Apprentissage par renforcement

Stéphane Airiau

Université Paris Dauphine

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

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

Value Iteration

```
for each s ∈ S and k ∈ ℕ
    V_k(s) ← 0

repeat for k = 0 to ...

    for each s ∈ S

        V_{k+1}(s) ← max_{a∈A} [ R_s^a + γ Σ_{s'∈S} T_{ss'}^a V_k(s') ]        /* mise à jour */

until convergence
```

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

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 ?

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 :

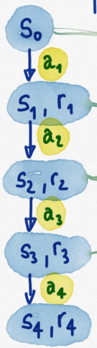
$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

- 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



fin de l'épisode

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

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

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

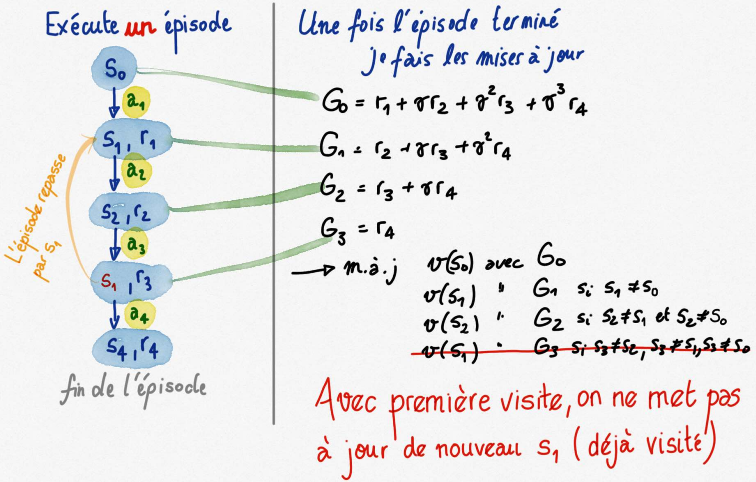
$$G_2 = r_3 + \gamma r_4$$

$$G_3 = r_4$$

→ 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
            $Acc(s) \leftarrow Acc(s) + G_t$
            $count(s) \leftarrow count(s) + 1$
            $v(s) \leftarrow \frac{Acc(s)}{count(s)}$

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

$$\pi_0 \stackrel{\text{évalue}}{\longrightarrow} q_{\pi_0} \stackrel{\text{améliore}}{\longrightarrow} \pi_1 \stackrel{\text{évalue}}{\longrightarrow} q_{\pi_1} \rightarrow \cdots \rightarrow \pi_\star \stackrel{\text{évalue}}{\longrightarrow} q_{\pi_\star}$$

- 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 $\sum_{a\in A}\left(\pi(s,a)-\frac{\epsilon}{|A|}\right)=1-\epsilon$

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

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

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

Une politique π est optimale dans $\mathcal{E}$ 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 :

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

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

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

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

Evaluer une politique tout en en suivant une autre

- Supposons qu'on suive une politique π' (ou bien qu'on a collecté des données)
- Peut-on calculer v_π pour une autre politique π ?

Evaluer une politique tout en en suivant une autre

- Supposons qu'on suive une politique π' (ou bien qu'on a collecté des données)
- Peut-on calculer v_π pour une autre politique π ?
- oui si $\pi(s,a) > 0 \Rightarrow \pi'(s,a) > 0$ ⇨ couverture

Evaluer une politique tout en en suivant une autre

- Supposons qu'on suive une politique π' (ou bien qu'on a collecté des données)
- Peut-on calculer v_π pour une autre politique π ?
- oui si $\pi(s,a) > 0 \Rightarrow \pi'(s,a) > 0$ ⇨ couverture
- l'idée est d'utiliser les techniques d'échantillonnage préférentiel ("importance sampling").
  - On a obtenu une trace d'exécution avec π'
  - On a obtenu des récompenses $r_{t+1}, r_{t+2}, \dots, r_T$.
  - Le gain $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots$ ne vaut que pour v_π et non $v_{\pi'}$
  - ? Peut-on ajouter un poid sur les récompenses pour estimer ce que serait le gain pour de trace si elle avait été générée par π et non π' ?

Etant donné un état de départ S_t, la probabilité d'observer une trace d'exécution $S_{t+1}, A_{t+1}, S_{t+2}, A_{t+2}, \ldots, S_T$ en suivant une politique π est

$$\mathbb{P}(A_t, S_{t+1}, A_{t+1}, S_{t+2}, A_{t+2}, \ldots, S_T \mid S_t, \pi)$$

$$\begin{aligned}
&= \pi(A_t \mid S_t) T_{S_t, S_{t+1}}^{A_t} \pi(A_{t+1} \mid S_{t+1}) \ldots T_{S_{T-1}, S_T}^{A_{T-1}} \\
&= \prod_{k=t}^{T-1} \pi(A_k \mid S_k) T_{S_k, S_{k+1}}^{A_k}
\end{aligned}$$

Etant donné un état de départ S_t, la probabilité d'observer une trace d'exécution $S_{t+1}, A_{t+1}, S_{t+2}, A_{t+2}, \ldots, S_T$ en suivant une politique π est

$$\begin{aligned}
&\mathbb{P}(A_t, S_{t+1}, A_{t+1}, S_{t+2}, A_{t+2}, \ldots, S_T \mid S_t, \pi) \\
&\qquad = \pi(A_t \mid S_t) T^{A_t}_{S_t, S_{t+1}} \pi(A_{t+1} \mid S_{t+1}) \ldots T^{A_{T-1}}_{S_{T-1}, S_T} \\
&\qquad = \prod_{k=t}^{T-1} \pi(A_k \mid S_k) T^{A_k}_{S_k, S_{k+1}}
\end{aligned}$$

La probabilité relative de la trace d'exécution (i.e. le ratio d'échantillonnage préférentiel) est

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k) T^{A_k}_{S_k, S_{k+1}}}{\prod_{k=t}^{T-1} \pi'(A_k \mid S_k) T^{A_k}_{S_k, S_{k+1}}} = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} \pi'(A_k \mid S_k)}$$

et il est donc indépendant du PDM !

Evaluer une politique tout en en suivant une autre

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

On estime $v_{\pi'}(s)$ la moyenne des retours G_t en suivant la politique π'

Pour estimer $v_\pi(s)$, on peut utiliser

$$v_\pi(s) = \mathbb{E}\left[\rho_{t:T-1} G_t \mid S_t = s\right]$$

Echantillonnage préférentiel "ordinaire"

Notations :

- On numérote chaque itération, indépendamment des épisodes
  ex : deux épisodes, le premier se termine au bout de 3 itérations, le second au bout de 4 itérations, chaque itération est identifiée par un entier dans $\{1, \ldots, 7\}$
- $\mathcal{T}(s)$
  - variante "chaque visite" : toutes les itérations où l'état courant est s.
  - variante "première visite" : $\mathcal{T}(s)$ chaque première itération d'un épisode où l'état courant est s.
- $T(t)$ représente l'itération où l'épisode dont t fait partie se termine.

Pour calculer une estimation $\tilde{v}$ de v_π, il suffit de calculer la moyenne

$$\tilde{v}(s) = \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}$$

Echantillonnage préférentiel pondéré

Pour calculer une estimation $\tilde{v}$ de v_π, on peut aussi calculer la moyenne pondérée

$$\tilde{v}(s) = \frac{\displaystyle\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{\displaystyle\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1}}$$

Echantillonnage préférentiel : ordinaire ou pondéré

- Pour la version "première visite"
  - variante ordinaire : non biaisée mais la variance peut être grande (voir infinie)
  - variante pondérée : biaisée, si les gains sont bornés, la variance converge vers 0
  - en pratique, la variante pondérée à une variance bien plus faible et est préférée.
- Pour la version "chaque visite", dans les deux cas, il y a un biais, mais il tend vers 0 avec le nombre d'échantillons.
  ➡ plus simple à implémenter.

Precup, Sutton, Dasgupta (2001) Off policy temporal-difference learning with function approximation, *Proceedings of ICML 2001*

Implémentation incrémentale

Comme pour les *bandits*, on ne veut pas stocker en mémoire toutes les informations pour calculer les moyennes !

Pour la version *ordinaire*, on peut faire comme dans les bandits, en utilisant les gains pondérés à la place de la récompense immédiate

Pour la version *pondérée*, on doit travailler un peu plus.

- Pour un état s donné, on a les valeurs de gains $G_1, G_2, \ldots, G_{n-1}$.
- pour simplifier $w_i = \rho_{t_i:T(t_i)-1}$

➩ supposons qu'on a $V_{n-1} = \frac{\sum_{i=1}^{n-1} w_i G_i}{\sum_{i=1}^{n-1} w_i}$

- On va maintenir, pour chaque état la somme des poids $C_n = \sum_{i=1}^{n} w_i$
- on a alors

$$V_n = V_{n-1} + \frac{w_n}{C_{n-1}} [G_n - V_{n-1}]$$

Evaluation "Off-Policy" par Monte Carlo

Input : an arbitrary target policy π
Initialize $Q(s,a) \in \mathbb{R}$
Initialize $C(s,a) = 0$

Répète éternellement
    $\pi' \leftarrow$ any policy with coverage of π.
    Simule un épisode en suivant la politique π'
    $G \leftarrow 0$
    $W \leftarrow 1$
    Pour chaque état s dans l'épisode et tant que $W \neq 0$
        $G \leftarrow \gamma G + R_{t+1}$
        $C(S_t,A_t) \leftarrow C(S_t,A_t) + W$
        $Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \dfrac{W}{C(S_t,A_t)}\left[G - Q(S_t,A_t)\right]$
        $W \leftarrow W \dfrac{\pi(A_t|S_t)}{\pi'(A_t|S_t)}$

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

On sépare ici

- la politique du comportement courant
- la politique que l'on cherche à optimiser : la politique estimée
- On utilise une politique gloutonne pour améliorer la politique estimée
- le choix de la politique courante va rendre la convergence plus ou moins rapide

Evaluation "Off-Policy" par Monte Carlo

Input : an arbitrary target policy π
Initialize $Q(s,a) \in \mathbb{R}$
Initialize $C(s,a) = 0$

Répète éternellement
- $\pi' \leftarrow$ any policy with coverage of π.
- Simule un épisode en suivant la politique π'
- $G \leftarrow 0$
- $W \leftarrow 1$
- Pour chaque état s dans l'épisode et tant que $W \neq 0$
  - $G \leftarrow \gamma G + R_{t+1}$
  - $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \dfrac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
  - $\pi(S_t) \leftarrow arg\max_a Q(S_t, a)$
  - Si $A_t \neq \pi(S_t)$, on stop l'épisode et passe au suivant
  - $W \leftarrow W \dfrac{\pi(A_t|S_t)}{\pi'(A_t|S_t)}$

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'effectuent pas de "bootstrap"