Agent Oriented Learning

Introduction apprentissage par renforcement - Résolutions

Stéphane Airiau

Université Paris-Dauphine

Definition (Processus décisionnel de Markov)

Un *Processus décisionnel de Markov* est un tuple $\langle S, A, T, R, \gamma \rangle$ où

- S est un ensemble fini d'états
- A est un ensemble fini d'actions
- T est une matrice de transition
  $T^a_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$ probabilité d'arriver dans l'état s' à l'instant +1 quand on a pris l'action a dans l'état s à l'instant t
- R est le vecteur de récompenses
  $R^a_s = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$ valeur moyenne obtenue après avoir pris l'action a dans l'état s
- un ensemble d'état initial
- parfois un ensemble d'états terminaux

Les composants d'un agent : politique

C'est ce qui gouverne le comportement de l'agent

politique déterministe : La fonction associe à chaque état **une action**

$$\pi : S \mapsto A$$

3	→	→	→	+1
2	↑		↑	-1
1	START ↑	→	↑	←
	1	2	3	4

politique optimale pour un pénalité de 0.03 par déplacement

Focus pour le cours

- problèmes itératifs en continue.
- objectif : maximiser la somme "avec dévaluation" $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$
  - Pour éviter une récompense infinie si on tombe dans des cycles
  - Le futur reste uncertain ! Bon compromis entre court et long terme
  - Tendance naturelle vers le court terme
  - Mathématiquement, c'est quand même pratique !
- la fonction de transition est stochastique
- la fonction de récompense est connue

Comment trouver la meilleure politique ?

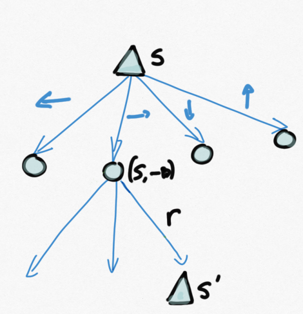
Evaluation

On va s'aider de deux quantités

$v^\star(s)$ quelle est la valeur de me trouver dans l'état s puis de continuer avec la politique optimale

$q^\star(s,a)$ quelle est la valeur de prendre l'action a dans l'état s puis de continuer avec la politique optimale

$\pi^\star(s)$ politique optimale pour l'état s (i.e. quelle est la meilleure action).



Valeur optimale d'un état $v^\star(s)$

- on calcule la valeur espérée en supposant qu'on suive la politique optimale
- on prend la moyenne pondérée des récompenses escomptée

➩ comme dans expectimax !

$$v^\star(s) = \max_{a \in A} q^\star(s,a)$$

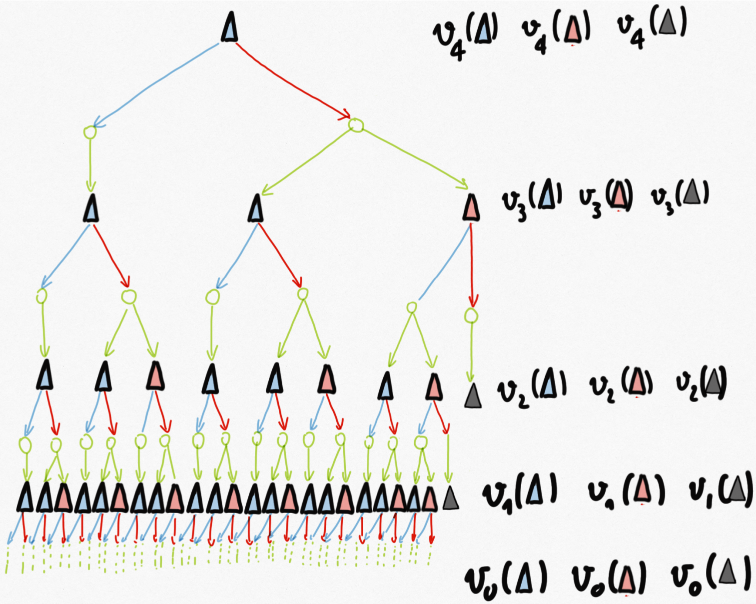
$$q^\star(s,a) = R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v^\star(s')$$

donc

$$v^\star(s) = \max_{a \in A} \left[R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v^\star(s') \right]$$

Cette équation est l'**équation de Bellman** pour la fonction de valeur optimale.

Idée d'itération sur les valeurs



Value Iteration

```
for each $s \in S$ and $k \in \mathbb{N}$
    $V_k(s) \leftarrow 0$

repeat for $k = 0$ to ...

    for each $s \in S$

        $V_{k+1}(s) \leftarrow \max_{a \in A} \left[ R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a V_k(s') \right]$        /* mise à jour */

until convergence
```

Value Iteration – plus efficace pour la mémoire

for each $s \in S$

  $V(s) \leftarrow 0$

repeat

  for each $s \in S$

$$V(s) \leftarrow \max_{a \in A} \left[R^a_s + \gamma \sum_{s' \in S} T^a_{ss'} V(s') \right]$$

/* mise à jour */

until convergence

Value Iteration – avec test de convergence

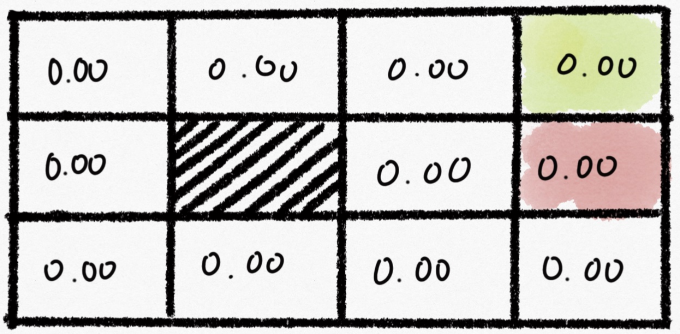
```
for each s ∈ S
    V(s) ← 0

repeat
    Δ ← 0                                              /* mesure le plus grand changement */
    for each s ∈ S
      v ← V(s)                   /* sauvegarde l'ancienne valeur pour mesurer le changement */
      V(s) ← max_{a∈A} [ R_s^a + γ Σ_{s'∈S} T_{ss'}^a V(s') ]                 /* mise à jour */
      Δ ← max(Δ, |v − V(s)|)                    /* mise à jour du plus grand changement*/
until Δ < ε                                                          /* test convergence */
```

$$V(s) \leftarrow \max_{a\in A}\left[R_s^a + \gamma \sum_{s'\in S} T_{ss'}^a V(s')\right]$$

- On n'a pas de politique explicite
- On a un théorème de convergence

Itération sur les valeurs : $k = 0$



$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 1$

0.00	0.00	0.00	+1.00
0.00		0.00	−1.00
0.00	0.00	0.00	0.00

$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 2$

0.00	0.00	0.72	+1.00
0.00		0.00	−1.00
0.00	0.00	0.00	0.00

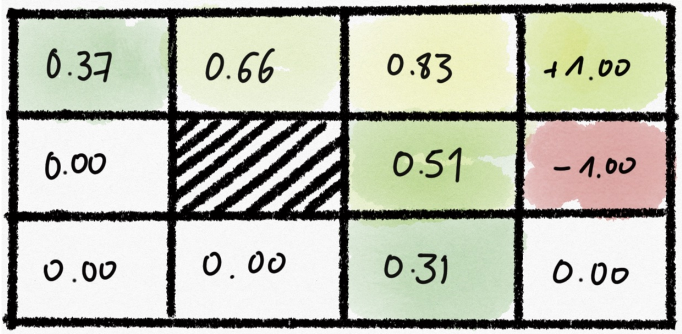
$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 3$

0.00	0.52	0.78	+1.00
0.00		0.43	−1.00
0.00	0.00	0.00	0.00

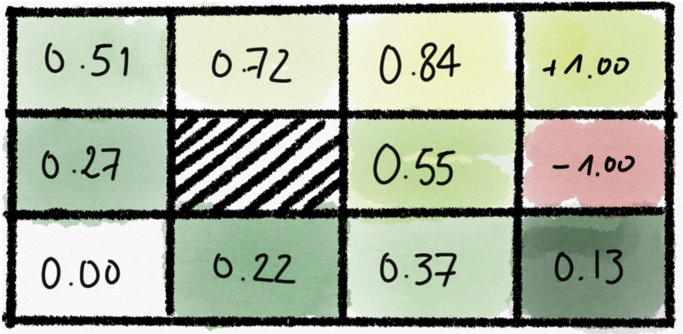
$r = 0$
$\gamma = 0.9$
$bruit = 0.2$

Itération sur les valeurs : $k = 4$



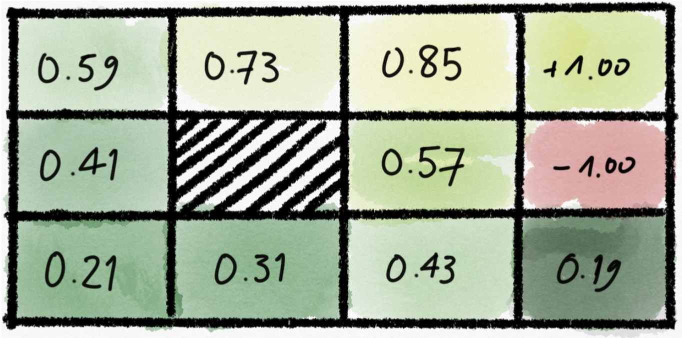
$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 5$



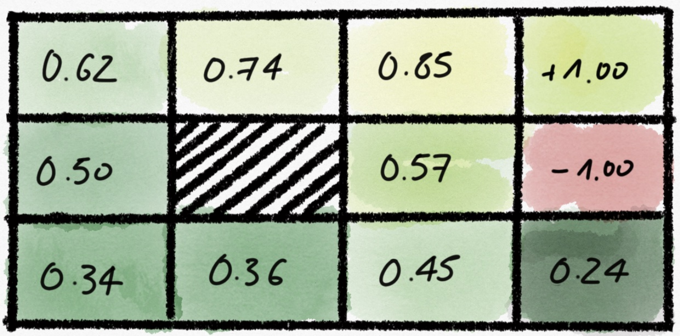
$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 6$



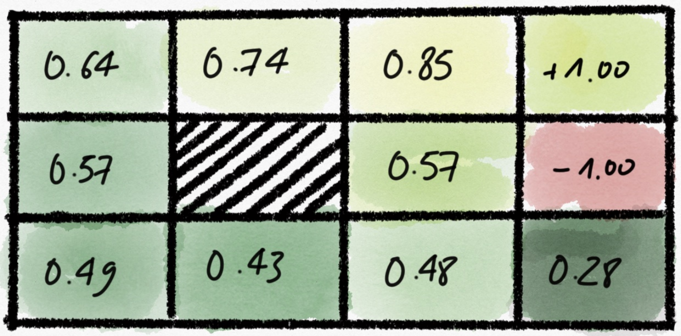
$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 7$



$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Itération sur les valeurs : $k = 100$



$$r = 0$$
$$\gamma = 0.9$$
$$bruit = 0.2$$

Limitations de Value Iteration

- la méthode est lente
- la valeur du max change rarement

➾ pourtant c'est cela qui va aider à changer toutes les valeurs !

- les valeurs peuvent mettre longtems à converger exactement alors que la politique, elle, est déjà optimale depuis longtemps

➾ on va essayer de travailler sur la politique.

Supposons qu'on connaise les valeurs optimales $v^\star(s)$

0.95	0.96	0.98	+1.00
0.94		0.89	-1.00
0.92	0.91	0.90	0.80

Comment devons-nous agir ?

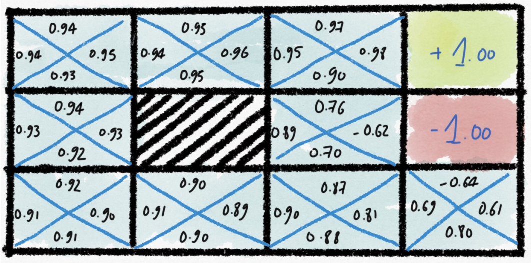
On doit faire une étape d'expectimax !

$$\pi^\star(s) = arg\max_a \left[R_s^a + \gamma \sum_{s'} T_{ss'}^a v^\star(s') \right]$$

⇰ on peut appeler cette étape faire une extraction de politique.

Comment déterminer une bonne action à partir de q ?

Supposons qu'on connaise les valeurs optimales $q^\star(s,a)$



Comment devons-nous agir ?

Trivial, on choisit la meilleure action ! (ou une des meilleurs actions).

$$\pi^\star(s) = arg\max_a q*(s,a)$$

⇨ morale de l'histoire : les actions sont plus faciles à obtenir à partir de q qu'à partir de v ! !

Equations de Bellman

En s'inspirant d'expectimax, on a écrit l'équation de Bellman

$$v_{\pi^\star} = \max_{a \in A} R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_{\pi^\star}(s')$$

On pourrait définir cela plus formellement.

On définit la fonction de valeur pour la politique π, $v_\pi : S \mapsto \mathbb{R}$ telle que

$$v_\pi(s) = \mathbb{E}\left[G_t \mid S_t = s, \pi\right]$$

où $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$

$v_\pi(s)$ est la valeur *à long terme* de se trouver dans l'état s puis d'utiliser la politique π pour choisir ses actions.

On peut montrer que v_π satisfait une équation de Bellman.

$$v_\pi = R_s^{\pi(s)} + \gamma \sum_{s' \in S} T_{ss'}^{\pi(s)} v_\pi(s')$$

Equations de Bellman

Ensuite, on pourrait définir $v^\star(s) = \max_\pi v_\pi(s)$
En on pourrait montrer que $v^\star$ satisfait l'équation de Bellman

$$v^\star(s) = \max_{a \in A} \left[R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v^\star(s') \right]$$

On peut montrer que :

- cette équation admet une unique solution $v^\star$
- on peut définir une politique optimale comme étant une politique $\pi^\star$ telle que $v_{\pi^\star} = v^\star$
- il peut y avoir plusieurs politiques optimales
- au moins une politique optimale est deterministe :

$$\pi(s) = arg \max_{a \in A} \left[R_s^a + \sum_{s' \in S} T_{ss'}^a v^\star(s') \right]$$

Autre stratégie de résolutions : améliorer une politique petit à petit

On peut partir d'une politique arbitraire π et essayer de la modifier pour améliorer les performances.

$$\pi_0 \stackrel{\textit{évalue}}{\rightarrow} v_{\pi_0} \stackrel{\textit{améliore}}{\rightarrow} \pi_1 \stackrel{\textit{évalue}}{\rightarrow} v_{\pi_1} \rightarrow \cdots \rightarrow \pi_\star \stackrel{\textit{évalue}}{\rightarrow} v_{\pi_\star}$$

On va améliorer la politique en se comportant de manière "gloutonne"

Une fois v_π évaluée : on peut calculer $q^\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_\pi(s')$

- si $q^\pi(s,a) > v_\pi(s)$: on a trouvé une amélioration ! [a]

➾ on peut regarder tous les états $s \in S$ et mettre à jour la politique $\pi'(s) = arg\max_{a \in A} q_\pi(s,a)$

Si aucune amélioration n'est trouvée, on a donc $v_\pi = v_{\pi'}$

➾ $v_{\pi'} = \max_{a \in A} R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_{\pi'}(s')$

On reconnait là l'équation de Bellman pour la fonction de valeurs *optimale*

On a donc trouvé $v^\star = v_{\pi'}$!

a. il faut une petite démonstration sur ce point

Itération de politique

L'idée est donc d'alterner

1- l'évaluation d'une politique
2- l'amélioration de la politique

*jusqu'à ce qu'*on converge vers une politique qui sera la politique optimale.

Pour les politiques deterministes, il y a un nombre fini de politiques, on va converger en un nombre fini d'itérations.

Variantes : quand arrêter l'évaluation ?

- convergence à un ϵ près
- après k itérations (k a une petite valeur)
- pourquoi pas après chaque itération ?

Comparaison

- Les deux méthodes calcule le même résultat : à la fin, on a les mêmes valeurs optimales ($v^\star$ et $q^\star$)
- Itération sur les valeurs
  - la politique est implicite. A chaque itération, on met à jour les valeurs
  - si on travaille avec v, on doit utiliser l'extraction d'une politique pour obtenir la politique optimale.
- Iteration sur les politiques
  - plusieurs itérations pour mettre à jour les valeurs d'une politique fixe (mais pour chaque itération, on ne considère qu'une seule action, ce qui devrait être rapide).
  - on met à jour la politique, on doit comparer toutes les actions (peut être lent)
  - soit on améliore la politique, soit on a terminé !

Résolution à l'aide de méthodes Monte Carlo pour des PDMs épisodiques

Apprendre un PDM inconnu

Pour les deux algorithmes vus précédemment ("iteration sur les valeurs” et "iteration sur les politiques”), on devait connaître :

- le modèle de transition $T^a_{ss'}$,
- le modèle de récompense R^a_s

Aujourd’hui, on va voir des méthodes qui **ne** nécessitent **pas** la connaissance des ces modèles.

- ➩ seule l’expérience va guider le choix
- ➩ véritablement de l’apprentissage

Environnement épisodique

On va se placer seulement dans des PDMs épisodique :

- chaque épisode doit se terminer
- on va apprendre d'un épisode en entier
- un épisode : une partie de black jack

Plan

1. Méthodes Monte Carlo
   - Evaluation d'une politique
   - Etimation de la valeur des actions
   - Contrôle par Monte Carlo ("on policy" et "off policy")

Méthode Monte Carlo : Evaluation d'une politique π

- apprendre v_π à partir des épisodes en suivant une politique π
- On veut apprendre la valeur *à long terme*
  Pour un épisode qui se termine à l'itération k, on a

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{k-1} R_{k+t}$$

- $v_\pi(s)$ est la valeur de passer par l'état s en utilisant la politique π :

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

- on va utiliser l'expérience de l'agent pour estimer $v_\pi(s)$ pour chaque état s.
- Attention, dans un épisode, on peut passer plusieurs fois par le même état !

Méthode Monte Carlo : Evaluation d'une politique

algorithme première visite

Exécute **un** épisode

$s_0 \xrightarrow{a_1} s_1, r_1 \xrightarrow{a_2} s_2, r_2 \xrightarrow{a_3} s_3, r_3 \xrightarrow{a_4} s_4, r_4$

fin de l'épisode

Une fois l'épisode terminé
je fais les mises à jour

$G_0 = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4$

$G_1 = r_2 + \gamma r_3 + \gamma^2 r_4$

$G_2 = r_3 + \gamma r_4$

$G_3 = r_4$

→ m.à.j $v(s_0)$ avec G_0

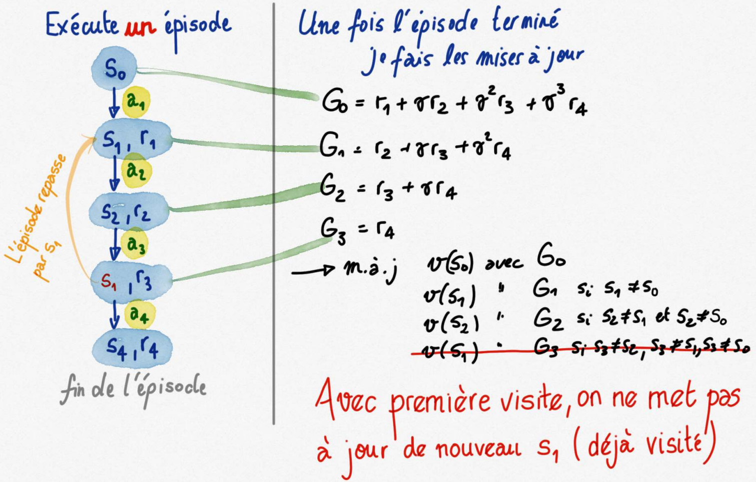
$v(s_1)$ " G_1 si $s_1 \neq s_0$

$v(s_2)$ " G_2 si $s_2 \neq s_1$ et $s_2 \neq s_0$

$v(s_3)$ " G_3 si $s_3 \neq s_2$, $s_3 \neq s_1$, $s_3 \neq s_0$

Méthode Monte Carlo : Evaluation d'une politique

algorithme première visite



Méthode Monte Carlo : Evaluation d'une politique

Algorithme "première visite"

$v \in \mathbb{R}^{|S|}$, $count \in \mathbb{N}^{|S|}$, $Acc \in \mathbb{R}^{|S|}$
initialise $v(s) = 0$ pour chaque état $s \in S$
initialise $count(s) = 0$ pour chaque état $s \in S$
initialise $Acc(s) = 0$ pour chaque état $s \in S$

Répète éternellement
    Simule un épisode en suivant la politique π
    Pour chaque transition t de l'épisode, on calcule G_t
    Pour chaque état s qui apparait dans l'épisode
        Pour la **première itération** t où s est visité dans l'épisode
            $Acc(s) \leftarrow Acc(s) + G_t$
            $count(s) \leftarrow count(s) + 1$
            $v(s) \leftarrow \dfrac{Acc(s)}{count(s)}$

chaque valeur de G_t est un échantillon tiré de manière indépendante et identiquement distribué, avec une variance finie
➩ avec la loi des grands nombres, on a

$$\lim_{count(s)\to\infty} v(s) = v_\pi(s)$$

Méthode Monte Carlo : Evaluation d'une politique

Algorithme "chaque visite"

$v \in \mathbb{R}^{|S|}$, $count \in \mathbb{N}^{|S|}$, $Acc \in \mathbb{R}^{|S|}$
initialise $v(s) = 0$ pour chaque état $s \in S$
initialise $count(s) = 0$ pour chaque état $s \in S$
initialise $Acc(s) = 0$ pour chaque état $s \in S$

Répète éternellement
    Simule un épisode en suivant la politique π
    Pour chaque transition t de l'épisode, on calcule G_t
    Pour **chaque itération** t qui visite l'état s
        $Acc(s) \leftarrow Acc(s) + G_t$
        $count(s) \leftarrow count(s) + 1$
        $v(s) \leftarrow \dfrac{Acc(s)}{count(s)}$

Ici, chacun des échantillons n'est pas forcément indépendant des autres. Mais on a quand même convergence vers $v_\pi(s)$.

(Singh & Sutton, 1996)

si un état s se répète, le fait de retomber dans l'état s n'est surement pas un hasard, et donc il y a une corrélation entre le premier passage et le second...

Méthode Monte Carlo : Evaluation d'une politique

Très différent de l'utilisation de la programmation dynamique

- toutes les transitions possibles / seulement les transitions de l'épisode
- une seule transition / toutes les transitions de l'épisode
- l'estimation de chaque état est fait de manière indépendante / l'estimation d'un état dépend de l'estimation des autres états

➯ l'estimation est indépendante du nombre d'états $|S|$.

➯ on peut partir d'un état et faire des simulations pour apprendre ces états sans se soucier des autres

Evaluation de la valeur des paires état-action

- Si on n'a pas le modèle $T^a_{ss'}$, on a beau avoir $v_\pi(s)$, on ne peut pas déduire l'action optimale !
  En effet, pour cela on a *besoin* de faire le calcul :

$$q_\pi(s,a) = R^a_s + \gamma \sum_{s' \in S} T^a_{ss'} v_\pi(s)$$

- Rappel $q_\pi(s,a)$ estime la valeur à long terme de prendre l'action a puis de suivre la politique π.

➾ au lieu d'estimer v_π avec notre algorithme de Monte Carlo, on va estimer q_π
➾ a priori même stratégie "première visite" et "chaque visite" possible

- attention, maintenant il faut tenir un compte sur chaque paire (action, état)

➾ il faudra faire attention à avoir suffisemment d'observations pour chaque paire !

Méthode Monte Carlo : Evaluation d'une politique

Algorithme version "première visite"

$q \in \mathbb{R}^{|S|\times|A|}$, $count \in \mathbb{N}^{|S|\times|A|}$, $Acc \in \mathbb{R}^{|S|\times|A|}$
initialise $q(s,a) = 0$ pour chaque état $s \in S$ et chaque action $a \in A$
initialise $count(s) = 0$ pour chaque état $s \in S$ et chaque action $a \in A$
initialise $Acc(s) = 0$ pour chaque état $s \in S$ et chaque action $a \in A$

Répète éternellement
    Simule un épisode en suivant la politique π
    Pour chaque transition t de l'épisode, on calcule G_t
    Pour chaque état s qui apparait dans l'épisode
        Pour la **première itération** t où s est visité dans l'épisode
            $Acc(s,a) \leftarrow Acc(s,a) + G_t$
            $count(s,a) \leftarrow count(s) + 1$
            $q(s,a) \leftarrow \dfrac{Acc(s,a)}{count(s,a)}$

chaque valeur de G_t est un échantillon tiré de manière indépendante et identiquement distribué, avec une variance finie

➩ avec la loi des grands nombres, on a

$$\lim_{count(s,a)\to\infty} q(s,a) = q_\pi(s,a)$$

Evaluation de la valeur des actions

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`
  on aura seulement des valeurs pour les paires $s,\pi(s)$

Evaluation de la valeur des actions

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`
  on aura seulement des valeurs pour les paires $s,\pi(s)$
- une stratégie : "exploring starts"
  on tire au hasard une paire (s_0,a_0) pour l'état initial et on utilise "première visite"
  ➥ on utilise la loi des grands nombres pour estimer $q_\pi(s_0,a_0)$

Evaluation de la valeur des actions

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`
  on aura seulement des valeurs pour les paires $s,\pi(s)$

- une stratégie : "exploring starts"
  on tire au hasard une paire (s_0,a_0) pour l'état initial et on utilise "première visite"
  ➩ on utilise la loi des grands nombres pour estimer $q_\pi(s_0,a_0)$
- Evidemment, ceci est problématique pour des interactions avec un environnement réel (on ne peut pas forcément choisir l'état initial !!)
- Supposons pour le moment qu'on puisse faire cela
  on verra comment le contourner plus tard.

Evaluation de la valeur des actions

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`
  on aura seulement des valeurs pour les paires $s,\pi(s)$

- une stratégie : "exploring starts"
  on tire au hasard une paire (s_0,a_0) pour l'état initial et on utilise "première visite"
  ➩ on utilise la loi des grands nombres pour estimer $q_\pi(s_0,a_0)$
- Evidemment, ceci est problématique pour des interactions avec un environnement réel (on ne peut pas forcément choisir l'état initial !!)
- Supposons pour le moment qu'on puisse faire cela
  on verra comment le contourner plus tard.
- On devrait avoir une bonne estimation de $q_\pi(s,a)$ pour chaque paire (s,a).

Evaluation de la politique **optimale**

Même idée que pour itération des politiques :
une approximation de la fonction de valeurs optimale et une approximation de la politique optimale

$$\pi_0 \overset{\textit{évalue}}{\rightarrow} q_{\pi_0} \overset{\textit{améliore}}{\rightarrow} \pi_1 \overset{\textit{évalue}}{\rightarrow} q_{\pi_1} \rightarrow \cdots \rightarrow \pi_\star \overset{\textit{évalue}}{\rightarrow} q_{\pi_\star}$$

- utilisation d'une infinité d'épisode pour estimer $q_\pi(s,a)$ avec "exploring starts"
- améliore de façon gloutonne la politique π comme dans itération des politiques : $\pi_{k+1}(s) = \text{argmax}_a\, q_{\pi_k}(s,a)$
- ⇨ même garantie de convergence que pour itération des politiques

peut-on utiliser moins d'épisodes pour évaluer une politique ?

Evaluation de la politique **optimale**

On reprend l'idée de l'algorithme itération sur les valeurs :

1. Utilisation d'une convergence à ϵ près pour estimer $q_\pi(s,a)$
   on peut calculer des bornes pour être sûr de faire suffisamment d'itérations
   ➩ on peut garantir la convergence, mais sûrement beaucoup trop d'itérations pour être une solution en pratique
2. plus extrême : utiliser seulement un <u>épisode</u> avant de faire une amélioration.
   ➩ mais pour le moment, on n'a pas de garantie de convergence ! cependant, on a du mal à se convaincre que cela ne va pas converger !

Evaluation de la politique optimale : "Monte Carlo Exploring Starts"

$q \in \mathbb{R}^{|S|\times|A|}$
$count \in \mathbb{N}^{|S|\times|A|}$
$Acc \in \mathbb{R}^{|S|\times|A|}$
initialise $v(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$
initialise $count(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$
initialise $Acc(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$

Répète éternellement
  Tire aléatoirement une paire $(s_0,a_0) \in S \times A$
  Simule un épisode en suivant la politique π en **partant** de (s_0,a_0)
  Pour **chaque paire** (s,a) qui est visitée dans l'épisode
    Si la première occurrence de (s,a) est à l'instant t
      $Acc(s,a) \leftarrow Acc(s,a) + G_t$
      $count(s,a) \leftarrow count(s,a) + 1$
      $q(s,a) \leftarrow \dfrac{Acc(s,a)}{count(s,a)}$
    Pour chaque état s dans l'épisode
      $\pi(s) \leftarrow arg\max_{a\in A} q(s,a)$

pas encore de démonstration que la convergence soit garantie !!!
(mais l'hypothèse exploring starts est trop forte !)

Eviter l'astuce "exploring starts"

Avec les méthodes Monte Carlo, on a fait deux hypothèses jusqu'ici :

1. on travaille dans un PDM épisodique
2. on peut choisir l'état initial au hasard pour garantir de visiter toutes les paires $(s,a) \in S \times A$

On veut trouver une technique pour éviter l'hypothèse "exploring starts"

- Comme pour le problème des bandits, il va falloir **explorer**
- soit on va essayer d'améliorer la politique qu'on utilise pour générer les données : approche "on policy"
- soit on va utiliser une politique pour générer les données, et mettre à jour une autre politique (qui n'est pas utilisée) : approche "off policy"

"on-policy" Monte Carlo

- estime et améliore une politique tout en l'utilisant
- utiliser une politique stochastique avec des probabilités strictement positives
  $\pi(s,a) > 0$ "politique soft"

➾ graduellement mettre à jour cette politique vers une politique deterministique (et optimale !)

- ajouter de l'**exploration** aléatoire

$$\text{ex : } \epsilon\text{-greedy} \begin{cases} \text{utiliser } arg\max_{a\in A} q(s,a) \text{ avec une probabilité } 1-\epsilon \\ \text{tirer une action avec une proabilité uniforme avec une probabilité } \epsilon \end{cases}$$

Evaluation de la politique optimale : "on policy Monte Carlo"

version première visite

$q : S \times A \to \mathbb{R}$ initialise $q(s,a) = 0 \ \forall s \in S$ et $\forall a \in A$

$count : S \times A \to \mathbb{N}$ initialise $count(s,a) = 0 \ \forall s \in S$ et $\forall a \in A$

$Acc : S \times A \to \mathbb{R}$ initialise $Acc(s,a) = 0 \ \forall s \in S$ et $\forall a \in A$

π une politique soft (par exemple uniforme)

Répète éternellement

- Simule un épisode en entier en suivant la politique π
- Compute G_t pour chaque transition de l'épisode
- Pour **chaque paire** (s,a) qui est visitée dans l'épisode
  - Si la première occurrence de (s,a) est à l'instant t
    - $Acc(s,a) \leftarrow Acc(s,a) + G_t$
    - $count(s,a) \leftarrow count(s,a) + 1$
    - $q(s,a) \leftarrow \frac{Acc(s,a)}{count(s,a)}$
  - Pour chaque état s dans l'épisode
    - $a^\star \leftarrow arg\max_{a \in A} q(s,a)$ (départage arbitraire des ex-æquos)
    - Pour chaque action a

$$\pi(s,a) \leftarrow \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A|} & \text{if } a = a^\star \\ \frac{\epsilon}{|A|} & \text{if } a \neq a^\star \end{cases}$$

Vérification de l'amélioration

On nomme π' la politique ϵ-greedy.

Pour n'importe quelle politique π, on a $\sum_{a\in A}\left(\pi(s,a) - \frac{\epsilon}{|A|}\right) = 1-\epsilon$

$$
\begin{aligned}
q(s,\pi'(s)) &= \sum_{a\in A}\pi'(s,a)q(s,a)\\
&= \frac{\epsilon}{|A|}\sum_{a\in A}q(s,a) + (1-\epsilon)\max_{a\in A}q(s,a)\\
&= \frac{\epsilon}{|A|}\sum_{a\in A}q(s,a) + (1-\epsilon)\sum_{a\in A}\frac{\pi(s,a)-\frac{\epsilon}{|A|}}{1-\epsilon}\max_{a\in A}q(s,a)\\
&\geq \frac{\epsilon}{|A|}\sum_{a\in A}q(s,a) + (1-\epsilon)\sum_{a\in A}\frac{\pi(s,a)-\frac{\epsilon}{|A|}}{1-\epsilon}q(s,a)\\
&\geq \frac{\epsilon}{|A|}\sum_{a\in A}q(s,a) + \sum_{a\in A}\pi(s,a)q(s,a) - \frac{\epsilon}{|A|}\sum_{a\in A}q(s,a)\\
&\geq \sum_{a\in A}\pi(s,a)q(s,a) = q_\pi(s,a) \text{ On a bien une amélioration !} \quad \square
\end{aligned}
$$

Vérification de la convergence

Il reste à démontrer la convergence vers une soft politique optimale.

Il faut montrer qu'on a une égalité quand π (où π') sont optimales sur les soft policies (i.e. aucune autre politique "soft" ne peut les dominer).

On considère un environnement ϵ-E dérivé de notre environnement E : si on exécute une action a dans ϵ-E, le résultat est celui de prendre une action de manière uniforme avec une probabilité ϵ dans E et de d'exécuter l'action a avec une probabilité $1 - \epsilon$ dans E.

Soit $\widetilde{v}^\star$ et $\widetilde{q}^\star$ les fonctions de valeurs optimales dans ϵ-E.

Une politique π est optimale dans E parmis les politiques softs ssi $v_\pi = \widetilde{v}^\star$

D'après la définition de $\widetilde{v^\star}$, ce doit être l'*unique* solution de :

$$\begin{aligned}
\widetilde{v^\star}(s) &= (1-\epsilon)\max_a \widetilde{q^\star}(s,a) + \frac{\epsilon}{|A|}\sum_{a\in A}\widetilde{q^\star}(s,a) \\
&= (1-\epsilon)\max_a \sum_{s',r} p(s',r\,|\,s,a)\left[r+\gamma\widetilde{v^\star}(s')\right] \\
&\quad + \frac{\epsilon}{|A|}\sum_{a\in A}\sum_{s',r} p(s',r\,|\,s,a)\left[r+\gamma\widetilde{v^\star}(s')\right]
\end{aligned}$$

Quand l'égalité est obtenue et qu'on ne peut pas améliorer π, on a :

$$\begin{aligned}
v_\pi(s) &= (1-\epsilon)\max_a q_\pi(s,a) + \frac{\epsilon}{|A|}\sum_{a\in A} q_\pi(s,a) \\
&= (1-\epsilon)\max_a \sum_{s',r} p(s',r\,|\,s,a)\left[r+\gamma v_\pi(s')\right] \\
&\quad + \frac{\epsilon}{|A|}\sum_{a\in A}\sum_{s',r} p(s',r\,|\,s,a)\left[r+\gamma v_\pi(s')\right]
\end{aligned}$$

Comme $\widetilde{v^\star}$ est l'*unique* solution, on doit avoir $\widetilde{v^\star} = v_\pi$

Bilan

- On parvient à trouver la meilleure politique parmi les politique ϵ-soft
- Sans avoir besoin de faire l'hypothèse (souvent peu réaliste) "exploring starts".

➾ apprentissage d'une politique presque optimale qui continue à explorer.

- une autre stratégie est d'avoir deux politiques :
  - une pour explorer et générer des données
  - l'autre que l'on optimise
  - mais pour notre cours, on ne va pas étudier ces autres algorithmes.

Conclusions

Dans des domaines où on peut répéter beaucoup d'épisodes

- on est capable d'évaluer une politique
- on est capable de trouver une politique optimale

⇒ on n'a plus besoin de connaître les modèles de transitions T et de récompenses R

- dans beaucoup de situations, il est facile de simuler des épisodes.
- si on a besoin d'étudier des états particuliers et qu'on peut les simuler ⇒on peut simplement faire les calculs pour ces états.
- ces méthodes n'effectue pas de "bootstrap"