Exemple

Environnement simplifié

2	R=-0.04	R=+1
1	R=-0.04	R=-1
	1	2

- Etat initial (1,1)
- Etats "buts": (2,1), (2,2)
- $\gamma = 1$
- Actions N,W,S,E.
- $U(1,1) = R(1,1) + \gamma \max_a \{0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1), 0.9U(1,1) + 0.1U(1,2), 0.9U(1,1) + 0.1U(2,1), 0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)\}$

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \Sigma_s (T(s, a, s') U_i(s'))$$

Exemple (1)

- itération 1
  - U(1,1) = R(1,1) + γ max { 0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1), N
    0.9U(1,1) + 0.1U(1,2), W
    0.9U(1,1) + 0.1U(2,1), S
    0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)} E
    = -0.04 + 1 x max { 0.8x(-0.04) + 0.1x(-0.04) + 0.1x(-1), N
    0.9x(-0.04) + 0.1x(-0.04), W
    0.9x(-0.04) + 0.1x(-1), S
    0.8x(-1) + 0.1x(-0.04) + 0.1x(-0.04)} E
    =-0.04 + max{ -0.136, -0.04, -0.136, -0.808}
    =-0.08
  - U(1,2) = R(1,2) + γ max {0.9U(1,2) + 0.1U(2,2), N
    0.9U(1,2) + 0.1U(1,1), W
    0.8U(1,1) + 0.1U(2,2) + 0.1U(1,2), S
    0.8U(2,2) + 0.1U(1,2) + 0.1U(1,1)} E
    = -0.04 + 1 x max {0.9x(-0.04)+ 0.1x1, N
    0.9x(-0.04) + 0.1x(-0.04), W
    0.8x(-0.04) + 0.1x1 + 0.1x(-0.04), S
    0.8x1 + 0.1x(-0.04) + 0.1x(-0.04)} E
    =-0.04 + max{ 0.064, -0.04, 0.064, 0.792}
    =0.752

2	U=-0.04 R=-0.04	U=+1 R=+1
1	U=-0.04 R=-0.04	U=-1 R=-1
	1	2

Exemple (2)

- itération 2
  - U(1,1) = R(1,1) + γ max { 0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1), N
    0.9U(1,1) + 0.1U(1,2), W
    0.9U(1,1) + 0.1U(2,1), S
    0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)} E
    = -0.04 + 1 x max { 0.8x(0.752) + 0.1x(-0.08) + 0.1x(-1), N
    0.9x(-0.08) + 0.1x(0.752), W
    0.9x(-0.08) + 0.1x(-1), S
    0.8x(-1) + 0.1x(-0.08) + 0.1x(0.752)} E
    =-0.04 + max{ 0.4936, 0.0032, -0.172, -0.3728}
    =0.4536
  - U(1,2) = R(1,2) + γ max {0.9U(1,2) + 0.1U(2,2), N
    0.9U(1,2) + 0.1U(1,1), W
    0.8U(1,1) + 0.1U(2,2) + 0.1U(1,2), S
    0.8U(2,2) + 0.1U(1,2) + 0.1U(1,1)} E
    = -0.04 + 1 x max {0.9x(0.752)+ 0.1x1, N
    0.9x(0.752) + 0.1x(-0.08), W
    0.8x(-0.08) + 0.1x1 + 0.1x(0.752), S
    0.8x1 + 0.1x(0.752) + 0.1x(-0.08)} E
    =-0.04 + max{ 0.7768, 0.6688, 0.1112, 0.8672}
    = 0.8272

2	U=0.752 R=-0.04	U=+1 R=+1
1	U=-0.08 R=-0.04	U=-1 R=-1
	1	2

Exemple (3)

- Iteration 3
  - U(1,1) = R(1,1) + γ max { 0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1), N
    0.9U(1,1) + 0.1U(1,2), W
    0.9U(1,1) + 0.1U(2,1), S
    0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)} E
    = -0.04 + 1 x max { 0.8x(0.8272) + 0.1x(0.4536) + 0.1x(-1), N
    0.9x(0.4536) + 0.1x(0.8272), W
    0.9x(0.4536) + 0.1x(-1), S
    0.8x(-1) + 0.1x(0.4536) + 0.1x(0.8272)} E
    =-0.04 + max{ 0.6071, 0.491, 0.3082, -0.6719}
    =0.5676
  - U(1,2) = R(1,2) + γ max {0.9U(1,2) + 0.1U(2,2), N
    0.9U(1,2) + 0.1U(1,1), W
    0.8U(1,1) + 0.1U(2,2) + 0.1U(1,2), S
    0.8U(2,2) + 0.1U(1,2) + 0.1U(1,1)} E
    = -0.04 + 1 x max {0.9x(0.8272)+ 0.1x1, N
    0.9x(0.8272) + 0.1x(0.4536), W
    0.8x(0.4536) + 0.1x1 + 0.1x(0.8272), S
    0.8x1 + 0.1x(0.8272) + 0.1x(0.4536)} E
    =-0.04 + max{ 0.8444, 0.7898, 0.5456, 0.9281}
    = 0.8881

2	U=0.8272 R=-0.04	U=+1 R=+1
1	U= 0.4536 R=-0.04	U=-1 R=-1
	1	2

Exemple (4)

- Itération suivante

U= 0.8881 R=-0.04	U=+1 R=+1
U=0.5676 R=-0.04	U=-1 R=-1

- Fin si $U_{i+1} \approx U_i$ à ε près.
- Ici écart max: 0.114.

Convergence

- Détection de la convergence:
  - Erreur de moindres carrés de la valeur d'utilité

$$RMS = \frac{1}{|S|} \cdot \sqrt{\sum_{i=1}^{|S|} (U(i) - U'(i))^2}$$

RMS(U, U') < ε

ε = écart maximum admis pour toutes les valeurs dans une itération

Exercice - énoncé (1)

- Soit l'environnement montré dans la figure dans lequel un robot doit naviguer. La cellule C3 est occupée par un obstacle et ne peut-être atteinte. Les récompenses sont indiquées dans les cellules. Un état du robot est représenté par une paire (p, o) où p et o représentent respectivement la position (cellule) et l'orientation (haut, bas, gauche, droite) du robot.

C1→ 2	C2 -50	C3
C4 2	C5 2	C6 10
C7 2	C8 2	C9↑ 20

Exercice - énoncé (2)

- Le robot peut exécuter trois actions.
  - a1 : Effectuer une rotation de 90° vers la gauche;
  - a2 : Effectuer une rotation de 90° vers la droite;
  - a3 : Avancer d'une case.
- Les actions de rotation sont déterministes, mais l'action d'avancer est incertaine. Ainsi, suite à l'exécution de l'action a3, le robot peut ne pas avoir bougé (P=5%), peut avoir avancé d'une seule case (P=85%), ou peut avoir avancé de deux cases (P=10%). S'il heurte un obstacle ou un mur le robot reste sur la case où a eu lieu le choc.
- En considérant un facteur d'escompte $\gamma = 0,8$, simulez deux itérations de l'algorithme d'itération de valeurs sur les états (C1, droite) et (C9, haut).

Solution (1)

- itération 1
  - U(C1,D) = R(C1) + γ max {U(C1), RG
    U(C1), RD
    0.05U(C1) + 0.95U(C2,D) AV
    }
    = 2 + 0.8 max { 2, RG
    2, RD
    0.05x2 + 0.95x-50 AV
    }
    = 2 + 0.8 max{ 2, 2, -47.4}
    =2+1,6 = **3,6**
  - U(C9,H) = R(C9) + γ max {U(C9), RG
    U(C9), RD
    0.05U(C9) + 0.95U(C6,H) AV
    }
    = 20 + 0.8 max {20, RG
    20, RD
    0.05x20+0.95x10 AV
    }
    =20 + 0.8 max{ 20, 20, 10.5}
    =**36**

C1→ 2	C2 -50	C3
C4 2	C5 2	C6 10
C7 2	C8 2	C9↑ 20

Attention: Pour calculer la seconde itération, on a besoin des utilités de C2 et de C6. Il faut donc aussi les calculer à la première itération

Solution (2)

- itération 1
  - U(C2,D) = R(C2) + γ max {U(C2), RG
    U(C2), RD
    U(C2) AV
    }
    = -50 + 0.8 max { -50, RG
    -50, RD
    -50 AV
    }
    = -50 + 0.8 max{ -50,-50, -50}
    =-50-40 = **-90**
  - U(C6,H) = R(C6) + γ max {U(C6), RG
    U(C6), RD
    U(C6) AV
    }
    = 10 + 0.8 max {10, RG
    10, RD
    10, AV
    }
    =10 + 0.8 max{ 10, 10, 10}
    =**18**

C1→ 2	C2 -50	C3
C4 2	C5 2	C6 10
C7 2	C8 2	C9↑ 20

Solution (3)

- itération 2
  - U(C1,D) = R(C1) + γ max {U(C1), RG
    U(C1), RD
    0.05U(C1) + 0.95U(C2,D) AV
    }
    = 2 + 0.8 max { 3.6, RG
    3.6, RD
    0.05x3.6 + 0.95x-90 AV
    }
    = 2 + 0.8 max{ 3.6, 3.6, -85.32}
    = 2+2.88 = **4.88**
  - U(C9,H) = R(C9) + γ max {U(C9), RG
    U(C9), RD
    0.05U(C9) + 0.95U(C6,H) AV
    }
    = 20 + 0.8 max {36, RG
    36, RD
    0.05x36+0.95x18 AV
    }
    =20 + 0.8 max{ 36, 36, 18.9}
    = **48.8**

C1→ U=3.6	C2 U=-90	C3
C4 2	C5 2	C6 U=18
C7 2	C8 2	C9↑ U=36

La politique

- Les utilités étant trouvées
- Recherche de la politique optimale:

  Pour tout s dans S:

  $\pi[s] = argmax_a \{\sum_{s'} T(s, a, s')U(s')\}$

Fournit π^*

Exemple: politique optimale

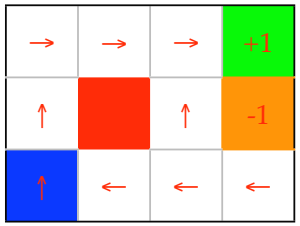
- Calcul de la politique optimale
  - $\Pi(1,1) = \text{argmax}_a \{ 0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1),$ N
    $0.9U(1,1) + 0.1U(1,2),$ W
    $0.9U(1,1) + 0.1U(2,1),$ S
    $0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)\}$ E
    $= \text{argmax}_a \{ 0.8x(0.8881) + 0.1x(0.5676) + 0.1x(-1),$ N
    $0.9x(0.5676) + 0.1x(0.8881),$ W
    $0.9x(0.5676) + 0.1x(-1),$ S
    $0.8x(-1) + 0.1x(0.5676) + 0.1x(0.8881)\}$ E
    $= \text{argmax}_a \{ 0.6672, 0.5996, 0.4108, -0.6512\}$
    $= N$
  - $\Pi(1,2) = \text{argmax}_a \{ 0.9U(1,2) + 0.1U(2,2),$ N
    $0.9U(1,2) + 0.1U(1,1),$ W
    $0.8U(1,1) + 0.1U(2,2) + 0.1U(1,2),$ S
    $0.8U(2,2) + 0.1U(1,2) + 0.1U(1,1)\}$ E
    $= \text{argmax}_a \{ 0.9x(0.8881) + 0.1x1,$ N
    $0.9x(0.8881) + 0.1x(0.5676),$ W
    $0.8x(0.5676) + 0.1x1 + 0.1x(0.8881),$ S
    $0.8x1 + 0.1x(0.8881) + 0.1x(0.5676)\}$ E
    $= \text{argmax}_a \{0.8993, 0.8561, 0.6429, 0.9456\}$
    $= E$

U= 0.8881 R=-0.04	U=+1 R=+1
U=-0.5676 R=-0.04	U=-1 R=-1

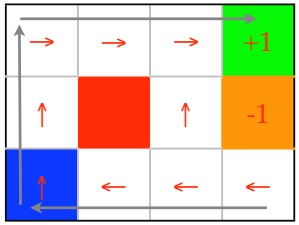
Résumé: itération de valeur

1. Environment et récompenses

2. Calcul itératif des utilités



3. Calcul de la politique optimale



4. Problème spécifique; ex: "Que doit faire le robot s'il est dans l'état (4,1)"?

2. Itération de Politique

- Choix d'une politique puis calcul de l'utilité de chaque état pour cette politique.
- Mise à jour de la politique à chaque état en utilisant les utilités des états successeurs.
- Répétition jusqu'à obtenir une politique stable.

Itération de politique

- Soit une politique π_i . Pour chaque état, à chaque étape:
  - Evaluation de la politique
    - Calculer l'utilité U_i de chaque état si π_i devait être exécutée
  - Amélioration de la politique
    - Calculer une nouvelle politique π_{i+1} à partir de π_i:

$$\pi_{i+1}[s] \leftarrow \text{argmax}_a\{\Sigma_{s'} T(s, a, s')U_i(s')\}$$

Itération de politique

ITERATION_POLITIQUE (mdp)

entrée: *mdp(S,A,T,R)*

Initialiser *U* (utilités des états de *S*) à *R*

π (politique) à une valeur arbitraire

répéter jusqu'à *π* inchangé

Calculer l'utilité *U* de chaque état si *π* devait être exécutée (évaluation de *π)*

Pour chaque *s* **faire**

si $\max_a\{\Sigma_{s'}T(s,a,s')\ U[s']\} > \Sigma_{s'}T(s,\pi[s],s')\ U[s']$ **alors**

$\pi[s] \leftarrow \text{argmax}_a\{\Sigma_{s'}T(s,a,s')\ U[j]\}$

retourner *π*

Exemple itération de politique

- Politique initialisée à N pour (1,1) et (1,2)
- Itération 1; calcul des utilités.
  - $U(1,1) = R(1,1) + \gamma\,(0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1))$

    $U(1,2) = R(1,2) + \gamma\,(0.9U(1,2) + 0.1U(2,2))$

    $U(2,1) = R(2,1) = -1$

    $U(2,2) = R(2,2) = 1$

    Peut s'écrire:

  - $0.04 = -0.9U(1,1) + 0.8\,U(1,2) + 0.1\,U(2,1) + 0\ \ U(2,2)$
    $0.04 = \ \ 0U(1,1) - 0.1\ \ U(1,2) + 0\ \ U(2,1) + 0.1\,U(2,2)$
    $-1 \ \ = \ \ 0U(1,1) + 0\ \ U(1,2) + 1\ \ U(2,1) + 0\ \ U(2,2)$
    $1 \ \ = \ \ 0U(1,1) + 0\ \ U(1,2) - 0\ \ U(2,1) + 1\ \ U(2,2)$

2	U=-0.04 R=-0.04	U=+1 R=+1
1	U=-0.04 R=-0.04	U=-1 R=-1
	1	2

Politique
Π(1,1) = *N*
Π(1,2) = *N*

$\gamma = 1$

Exemple - suite

$$\begin{bmatrix} -0.9 & 0.8 & 0.1 & 0 \\ 0 & -0.1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U(1,1) \\ U(1,2) \\ U(2,1) \\ U(2,2) \end{bmatrix} = \begin{bmatrix} 0.04 \\ 0.04 \\ -1 \\ 1 \end{bmatrix}$$

- D'où:

  U(1,1) = 0.3778

  U(1,2) = 0.6

  U(2,1) = -1

  U(2,2) = 1

Exemple - suite

- Première itération - Amélioration de la politique
  - $\Pi(1,1) = \text{argmax}_a\{0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1),$ N
    $0.9U(1,1) + 0.1U(1,2),$ W
    $0.9U(1,1) + 0.1U(2,1),$ S
    $0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)\}$ E
    $= \text{argmax}_a\{0.8x(0.6) + 0.1x(0.3778) + 0.1x(-1),$ N
    $0.9x(0.3778) + 0.1x(0.6),$ W
    $0.9x(0.3778) + 0.1x(-1),$ S
    $0.8x(-1) + 0.1x(0.3778) + 0.1x(0.6)\}$ E
    $= \text{argmax}_a\{0.4178, 0.4, 0.24, -0.7022\}$
    = N ; pas de mise à jour.
  - $\Pi(1,2) = \text{argmax}_a\{0.9U(1,2) + 0.1U(2,2),$ N
    $0.9U(1,2) + 0.1U(1,1),$ W
    $0.8U(1,1) + 0.1U(2,2) + 0.1U(1,2),$ S
    $0.8U(2,2) + 0.1U(1,2) + 0.1U(1,1)\}$ E
    $= \text{argmax}_a\{0.9x(0.6) + 0.1x1,$ N
    $0.9x(0.6) + 0.1x(0.3778),$ W
    $0.8x(0.3778) + 0.1x1 + 0.1x(0.6),$ S
    $0.8x1 + 0.1x(0.6) + 0.1x(0.3778)\}$ E
    $= \text{argmax}_a\{0.64, 0.5778, 0.4622, 0.8978\}$
    = E ; **mise à jour**.

2	U= 0.6 R=-0.04	U=+1 R=+1
1	U=0.3778 R=-0.04	U=-1 R=-1
	1	2

Politique
Π(1,1) = *N*
Π(1,2) = *N*

Exemple - suite

- Seconde itération – évaluation de politique (calcul des utilités)
  - U(1,1) = R(1,1) + γ x (0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1))
    U(1,2) = R(1,2) + γ x (0.1U(1,2) + 0.8U(2,2) + 0.1U(1,1))
    U(2,1) = R(2,1)
    U(2,2) = R(2,2)
  - U(1,1) = -0.04 + 0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1)
    U(1,2) = -0.04 + 0.1U(1,2) + 0.8U(2,2) + 0.1U(1,1)
    U(2,1) = -1
    U(2,2) = 1
  - 0.04 = -0.9U(1,1) + 0.8U(1,2) + 0.1U(2,1) + 0U(2,2)
    0.04 = 0.1U(1,1) – 0.9U(1,2) + 0U(2,1) + 0.8U(2,2)
    -1 = 0U(1,1) + 0U(1,2) - 1U(2,1) + 0U(2,2)
    1 = 0U(1,1) + 0U(1,2) - 0U(2,1) + 1U(2,2)
  - U(1,1) = 0.5413
    U(1,2) = 0.7843
    U(2,1) = -1
    U(2,2) = 1

2	U= 0.6 R=-0.04	U=+1 R=+1
1	U=0.3778 R=-0.04	U=-1 R=-1
	1	2

Politique
Π(1,1) = *N*
Π(1,2) = *E*

$\gamma = 1$

$$\begin{bmatrix} -0.9 & 0.8 & 0.1 & 0 \\ 0.1 & -0.9 & 0 & 0.8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U(1,1) \\ U(1,2) \\ U(2,1) \\ U(2,2) \end{bmatrix} = \begin{bmatrix} 0.04 \\ 0.04 \\ -1 \\ 1 \end{bmatrix}$$

Exemple - suite

- Seconde itération – amélioration de politique
  - Π(1,1) = argmax_a\{0.8U(1,2) + 0.1U(1,1) + 0.1U(2,1), N
    0.9U(1,1) + 0.1U(1,2), W
    0.9U(1,1) + 0.1U(2,1), S
    0.8U(2,1) + 0.1U(1,1) + 0.1U(1,2)\} E
    = argmax_a\{0.8x(0.7843) + 0.1x(0.5413) + 0.1x(-1),
    0.9x(0.5413) + 0.1x(0.7843),
    0.9x(0.5413) + 0.1x(-1),
    0.8x(-1) + 0.1x(0.5413) + 0.1x(0.7843)\}
    = argmax_a\{0.5816, 0.5656, 0.3871, -0.6674\}
    = N; pas de mise à jour
  - Π(1,2) = argmax_a\{0.9U(1,2) + 0.1U(2,2), N
    0.9U(1,2) + 0.1U(1,1), W
    0.8U(1,1) + 0.1U(2,2) + 0.1U(1,2), S
    0.8U(2,2) + 0.1U(1,2) + 0.1U(1,1)\} E
    = argmax_a\{0.9x(0.7843)+ 0.1x1,
    0.9x(0.7843) + 0.1x(0.5413),
    0.8x(0.5413) + 0.1x1 + 0.1x(0.7843),
    0.8x1 + 0.1x(0.7843) + 0.1x(0.5413)\}
    = argmax_a\{0.8059, 0.76, 0.6115, 0.9326\}
    = **E; pas de mise à jour**

2	U= 0.7843 R=-0.04	U=+1 R=+1
1	U=0.5413 R=-0.04	U=-1 R=-1
	1	2

Politique
Π(1,1) = *N*
Π(1,2) = *E*

- Pas de changement de politique: fin.
- Politique optimale: N,E.

Itération de politique

- Equations linéaires facile à résoudre.
- Convergence rapide en général; Nombre de politiques fini.
- Optimalité établie.

Itération de valeur ou de politique?

- Itération de politique est plus gourmande en calcul par itération.
- Mais requiert en général moins d'itérations qu'itération de valeur.

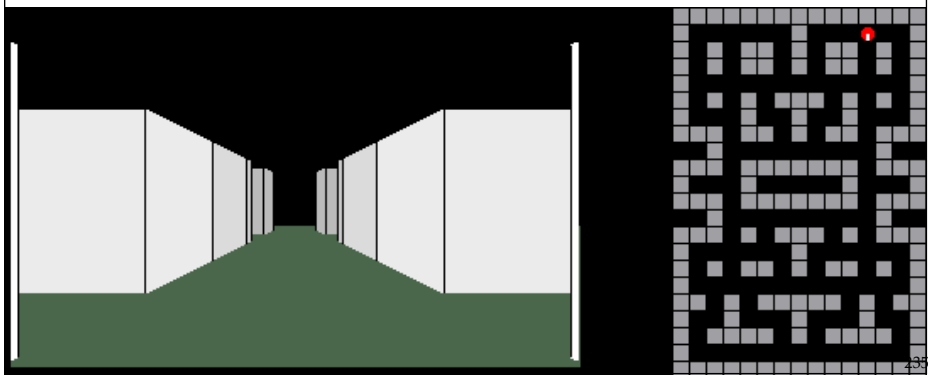
Processus Décisionnels de Markov Partiellement Observables (POMDP-PDMPO)

POMDP

- L'agent, le robot, ne connaît pas parfaitement l'état de l'environnement: décision avec observabilité partielle.
- Contrairement au MDP, après une action donnée, l'agent ne sait pas dans quel état il se trouve.
- *L'observation* de l'état va lui fournir une information incertaine.

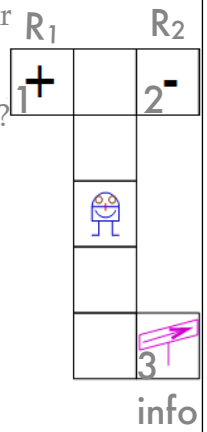
Observabilité partielle

- Les agents ne peuvent pas observer l'état directement.
- Les capteurs fournissent une information partielle et bruitée et l'état.



Valeur de l'Information

- Un POMDP permet de tenir compte de la valeur de l' information.
- Exemple: faut-il aller vers le haut ou vers le bas?
  - l'état ayant la récompense la plus grande n'est pas connu. Quelle action effectuer?
  - Solution dépend de:
    - la différence entre les récompenses des états 1 et 2.
    - le coût de l'information obtenue à l'état 3).
    - la qualité de l'information obtenue à état 3.
- Le POMDP fournit une politique optimale.

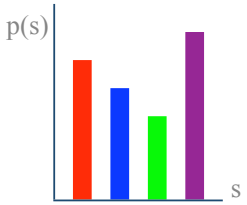


Définition d'un POMDP

- Tuple < ***S***, ***A***, ***T***, ***R***, Ω, ***O*** >
- < ***S***, ***A***, ***T***, ***R*** > **:** comme pour les MDP.
- Ω: Ensemble fini d'observations o que l'agent peut effectuer.
- ***O: S***x***A***→***π***(Ω) fonction d'observation *O(s',a,o)*. Probabilité d'obtenir l'observation *o* après avoir effectué l'action *a* amenant à l'état *s'*.

Croyance

- L'agent représente les états du monde par des états de *croyance*.
- Un état de croyance *(b)* est une distribution de probabilités sur l'ensemble des états *S*.

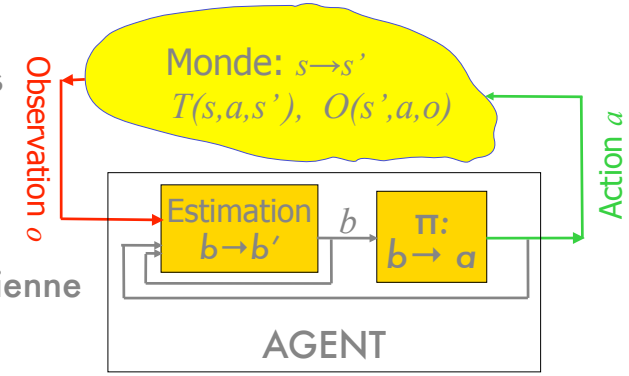


- *b(s)* est la probabilité d'être dans l'état *s* quand l'état de croyance est *b*: $b(s_t) = Pr(s_t = s \mid o_1,a_1,o_2,a_2, \ldots, o_{t-1},a_{t-1},o_t)$
- *b(s')*, avec *s'* successeur de *s*, peut être calculé à partir *b(s)*.
- b_0 est supposé connu.

MDP / POMDP

MDP:
Etat connu après exécution d'une action

POMDP:
Estimation bayesienne de l'état



POMDP

- La décision est prise sur la base de l'espace des croyances qui est une distribution de probabilités *a posteriori* sur les états.
- le POMDP calcule une fonction de valeur sur l'espace des croyances.

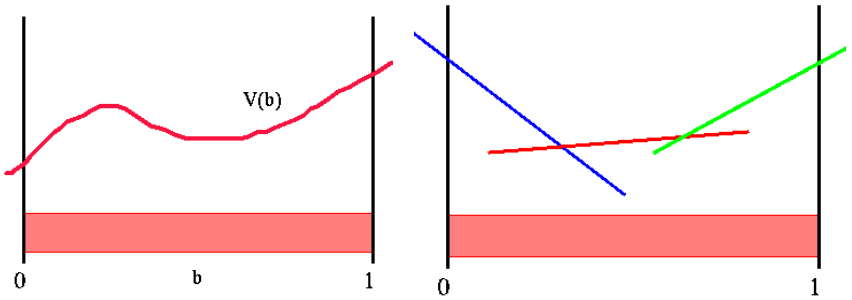
$$V_T(b) = \gamma \max_u \left[r(b,u) + \int V_{T-1}(b') p(b' \mid u, b)\, db' \right]$$

Problèmes

- Chaque croyance est une distribution de probabilités. Donc chaque valeur dans le POMDP est fonction de toute une distribution de probabilités.
- Or les distributions sont continues.
- De plus, l'espace des croyances est grand (fonction du nombre d'états).

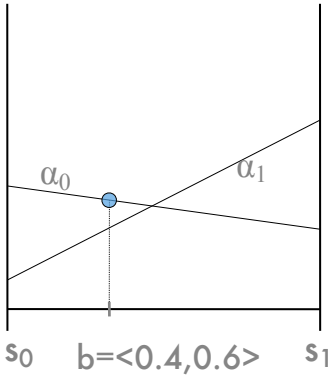
Approche

- Dans le cas d'espaces finis (états, actions, mesure) et d'horizons finis, les fonctions de valeurs peuvent être représentées par des fonctions linéaires par morceaux.

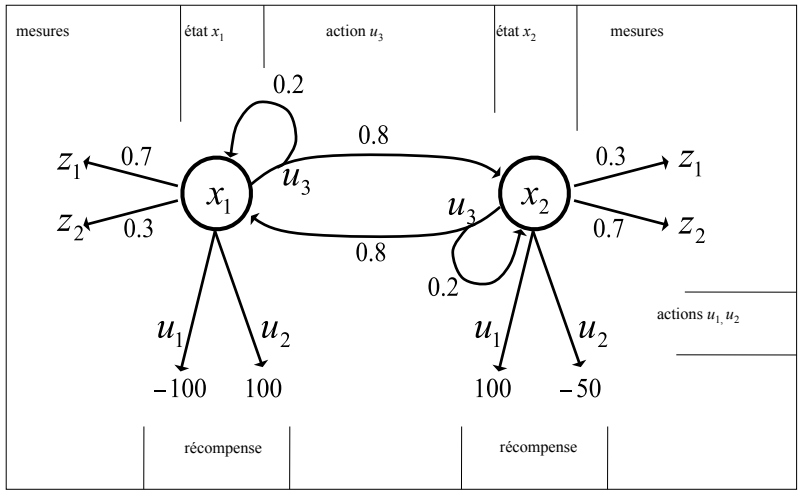


Fonction de Valeur (Sondik 1973)

- Une fonction de valeur V assigne une valeur à un état de croyance b.
- Soit V^* la fonction valeur optimale.
- $V^*(b)$ est la récompense attendue si l'agent effectue les actions optimales à partir de l'état de croyance b.
- V est représenté par un ensemble de vecteurs α (linéarité par morceaux)
- $V(b) = max_\alpha \alpha \cdot b$ (enveloppe supérieure).
- $\alpha \cdot b = \sum_s \alpha(s) b(s)$



Exemple



Exemple

- Les actions u_1 et u_2 sont terminales.
- L'action u_3 est une perception qui peut éventuellement amener à une transition d'état.
- L'horizon T est fini. Soit T=1 et γ=1.

$$
\begin{array}{llll}
r(x_1,u_1) = -100 & & r(x_2,u_1) = +100 \\
r(x_1,u_2) = +100 & & r(x_2,u_2) = -50 \\
r(x_1,u_3) = -1 & & r(x_2,u_3) = -1
\end{array}
$$

$$
\begin{array}{llll}
p(x_1'|x_1,u_3) = 0.2 & & p(x_2'|x_1,u_3) = 0.8 \\
p(x_1'|x_2,u_3) = 0.8 & & p(z_2'|x_2,u_3) = 0.2
\end{array}
$$

$$
\begin{array}{llll}
p(z_1|x_1) = 0.7 & & p(z_2|x_1) = 0.3 \\
p(z_1|x_2) = 0.3 & & p(z_2|x_2) = 0.7
\end{array}
$$

Récompense

- Dans les MDPs, la récompense dépend de l'état.
- Dans les POMDPs, l'état n'est pas exactement connu.
- La récompense attendue est calculée en intégrant sur l'ensemble des états:

$$
\begin{aligned}
r(b,u) &= E_x[r(x,u)] \\
&= \int r(x,u)p(x)\,dx \\
&= p_1\, r(x_1,u) + p_2\, r(x_2,u)
\end{aligned}
$$

Dans l'exemple

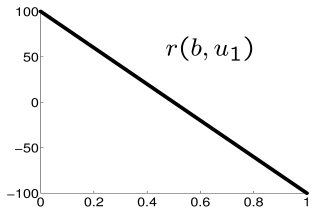
- Si l'agent est certain d'être dans l'état x_1 et qu'il exécute l'action u_1, il reçoit une récompense de -100
- S'il sait qu'il est dans l'état x_2 et qu'il exécute l'action u_1, il reçoit une récompense de +100.
- Dans une situation intermédiaire, la récompense attendue est la combinaison linéaire des valeurs extrêmes pondérées par leur probabilités.

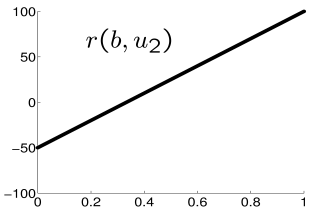
$$\begin{aligned} r(b,u_1) &= -100\,p_1 + 100\,p_2 \\ &= -100\,p_1 + 100\,(1-p_1) \end{aligned}$$

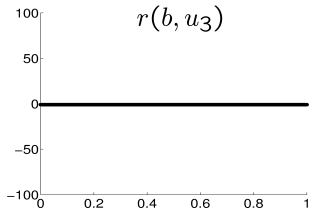
$$r(b,u_2) = 100\,p_1 - 50\,(1-p_1)$$

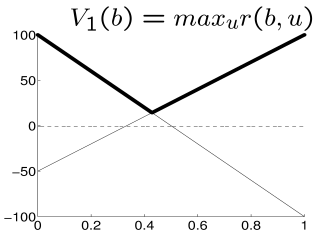
$$r(b,u_3) = -1$$

Dans l'exemple









Politique optimale

- Puisqu'on a un POMDP fini avec T=1, on utilise $V_1(b)$ pour déterminer la politique optimale.
- Ici, la politique optimale pour T=1 est

$$\pi_1(b) = \begin{cases} u_1 & \text{if } p_1 \leq \frac{3}{7} \\ u_2 & \text{if } p_1 > \frac{3}{7} \end{cases}$$

- Enveloppe supérieure dans le diagramme

Linéarité par morceaux et convexité

- La fonction valeur résultante $V_1(b)$ est le maximum des trois fonctions en chaque point:

$$\begin{aligned} V_1(b) &= \max_u r(b,u) \\ &= \max \begin{Bmatrix} -100\,p_1 & +100\,(1-p_1) \\ 100\,p_1 & -50\,(1-p_1) \\ -1 & \end{Bmatrix} \end{aligned}$$

- Elle est linéaire par morceaux et convexe.

Elagage

- En regardant $V_1(b)$, on remarque que seules les deux premières composantes contribuent.
- La troisième composante peut être élaguée de $V_1(b)$.

$$V_1(b) = \max \left\{ \begin{array}{cc} -100\, p_1 & +100\,(1-p_1) \\ 100\, p_1 & -50\,(1-p_1) \end{array} \right\}$$

Horizon

- Si l'on augmente l'horizon à T=2, l'agent peut effectuer l'action u_3.
- Supposons que la perception est z_1 avec $p(z_1 \mid x_1)=0.7$ et $p(z_1 \mid x_2)=0.3$.
- Etant donnée l'observation z_1 la valeur $V_1(b \mid z_1)$ peut être mise à jour par Bayes :

$$V_1(b \mid z_1) = \max \left\{ \begin{array}{cc} -100 \cdot \frac{0.7\, p_1}{p(z_1)} & +100 \cdot \frac{0.3\,(1-p_1)}{p(z_1)} \\ 100 \cdot \frac{0.7\, p_1}{p(z_1)} & -50 \cdot \frac{0.3\,(1-p_1)}{p(z_1)} \end{array} \right\}$$

$$= \frac{1}{p(z_1)} \max \left\{ \begin{array}{cc} -70\, p_1 & +30\,(1-p_1) \\ 70\, p_1 & -15\,(1-p_1) \end{array} \right\}$$

Valeur espérée après observation

- D'abord mise à jour de la croyance attendue

$$
\begin{aligned}
\bar{V}_1(b) &= E_z[V_1(b \mid z)] \\
&= \sum_{i=1}^{2} p(z_i)\, V_1(b \mid z_i) \\
&= \max \left\{ \begin{array}{cc} -70\, p_1 & +30\,(1-p_1) \\ 70\, p_1 & -15\,(1-p_1) \end{array} \right\} \\
&\quad + \max \left\{ \begin{array}{cc} -30\, p_1 & +70\,(1-p_1) \\ 30\, p_1 & -35\,(1-p_1) \end{array} \right\}
\end{aligned}
$$

Fonction valeur

- Les quatre combinaisons possibles produisent une fonction qui peut être simplifiée et élaguée à son tour.

$$
\begin{aligned}
\bar{V}_1(b) &= \max \left\{ \begin{array}{cccc} -70\, p_1 & +30\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ -70\, p_1 & +30\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \end{array} \right\} \\
&= \max \left\{ \begin{array}{cc} -100\, p_1 & +100\,(1-p_1) \\ +40\, p_1 & +55\,(1-p_1) \\ +100\, p_1 & -50\,(1-p_1) \end{array} \right\}
\end{aligned}
$$

Transitions d'état (Prédiction)

- Quand l'agent décide u_3 son état est potentiellement modifié.
- En calculant la fonction de valeur, ce changement doit être pris en compte.

$$\begin{aligned} p_1' &= E_x[p(x_1 \mid x, u_3)] \\ &= \sum_{i=1}^{2} p(x_1 \mid x_i, u_3) p_i \\ &= 0.2p_1 + 0.8(1 - p_1) \\ &= 0.8 - 0.6p_1 \end{aligned}$$

Fonction Valeur après exécution de u_3

- En tenant compte des transitions d'états on obtient finalement:

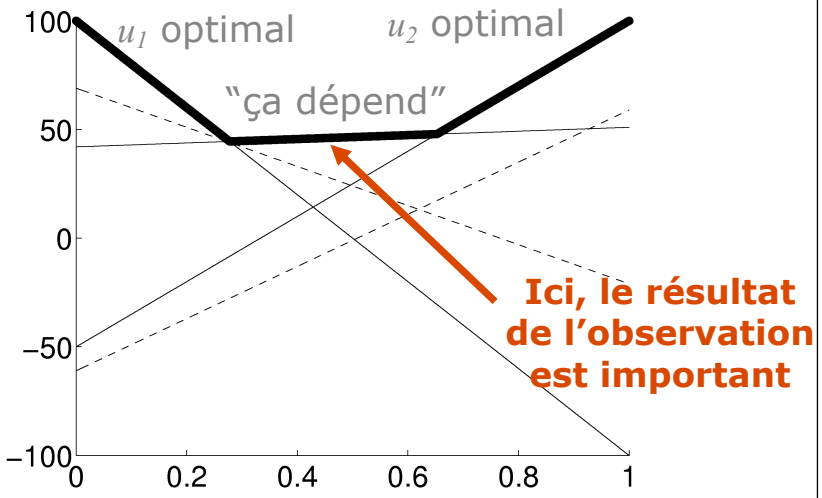
$$\bar{V}_1(b \mid u_3) = \max \left\{ \begin{array}{rr} 60\, p_1 & -60\,(1 - p_1) \\ 52\, p_1 & +43\,(1 - p_1) \\ -20\, p_1 & +70\,(1 - p_1) \end{array} \right\}$$

Fonction de valeur pour T=2

- L'agent peut exécuter directement u_1 ou u_2, ou bien exécuter d'abord u_3 puis u_1 ou u_2, on obtient (après élagage):

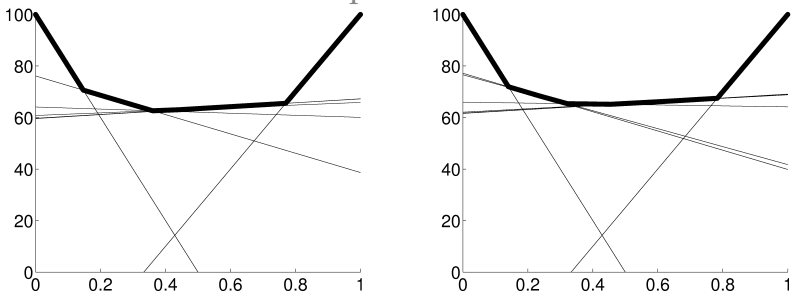
$$\bar{V}_2(b) = \max \left\{ \begin{array}{rr} -100\, p_1 & +100\,(1-p_1) \\ 100\, p_1 & -50\,(1-p_1) \\ 51\, p_1 & +42\,(1-p_1) \end{array} \right\}$$

Représentation Graphique de $V_2(b)$



Horizons lointains et élagage

- Nous avons effectué une itération entière sur l'espace des croyances.
- Ceci peut être appliqué récursivement.
- Fonctions de valeur pour T=10, T=20 :



Importance de l'élagage

- Chaque mise à jour introduit des composantes linéaires supplémentaires à V.
- Chaque observation élève le nombre de composantes linéaires au carré.
- Une fonction de valeur non élaguée pour T=20 inclut plus de 10^{547864} fonctions linéaires.
- A T=30 on a $10^{561012337}$ fonctions linéaires.
- Avec élagage, le nombre de composantes linéaires à T=20 n'est que de 12.
- L'explosion combinatoire du nombre de composantes linéaires dans la fonction valeur est la raison principale qui rend les POMDP non pratiques dans les cas réels.

Conclusions sur les POMDPs

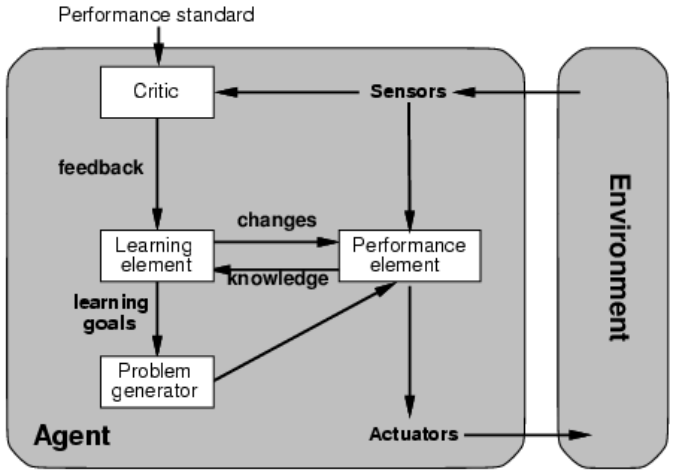
- Les POMDPs calculent l'action optimale dans des domaines stochastiques partiellement observables.
- Si l'horizon est fini, les fonctions de valeur sont linéaires par morceaux et convexes.
- A chaque itération le nombre de composantes linéaires croît exponentiellement.
- Les POMDPs ne sont utilisés que pour des espaces de petite taille.

Apprentissage par renforcement

Apprentissage

- Amélioration des capacités de l'agent sur la base de son expérience
- Apprentissage de l'environnement, de concepts, d'actions
- Apprentissage et adaptation

Agent apprenant



Apprentissage supervisé

Objectif: Réduire l'erreur (Sortie désirée - Sortie)

Apprentissage supervisé

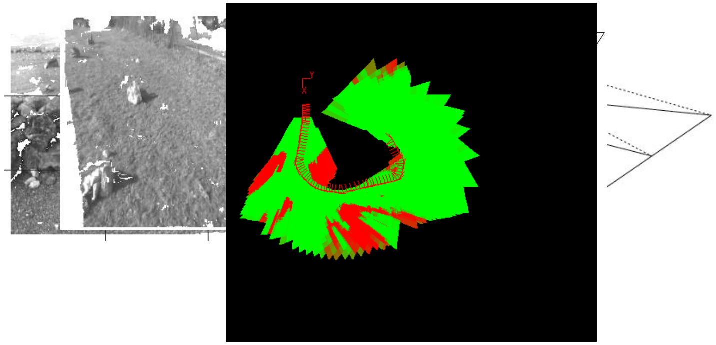
- A partir d'exemples, d'observations
- l'environnement (ou le tuteur) fournit un nombre suffisant de couples entrée/ sortie (pas toujours possible).
- Apprentissage de distribution de probabilités.

Exemple:
Navigation en environnement naturel

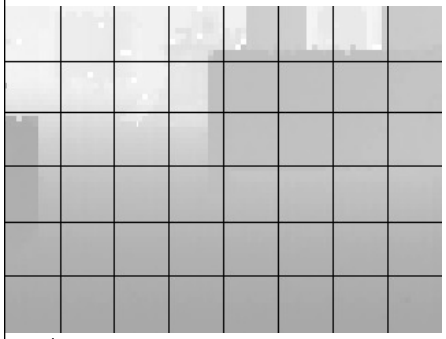
Modélisation incrémentale

Terrain plutôt plat avec obstacles

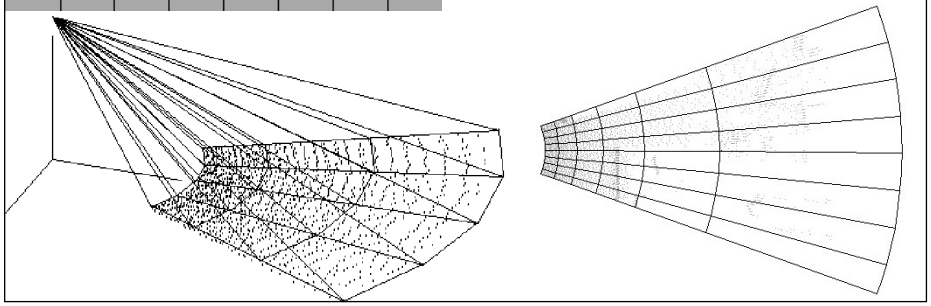
4 catégories: zones plates, accidentées mais franchissables, obstacles, indéterminées



Discrétisation de la scène perçue



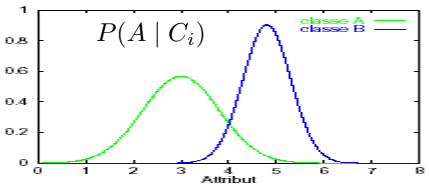
- Image de points 3D (Laser, Stereo)
- Grille régulière dans l'image
- Projection au sol



Classification par apprentissage supervisé

- Choix d'attributs caractéristiques pour chaque cellule:
  - Densité des points
  - Différence de l'élévation et variance
  - Orientation moyenne de la normale et variance

- Classification supervisée bayésienne
  - 4 classes: obstacle, accidenté, plat, inconnu
  - Elaboration des associations Classes-Attributs pour construire les *P(A | Ci)* à partir d'exemples.



Utilisation en ligne

$$P(C_i \mid A) = \frac{P(A \mid C_i)P(C_i)}{P(A)}$$

$$P(A) = \sum_i P(A \mid C_i)P(C_i)$$

Modèle du terrain

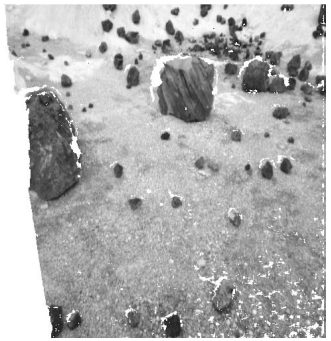
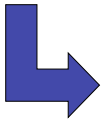
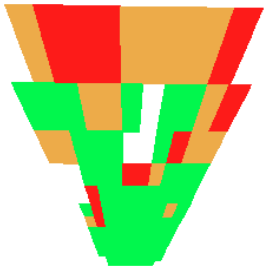


Image stéréo

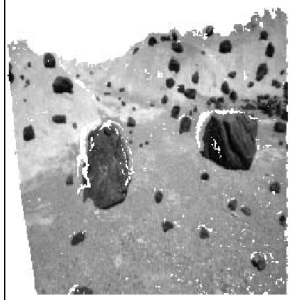


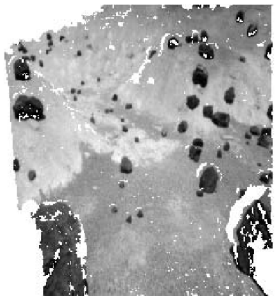
- Classification

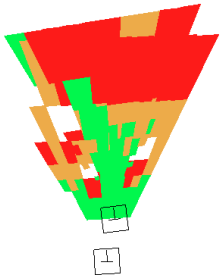


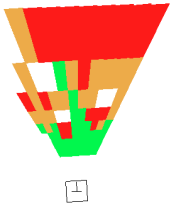
- Reprojection dans l'image

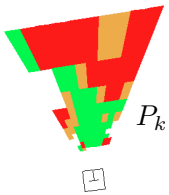
Fusion après nouvelle observation





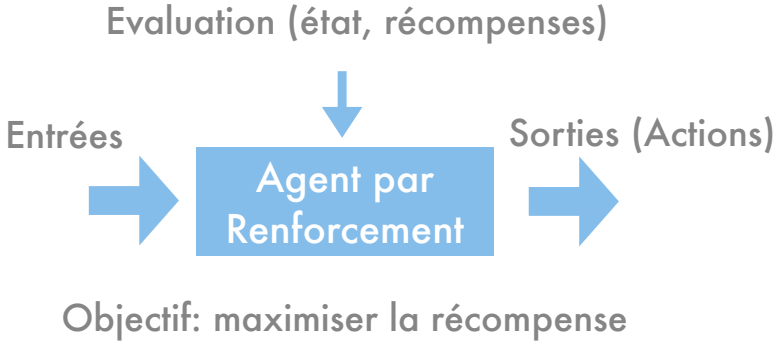






$$P_k(C_i) = \frac{P_{k_{\text{local}}}(C_i) P_{k-1}(C_i)}{\sum_{C_i} P_{k_{\text{local}}}(C_i) P_{k-1}(C_i)}$$

Apprentissage par renforcement

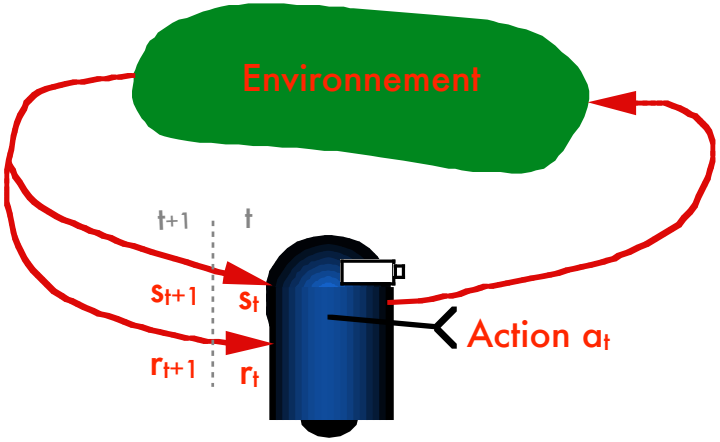


Objectif: maximiser la récompense

RL - Principes

- L'agent ne sait pas *a priori* quelles actions exécuter
- Sélection des actions par essai-erreur.
- Récompenses. Gains à court terme v.s. gains à long terme
  - Les récompenses peuvent être retardées (ne pas suivre immédiatement l'action qui les a provoquées).
- Exploitation des actions apprises v.s. exploration de nouvelles actions.

RL - Illustration



App. Supervisé vs.RL

- Exemple: jeu d'échec
  - Apprentissage supervisé: couples {configuration de jeu/meilleur mouvement}
  - Apprentissage par renforcement
    - essais
    - observation de l'état de l'échiquier et du jeu de l'adversaire
    - Récompense (gain ou perte de pièce, mat, ...)

RL - formulation

- Contexte markovien.
- Agent représenté par un MDP {S,A,T,R}.
  - *Table de transitions non connue a priori*.
  - *Récompenses: issues de l'interaction avec l'environnement; non connues a priori.*
- Fonction de valeur: prédiction de la récompense.
- Politique: détermine l'action à effectuer dans chaque état.
- Modèle de l'environnement: lien entre actions et récompenses.

RL: Apprentissage

- Exécution d'une action: résultat (état, récompense).
- Apprendre le modèle de transition: T(s,a,s') et les récompenses r(s') pour tous les états en exécutant les actions jusqu'à atteindre le but.
- Choisir à chaque étape l'action de manière à maximiser les récompenses (utilités) apprises.
- Les utilités à apprendre sont définies par la politique optimale:

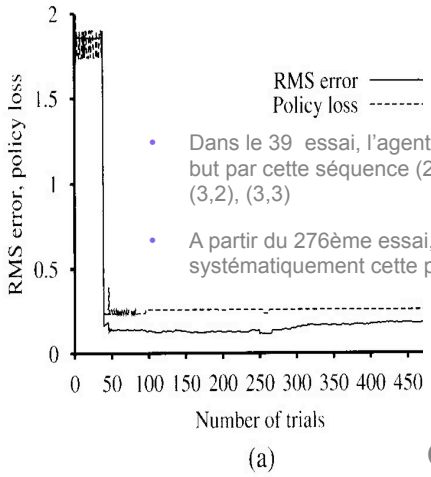
$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

- Résolution par itération de valeur ou de politique

Démarche

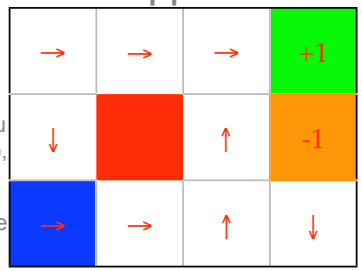
- A chaque étape: choix de l'action maximisant l'utilité (agent glouton).
- Si itération de valeur: $a = \text{argmax}_a(\sum_{s'} T(s,a,s')U(s))$
- Si itération de politique, action déjà définie
- **MAIS**: ces valeurs sont issues d'un calcul partiel et d'une séquence d'actions qui a inclut une exécution aléatoire.
- Elle ne représentent pas la politique optimale.

Non optimalité

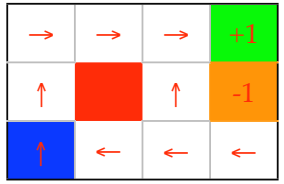


- Dans le 39 essai, l'agent arrive au but par cette séquence (2,1), (3,1), (3,2), (3,3)
- A partir du 276ème essai, il adopte systématiquement cette politique

Appris



Optimal



Exploration/exploitation

- **Combiner**
  - **Exploitation**: maximise la récompense globale dans l'état d'estimation actuel des utilités.
  - **Exploration**: pour éviter l'application systématique de politiques sous-optimales. Tirage aléatoire d'une action pour explorer d'autres voies.

Q-Learning

- Apprendre les représentations actions-valeurs au lieu des utilités
- Q(s,a) : Valeur résultant de l'exécution de l'action a dans l'état s
  - $U(s) = \max_a Q(s,a)$
- Les contraintes sur les valeurs doivent être respectées quand les Q-valeurs sont correctes
  - $Q(s,a) = R(s) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q(s',a')$

Q-Learning

- A chaque étape
  - Sélectionner une action dans l'état s
  - Observer s' – état résultant
  - Mettre à jour Q(a,s)
    - Choisir la valeur maximale

Q-Learning

- Choix de l'action:
  - Choix aléatoire
  - fraction décroissante dans le temps
  - a ← $\text{argmax}_{a'} f(Q(s',a'), N_{sa}[a',s'])$
    - $N_{sa}[a',s']$ nombre de fois que l'action a été choisie.
    - Cette information est à apprendre
  - a ← $\text{argmax}_{a'} Q(s',a')$
    - Fournit la plus grande valeur de Q

Q-Learning

- Calculer quand l'action a exécutée en s, conduisant à s': $Q(s,a) \leftarrow Q(s,a)+\alpha(R(s)+\gamma \max_{a'} Q(s',a') - Q(s,a))$
- Mettre à jour selon la valeur maximale reçue à l'étape suivante.
  - α: taux d'apprentissage.

- http://homepages.laas.fr/raja/Cours/P8-M2/

Examen

- Se mettre par binômes ou monômes. Chaque binôme choisit un article parmi ceux proposés sur http://homepages.laas.fr/raja/Cours/Articles/
- Si des binômes différents choisissent le même article, ils ne doivent pas communiquer.
- Chaque binôme rédige 4 à 5 pages (en français ou en anglais) selon le plan donné. Les coupé-collés pris dans l'article, (sauf les figures éventuellement ou les algorithmes), sont interdits. Les traductions automatiques (google...) seront éliminatoires.

Plan général du devoir

- 1. Objectif de l'article
- 2. Contexte général (motivation des auteurs: pourquoi ce sujet a été traité? Ceci se trouve normalement dans l'introduction de l'article lui-même, mais vous pouvez élaborer là-dessus, faire une recherche internet.
- 3. Méthode suivie par les auteurs et son explication. S'il y a un algorithme, en expliquer le déroulement.
- 4. Montrer les résultats obtenus.
- 5. Donner votre avis personnel sur l'article, au delà des conclusions des auteurs: critiques et propositions d'évolution.

- M'envoyer votre travail par mail sou sla forme d'un ficher **pdf** (Word accepté). Intituler les fichiers de la manière suivante: NOM_R2_MIME2011. Dans le cas d'un binôme, NOM devient NOM1_NOM2.
- En sujet du mail: Examen M2_MIME2011
- Indiquer le titre de l'article et ses auteurs ainsi que votre nom et numéro en première page du document.