweeping
- **sans modèle** : L'agent apprend directement un comportement optimal sans apprendre T et R : AR direct

Apprentissage par essais-erreurs/par interaction : à chaque interaction avec l'environnement, on a des tuples (s_t, a_t, s_{t+1}, r_t) utilisés pour la mise à jour de l'estimation courante de la fonction de valeur.

Q-learning [Watkins,1989]

- Adaptation de *value iteration* en remplacant mise à jour de T et R par une différence temporelle

$$Q_{t+1}(s,a) = Q_t(s,a) + \delta$$

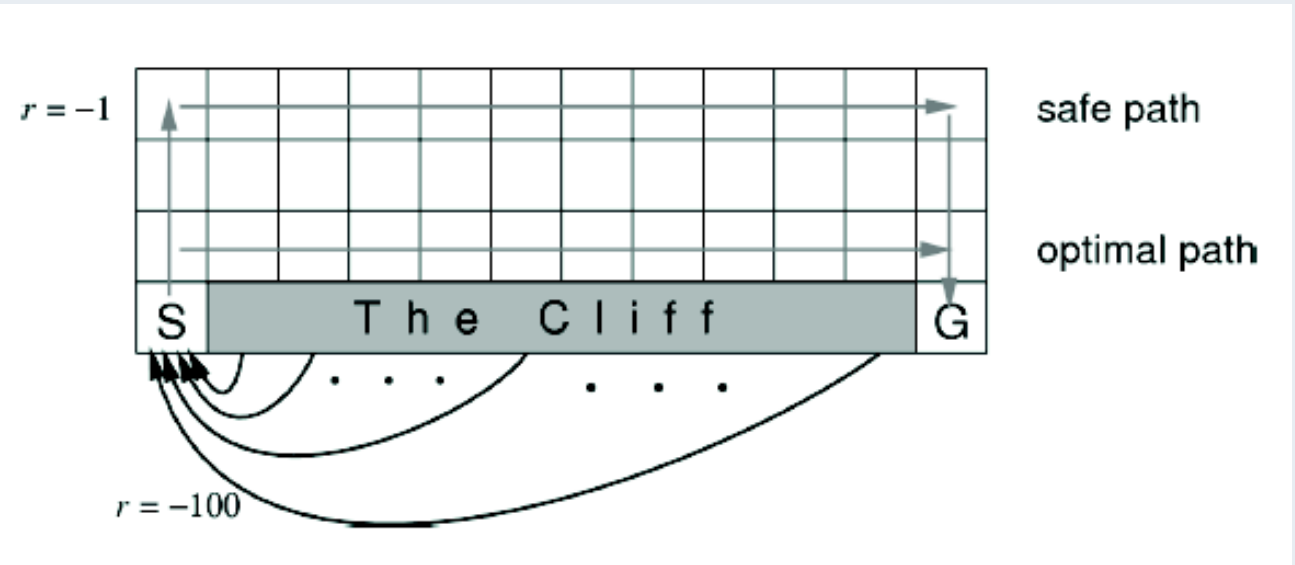
avec δ = amélioration - ancienne estimation

$$\delta = [r_t + \gamma \sum_{s' \in S} T(s,a,s') max_{b \in A} Q_t(s',b)] - Q_t(s,a)$$

- $Q_{t+1}(s,a) = Q_t(s,a) + \alpha[r_t + \gamma max_{b \in A} Q_t(s',b) - Q_t(s,a)]$ avec $\alpha \in]0;1]$ taux d'apprentissage
- Q-learning maintient une table des Q-valeurs
- Dans s_t, choix de l'action a_t selon méthode ϵ-greedy, softmax, ...

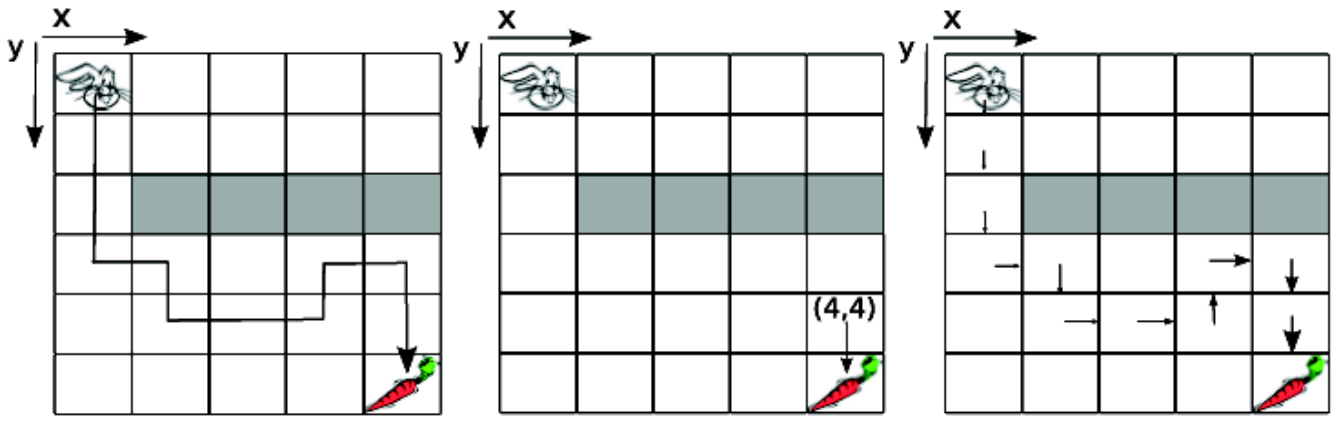
Autres algorithmes

- apprentissage en-ligne : utilise les actions choisies pour estimer la valeur du prochain état (Sarsa).



Autres algorithmes

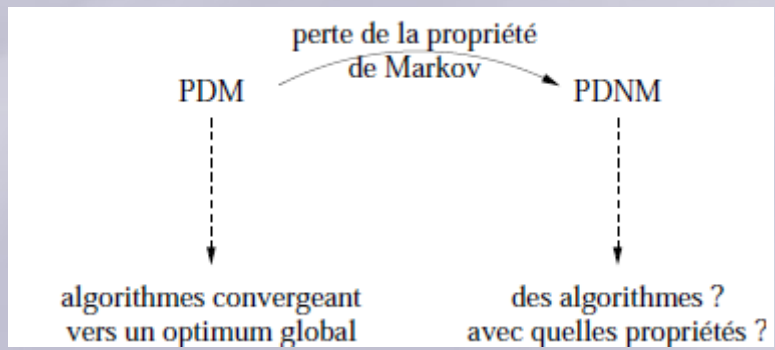
- traces d'éligibilité : mise à jour des Q-valeurs sur tout le chemin suivi (algorithmes $TD(\lambda)$)



(a) chemin parcouru (b) couple état-action mis à jour avec le Q-learning (c) couples état-action mis à jour avec la trace d'éligibilité

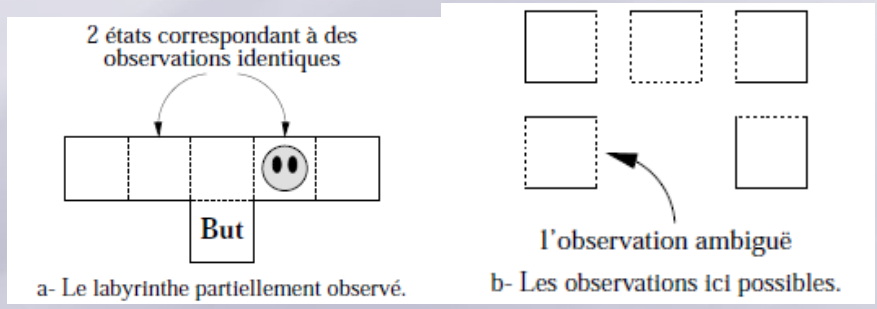
Apprentissage par renforcement
Quelques limitations des MDP

- algorithmes coûteux : représentations factorisées cherchent à diminuer le nombre d'états et actions, approximation de la fonction de valeur
- finitude : trouver une discrétisation adaptée et pertinente de l'espace d'états ou approximation de la fonction de valeur
- actions de mêmes durées : processus décisionnel Semi-Markovien
- propriété de Markov : perceptions partielles, SMA



POMDP

- $< S, A, T, R >$ MDP “sous-jacent”
- Ω ensemble d’observations
- $O : S \times \Omega \rightarrow [0; 1]$ fonction d’observation qui donne $P(o_{t+1} = o' | s_{t+1} = s')$



Les algorithmes valables dans les MDP peuvent être inadéquats pour apprendre un comportement basé sur les observations immédiates du POMDP.

Avec estimation d'état

Essayer de se ramener à un MDP sous-jacent.

- avec modèle : calcul des états de croyance (*belief state*) qui donne une probabilité de distribution sur l'ensemble des états possibles pour travailler sur un nouveau MDP défini sur les états de croyance.

$$b(s_{t+1}) = P(s_{t+1}|b_t, a_t, o_{t+1})$$

  Problème : besoin de connaitre le modèle, et les états de croyance forment un ensemble continu.

- sans modèle : construire un observable exhaustif permettant de construire un processus markovien : AR basé sur la mémoire

Sans estimation d'état

Chercher une politique stochastique qui peut n'être qu'un optimum local : recherche directe de politique (Montée de gradient de Baxter ...)

AR basé sur la mémoire [Dutech,2003]

- trouver la politique adaptée optimale défini sur un observable exhaustif composé d'éléments non-ambigüs
- détecter les éléments ambigüs : la convergence des Q-valeurs pour les éléments ambigüs est plus lente ... mais les variations des Q-valeurs sont influencées par beaucoup de facteurs
- construction incrémentale de l'observable : remplacer les éléments ambigüs par une trace d'action-observation
- critère d'arret : chaque nouvel observable est évalué en estimant la valeur moyenne de la meilleure politique construite sur cet observable

Apprendre une représentation pertinente du monde ?

- **Qualité de la représentation** fondamentale - si trop pauvre : incapable d'apprendre une politique optimale ; si trop riche : incapable de gérer et de manipuler cette représentation
- L'agent doit se construire lui-même une **représentation de qualité** de son environnement.
- Quels sont les éléments pertinents à prendre en compte ? Comment faire émerger les représentations pertinentes ?

Wiki de l'équipe (connexion avec login/mdp de l'intranet du LIRIS) :

`http://liris.cnrs.fr/sycosma/wiki/doku.php?id=reunions_d_equipe_sycosma`