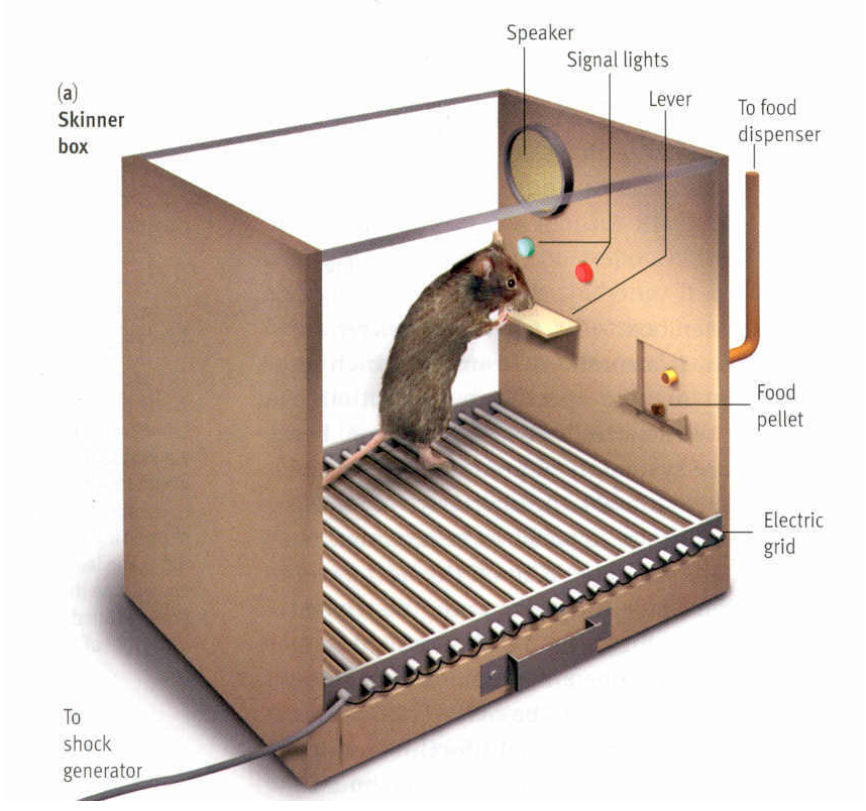
Apprentissage par Renforcement (Efficace)

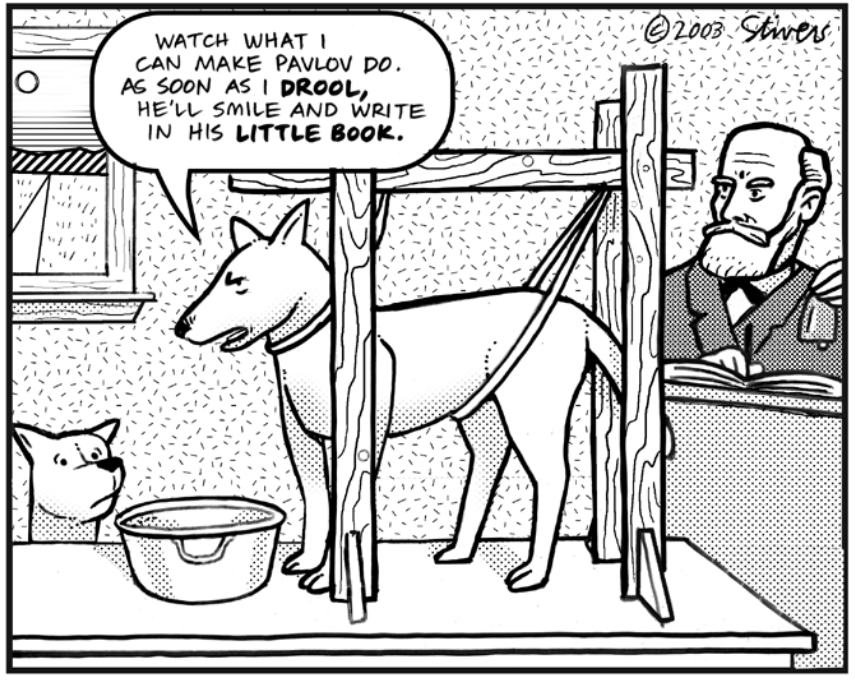
Alain Dutech

Equipe MAIA - LORIA
Nancy, France
Web : http://maia.loria.fr
Mail : Alain.Dutech@loria.fr

Séminaire MIS - 11 juillet 2008

Inspiration : le conditionnement



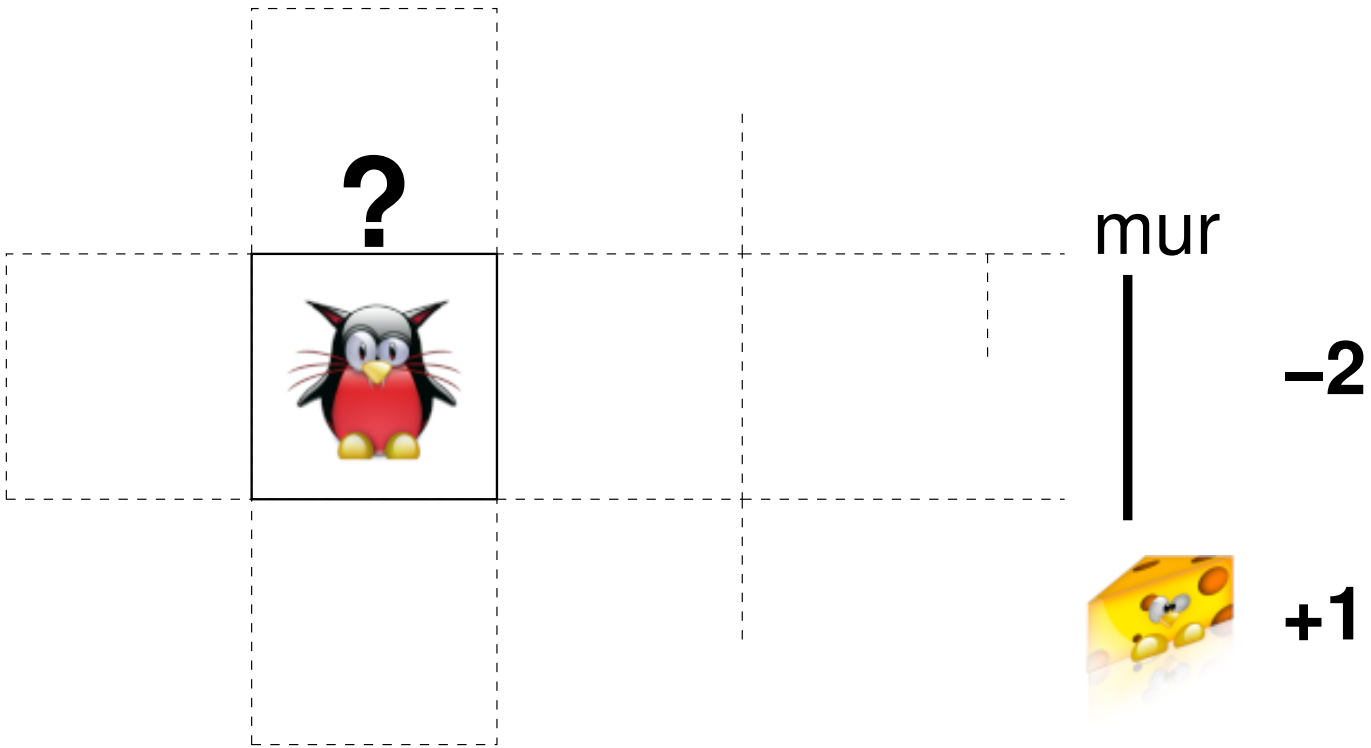


Rat dans labyrinthe

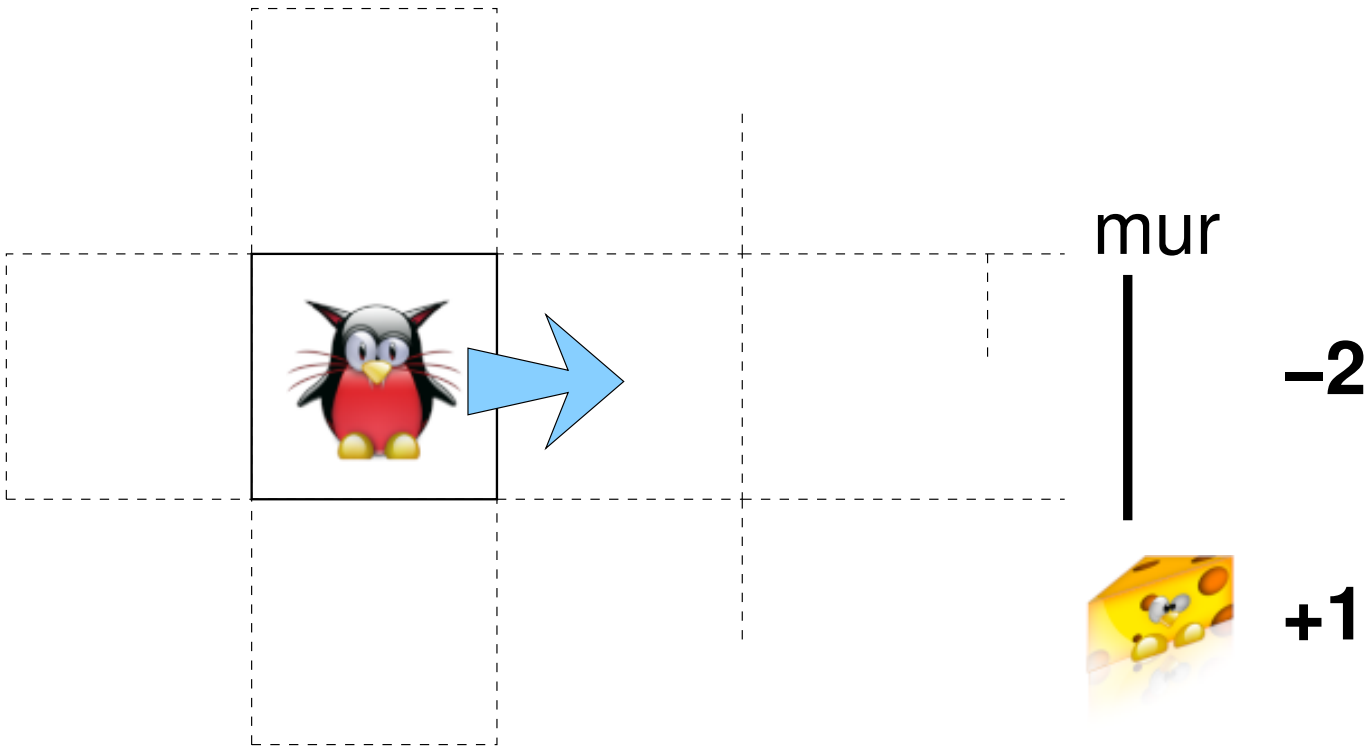
?

Rat dans labyrinthe

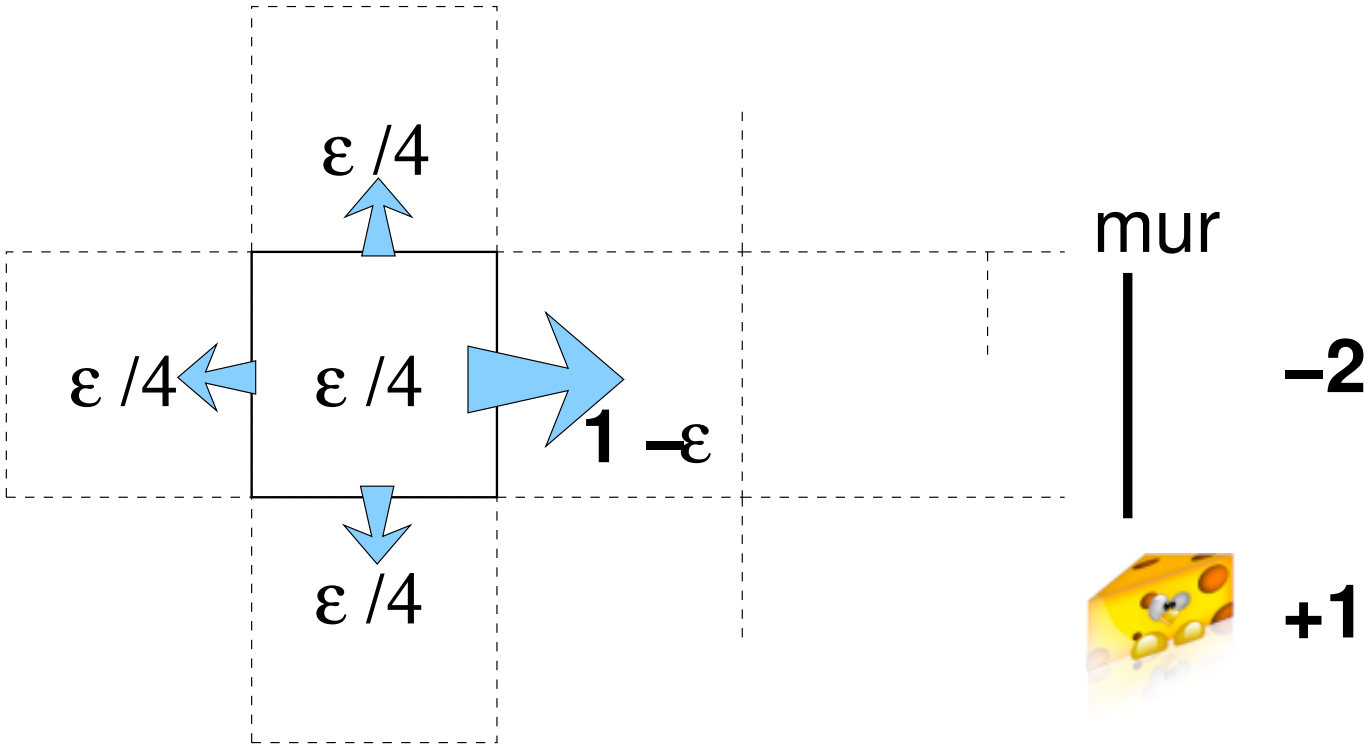
Rat dans labyrinthe



Rat dans labyrinthe



Rat dans labyrinthe



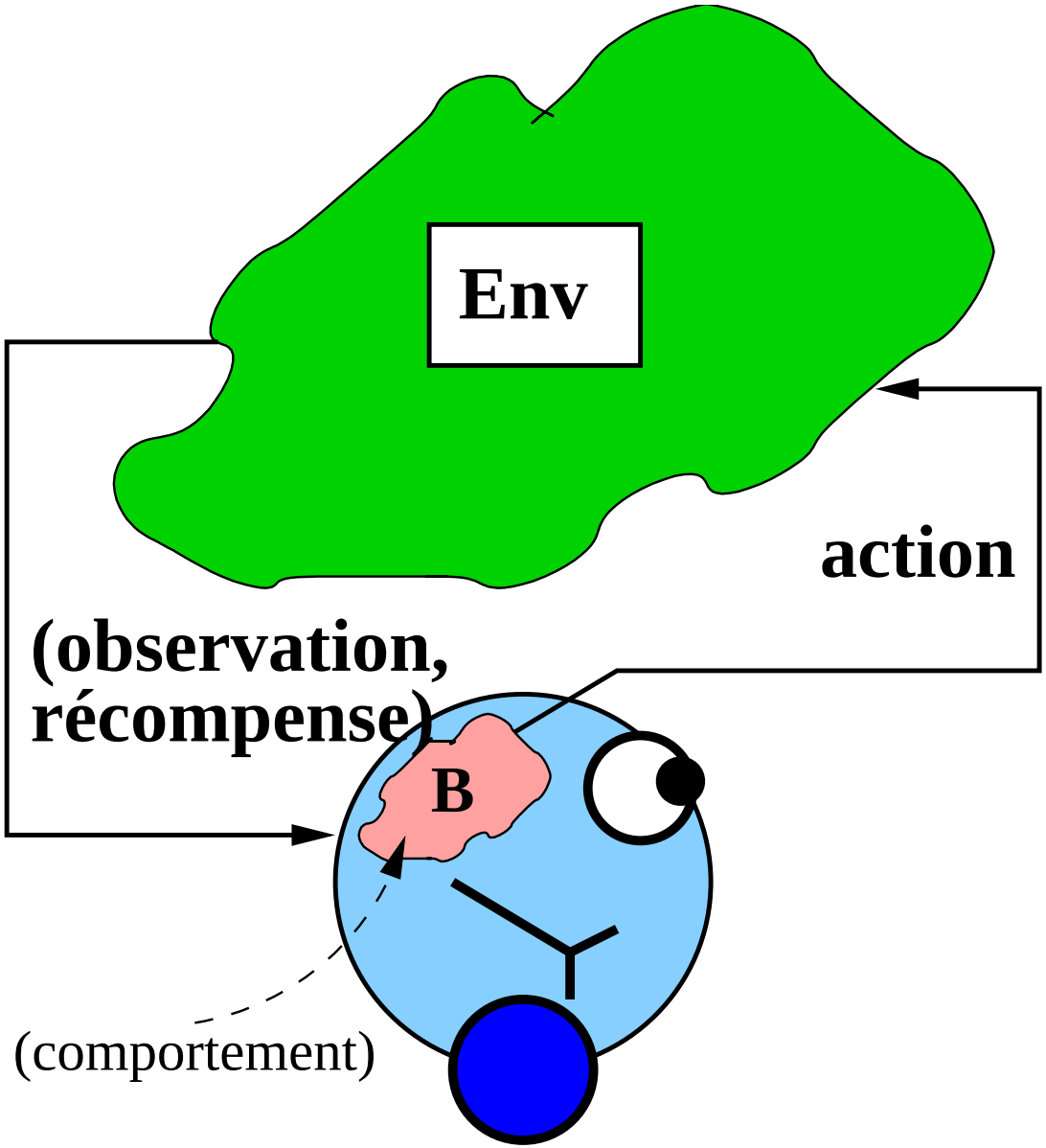
Propriétés intéressantes de l'apprentissage par renforcement

- **Récompense Scalaire**. Moins exigeant que apprentissage supervisé et plus dirigé que non-supervisé.
- **Récompenses Retardées**. Permet de tenir compte des conséquences à long terme des actions.
- **Environnement Incertain**. Plusieurs conséquences possibles pour une action.
- **Solution déterministe**. Malgré l'incertain, une solution déterministe existe.
- **Sans modèle**. On peut apprendre sans connaître la dynamique de l'environnement.

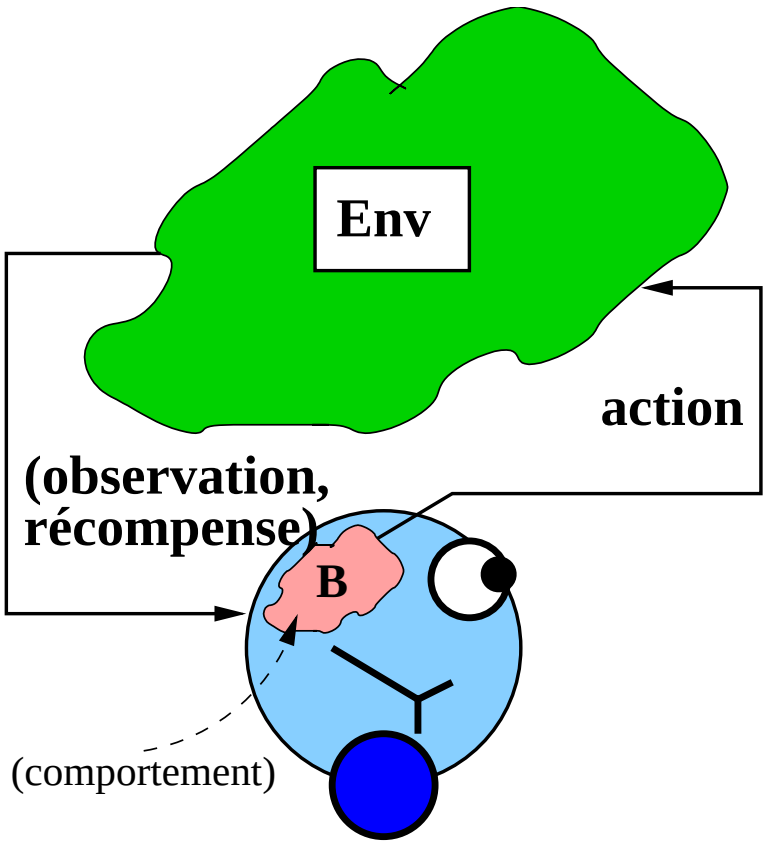
Plan de l'exposé

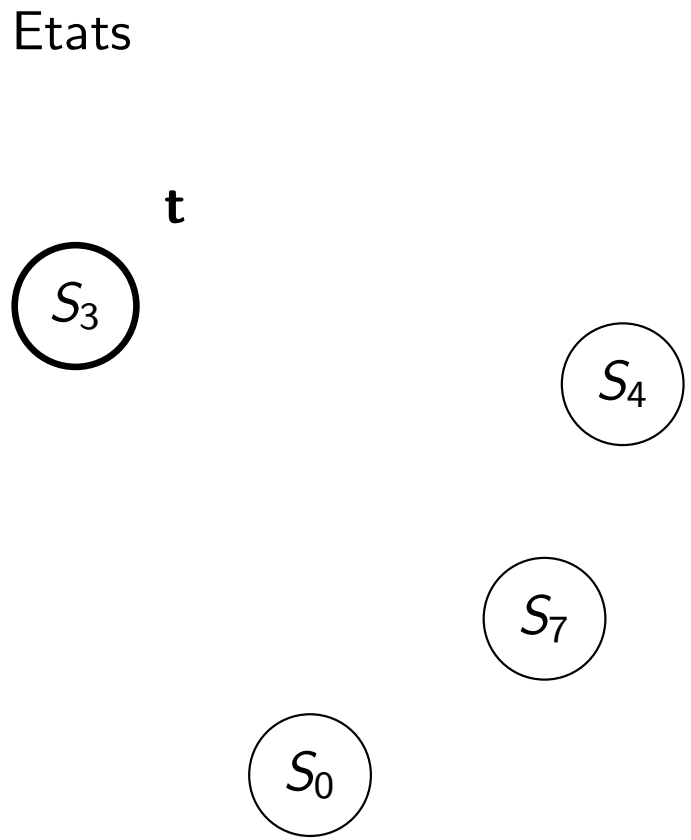
- Introduction
- Formalisme de l'Apprentissage par Renforcement
- Algorithmes pour l'Apprentissage par Renforcement
- Apprendre Efficacement
- Traces d'éligibilité
- Discussion

Cadre formel de l'Apprentissage par Renforcement

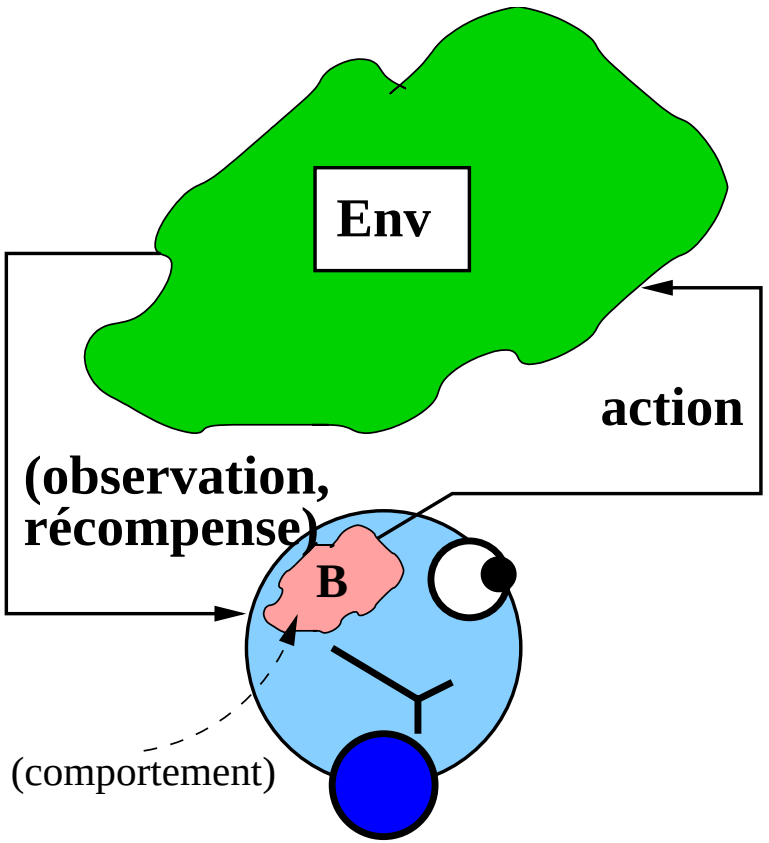


Cadre formel de l'Apprentissage par Renforcement

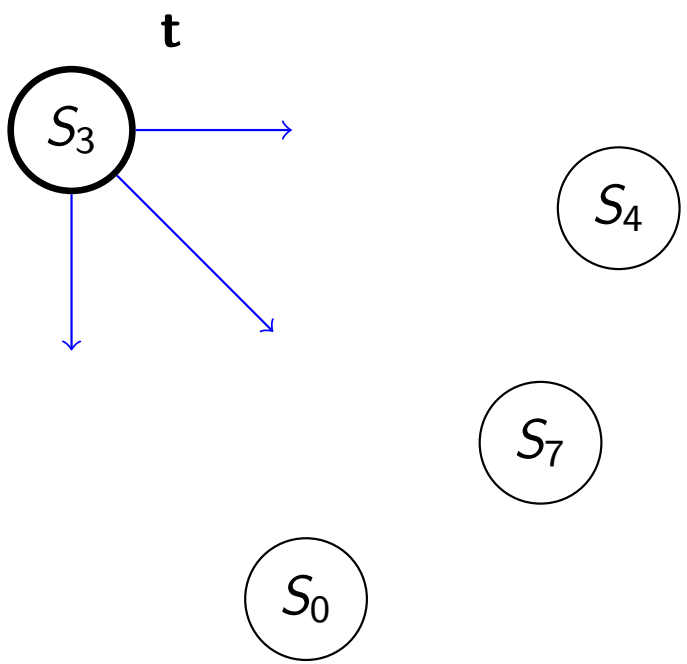




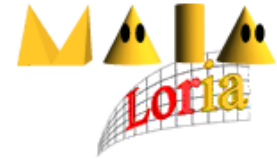
Cadre formel de l'Apprentissage par Renforcement



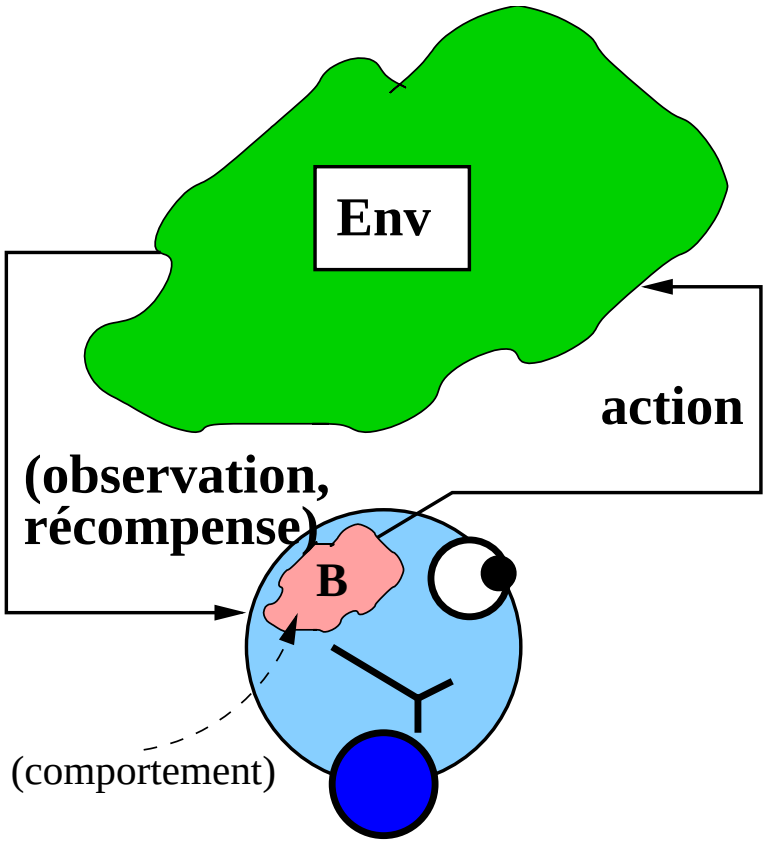
Etats , Actions

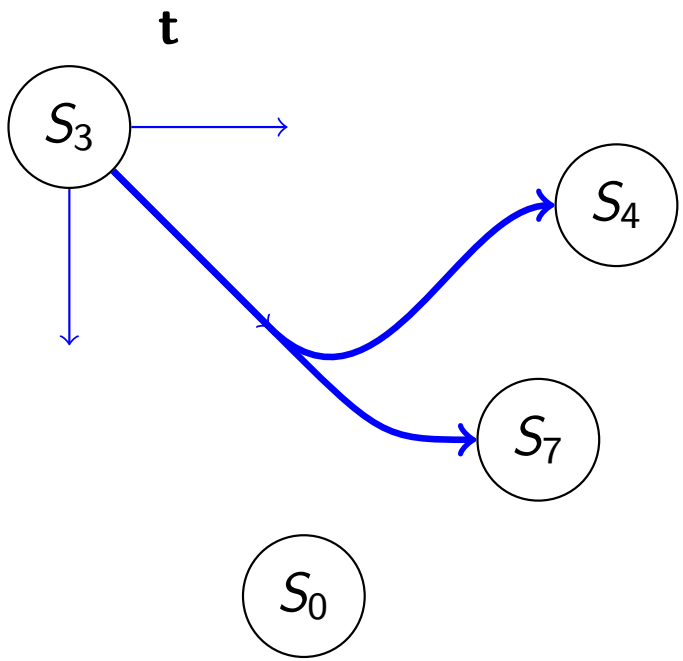


Cadre formel de l'Apprentissage par Renforcement



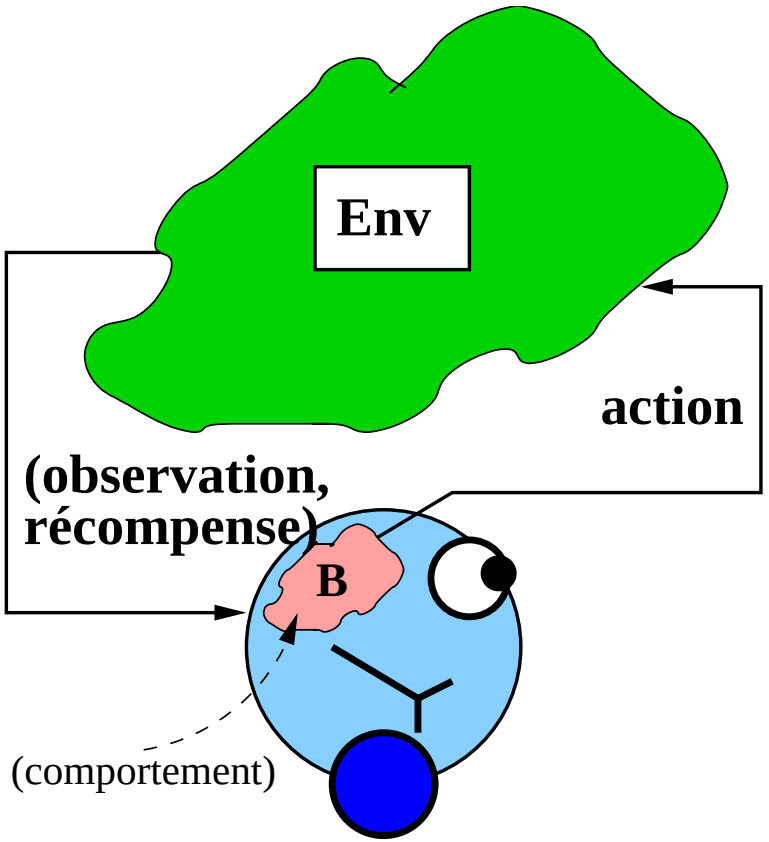
Etats , Actions , Transitions

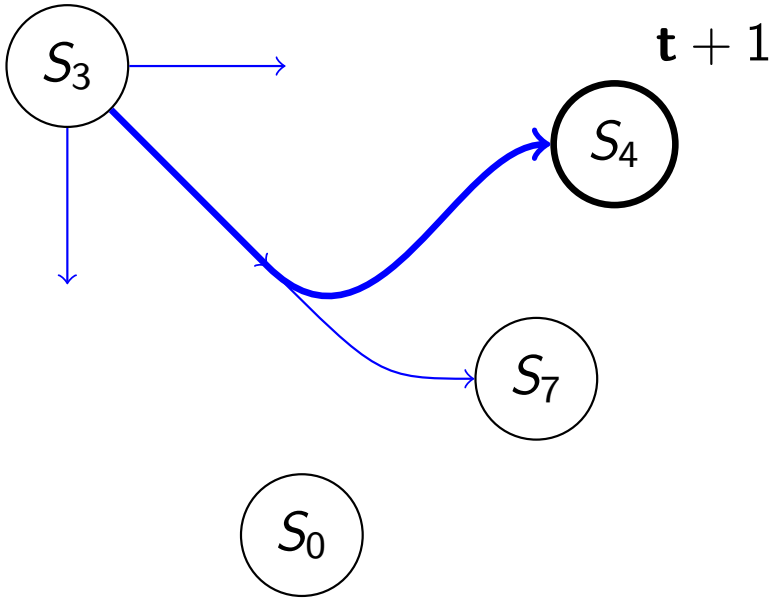




Cadre formel de l'Apprentissage par Renforcement

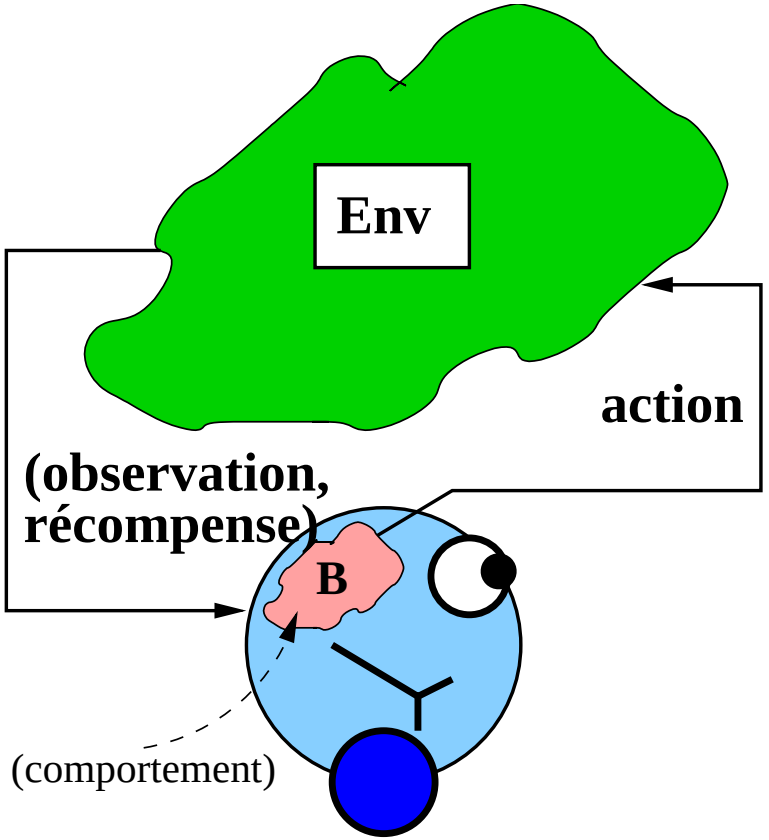
Etats , Actions , Transitions

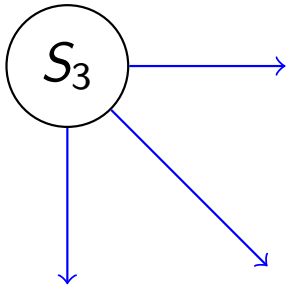


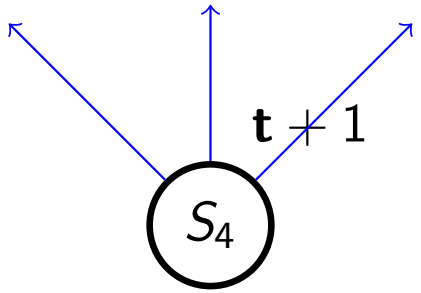


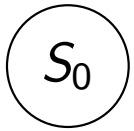
Cadre formel de l'Apprentissage par Renforcement

Etats , Actions , Transitions



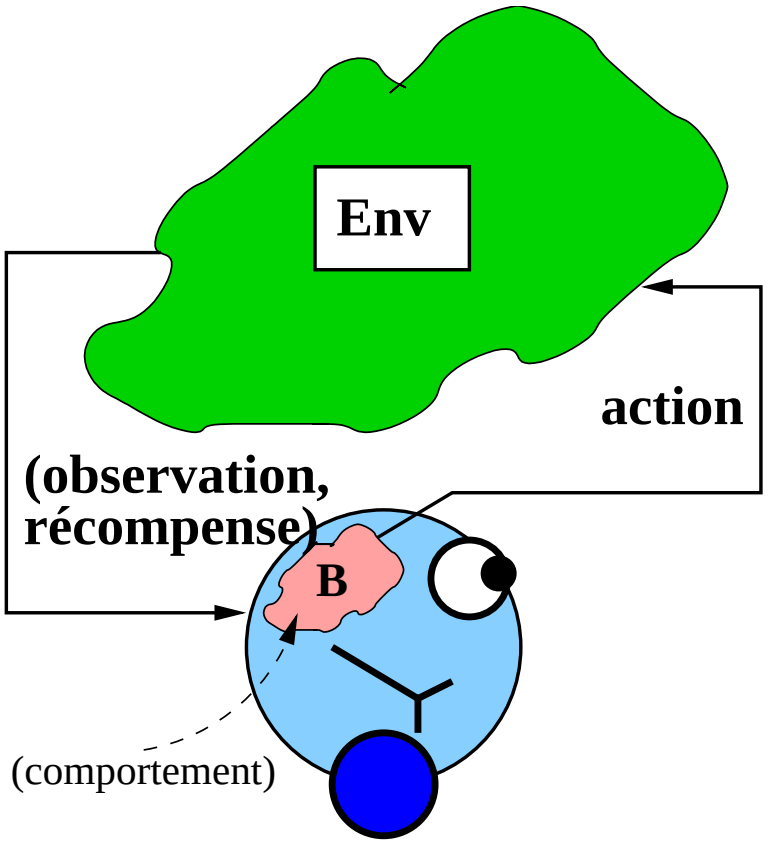


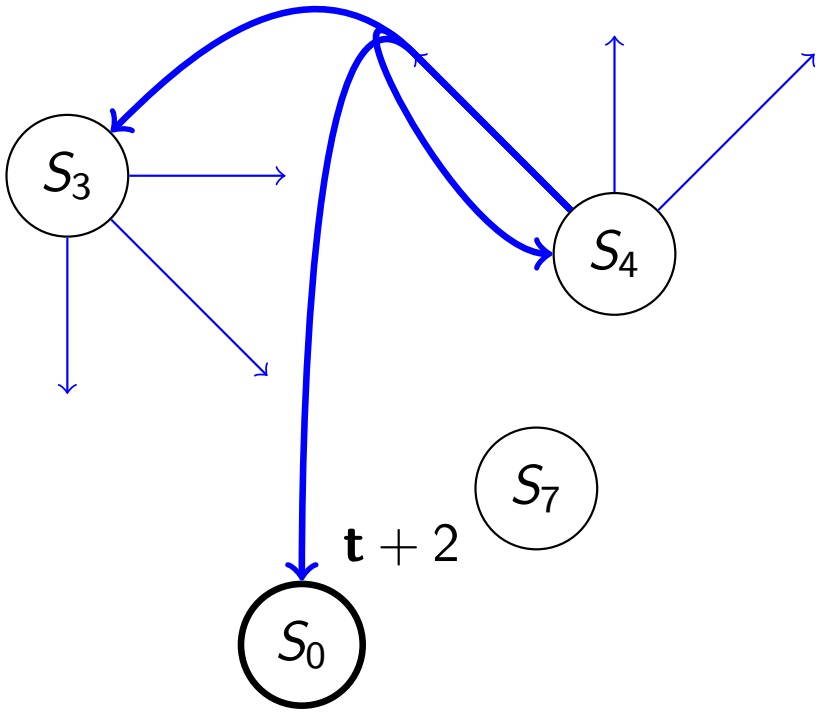




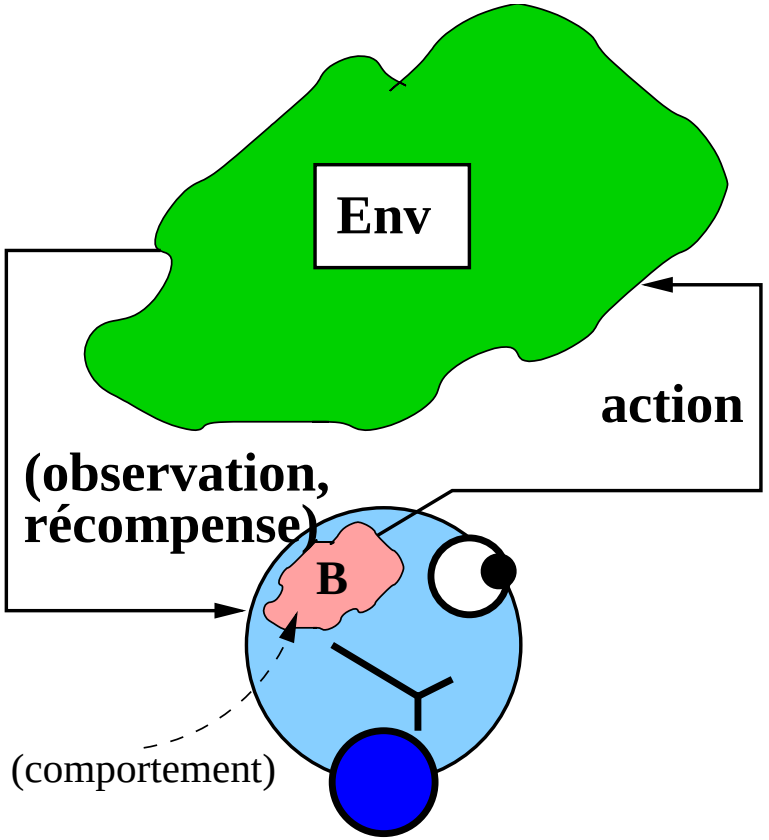
Cadre formel de l'Apprentissage par Renforcement

Etats , Actions , Transitions

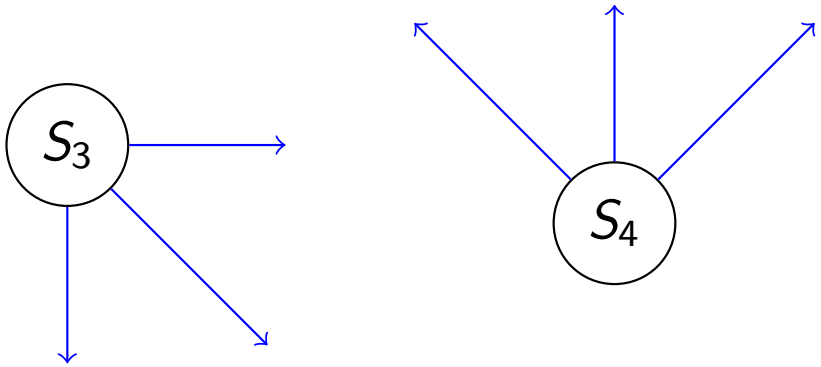


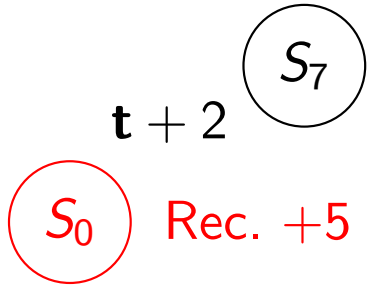


Cadre formel de l'Apprentissage par Renforcement

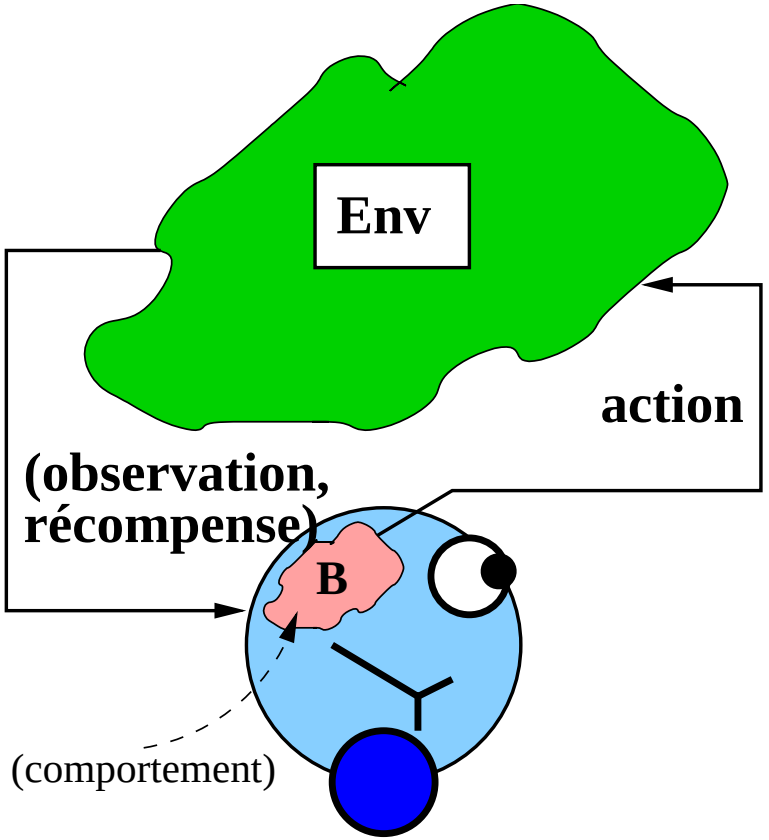


Etats , Actions , Transitions , Récompenses

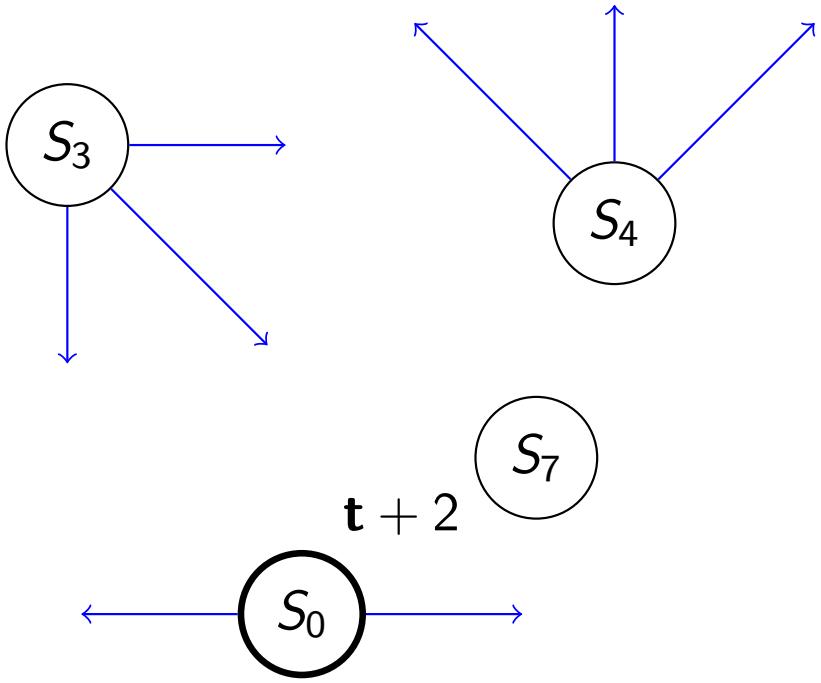




Cadre formel de l'Apprentissage par Renforcement



Etats , Actions , Transitions , Récompenses

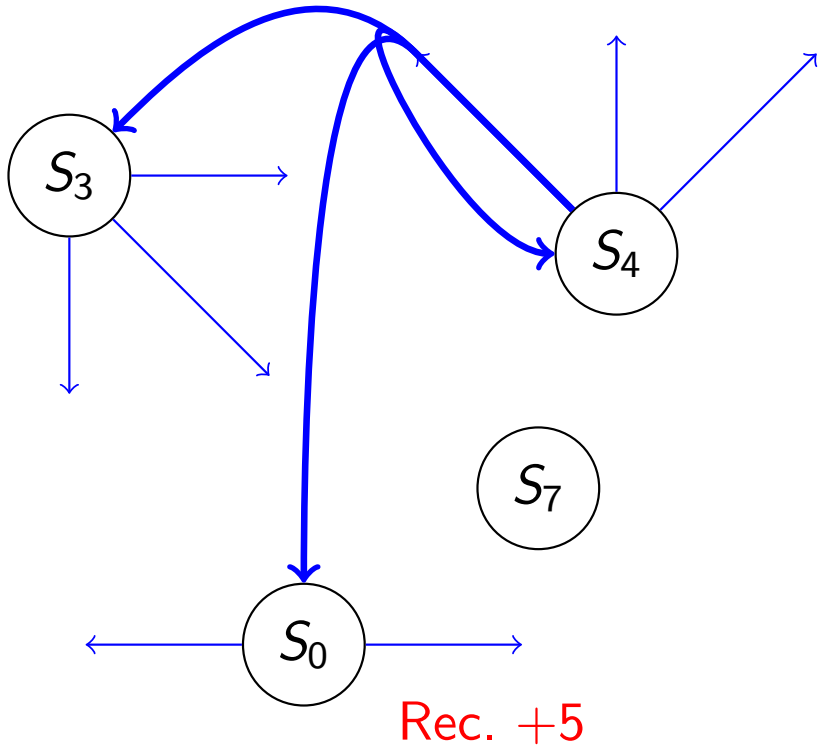


Notations

- $\mathcal{S}$, ens. d'états.
- $\mathcal{A}$, ens. d'actions.
- $p(s'|a,s)$, Transitions.
- $r(s,a)$, récompenses

Critère à maximiser

$$\sum_{t=0}^{\infty} \gamma^t r_t$$



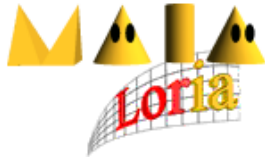
Les hypothèses idéales mais restrictives

- **Environnement Stationnaire**. Les probabilités de transitions restent constantes au cours du temps.
- **Etats complets**. L'état doit contenir assez d'information pour prédire le futur - Propriété de Markov.
- **(état,action) en nombre fini et "raisonnable"**. Extension au continu, mais pas en ce qui concerne l'apprentissage.
- **Convergence très très lente**.

Plan de l'exposé

- Introduction
- Formalisme de l'Apprentissage par Renforcement
- Algorithmes pour l'Apprentissage par Renforcement
- Apprendre Efficacement
- Traces d'éligibilité
- Discussion

Fonction valeur et politique



Critère γ-pondéré : $\sum_{t=0}^{\infty} \gamma^t r_t$

Politique : $\pi : \mathcal{S} \longrightarrow \Delta(\mathcal{A})$

Fonction valeur et politique

Critère γ-pondéré : $\sum_{t=0}^{\infty} \gamma^t r_t$

Politique : $\pi : \mathcal{S} \longrightarrow \Delta(\mathcal{A})$

Fonction valeur

$$V(s) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^t r_t + \cdots$$

Fonction valeur et politique

Critère γ-pondéré : $\sum_{t=0}^{\infty} \gamma^t r_t$

Politique : $\pi : \mathcal{S} \longrightarrow \Delta(\mathcal{A})$

Fonction valeur

$$V(s) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^t r_t + \cdots$$

En déduire une (meilleure) politique

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|a,s) V(s') \right\}$$

Opérateur de Bellman

$$V(s) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^t r_t + \cdots$$

Pour une politique π donnée :

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

Opérateur de Bellman

$$V(s) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^t r_t + \cdots$$

Pour une politique π donnée :

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

Calculer directement la fonction valeur **optimale** :

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

Algorithme Value Iteration

1. $\forall s \in \mathcal{S}$, Initialiser $V_0(s)$
2. Répéter

$$V_{i+1}(s) \longleftarrow \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|a,s) V_i(s') \right\}$$

3. Jusqu'à $||V_{i+1} - V_i|| < \epsilon$

Et ensuite, on en déduit la politique optimale.

Algorithme Value Iteration

1. $\forall s \in \mathcal{S}$, Initialiser $V_0(s)$
2. Répéter

$$V_{i+1}(s) \longleftarrow \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|a,s) V_i(s') \right\}$$

3. Jusqu'à $||V_{i+1} - V_i|| < \epsilon$

Et ensuite, on en déduit la politique optimale.

En déduire une (meilleure) politique

$$\pi(s) = \operatorname{argmax}_{a \in \mathcal{A}} \left\{ \mathbf{r(s,a)} + \gamma \sum_{s' \in \mathcal{S}} \mathbf{p(s'|a,s)} V(s') \right\}$$

Algorithme Value Iteration

1. $\forall s \in \mathcal{S}$, Initialiser $V_0(s)$
2. Répéter

$$V_{i+1}(s) \longleftarrow \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|a,s) V_i(s') \right\}$$

3. Jusqu'à $||V_{i+1} - V_i|| < \epsilon$

Et ensuite, on en déduit la politique optimale.

Caractéristiques

- Doit connaître la dynamique du MDP.
- Simple à mettre en oeuvre
- Complexité en $O(|\mathcal{S}|^2|\mathcal{A}|)$

Estimation par Monte Carlo

A partir d'une politique π_i :

1. Pour chaque état s de $\mathcal{S}$
   - 1.1 Générer m séquences de récompense de longueur T $(r_0^0, r_1^0, \cdots, r_T^0)\,;\,\cdots\,;\,(r_0^m, r_1^m, \cdots, r_T^m)$
   - 1.2 V est la moyenne de ces séquences

$$V(s) \quad = \quad \frac{1}{m} \sum_{k=0}^{m} \sum_{t=0}^{T} \gamma^t r_t^k$$

2. En déduire une nouvelle politique π_{i+1}

Estimation par Monte Carlo

A partir d'une politique π_i :

1. Pour chaque état s de $\mathcal{S}$
   - 1.1 Générer m séquences de récompense de longueur T $(r_0^0, r_1^0, \cdots, r_T^0)\,;\,\cdots\,;\,(r_0^m, r_1^m, \cdots, r_T^m)$
   - 1.2 V est la moyenne de ces séquences

$$V(s) \quad = \quad \frac{1}{m} \sum_{k=0}^{m} \sum_{t=0}^{T} \gamma^t r_t^k$$

2. En déduire une nouvelle politique π_{i+1}

Caractéristiques

- La connaissance de la dynamique n'est pas nécessaire
- Peut se concentrer sur un sous-ensemble des états
- Implémentation incrémentale

Différences temporelles

En suivant une politique π, après chaque transition (s_t, r_{t+1}, s_{t+1})

$$V(s_t) \longleftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

Permet d'estimer la fonction valeur de π.

Différences temporelles

En suivant une politique π, après chaque transition (s_t, r_{t+1}, s_{t+1})

$$V(s_t) \longleftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

Permet d'estimer la fonction valeur de π.

Caractéristiques

- En ligne, avec une politique fixée.
- La connaissance de la dynamique n'est pas nécessaire
- Convergence
- MAIS ne peut pas calculer une (meilleure) politique.

SARSA et Q-Learning

Fonction valeur des couples **état-action**

SARSA, transition $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha[\mathbf{r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})} - Q(s_t, a_t)]$$

Q-Learning, transition $(s_t, a_t, r_{t+1}, s_{t+1})$

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha[\mathbf{r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a)} - Q(s_t, a_t)]$$

SARSA et Q-Learning

Fonction valeur des couples **état-action**

SARSA, transition $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha[\mathbf{r_{t+1}} + \gamma \mathbf{Q(s_{t+1}, a_{t+1})} - Q(s_t, a_t)]$$

Q-Learning, transition $(s_t, a_t, r_{t+1}, s_{t+1})$

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha[\mathbf{r_{t+1}} + \gamma \mathbf{\max_{a \in \mathcal{A}} Q(s_{t+1}, a)} - Q(s_t, a_t)]$$

Caractéristiques

- En ligne, avec une politique d'exploration.
- La connaissance de la dynamique n'est pas nécessaire
- Convergence
- MAIS lent (exploration/exploitation)

Plan de l'exposé

- Introduction
- Formalisme de l'Apprentissage par Renforcement
- Algorithmes pour l'Apprentissage par Renforcement
- Apprendre Efficacement
- Traces d'éligibilité
- Discussion

Apprentissage "efficace"

Complexité exploration

Pour tout $\epsilon > 0$, complexité d'exploration est le nombre de pas de temps (moyen) t tel que la politique π_t à un instant t devienne ϵ-optimale : $||V^{\pi_t}(.) - V^*(.)|| \leq \epsilon$

Algorithme PAC-efficace

Algoritme est PAC-efficace si, pour tout $\epsilon > 0$ et $0 \leq \delta > 1$, sa complexité d'exploration peut s'exprimer sous la forme d'un polynôme en fonction de $|\mathcal{S}|$, $|\mathcal{A}|$, $1/\epsilon$, $1/\delta$ et $1/(1-\gamma)$ avec une probabilité supérieure à $1-\delta$.

Méthodes d'exploration non dirigées

Politique `greedy`

$$\pi_{\text{greedy}}(s) \quad = \quad \underset{a \in \mathcal{A}}{\text{argmax}}\, Q(s,a)$$

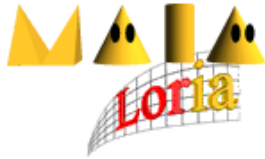
Politique ϵ-`greedy`

$$\pi_\epsilon(s) \quad = \quad \begin{cases} \pi_{\text{greedy}}(s) & \text{avec proba } (1-\epsilon) \\ \text{action } a \text{ aléatoire} & \text{avec proba } \epsilon \end{cases}$$

Politique `softmax`

$$\overset{\pi}{\Pr}(s,a) \quad = \quad \frac{e^{Q(s,a)}/T}{\sum_{a' \in \mathcal{A}} e^{Q(s,a')}/T}$$

Exploration dirigée de manière "comptable"



Visiter les états les moins fréquents [Barto and Sutton, 1990]

$$\mathrm{eval}(s_t, a) \quad = \quad \beta Q(s_t, a) + \frac{c(s_t)}{\mathbb{E}[c(s_{t+1})]}$$

(extension) décroissance du compteur

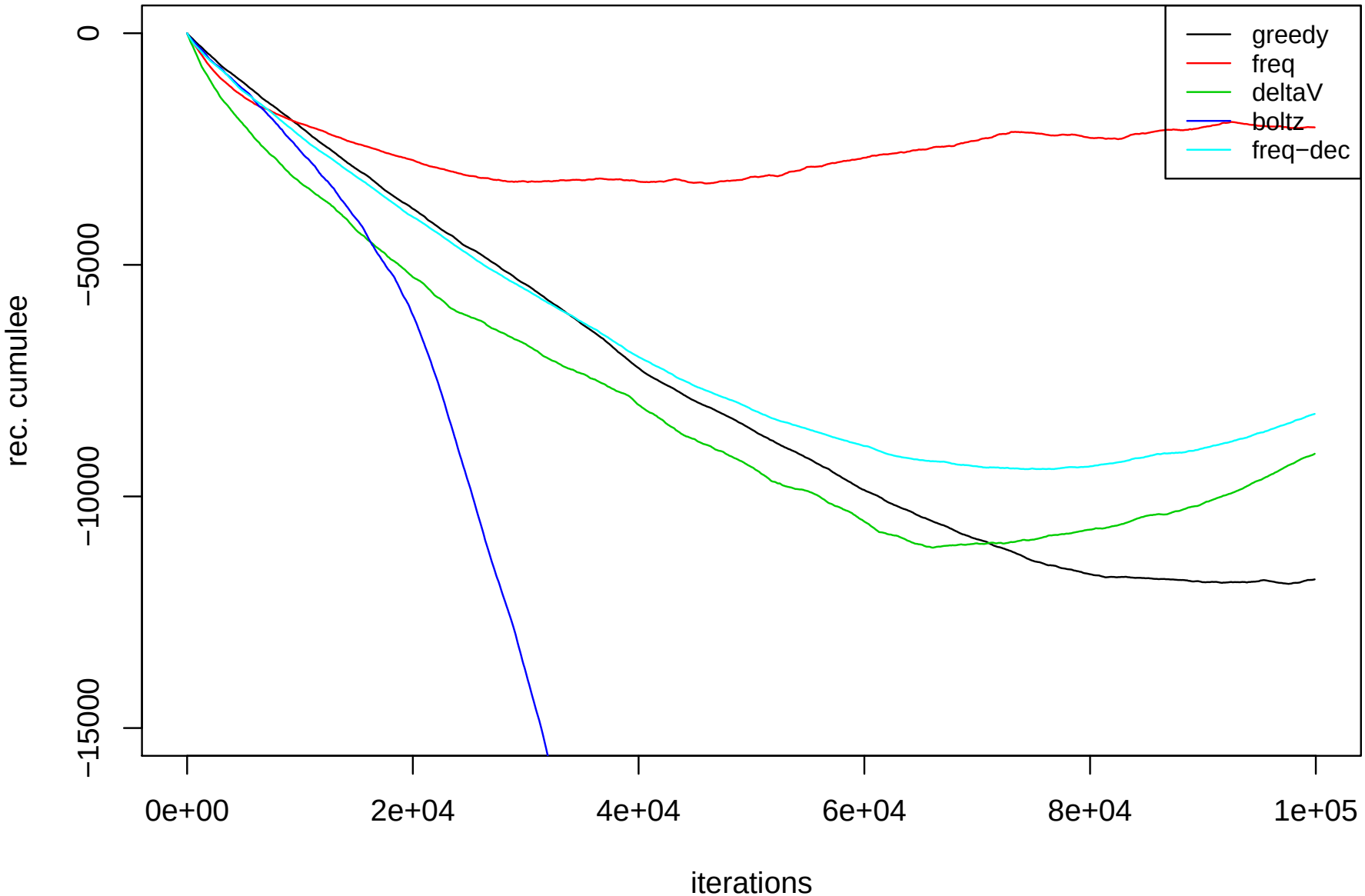
$$c(s) \quad \longleftarrow \quad \lambda c(s)$$

Visiter les états les moins récents [Sutton, 1990]

$$\mathrm{eval}(s_t, a) \quad = \quad \beta Q(s_t, a) + \sqrt{\mathbb{E}[\rho_{s\{t+1}]}$$

Visiter les états apportant le plus [Schmidhuber, 1991]

$$\mathrm{eval}(s_t, a) \quad = \quad \beta Q(s_t, a) + \mathbb{E}[\Delta V_{\mathrm{last}}(s_{t+1})]$$

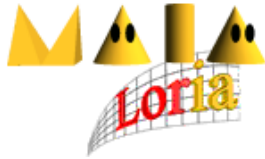

greedy
freq
deltaV
boltz
freq−dec
rec. cumulee
0
−5000
−10000
−15000
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
iterations

Algorithme E^3, [Kearns and Singh, 1998]

E^3 pour `Explicit Explore or Exploit`
S est l'ensemble des états *connus*

1. Si état courant $s \notin S$, choisir action moins explorée
2. Si s exploré m fois, alors $S \leftarrow S \cup \{s\}$
3. Si état courant $s \in S$ et $||V(s) - V^*(s)|| < \epsilon$
   - Alors **exploiter** MDP avec politique `greedy`
   - Sinon **explorer** en maximisant proba d'aller vers état inconnu

Algorithme E^3, [Kearns and Singh, 1998]



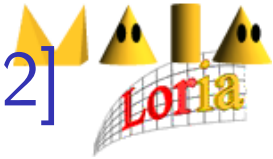
E^3 pour `Explicit Explore or Exploit`
S est l'ensemble des états *connus*

1. Si état courant $s \notin S$, choisir action moins explorée
2. Si s exploré m fois, alors $S \leftarrow S \cup \{s\}$
3. Si état courant $s \in S$ et $||V(s) - V^*(s)|| < \epsilon$
   - Alors **exploiter** MDP avec politique `greedy`
   - Sinon **explorer** en maximisant proba d'aller vers état inconnu

Caractéristiques

- Nécessite de *connaître* le modèle
- Résoudre *deux* MDP par itération

Algorithme R-MAX, [Brafman and Tennenholtz, 2002]



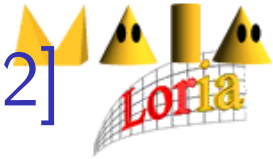
Utilise un *a priori* optimiste.

- Ajouter un état fictif $s_{\max}$
- Pour tout les états, $\hat{r} = R_{\max}$
- Initialement, pour tous les états, $\hat{p}(s_{\max}|a, s) = 1$

A chaque pas de temps,

1. **exploiter** le modèle actuel $\hat{p}(), \hat{r}()$
2. Mettre à jour $\hat{p}()$ et $\hat{r}()$ si état devient **connu**

Algorithme R-MAX, [Brafman and Tennenholtz, 2002]



Utilise un *a priori* optimiste.

- Ajouter un état fictif $s_{\max}$
- Pour tout les états, $\hat{r} = R_{\max}$
- Initialement, pour tous les états, $\hat{p}(s_{\max}|a, s) = 1$

A chaque pas de temps,

1. **exploiter** le modèle actuel $\hat{p}(), \hat{r}()$
2. Mettre à jour $\hat{p()}$ et $\hat{r()}$ si état devient **connu**

Caractéristiques

- Nécessite de mémoriser le modèle
- Résoudre *un* MDP par itération
- Propriétés prouvées.
- Existe des versions plus efficaces (*RTDP-RMAX*).

Algorithme MBIE, [Strehl and Littman, 2004]

MBIE pour *Model Based Interval Estimation*

A chaque instant, estimer

- ens. $CI(\hat{r})$ des fonctions récompense qui appartiennent à $[\hat{r}(s,a) - \epsilon_r, \hat{r}(s,a) + \epsilon_r]$
- ens. $CI(\hat{p})$ des distributions de proba p qui vérifient $||p(.|s,a) - \hat{p}(.|s,a)||_1 < \epsilon_p$

$$\hat{Q}(s,a) \quad = \quad \max_{\mathbf{r()\in CI(\hat{r})}} r(s,a) + \max_{\mathbf{p()\in CI(\hat{p})}} \gamma \sum_{s' in \mathcal{S}} p(s'|a,s) \max_{a' \in \mathcal{A}} \hat{Q}(s,a)$$

Algorithme MBIE, [Strehl and Littman, 2004]

MBIE pour *Model Based Interval Estimation*

A chaque instant, estimer

- ens. $CI(\hat{r})$ des fonctions récompense qui appartiennent à $[\hat{r}(s,a) - \epsilon_r, \hat{r}(s,a) + \epsilon_r]$
- ens. $CI(\hat{p})$ des distributions de proba p qui vérifient $||p(.|s,a) - \hat{p}(.|s,a)||_1 < \epsilon_p$

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

Caractéristiques

- Nécessite de mémoriser le modèle
- Résoudre *un* MDP par itération
- Propriétés prouvées.
- Existe des versions plus efficaces (*RTDP-MBIE*) ou plus simples (*MBIE-EB*).

Exploration dirigée par estimarion bayésienne

Approche bayésienne pour évaluer la “confiance” dans l’estimation actuelle de la fonction valeur.

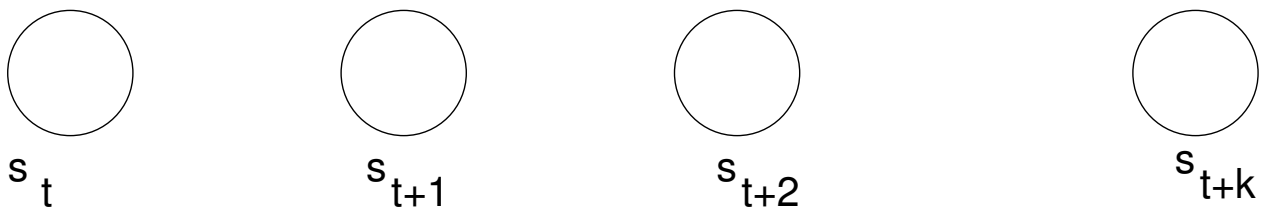
- Avec `Value Iteration` : [Mannor et al., 2004]

- Avec `Q-Learning` : [Dearden et al., 1998]

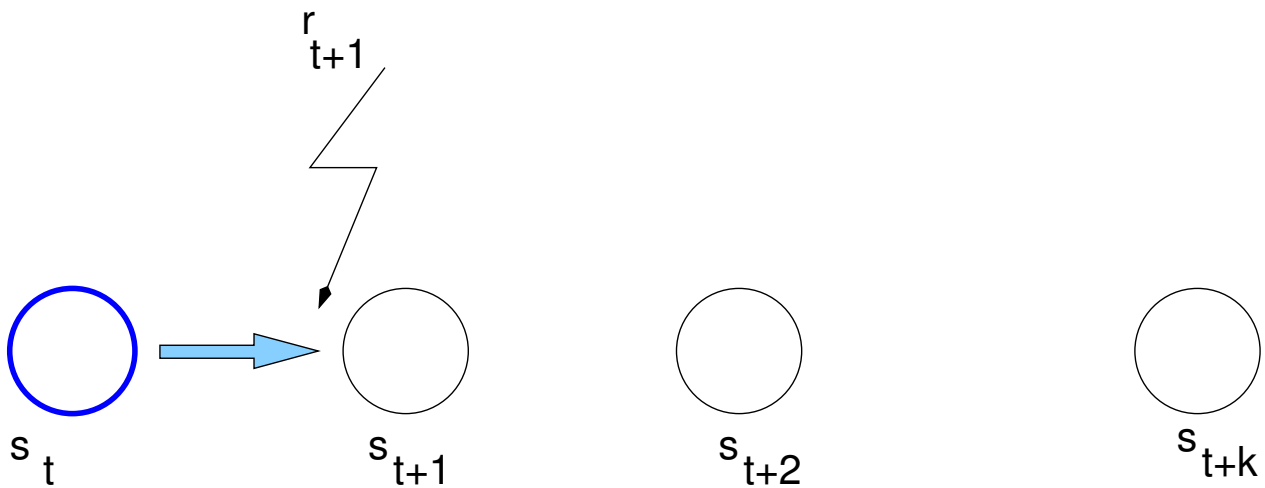
Plan de l'exposé

- Introduction
- Formalisme de l'Apprentissage par Renforcement
- Algorithmes pour l'Apprentissage par Renforcement
- Apprendre Efficacement
- Traces d'éligibilité
- Discussion

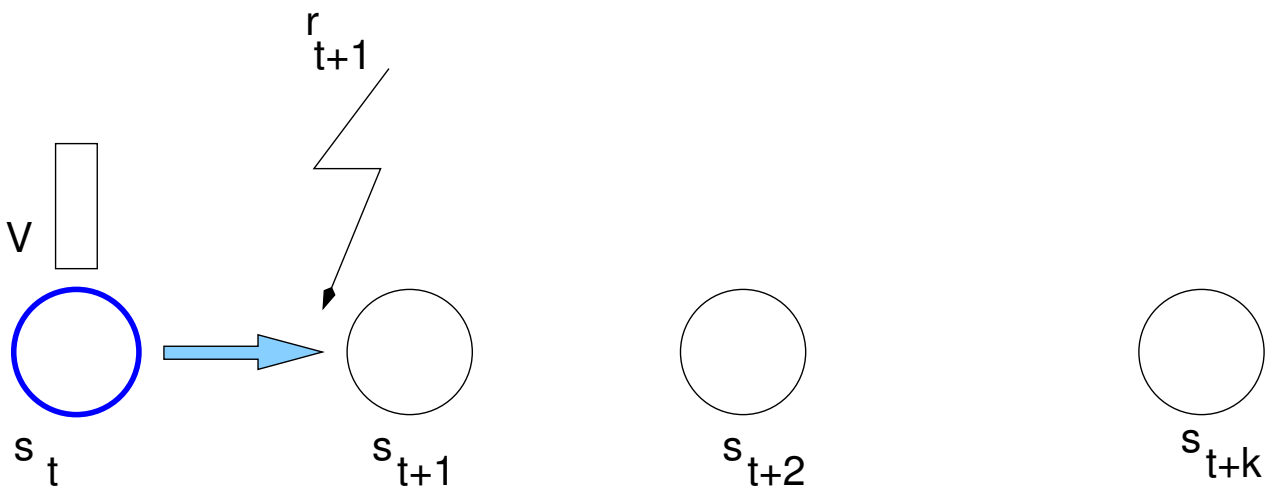
Traces d'éligibilité



Traces d'éligibilité

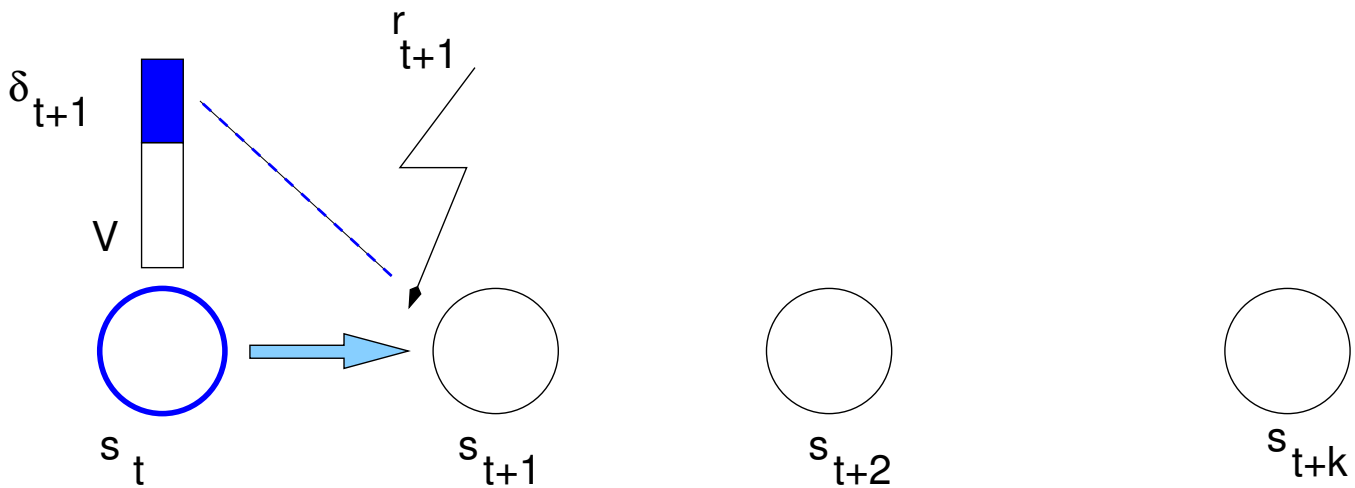


Traces d'éligibilité



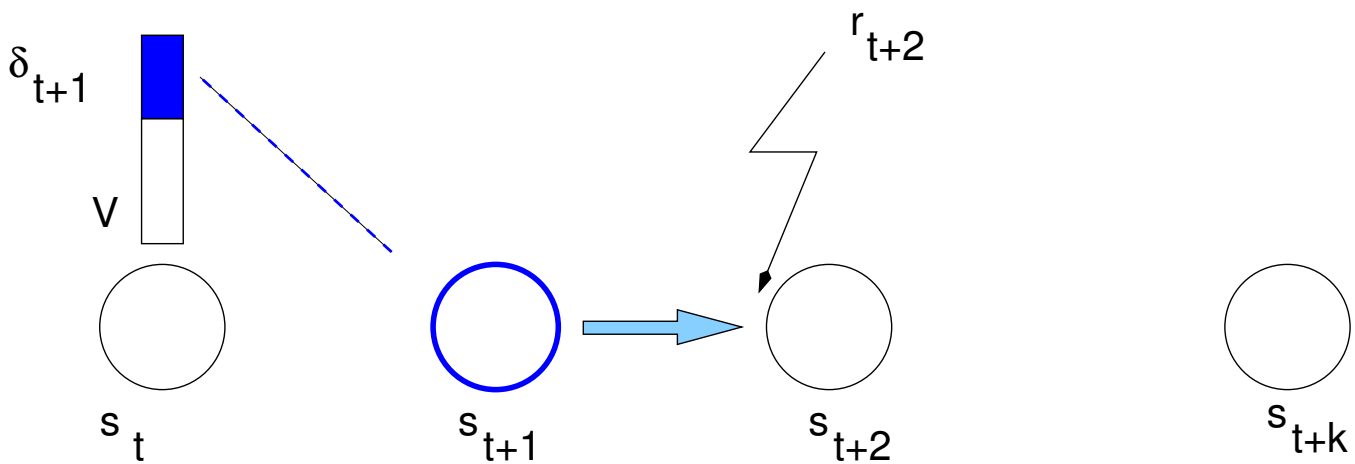
Traces d'éligibilité

$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$



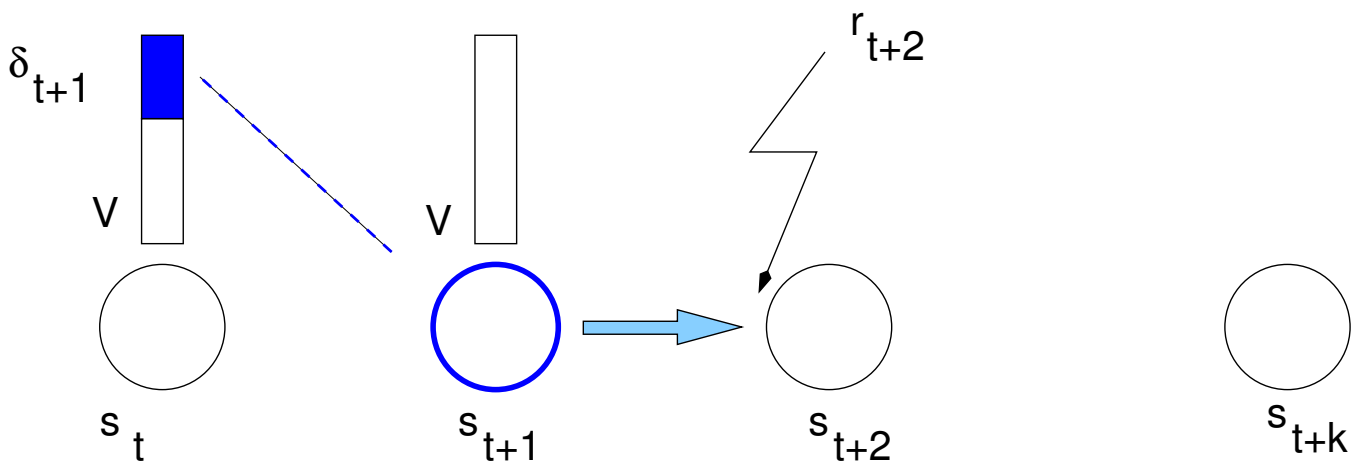
Traces d'éligibilité

$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$



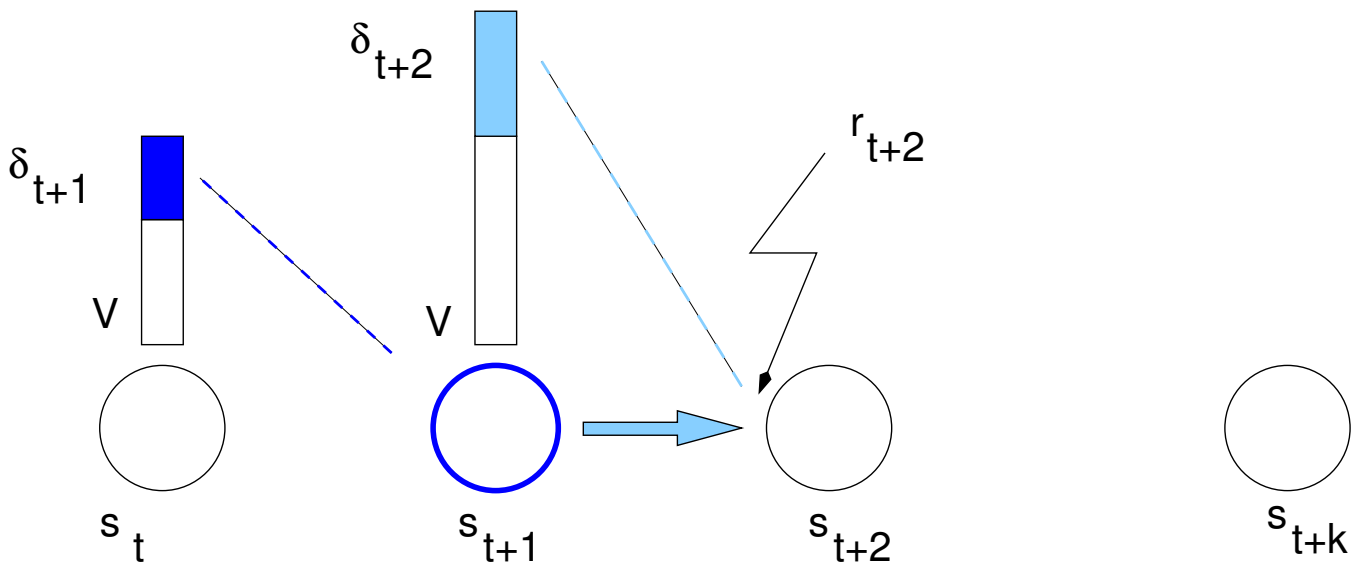
Traces d'éligibilité

$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$



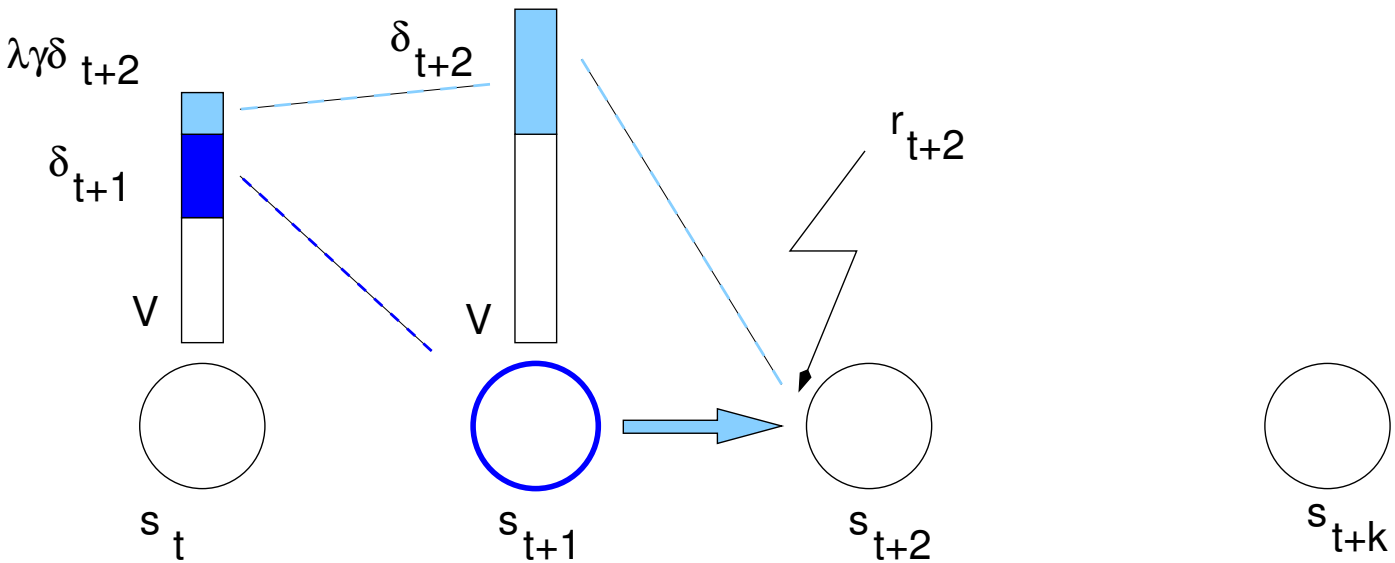
Traces d'éligibilité

$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$



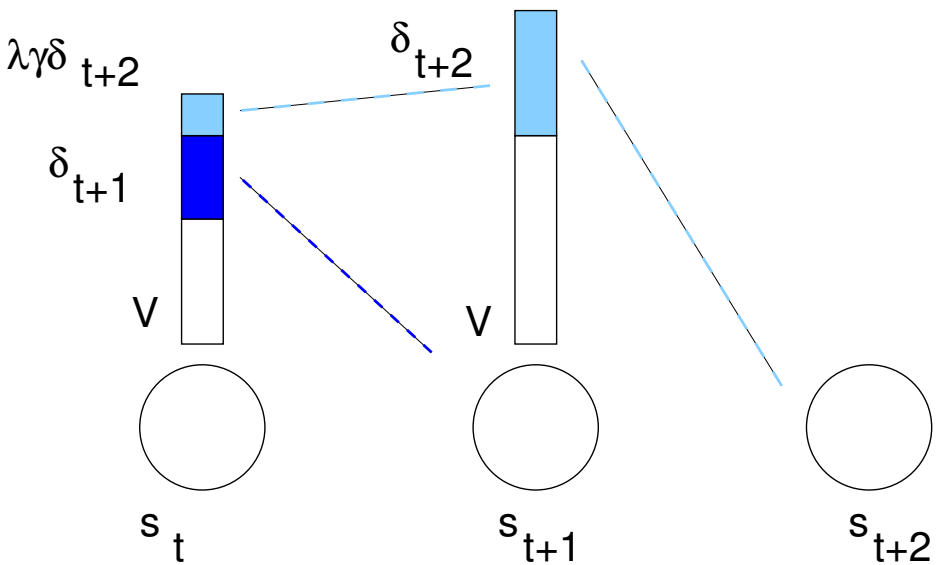
Traces d'éligibilité

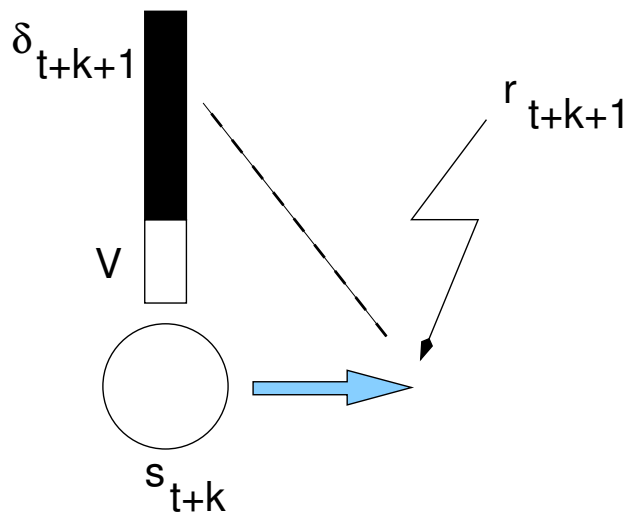
$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$



Traces d'éligibilité

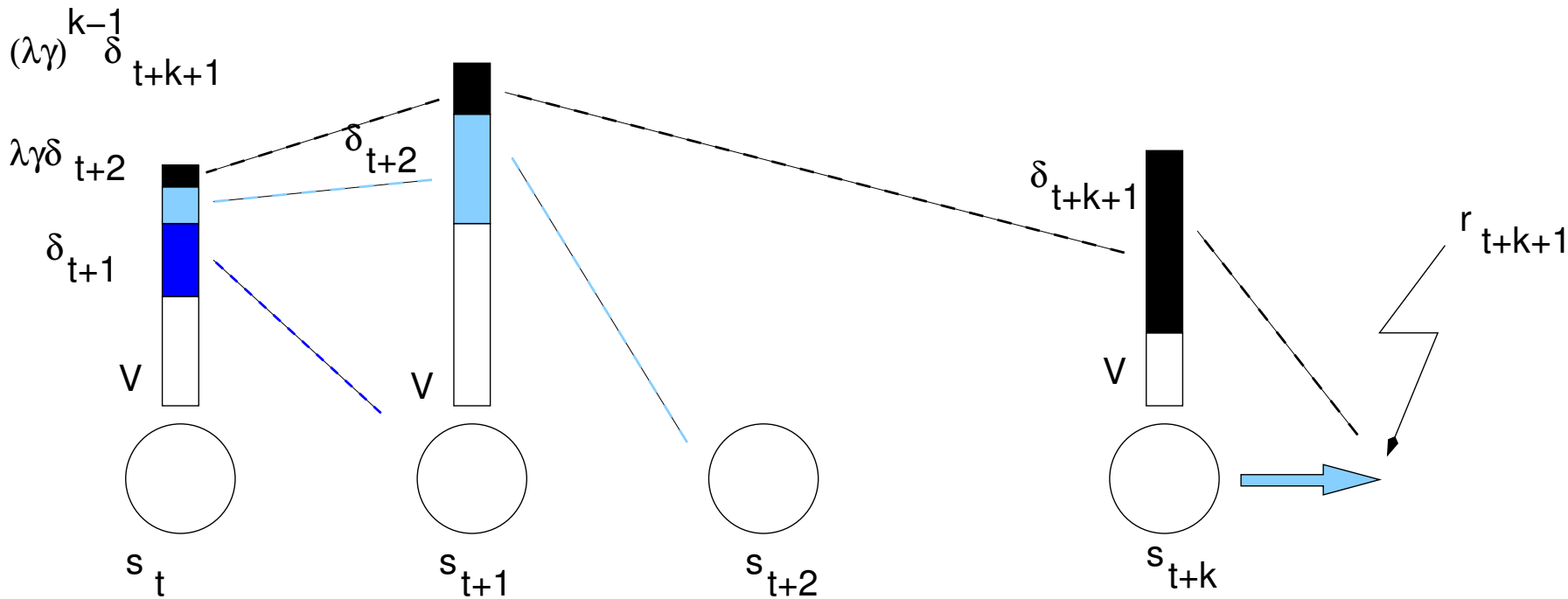
$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$





Traces d'éligibilité

$$\delta_{t+1} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$



Mise en œuvre

1. Transition (s, a, r, s')
2. $\delta = r + \gamma V(s') - V(s)$
3. $e(s) \leftarrow e(s) + 1$
4. Pour tout $s \in \mathcal{S}$
   - 4.1 $V(s) \leftarrow V(s) + \alpha\delta e(s)$
   - 4.2 $e(s) \leftarrow \gamma\lambda e(s)$

S'applique à tous les algo.

- TD(λ)
- Sarsa(λ)
- Q(λ)-Learning
- ...

Mise en œuvre

1. Transition (s, a, r, s')
2. $\delta = r + \gamma V(s') - V(s)$
3. $e(s) \leftarrow e(s) + 1$
4. Pour tout $s \in \mathcal{S}$
   - 4.1 $V(s) \leftarrow V(s) + \alpha\delta e(s)$
   - 4.2 $e(s) \leftarrow \gamma\lambda e(s)$

S'applique à tous les algo.

- TD(λ)
- SARSA(λ)
- Q(λ)-Learning
- ...

Caractéristiques

- Accélère la convergence.
- Unification des méthodes de MonteCarlo et des différences temporelles.
- MAIS Attention aux méthodes "hors"-politiques.

Plan de l'exposé

- Introduction
- Formalisme de l'Apprentissage par Renforcement
- Algorithmes pour l'Apprentissage par Renforcement
- Apprendre Efficacement
- Traces d'éligibilité
- Discussion

Mais “efficace”, ça veut dire quoi ?

On est *efficace* par rapport à un objectif, une tâche

- **Contrôler**
  - on parle bien d’efficacité dans l’apprentissage.
  - Importance/influence de $r()$, γ, α.

Mais "efficace", ça veut dire quoi ?

On est *efficace* par rapport à un objectif, une tâche

- **Contrôler**
  - on parle bien d'efficacité dans l'apprentissage.
  - Importance/influence de $r()$, γ, α.
- **Simuler**
  - Coller "au mieux" à des données réelles.
  - Nombreux paramètres : $\mathcal{S}$, $\mathcal{A}$, $p()$, $r()$, γ, π,...
  - Efficacité pour trouver les "bons" paramètres ?
  - Efficacité pour trouver les "bonnes" politiques ?
  - Autre ?

Les “vraies” difficultés

Point de vue “Sciences Cognitives” mais aussi “Contrôle”.

- Définir et identifier les “états”.
  - quels sont les états pertinents de l’environnement (Markov) ?
  - reconnaître un état mémorisé
  - représentation axée “action”, “comportement”. ?
- Comment mémoriser, représenter la dynamque, le modèle
- Comment mémoriser, représenter les incertitudes.
- ...

Références I

Barto, A. and Sutton, R. (1990).
On the computationnal economics of reinforcement learning.
In Kaufmann, M., editor, *Connectionist Models, Proc. of the 1990 Summer School*, pages 35–44.

Brafman, R. and Tennenholtz, M. (2002).
R-MAX. A general polynomial time algorithm for near-optimal reinforcement learning.
Journal of Machine Learning Research, 3(2) :213–231.

Références II

Dearden, R., Friedman, N., and Russell, S. (1998).
Bayesian q-learning.
In *AAAI '98/IAAI '98 : Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 761–768, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Kearns, M. and Singh, S. (1998).
Near-optimal reinforcement learning in polyniomial time.
In *Proc. of the 15th Int. Conf. on Machine Learning (ICML'98)*, pages 260–268.

Références III


Mannor, S., Simester, D., Sun, P., and Tsitsiklis, J. (2004).
Bias and variance in value function estimation.
In *Proc. of the 21st Int. Conf. on Machine Learning (ICML)*, pages 568–575.


Myerson, R. (1991).
Game Theory : Analysis of Conflict.
Harvard University Press.


Puterman, M. (1994).
Markov Decision Processes : discrete stochastic dynamic programming.
John Wiley & Sons, Inc. New York, NY.

Références IV

Schmidhuber, J. (1991).
Adaptive confidence and adaptive curiosity.
Technical Report FKI-149-91, Fakultat fur Informatik, Technische Universitat, München.

Strehl, A. and Littman, M. (2004).
An empirical evaluation of interval estimation for Markov decision processes.
In *Proc. of the IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI-2004)*, pages 128–135.

Références V


Sutton, R. (1990).
Integrated architectures for learning, plannin and reacting based on approcimate dynamic programming.
In *Proc. if the 7th Int. Conf. on Machine Learning (ICML)*, pages 216–224.


Sutton, R. and Barto, A. (1998).
Reinforcement Learning.
Bradford Book, MIT Press, Cambridge, MA.