Reinforcement Learning

Reinforcement Learning (RL)

Différent de :

- l'apprentissage supervisé (Données et Résultats)
-l'apprentissage non supervisé (données)

Car repose sur l'interconnexion d'un agent et d'un environnement

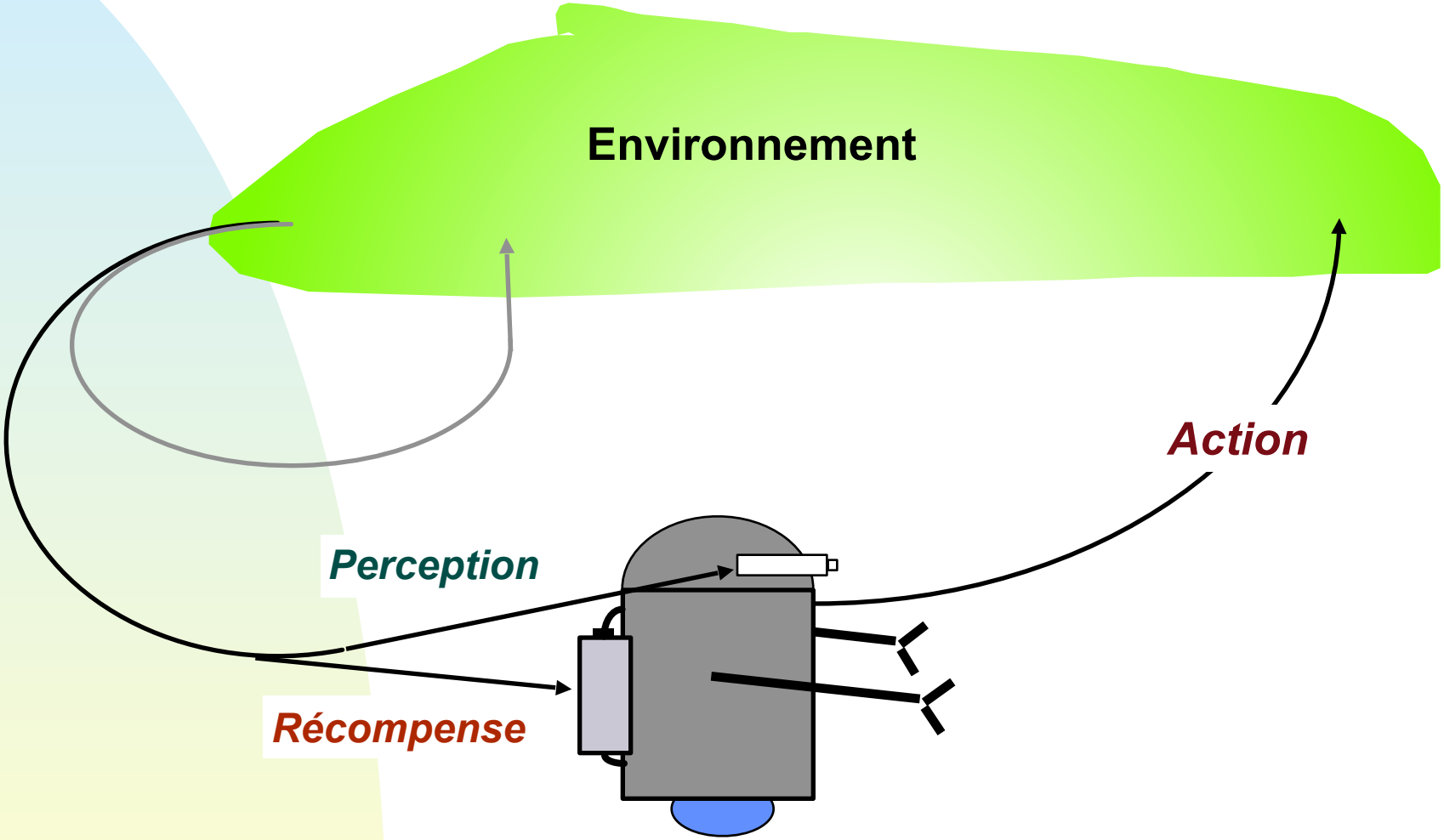
L'environnement fournit un état (généralement appelé s) à l'agent qui retourne une action (généralement appelée a) ainsi de suite.

De plus l'environnement retourne aussi une récompense (reward) qui modélise la qualité de l'action.

Une police correspond à la suite d'action à entreprendre pour réaliser un but.

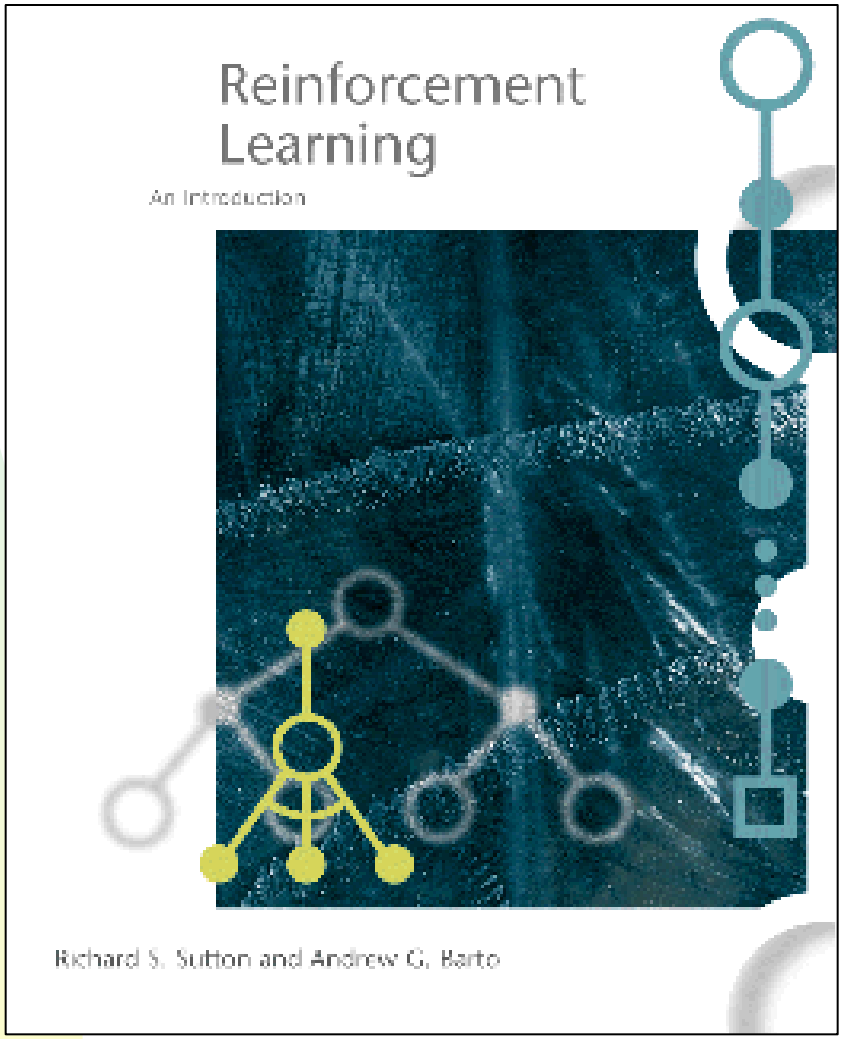
Le problème est de déterminer la police optimale (qui obtient les récompenses maximum)

Reinforcement Learning
Schema Général



Reinforcement Learning

Adapté de Sutton et Barto



Reinforcement Learning

Principe du RL :

-un agent (logiciel) fait des observations et décide des actions qu'un environnement doit exécuter. L'environnement retourne à l'agent des récompenses associées aux actions.

L'objectif est que l'agent apprenne à agir de façon à maximiser les récompenses à long termes.

Donc l'agent agit en relation avec l'environnement et apprend en essayant afin de maximiser les récompenses positives et minimiser les récompenses négatives.

Reinforcement Learning

Basé sur l'idée suivante:

Commençons par une analogie. Supposons que nous apprenions à un chien (agent) à attraper une balle. Au lieu d'apprendre explicitement au chien à attraper une balle, nous lui lançons simplement une balle et chaque fois que le chien attrape la balle, nous donnons au chien un cookie (récompense). Si le chien ne parvient pas à attraper la balle, nous ne lui donnons pas de cookie. Ainsi, le chien déterminera quelle action l'a amené à recevoir un cookie et répétera cette action.

Reinforcement Learning

Ainsi, le chien comprendra que le fait d'avoir attrapé la balle lui a fait recevoir un cookie et tentera de répéter l'attrape de la balle. Ainsi, de cette manière, le chien apprendra à attraper une balle tout en visant à maximiser les cookies qu'il peut recevoir.

De même, dans un cadre RL, nous n'enseignerons pas à l'agent quoi faire ou comment le faire ; à la place, nous donnerons une récompense à l'agent pour chaque action qu'il fera.

Reinforcement Learning

Nous donnerons une récompense positive à l'agent lorsqu'il accomplit une bonne action et nous donnerons une récompense négative à l'agent lorsqu'il accomplira une mauvaise action. L'agent commence par effectuer une action aléatoire et si l'action est bonne, nous donnons alors à l'agent une récompense positive afin que l'agent comprenne qu'il a effectué une bonne action et qu'il répétera cette action. Si l'action effectuée par l'agent est mauvaise, alors nous donnerons à l'agent une récompense négative afin que l'agent comprenne qu'il a effectué une mauvaise action et qu'il ne répétera pas cette action.

Reinforcement Learning

Ainsi, RL peut être considéré comme un processus d'apprentissage par essais et erreurs où l'agent essaie différentes actions et apprend la bonne action, ce qui donne une récompense positive.

Reinforcement Learning

ALGORITHME RL :

1. First, the agent interacts with the environment by performing an action.
2. By performing an action, the agent moves from one state to another.
3. Then the agent will receive a reward based on the action it performed.
4. Based on the reward, the agent will understand whether the action is good or bad.
5. If the action was good, that is, if the agent received a positive reward, then the agent will prefer performing that action, else the agent will try performing other actions in search of a positive reward.

Reinforcement Learning

PRENONS UN EXEMPLE SIMPLE
(Grid World suivant):

Les positions A à I dans l'environnement sont appelés les états de l'environnement. Le but de l'AGENT est d'atteindre l'état I en partant de A et en évitant les états hachurés (B,C,G, H).

Ainsi afin d'atteindre le but, chaque fois que l'agent visite un état hachurés, il recevra une recompense negative (disons -1) et quand il visite un état non hachuré il recevra une récompense positive (+1).

Reinforcement Learning

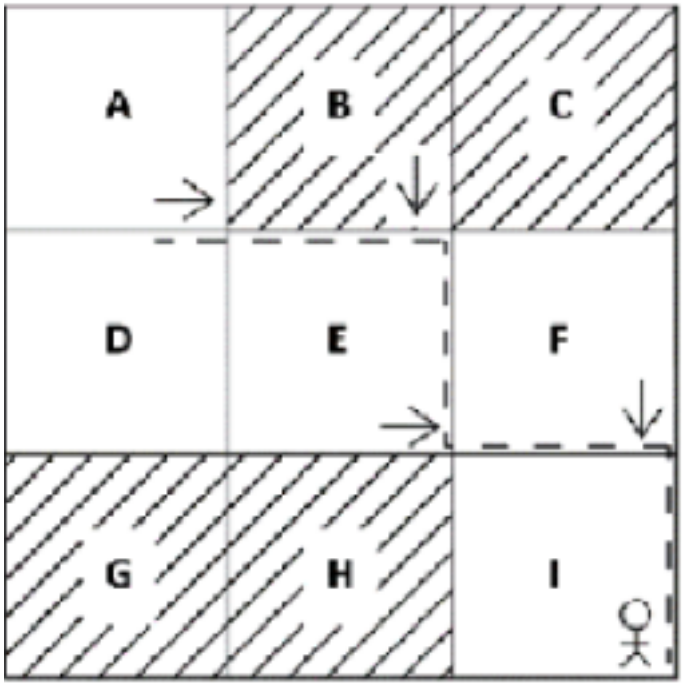
Les actions dans l'environnement sont : UP, DOWN, RIGHT et LEFT.
L'agent peut executer ces actions pour atteindre I à partir de A.

La première fois que l'agent interagit avec l'environnement (la première itération), il est peu probable que l'agent effectue l'action correcte dans chaque état, et il reçoit donc une récompense négative. Autrement dit, lors de la première itération, l'agent effectue une action aléatoire dans chaque état, ce qui peut conduire l'agent à recevoir une récompense négative. Mais au cours d'une série d'itérations, l'agent apprend à effectuer l'action correcte dans chaque état grâce à la récompense qu'il obtient, l'aidant à atteindre l'objectif.

Reinforcement Learning

1ere itération :

A la premiere iteration, l'agent execute une action aléatoire dans chaque état. Exemple:



L'agent fait l'action RIGHT à partir de l'état A et atteint le nouvel état B. Puisque B est un état hachuré, l'état B recevra un reward négatif et donc l'agent comprendra que passer de l'état A à l'état B est une mauvaise action. Quand il visitera l'état A la prochaine fois, l'agent essaiera une autre action que RIGHT.

Reinforcement Learning

De l'état B, l'agent execute DOWN et atteint le nouvel état E non hachuré. Puisque E est non hachuré, l'agent recevra un reward positif et l'agent comprendra que executer l'action DOWN à partir de l'état B est une bonne action.

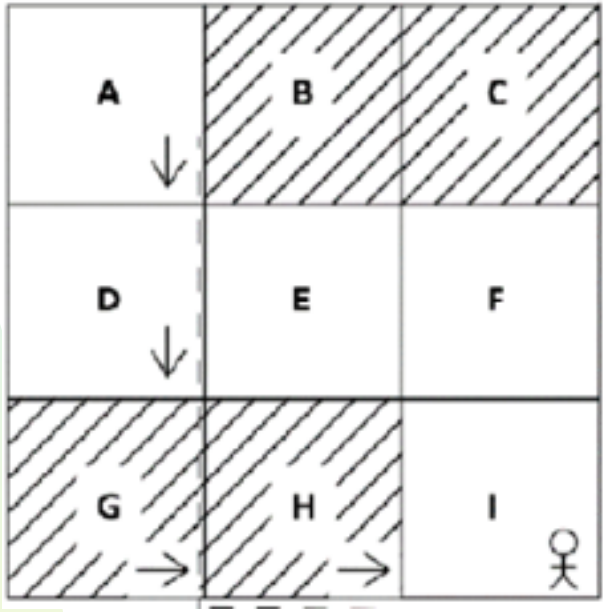
De l'état E, l'agent execute RIGHT et atteint le nouvel état F non hachuré. Puisque F est non hachuré, l'agent recevra un reward positif et l'agent comprendra que executer l'action RIGHT à partir de l'état E est une bonne action.

De l'état F, l'agent execute DOWN et atteint l'état BUT I et recevra un reward positif et l'agent comprendra que executer l'action DOWN à partir de l'état F est une bonne action.

Reinforcement Learning

2eme itération

De l'état A, au lieu de choisir RIGHT, l'agent essaie une action différente puisque il a apprit lors d cela précédente itération que RIGHT n'est pas une bonne action à partir de A.



Dans cette itération, l'agent execute DOWN à partir de A et atteint D non hachuré d'ou positif reward.

Reinforcement Learning

De l'état D, l'agent execute DOWN et atteint le nouvel état G hachuré. Puisque G est hachuré, l'agent recevra un reward négatif et l'agent comprendra que executer l'action DOWN à partir de l'état D n'est pas une bonne action (et il essaiera plus tard une action différente).

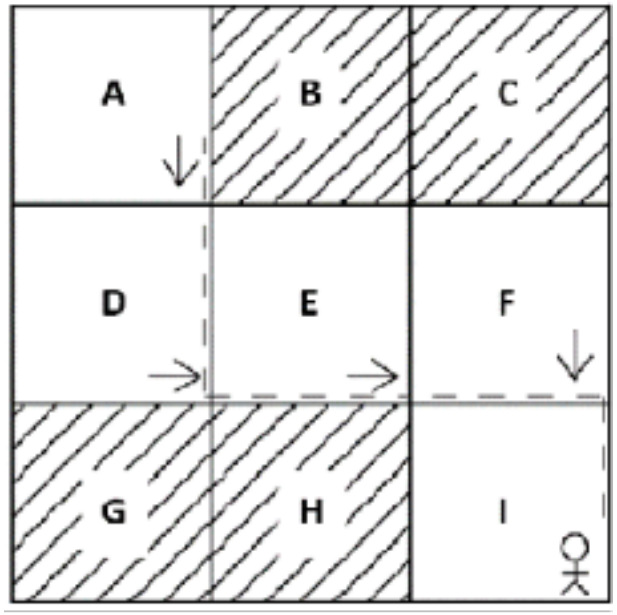
De l'état G, l'agent execute RIGHT et atteint le nouvel état H hachuré. Puisque H est hachuré, l'agent recevra un reward négatif et l'agent comprendra que executer l'action RIGHT à partir de l'état G n'est pas une bonne action.

De l'état H, l'agent execute RIGHT et atteint le état BUT I et reçoit un reward positif et l'agent comprendra que executer l'action RIGHT à partir de l'état H est une bonne action.

Reinforcement Learning

3eme itération

De l'état A, l'agent execute l'action DOWN et atteint D puisque lors de la seconde itération l'agent a apprit que exécuter DOWN est une bonne action à partir de A.



A partir de D, l'agent essaie une action différente au lieu de choisir DOWN puisque l'agent a apprit que DOWN était une mauvaise action à partir de D. L'agent choisit RIGHT et atteint l'état E.

Reinforcement Learning

A partir de l'état E, l'agent execute l'action RIGHT puisque lors de la première itération l'agent a apprit que exécuter RIGHT à partir de E est une bonne action. Il atteint F ensuite.

A partir de F, l'agent execute l'action DOWN puisque l'agent a apprit lors de la premiere itération que DOWN était une bonne action à partir de F. L'agent choisit RIGHT et atteint l'état I.

L'agent a donc apprit avec succès à atteindre l'état but I à partir de l'état A sans passer par les états hachurés.

De cette façon l'agent essaiera différentes actions dans chaque état et comprendra quand une action est bonne ou mauvaise en fonction du reward qu'il reçoit. Le but de l'agent est de maximiser les rewards. Donc l'agent exécutera toujours de bonnes actions qui retournent un reward positif et quand l'agent execute de bonnes actions dans chaque état, cela conduira à satisfaire le but.

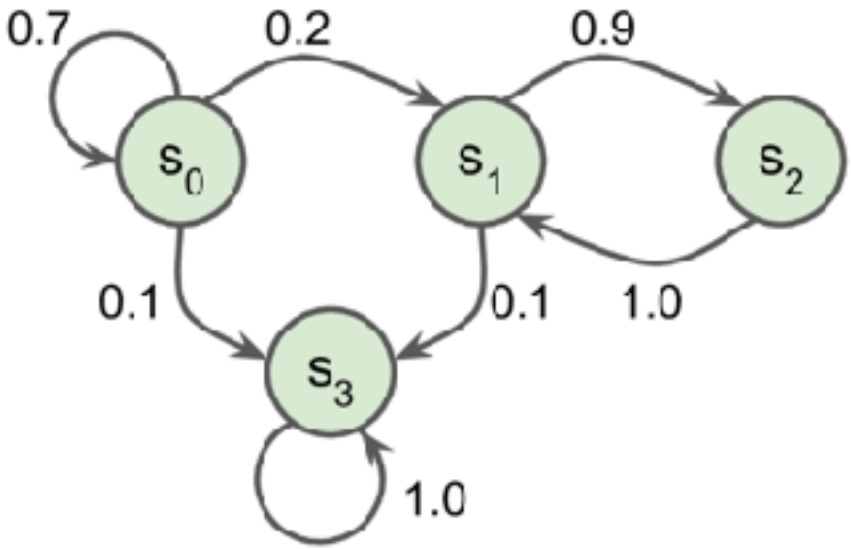
Ces différentes itérations sont appelées EPISODES.

Reinforcement Learning

Concepts de base issus des Markov Decision process (MDP)

Basé sur les chaines de Markov

Exemple de chaine de Markov



Reinforcement Learning

Propriété de base des chaines de Markov:

La propriété de Markov stipule que le futur ne dépend que du présent et non du passé. La chaîne de Markov, également connue sous le nom de processus de Markov, consiste en une séquence d'états qui obéissent strictement à la propriété de Markov ;

La chaine de Markov est un modèle probabiliste qui dépend uniquement de l'état actuel pour prédire l'état suivant (le futur est conditionnellement indépendant du passé)

Reinforcement Learning

EXEMPLE:

Par exemple, si nous voulons prédire le temps et que nous savons que l'état actuel est nuageux, nous pouvons prédire que l'état suivant pourrait être pluvieux. Nous avons conclu que le prochain état est susceptible d'être pluvieux uniquement en considérant l'état actuel (nuageux) et non les états précédents, qui auraient pu être ensoleillés, venteux, etc.

Reinforcement Learning

Le passage d'un état à un autre s'appelle une tradition et sa probabilité s'appelle une probabilité de transition. On noyer P(sIs') la probabilité de passer de l'état s à l'état suivant s'.

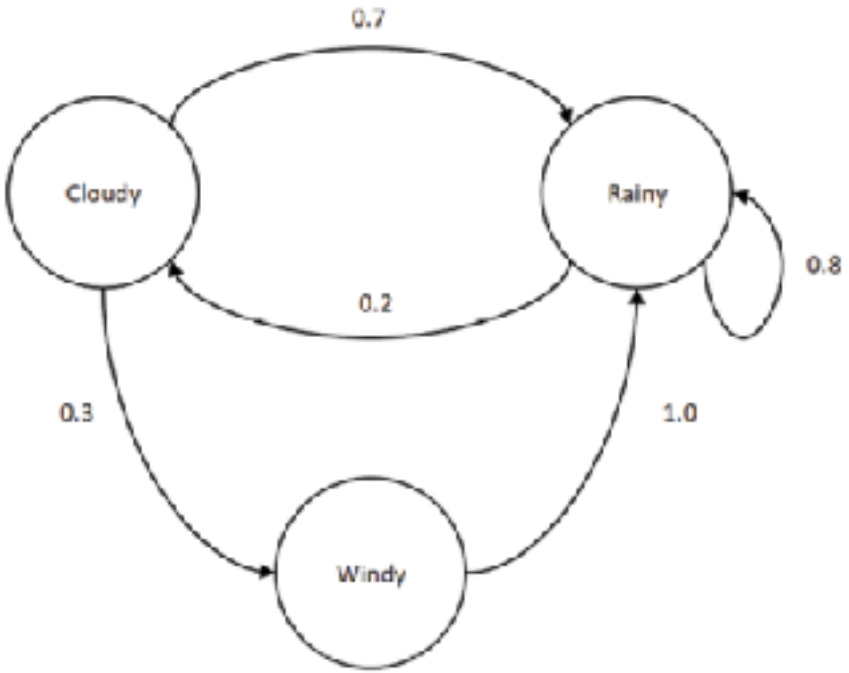
Prenons l'exemple météo à 3 états : nuageux, venteux, pluvieux.

Les probabilités de transition entre états peuvent être représentées par une table:

Current State	Next State	Transition Probability
Cloudy	Rainy	0.7
Cloudy	Windy	0.3
Rainy	Rainy	0.8
Rainy	Cloudy	0.2
Windy	Rainy	1.0

Reinforcement Learning

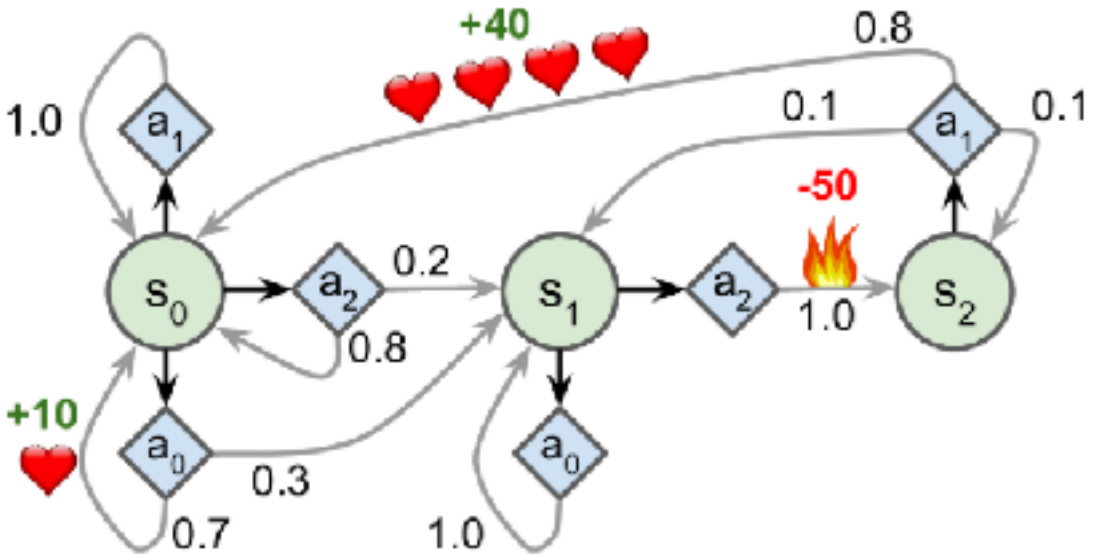
Autre forme de représentations (graphes ou matrices):



	Cloudy	Rainy	Windy
Cloudy	0.0	0.7	0.3
Rainy	0.2	0.8	0.0
Windy	0.0	1.0	0.0

Reinforcement Learning

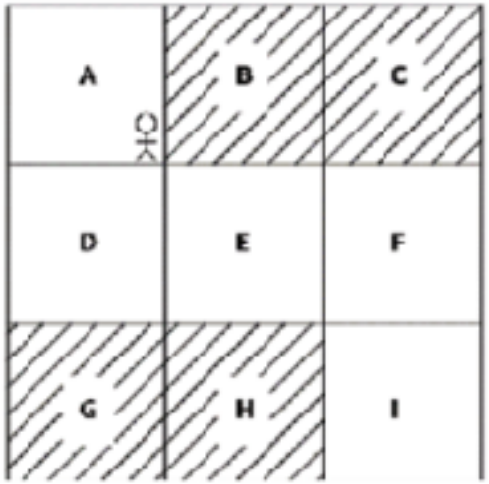
Markov REWARD process : extension des chaines de Markov incluant les rewards.



Reinforcement Learning

Les Markov Decision Process sont donc des MRP étendus avec la notion d'actions.

Reprenons l'exemple



Ensemble des états : A à I

Ensemble des actions : RIGHT, DOWN, LEFT, UP.

Probabilités de transitions notées P(s|s', a) : Probabilité de passer

Reinforcement Learning Concepts Fondamentaux

Rappel : notions de base en mathématique: la notion d'espérance.

Soit une variable X pouvant prendre elles valeurs 1,2,3,4,5,6.

Pour calculer la valeur moyenne : Somme de 1 à 6 / 6 = 3,5.

Supposons maintenant que X soit une variable aléatoire (elle prend ses valeurs sur une experience aléatoire telle que lancer de dès, lancer de pieces de monnaies, etc. Les variables aléatoires prennent des valeurs différentes en fonction de probabilités.

Exemple

X	1	2	3	4	5	6
P(x)	1/6	1/6	1/6	1/6	1/6	1/6

Reinforcement Learning Concepts Fondamentaux

Comment calculer la valeur moyenne d'une variable aléatoire?

Etant donné que chaque valeur a une probabilité d'occurence , on ne peut pas simplement prendre la moyenne.

Donc à la place, on calcule la moyenne pondérée c-a-d la somme des valeurs de X multipliées par leurs probabilités respectives : c'est l'espérance.

L'espérance d'une variable aléatoire X est définie par:

$$E(X) = \sum_{i=1}^{N} x_i p(x_i)$$

Reinforcement Learning Concepts Fondamentaux

Autre exemple

Espérance d'une fonction d'une variable aléatoire.

$f(x) = x^2$

X	1	2	3	4	5	6
f(x)	1	4	9	16	25	36
P(x)	1/6	1/6	1/6	1/6	1/6	1/6

$$\mathbb{E}_{x\sim p(x)}[f(X)] = \sum_{i=1}^{N} f(x_i)p(x_i)$$

Reinforcement Learning Concepts Fondamentaux

NOTION d'espace d'actions.

Dans l'exemple précédent, l'espace d'actions est défini par les 4 actions : UP, DOWN, RIGHT, LEFT.

On peut distinguer deux types d'espace d'actions :

- Espace d'actions discrètes: exemple les 4 actions précédentes
- Espace d'actions continues: exemple d'un agent qui conduit une voiture, alors l'espace d'actions consisterait un ensemble d'actions qui ont des valeurs continues telles que la vitesse à laquelle on doit conduire, le nombre de degrés requis pour tourner etc…

Reinforcement Learning Concepts Fondamentaux

NOTION de POLICY (Stratégie).

Une stratégie (Policy) définit le comportement de l'agent dans un environnement.

La stratégie indique à l'agent quelle action effectuer dans chaque état.

Exemple : dans le grid world, nous avons les watts de A à I et les 4 actions possibles.

La stratégie indique à l'agent de descendre dans l'état A de déplacer vers la droite dans l'état D, etc…

Reinforcement Learning Concepts Fondamentaux

Pour interagir avec l'environnement la première fois, la stratégie est initialisée de manière aléatoire (cette stratégie aléatoire indique à l'agent d'effectuer une action aléatoire dans chaque état.

Donc dans cette itération initiale, l'agent effectue une action aléatoire dans chaque état afin de savoir si l'action est bonne ou mauvaise en fonction du reward obtenu.

Au cours d'une série d'itérations, un agent apprendra à effectuer de bonnes actions dans chaque état (récompense positive).

Reinforcement Learning Concepts Fondamentaux

Pour interagir avec l'environnement la première fois, la stratégie est initialisée de manière aléatoire (cette stratégie aléatoire indique à l'agent d'effectuer une action aléatoire dans chaque état.

Donc dans cette itération initiale, l'agent effectue une action aléatoire dans chaque état afin de savoir si l'action est bonne ou mauvaise en fonction du reward obtenu.

Au cours d'une série d'itérations, un agent apprendra à effectuer de bonnes actions dans chaque état (récompense positive).

Cela fournira une bonne stratégie.

Reinforcement Learning Concepts Fondamentaux

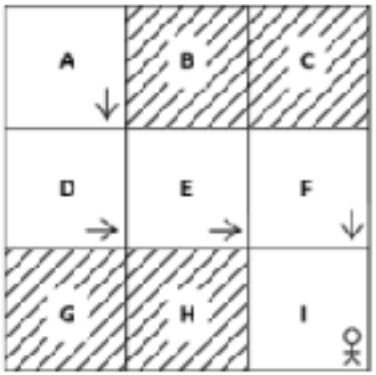
Cette bonne stratégie est appelée la politique (stratégie) optimale.

La politique optimale est la politique qui procure à l'agent une bonne récompense et l'aide à atteindre son objectif.

Exemple du grid world: la politique optimale indique à l'gent d'effectuer une action dans chaque état afin que l'agent puisse atteindre l'état I à partir de l'état A sans visiter les états hachurés.

Optimal Policy

State	Action
A	Down
D	Right
E	Right
F	Down



Reinforcement Learning Concepts Fondamentaux

Une stratégie (politique) peut être :

•une politique déterministe: elle indique à l'agent d'effectuer une action particulière dans un état. Elle associe l'état à une action. Elle est désignée généralement par:

$$a_t = \mu(s_t)$$

dans l'exemple du grid world: étant donné l'état A, la politique déterministe optimale μ donne l'action down à effectuer

μ(A) = DOWN

•une politique aléatoire: elle n'associe pas directement un état à une action particulière. Elle va associer un état à une distribution de probabilité sur un espace d'actions. Au lieu d'effectuer la même action chaque fois que l'agent visite l'état, l'gent effectue des actions différentes à chaque fois en fonction d'une distribution de probabilité renvoyée par la politique stochastique.

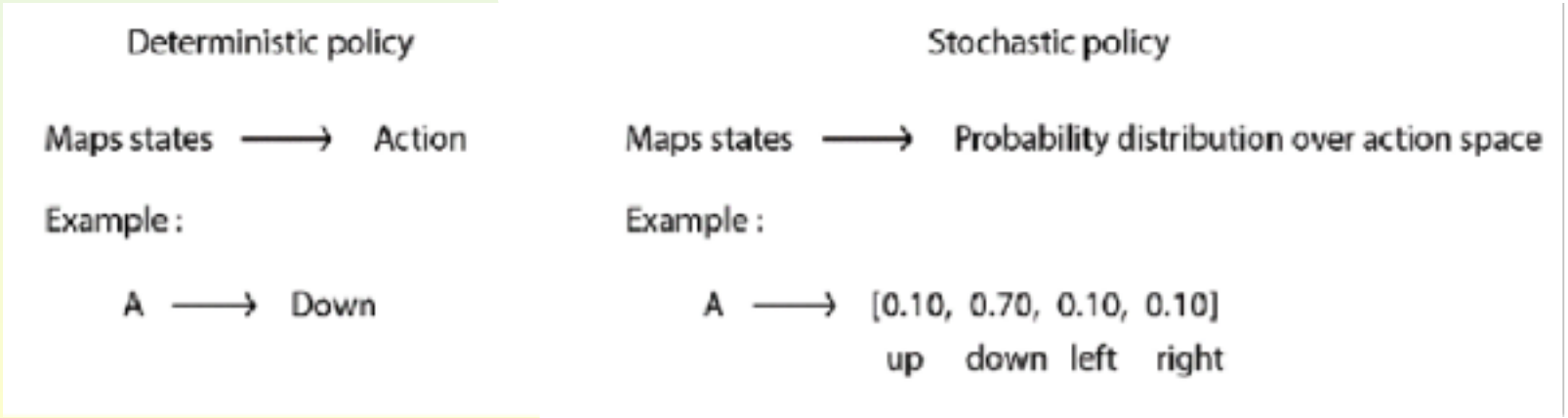
Reinforcement Learning Concepts Fondamentaux

Exemple le grid world: actions : (UP, DOWN, LEFT, RIGHT) .

Supposons que étant donné l'état A, une stratégie stochastique retourne la distribution de probabilité sur l'espace d'actions suivant: [0.10, 0.70, 0.10, 0.10]

Donc chaque fois que l'agent visite l'état A, au lieu de sélectionner la même action, l'agent sélectionne UP dans 10% du temps, DOWN dans 70% du temps, LEFT dans 10% du temps et RIGHT dans 10% du temps.

Différence en les deux types de stratégie



Reinforcement Learning Concepts Fondamentaux

Donc une politique stochastique associe l'eat à une distribution de probabilité sur l'espace d'actions notée généralement π.

Supposons que un état s et une action a au temps t, alors la stratégie stochastique peut exprimer par:

$$a_t \sim \pi(s_t)$$

Elle peut aussi être notée par $\pi(a_t|s_t)$.

2 types de stratégie stochastique existent:

- Stratégie catégorielle
- Stratégie gaussienne

Reinforcement Learning Concepts Fondamentaux

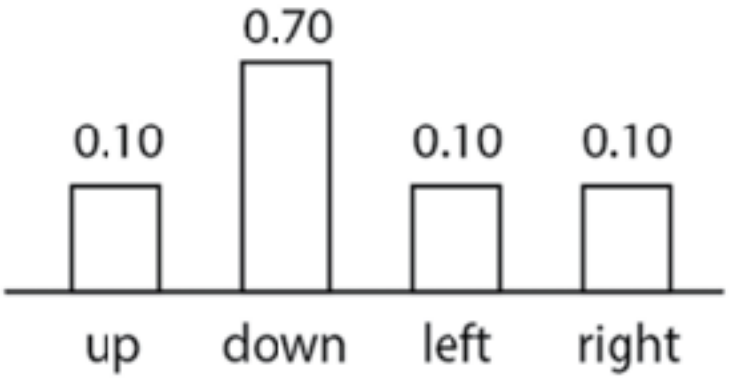
Stratégie catégorielle

Une politique stochastique est dite catégorielle lorsque l'espace d'actions est discret.

Donc la politique stochastique va utiliser dans ce cas la une distribution de probabilités catégorique sur l'espace d'actions pour sélectionner des actions.

Exemple du Grid World

Dans l'état A, une action sera sélectionnée sur la base de la distribution de probabilité catégorique sur l'espace d'actions



Reinforcement Learning Concepts Fondamentaux

Stratégie gaussienne

Une politique stochastique gaussienne concerne un espace d'actions continu.

Donc la politique stochastique va utiliser une distribution de probabilité gaussienne sur l'espace des actions pour sélectionner des actions.

Exemple simple: un agent est dédié à conduire une voiture. Actions continue comme par exemple la vitesse de la voiture dont la valeur évolue de 0 à 150 km/h.

La sélection de la valeur de la vitesse sera basée sur la distribution gaussienne

Reinforcement Learning Concepts Fondamentaux

Notion d'episode

Rappel : l'agent interagit avec l'environnement en effectuant des a tiens à partir de l'état initial jusqu'à atteindre l'état final.

Un episode est aussi appelé une trajectoire (le chemin emprunté par l'agent. Il est noté tau (†).

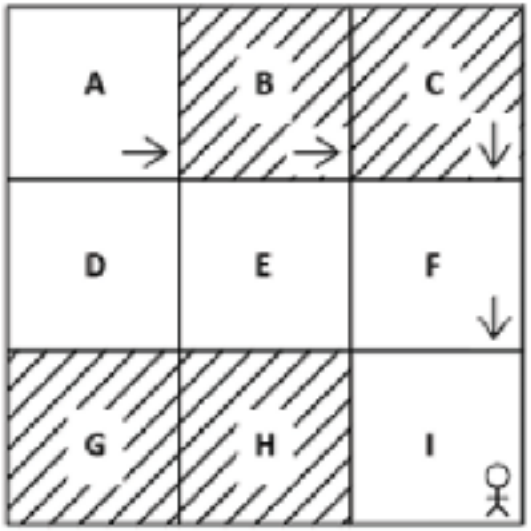
Un agent peut executer plusieurs episodes et chaque episode est indépendant des autres.

Pourquoi utiliser de nombreux episodes? Le but est d'apprendre la stratégie optimale et cela nécessite l'execution de nombreux episodes.

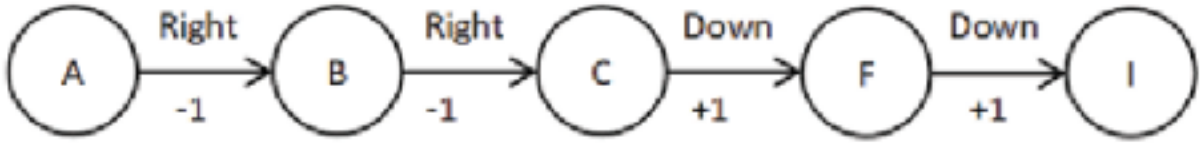
Reinforcement Learning Concepts Fondamentaux

Exemple Grid World

Episode 1 : l'agent utilise une politique aléatoire et sélectionne une action aléatoire dans chaque état de l'eat initial A à l'état final I et observe la récompense

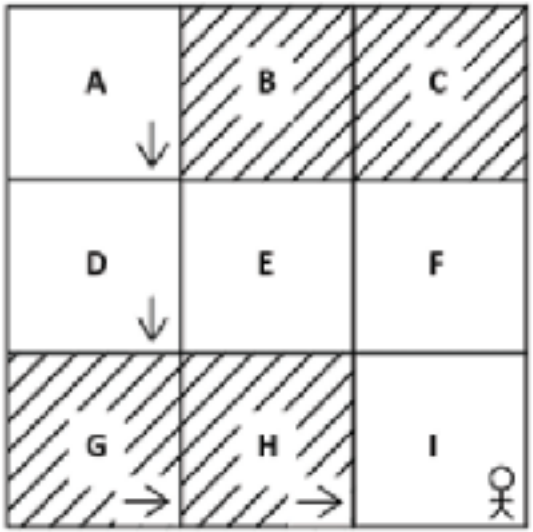


$\tau_1 =$

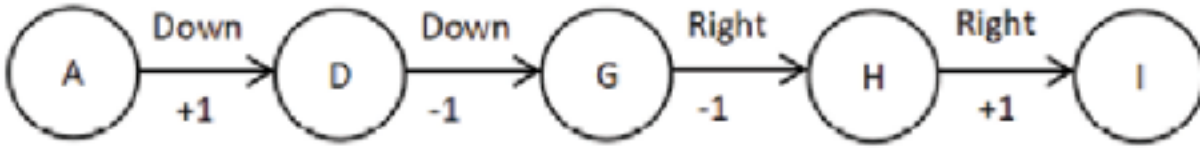


Reinforcement Learning Concepts Fondamentaux

Episode 2: l'agent essaie une politique différente pour éviter les récompenses négatives qu'il a observé dans l'episode précédent.

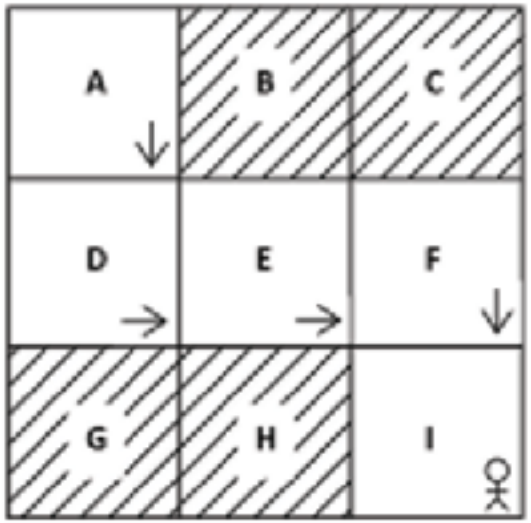


$\tau_2 =$



Reinforcement Learning Concepts Fondamentaux

Episode n: donc après une série d'episodes, l'agent apprend la politique optimale c-a-d la politique qui amène l'agent à l'état I sans visiter des états hachurés.



Reinforcement Learning Concepts Fondamentaux

Notion de taches épisodiques et taches continues

Une tache épisodique est une tache qui a un état final.

Donc les taches épisodiques sont des taches composées d'episodes et donc ont un etat final (exemple un jeu de course de voiture).

Les taches continues ne contiennent aucun épisodes et donc n'ont pas d'état final (exemple un robot d'assistance à une personne).

Reinforcement Learning Concepts Fondamentaux

Notion d'horizon

L'horizon est le pas de temps jusqu'auquel l'agent interagit avec l'environnement.

Deux catégories : horizon fini et horizon infini.

Horizon fini: l'interaction agent-environnement s'arrête à un pas de temps particulier. Exemple dans les taches épisodiques, un agent interagit avec l'environnement en partant de l'état initial pas de temps t=0 et atteint l'état final au pas de temps T. Puisque l'interaction s'arrête au pas de temps T, on dit que l'horizon est fini.

Horizon infini: si l'interaction ne s'arrete jamais, on parle d'horizon infini. Exemple tache continue n'a pas d'états terminaux.

Reinforcement Learning Concepts Fondamentaux

Notion de Return et Discount factor

La notion de Return peut être définie comme la somme des rewards obtenues par l'agent dans un episode. Il est noté R.

$$R(\tau) = r_0 + r_1 + r_2 + \ldots + r_T$$

$$R(\tau) = \sum_{t=0}^{T} r_t$$

Considérant la trajectoire (episode) † (tau), le return est la somme des rewards du pas de temps 0 au pas de temps T.

Reinforcement Learning Concepts Fondamentaux

Comment maximiser? Il faut effectuer une action correcte dans chaque état en choisissant la stratégie optimale. Cela dans le cas de taches épisodiques.

Comment définir le return pour les taches continues?

Comme il n'y pas d'état final:

$$R(\tau) = r_0 + r_1 + r_2 + \ldots + r_\infty$$

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$$

Pour maximiser le return, on introduit un nouveau terme appelé discount factor (gamma):

Reinforcement Learning Concepts Fondamentaux

Le discount factor permet de décider de l'importance qui sont accordées aux rewards futurs et aux rewards immédiats.

La valeur du discount factor varie entre 0 et 1.

Lorsque nous fixons le discount factor à une petite valeur (proche de 0), cela implique que nous accordons plus d'importance aux recompenses immédiates qu'aux récompenses futures.

Lorsque nous fixons le discount factor a une valeur élevée (proche de 1), cela implique que nous accordons plus d'importance aux recompenses futures qu'aux recompenses immédiates.

Reinforcement Learning Concepts Fondamentaux

Notion de Value Function

La fonction Value Function (aussi appelée State Value Function) indique la valeur de l'état.

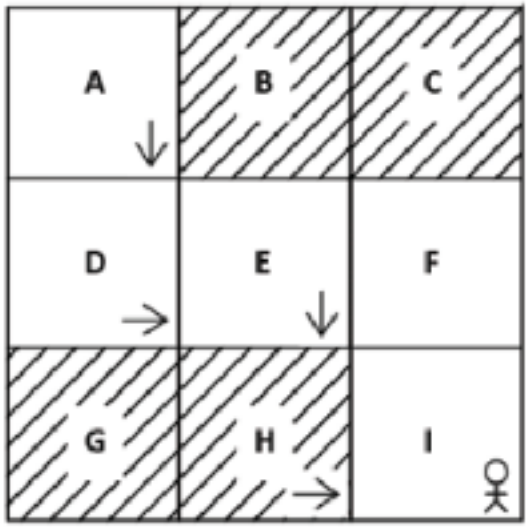
La valeur d'un état est le return qu'un agent obtiendrait à partir de cet état suivant la politique. Elle est notée V(s):

$$V^{\pi}(s) = [R(\tau)|s_0 = s]$$

s_0 = s implique de l'état initial est s_0

Reinforcement Learning Concepts Fondamentaux

Soit la trajectoire tau (†) générée en suivant une politique π dans l'exemple du Grid World:



La state value de l'état A ; v(A) = 1 +1 + -1 + 1 = 2

La state value de l'état D; V(D) = = 1 - 1 + 1 = 1

La state value de l'état E; V(E) = = - 1 + 1 = 0

La state value de l'état H; V(H) = = 1

Pas de valeur pour l'état I puisque état final.

En résume la valeur d'un état (state Value) est le return de la

Reinforcement Learning Concepts Fondamentaux

On peut donc définir le return attendu comme valeur d'un état; il est défini comme le retour attendu que l'agent obtiendrait à partir de l'état s en suivant la politique π:

$$V^{\pi}(s) = \mathop{\mathbb{E}}_{\tau \sim \pi} [R(\tau) | s_0 = s]$$

Pourquoi le retour attendu et pas calculer directement la valeur d'un état en tant que retour?

Le problème est que le retour est une valeur aléatoire impliquant des probabilités.

Reinforcement Learning Concepts Fondamentaux

Exemple:Supposons que nous ayons une politique stochastique.

Contrairement à une politique déterministe (qui relie directement l'état à une action), la politique stochastique relie l'état à la distribution de probabilité sur l'espace d'actions.

Donc mal politique stochastique sélectionne les actions en fonction d'une distribution de probabilité.

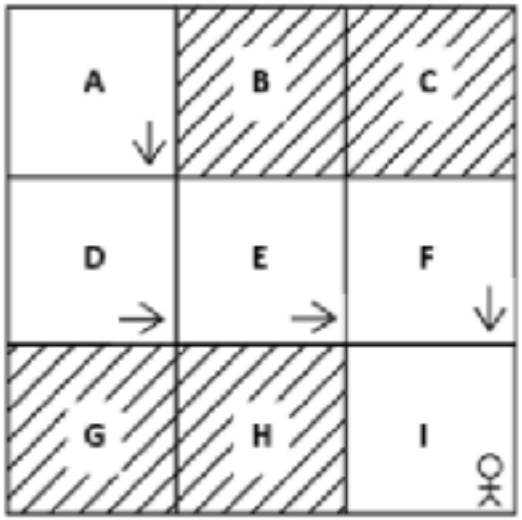
Si nous sommes dans l'état A et que la politique stochastique renvoie la distribution de probabilité suivante [0.0, 0.80,0.0,0.20]

Cela implique qu'avec la politique stochastique nous effectuions l'action DOWN 80% du temps et l'action RIGHT 20% du temps:
$\pi(\text{DOWN I A}) = 0.8$ et $\pi(\text{RIGHT I A}) = 0.20$

Supposons aussi que la politique choisisse l'action RIGHT dans les état D et E et l'action DONW dans les états B et F dans 100% des cas

Reinforcement Learning Concepts Fondamentaux

Exemple suite : Générons un premier episode avec la politique π:



$\tau_1 =$ A —Down, +1→ D —Right, +1→ E —Right, +1→ F —Down, +1→ I

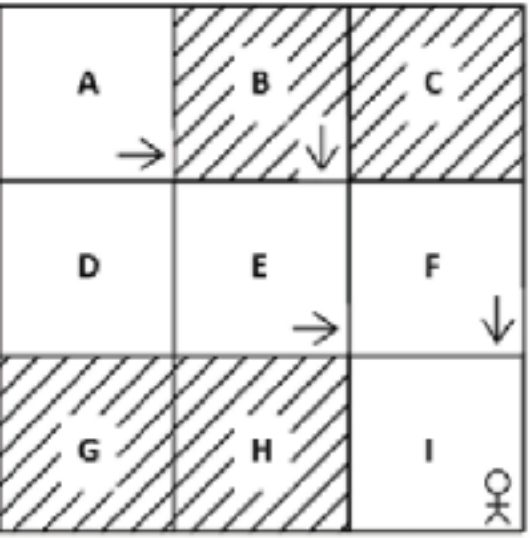
Valeur de état A:

$V(A) = R(\dagger_1) = 1 + 1 + 1 + 1 = 4$

Générons un autre episode $\dagger_2$ avec la même politique stochastique π

Reinforcement Learning Concepts Fondamentaux

Exemple suite : générons un episode en utilisant cette politique



$\tau_2 =$ A —Right, -1→ B —Down, +1→ E —Right, +1→ F —Down, +1→ I

Valeur de état A:

V(A) = R(c) = -1 + 1 +1 + 1 = 2

On peut donc voir que bien que nous utilisons la même politique, les valeur de l'etat A dans les deux trajectoires $\dagger_1$ et $\dagger_2$ sont différentes (evidemment politique stochastique 80% du temps on prend la trajectoire $\dagger_1$ et 20% du temps $\dagger_2$)

Reinforcement Learning Concepts Fondamentaux

Donc on prend pour valeur d'un état le retour attendu (puisque les retours ont des valeurs différentes a cause des probabilités).

Le retour attendu est la moyenne pondérée donc la somme des retours multiplié par leur probabilité. on peut donc écrire:

$$V^{\pi}(s) = \mathop{\mathbb{E}}_{\tau \sim \pi} [R(\tau) | s_0 = s]$$

$$V^{\pi}(A) = \mathop{\mathbb{E}}_{\tau \sim \pi} [R(\tau) | s_0 = A]$$

Valeur de l'état A:

$$= \sum_i R(\tau_i)\pi(a_i|A)$$

$$= R(\tau_1)\pi(\text{down}|A) + R(\tau_2)\pi(\text{right}|A)$$
$$= 4(0.8) + 2(0.2)$$
$$= 3.6$$

Reinforcement Learning Concepts Fondamentaux

Bien sur ce calcul de la valeur de état dépend de la politique choisie.

Il y a donc de nombreuses value fonctions différentes en fonction des différentes politiques.

La Value Function optimale notée V*(s) donne la valeur maximale par rapport à toutes les autres fonctions et exprimée par

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

Exemple : supposons que nous ayons 2 politiques π_1 et π_2.

Supposons que la state value de l'état A avec π_1 soit

$V^{\pi 1}(s) = 13$ et pour π_2 $V^{\pi 2}(s) = 11$.

La politique qui donne la valeur maximale est appelée la politique optimale π^*.

Reinforcement Learning Concepts Fondamentaux

Une façon pratique de visualiser la value function est une table.

Imaginons que nous ayons 2 états s_0 et s_1, alors la value function peut être représentée par :

State	Value
s_0	7
s_1	11

Grace a la table on s'aperçois qu'il est meilleur d'être dans l'état s_1 plutôt que dans l'état s_0 et donc l'état optimal est s_1.

Reinforcement Learning Concepts Fondamentaux

La Q Function (aussi appelée state-action value function) indique la valeur d'une paire état-action.

La valeur d'une paire état-action est le retour que l'agent obtiendrait en partant dun état s et en effectuant une action a suivant la politique π.

La valeur d'une paire état-action (Q function) est généralement désignée par Q(s,a) et connue sous le nom Q value ou state-action value.

$$Q^{\pi}(s,a) = [R(\tau)|s_0 = s, a_0 = a]$$

Seule différence avec ce qu'on a vu précédemment (entre la value function et la Q function:

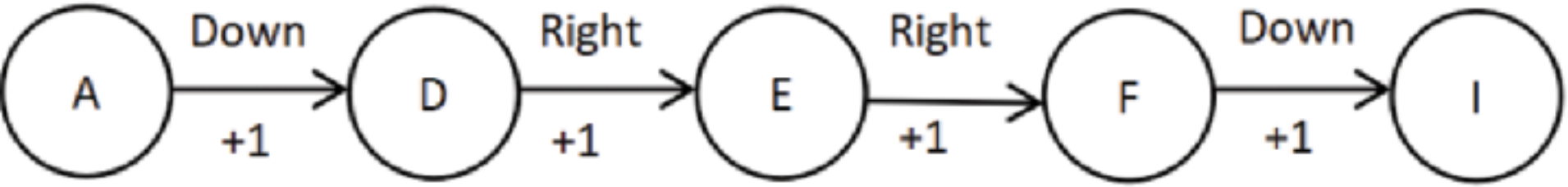
Value function : on calcule la valeur d'un état

Q function : on calcule la valeur dune paire état-action.La Q Function (aussi appelée stat-action value function) indique la valeur d'une paire état-action.

La valeur d'une paire état-action est le retour que l'agent

Reinforcement Learning Concepts Fondamentaux

Exemple: soit la trajectoire suivante



Nous avons vu que la Q function calculait la valeur dune paire état-action.

Nous devons donc calculer la Q value de la paire A-DOWN.

Donc la Q

$$Q^{\pi}(A, \text{down}) = [R(\tau)|s_0 = A, a_0 = \text{down}]$$ t A.

$$Q(A, \text{down}) = 1 + 1 + 1 + 1 = 4$$

Reinforcement Learning Concepts Fondamentaux

Calculer la Q value de la paire D-RIGHT.

C'est donc la Q value de faire RIGHT dans l'état D:

$$Q^{\pi}(A, \text{right}) = [R(\tau)|s_0 = D, a_0 = \text{right}]$$

$$Q(A, \text{right}) = 1 + 1 + 1 = 3$$

De même on peut calculer la Q value pour toutes les paires état-action.

Comme pour la value function, il faut utiliser le retour attendu car le retour est une variable aléatoire. On peut écrire:

$$Q^{\pi}(s, a) = \mathop{\mathbb{E}}_{\tau \sim \pi} [R(\tau)|s_0 = s, a_0 = a]$$

La Q Value est le retour attendu que l'agent obtiendrait à partir de l'eat s et en effectuant une action suivant la politique π

Reinforcement Learning Concepts Fondamentaux

Comme précédemment, la Q value varie en fonction de la politique choisie.

Il peut y avoir de nombreuses Q functions différentes selon différentes politiques.

La Q function optimale est celle qui a la Q value maximale par rapport aux autres Q fonctions.

Donc:

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$$

La politique optimale π* est la politique qui retourne la Q value maximale.

Comme pour la Value function, la Q function peut être visualiser avec une table.

Reinforcement Learning Concepts Fondamentaux

Considérons 2 états s_0 et s_1 ainsi que 2 actions 0 et 1. La Q function peut être représentée par la table Q:

State	Action	Value
s_0	0	9
s_0	1	11
s_1	0	17
s_1	1	13

La table Q représente les Q value de toutes les paires état-action possible.

La politique optimale est celle qui procure à l'agent le rendement (retour = somme des rewards) maximal.

Cette politique optimale peut être extraite de la table Q en sélectionnant tout simplement l'action qui a la Q value maximale dans chaque état.

Reinforcement Learning Concepts Fondamentaux

Dans l'exemple, on choisit l'action 1 dans l'état s_0 et l'action 0 dans l 'etat s_1

Q Table

State	Action	Value
s_0	0	9
s_0	1	11
s_1	0	17
s_1	1	13

Optimal policy

State	Action
s_0	1
s_1	0

Reinforcement Learning Concepts Fondamentaux

2 types d'apprentissage: Model-based et Model-free.

Apprentissage de type Model-based

un agent possède une description complète de l'environnement.

Donc l'agent connait la dynamique du modèle de l'environnemental (c-a-d il connait la probabilité de transition de l'environnement).

Apprentissage de type Model-free

un agent ne connait pas la dynamique du modèle de l'environnemental. il va essayer de trouver la politique optimale sans la dynamique du modèle.