Intelligence artificielle

Théorie des décisions

Motivation

- Très souvent, un agent ne peut pas déterminer avec certitude quel sera l'effet de ses actions
- Par contre, on peut estimer la probabilité de chaque situation pouvant résulter d'une action
- On peut aussi évaluer dans quelle mesure chaque état résultant est utile ou désirable
- Alors, comment choisir la meilleure action?

Principe du maximum d'utilité espérée

- Soit :
  - *U(S)* une fonction qui retourne la valeur d'utilité de l'état *S*
  - *Résultat*$_i$(*A*) le *i*ème état résultat possible de l'action A
  - *E* l'ensemble des informations sur l'état actuel dont l'agent dispose
- L'action choisie devrait être celle qui maximise EU(A|E) définie ainsi:

$$EU(A|E) = \sum_i P(Résultat_i(A)|faire(A),E) \times U(Résultat_i(A))$$

Loterie

- Une loterie L est un ensemble d'états C_1, C_2, ..., C_n pouvant se produire avec des probabilités p_1, p_2, ... p_n, respectivement;
- On écrira L = [p_1,C_1; p_2,C_2 ; ... p_n,C_n]
- Chaque état C_i peut lui-même être une loterie
- On écrira
  - A > B si l'agent préfère A à B
  - A ~ B si l'agent est indifférent entre A et B
  - A >~ B si l'agent préfère A ou est indifférent

Conditions sur la fonction d'utilité

- Pour que le principe de maximum d'utilité espérée nous assure un comportement rationnel, certaines conditions doivent être respectées par la fonction d'utilité:
  - A > B ou B > A ou A ~ B
  - Si A > B et B > C, alors A > C
  - Si A > B > C alors il existe une probabilité *p* telle que [*p*,A; *1-p*,C] ~ B
  - Si A ~ B, alors ils sont interchangeable dans une loterie
  - Soit A > B et deux loteries impliquant seulement A et B. L'agent accordera la préférence à celle où A a la plus forte probabilité (et vice versa)
  - [*p*,A; *1-p*,[*q*,B; *1-q*,C]] ~ [*p*,A; (*1-p*)*q*,B; (*1-p*)(*1-q*),C]

Comment déterminer la valeur d'utilité?

- La valeur absolue n'est pas importante
- Ce qui importe c'est la valeur d'un état comparé aux autres
- Exemple: valeur monétaire
- On choisit quoi entre [0.5, 0$; 0.5, 2500$] et 1000$?
- Si U(e) = montant, alors on choisit la loterie
- Si U(e) = log(avoir total), alors on décline la loterie si on est pauvre
- S'il nous manque exactement 2500$ pour faire l'achat de l'objet de nos rêve, on pourrait avoir U(e) = 1 si montant=2500$, 0 sinon. Dans ce cas, on prendra la loterie

Fonction d'utilité multi-attributs

- $U(x_1,...,x_n) = f\,[f_1(x_1),...,f_n(x_n)]$
- Un cas particulier souvent utilisé: *f* est la somme des valeurs (si les préférences sont indépendantes)

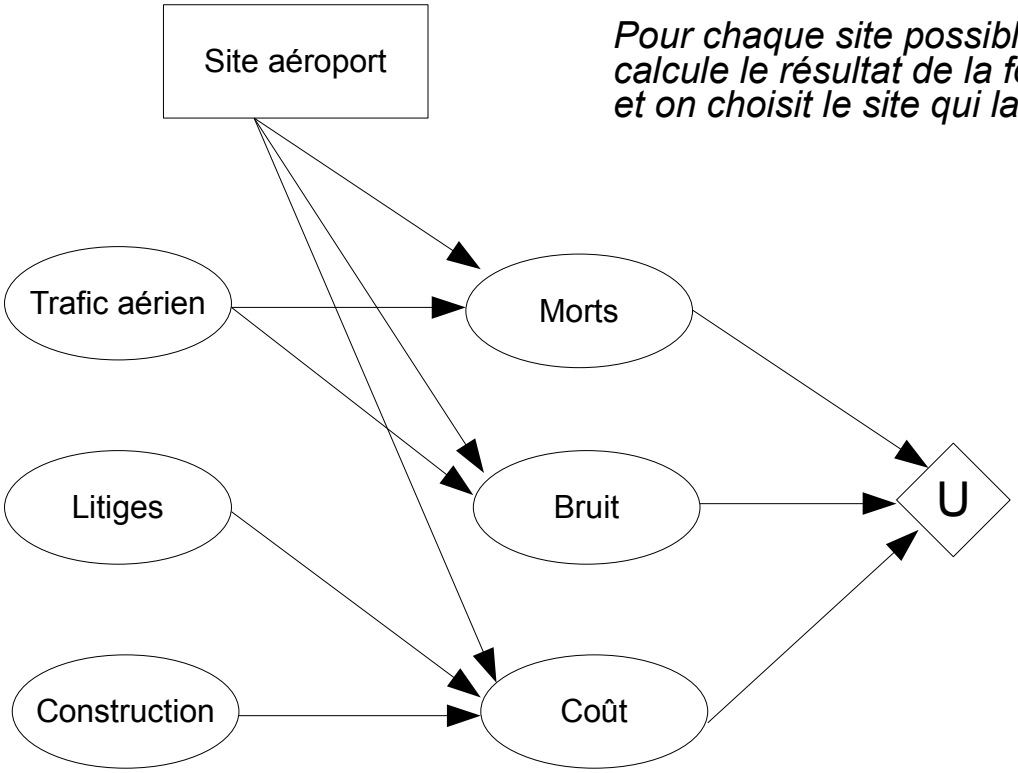
Réseaux de décision

- Il s'agit essentiellement de réseaux bayesiens
- On distingue trois types de noeuds:
  - *noeuds réguliers,* représentant les variables aléatoires
  - *noeuds de décision,* représentant des variables dont la valeur doit être sélectionnée parmi toutes celles possibles
  - *noeuds d'utilité*, représentant la valeur d'utilité pour un ensemble de variables

Évaluation d'un réseau de décision

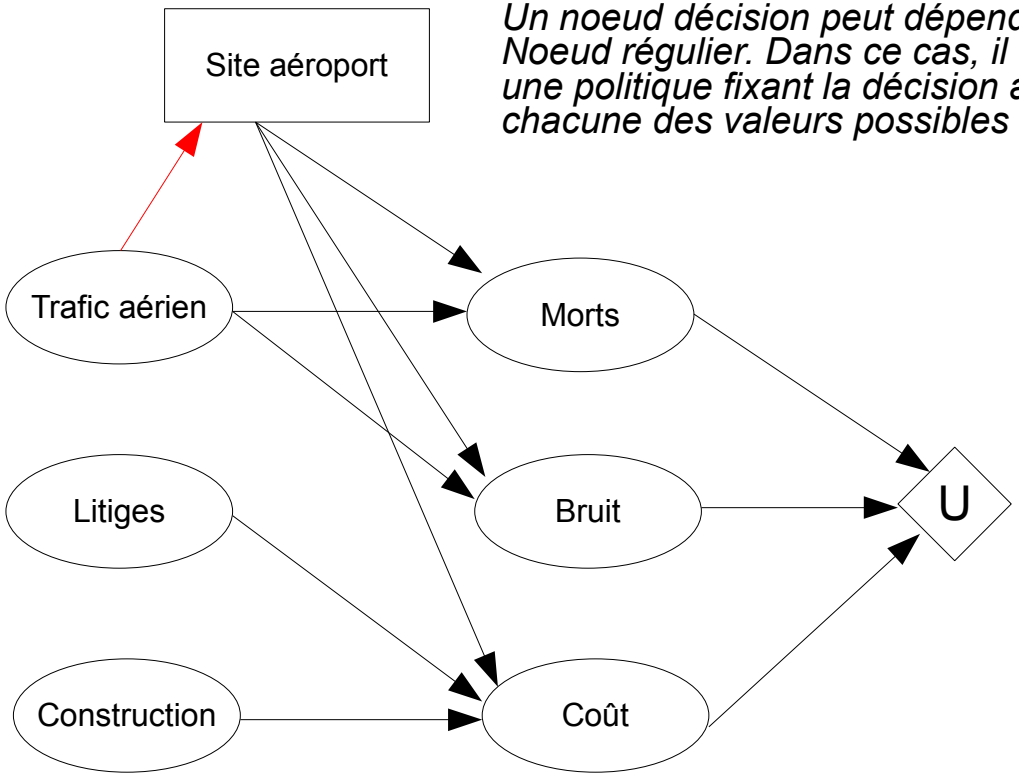
1. Établir les valeurs connues parmi les noeuds du réseau
2. Pour chaque valeur possible du noeud de décision:
   a) Fixer le noeud à cette valeur
   b) Calculer les probabilités de tous les noeuds reliés au noeud d'utilité
   c) Calculer la valeur d'utilité
3. Choisir l'action ayant la valeur d'utilité la plus élevée

Exemple de réseau de décision



Pour chaque site possible, on calcule le résultat de la fonction d'utilité et on choisit le site qui la maximise.

Exemple de réseau de décision



Un noeud décision peut dépendre d'un Noeud régulier. Dans ce cas, il faut spécifier une politique fixant la décision appliqué pour chacune des valeurs possibles

Décisions complexes

- Jusqu'à maintenant, on a supposé que l'agent n'a qu'une seule décision à prendre
- Que fait-on lorsque plusieurs actions consécutives sont nécessaires pour atteindre le but visé?
- Nous verrons que, dans ce cas, l'agent doit avoir une politique pour choisir la meilleure action à chaque état

Processus de décision Markovien

- Modèle de transition: *T(s,a,s')* représente la probabilité d'obtenir l'état *s'* si on exécute l'action *a* à l'état *s*
- État initial: S_0
- Fonction de récompense: *R*(*s*)
- La solution est une *politique* $\pi(s)$, qui spécifie l'action recommandée pour chaque état *s*

Politique optimale

- Une politique optimale, notée $\pi^*(s)$, est une politique qui retourne la valeur la plus élevée pour l'utilité espérée
- Si l'horizon n'est pas infini, elle sera non-stationnaire (c'est-à-dire qu'elle ne retournera pas toujours la même action pour le même état)

Comment calculer l'utilité?

- On supposera que la préférence est stationnaire (si l'agent préfère la séquence $[s_0,s_1,s_2,...]$ à la séquence $[s_0,s_1',s_2',...]$, alors il préfère $[s_1,s_2,...]$ à $[s_1',s_2',...]$)
- Dans ce cas, on a une seule fonction d'utilité raisonnable:

$$U([s_0,s_1,s_2,...]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + ...$$

où γ est compris entre 0 et 1

Comment choisir entre différentes politiques?

- Une politique optimale π^* est une politique π qui maximise la valeur espérée suivante:

$$E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi\right]$$

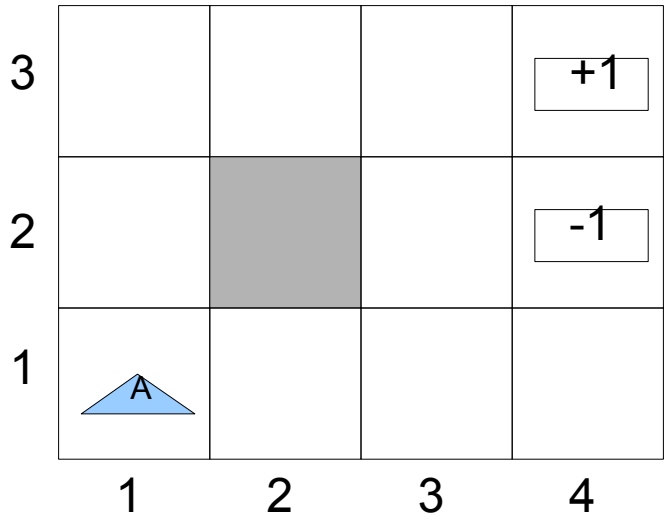
- Par le principe de l'utilité maximale espérée, on déduit que π^* est la politique suivante:

$$\pi^*(s) = \operatorname{argmax}_a \sum_{s'} T(s,a,s')\, U(s')$$

- On a aussi:

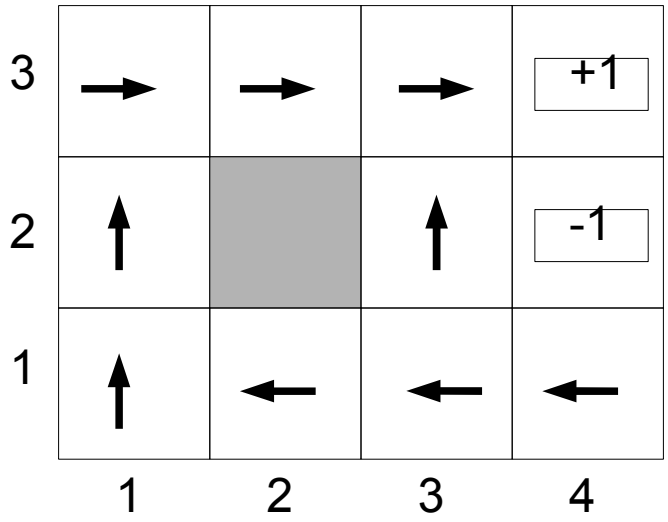
$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')\, U(s')$$

Exemple



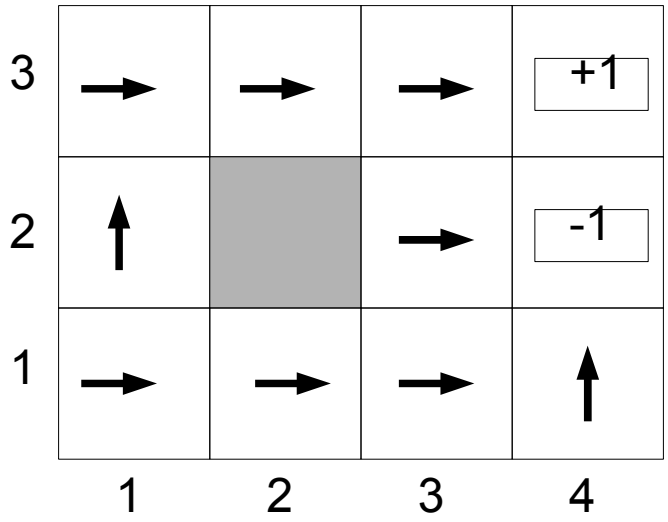
Récompense de -0,04 pour chaque case non finale
Probabilité d'avancer à la case suivante: 0,8
Probabilité de se déplacer sur une case latérale: 0,1

Exemple – politique optimale



Récompense de -0,04 pour chaque case non finale
Probabilité d'avancer à la case suivante: 0,8
Probabilité de se déplacer sur une case latérale: 0,1

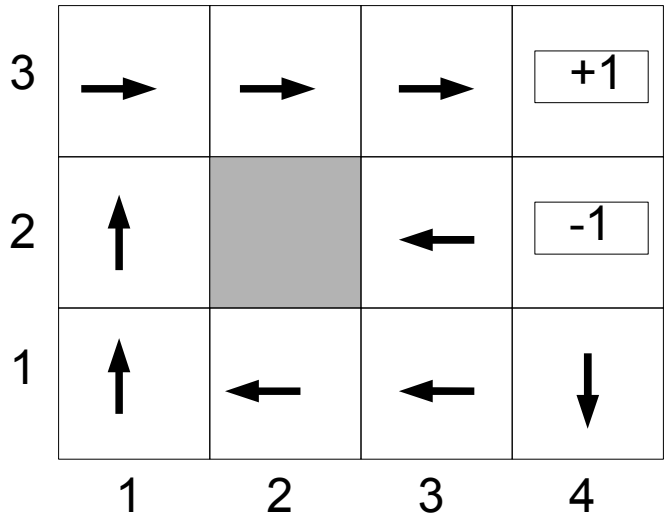
Exemple – politique optimale



L'agent se précipite le plus vite possible vers l'état final le plus proche, même s'il est le moins intéressant

Récompense inférieure à -1,6284 pour chaque case non finale
Probabilité d'avancer à la case suivante: 0,8
Probabilité de se déplacer sur une case latérale: 0,1

Exemple – politique optimale



L'agent ne prend aucun risque

Récompense entre -0,0221 et 0 pour chaque case non finale
Probabilité d'avancer à la case suivante: 0,8
Probabilité de se déplacer sur une case latérale: 0,1

Algorithme – itération de la valeur

fonction ITERATION-VALEUR(*pdm,*ε) **retourne** fonction d'utilité

entrées: *pdm,* un processus de décision markovien

ε, erreur maximale permise

U' est une fonction d'utilité initiale telle que U'(x) = 0

S est l'ensemble des états de *pdm*

répéter:

$U \leftarrow U'$

$\delta \leftarrow 0$

pour chaque $s \in S$ faire:

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

si $|U'[s] - U[s]| > \delta$ **alors** $\delta \leftarrow |U'[s] - U[s]|$

jusqu'à $\delta < \varepsilon(1-\gamma)/\gamma$

retourner U

Itération de la valeur -simulation

3	0,0	0,0	0,0	0,0
2	0,0		0,0	0,0
1	0,0	0,0	0.0	0,0
	1	2	3	4

$\gamma = 1$

Itération de la valeur -simulation

meilleure action: *droite*

U = -0,04 + 0,8*1 -

-0,1*0,04 – 0,1*0,04

= 0,752

meilleure action: *gauche*

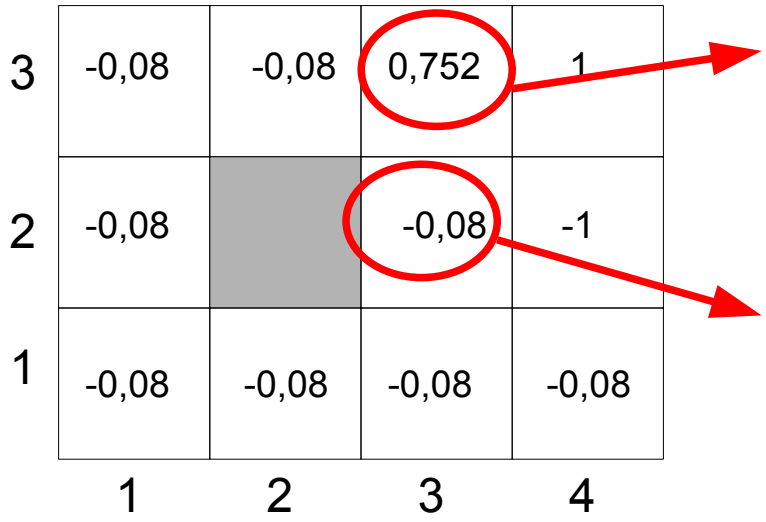
U = -0,04 – 0,8*0,04

-0,1*0,04 – 0,1*0,04

= -0,08

$\gamma = 1$

Itération de la valeur -simulation



meilleure action: *droite*

U = -0,04 + 0,8*1 -

+0,1*0,752 – 0,1*0,08

= 0,827

meilleure action: *haut*

U = -0,04 + 0,8*0,752

-0,1*1 – 0,1*0,08

= 0,454

$\gamma = 1$

Itération de la valeur -simulation

3	-0,12	0,546	0,827	1
2	-0,12		0,454	-1
1	-0,12	-0,12	-0,12	-0,12
	1	2	3	4

$\gamma = 1$

Itération de la valeur -simulation

3	0,372	0,731	0,888	1
2	-0,16		0,567	-1
1	-0,16	-0,16	0,299	-0,16
	1	2	3	4

$\gamma = 1$

Itération de la valeur -simulation

On continue tant que la plus grande variation de U(s) est supérieure au seuil minimal...

Itération de la valeur -simulation

Valeurs finales:

3	0,812	0,868	0,918	1
2	0,762		0,660	-1
1	0,705	0,655	0,611	0,388
	1	2	3	4

$\gamma = 1$

Algorithme – itération de la politique

fonction ITERATION-POLITIQUE(*pdm*, ε) **retourne** fonction d'utilité

entrées: *pdm,* un processus de décision markovien

ε, erreur maximale permise

π est une politique initialisée aléatoirement

S est l'ensemble des états de *pdm*

répéter:

$U \leftarrow$ EVALUATION-POLITIQUE(π, U, *pdm*)

inchangé $\leftarrow$ *true*

pour chaque s $\in$ S faire:

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

$\pi[s] = \text{argmax}_a \sum T(s,a,s')\, U[s']$

inchangé = *false*

jusqu'à *inchangé*

retourner π

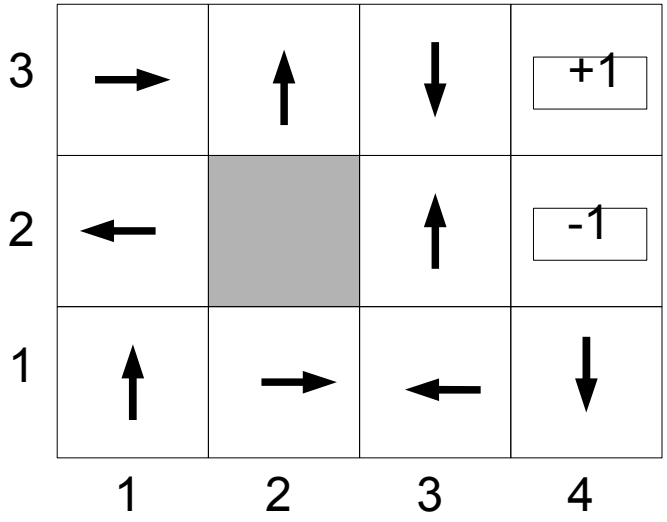
Algorithme – itération de la politique

- En principe l'évaluation d'une politique exige la résolution d'une série d'équations linéaires
- On peut simplifier en itérant *k* fois la mise à jour suivante:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} T(s,\pi_i(s),s') \, U_i[s']$$

Itération de la politique - simulation

État initial



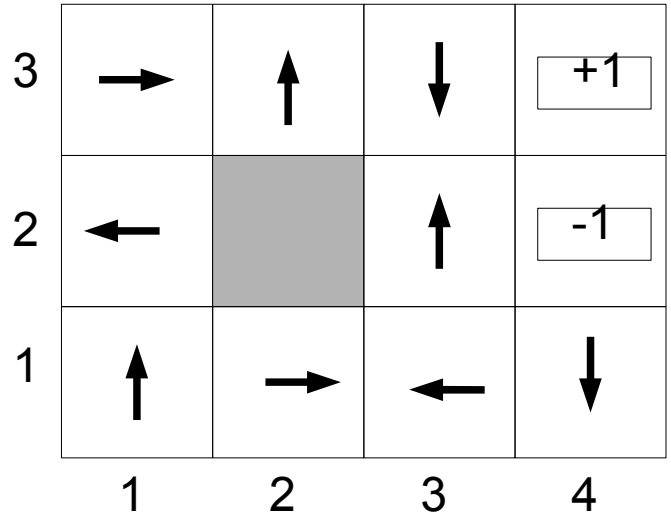
3	0,0	0,0	0,0	0,0
2	0,0		0,0	0,0
1	0,0	0,0	0.0	0,0
	1	2	3	4

Politique fixée au hasard

$\gamma = 1$

Itération de la politique - simulation

Évaluation de la politique

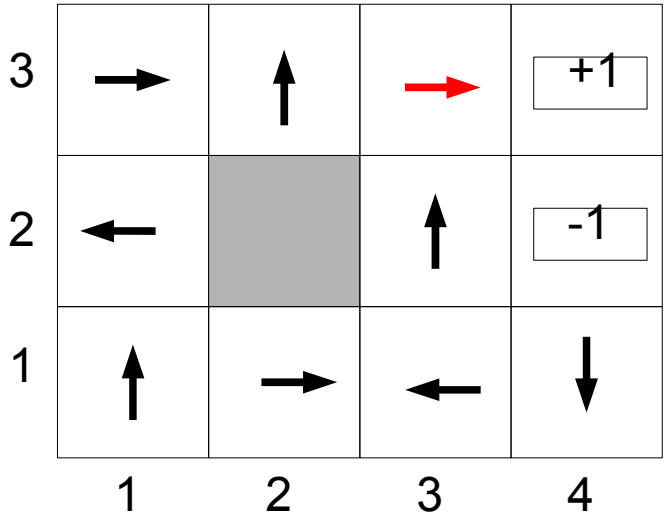


3	-1,21	-1,10	-0,60	+1
2	-1,63		-0,69	-1
1	-1,67	-1,54	-1,49	-1,88
	1	2	3	4

$\gamma = 1$

Itération de la politique - simulation

Modification de la politique

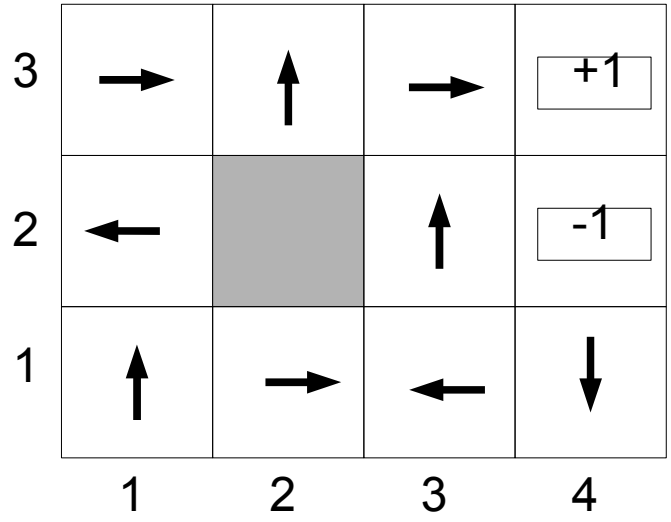


3	-1,21	-1,10	-0,60	+1
2	-1,63		-0,69	-1
1	-1,67	-1,54	-1,49	-1,88
	1	2	3	4

$\gamma = 1$

Itération de la politique - simulation

Évaluation de la politique

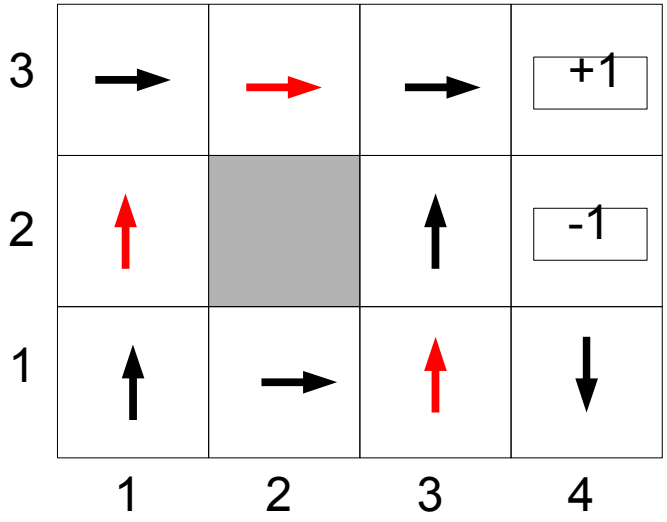


3	0,30	0,41	0,92	+1
2	-0,15		0,66	-1
1	-0,20	-0,19	-0,14	-0,54
	1	2	3	4

$\gamma = 1$

Itération de la politique - simulation

Modification de la politique

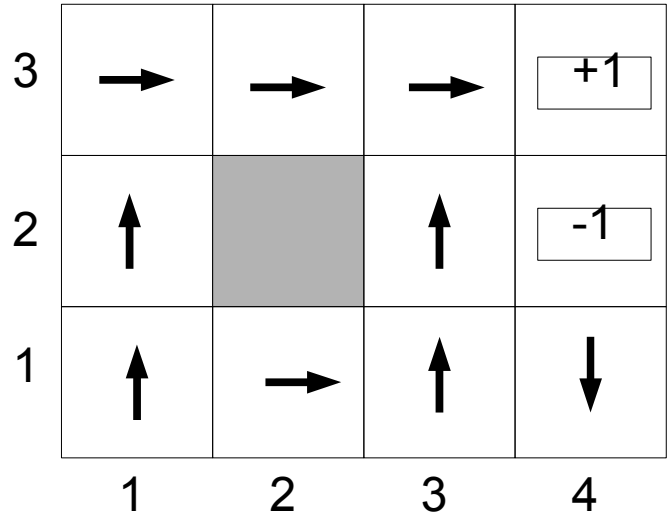


3	0,30	0,41	0,92	+1
2	-0,15		0,66	-1
1	-0,20	-0,19	-0,14	-0,54
	1	2	3	4

$\gamma = 1$

Itération de la politique - simulation

Évaluation de la politique



3	0,811	0,87	0,92	+1
2	0,761		0,66	-1
1	0.688	0,5	0,55	0,15
	1	2	3	4

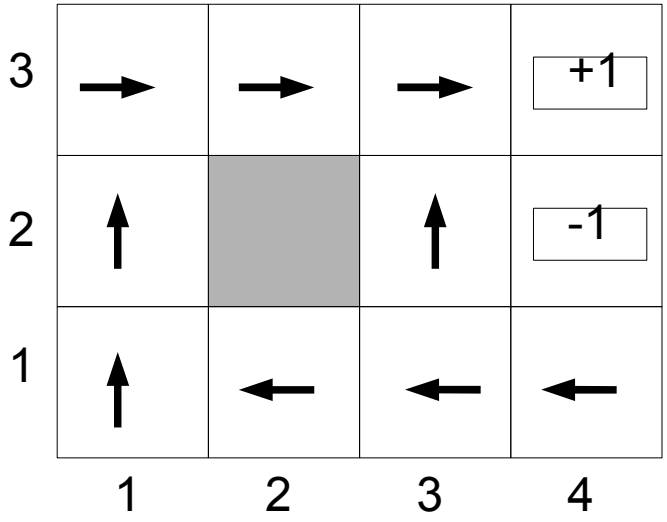
$\gamma = 1$

Itération de la politique - simulation

On continue tant que la politique change...

Itération de la politique - simulation

État final



3	0,0	0,0	0,0	0,0
2	0,0		0,0	0,0
1	0,0	0,0	0.0	0,0
	1	2	3	4

$\gamma = 1$