Apprentissage par renforcement

Stéphane Airiau

Université Paris Dauphine

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"

1 $v \in \mathbb{R}^n$
2 $n \in \mathbb{N}^n$
3 $Acc \in \mathbb{R}^n$
4 initialise $v(s) = 0$ pour chaque état $s \in S$
6 initialise $n(s) = 0$ pour chaque état $s \in S$
7 initialise $Acc(s) = 0$ pour chaque état $s \in S$
8
9 Répète éternellement
10 Simule un épisode en suivant la politique π
11 Pour chaque état s qui apparait dans l'épisode
12 Pour la **première itération** t où s est visité dans l'épisode
13 $Acc(s) \leftarrow Acc(s) + G_t$
14 $n(s) \leftarrow n(s) + 1$
15 $v(s) \leftarrow \dfrac{Acc(s)}{n(s)}$

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

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

Méthode Monte Carlo : Evaluation d'une politique

Algorithme "chaque visite"

1 $v \in \mathbb{R}^n$
2 $n \in \mathbb{N}^n$
3 $Acc \in \mathbb{R}^n$
4 initialise $v(s) = 0$ pour chaque état $s \in S$
6 initialise $n(s) = 0$ pour chaque état $s \in S$
7 initialise $Acc(s) = 0$ pour chaque état $s \in S$
8
9 Répète éternellement
10 Simule un épisode en suivant la politique π
11 Pour **chaque itération** t qui visite l'état s
12 $Acc(s) \leftarrow Acc(s) + G_t$
13 $n(s) \leftarrow n(s) + 1$
14 $v(s) \leftarrow \frac{Acc(s)}{n(s)}$

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

(Singh & Barto, 1996)

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 actions

- 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 !

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

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

⇨ même stratégie "première visite" et "chaque visite" possible

Evaluation de la valeur des actions

- 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 !

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

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

⇨ même stratégie "première visite" et "chaque visite" possible

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`

Evaluation de la valeur des actions

- 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 !

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

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

⇨ même stratégie "première visite" et "chaque visite" possible

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`

- une stratégie : "exploring starts"
  on tire au hasard une paire `(état,action)` pour l'état initial et on utilise "première visite"

Evaluation de la valeur des actions

- 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 !

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

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

⇨ même stratégie "première visite" et "chaque visite" possible

- petit problème : si π est déterministe, on n'a pas la valeur de toutes les paires `(état,action)`

- une stratégie : "exploring starts"
  on tire au hasard une paire `(état,action)` pour l'état initial et on utilise "première visite"
- Evidemment, ceci est problématique pour des interactions avec un environnement réel (on ne peut pas forcément choisir l'état initial ! !)

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

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

  - la fonction de valeur approxime de mieux en mieux la politique courante
  - mais on améliore la politique courante

  ⇨ à la limite, le processus va converger vers l'optimal

- utiliser 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 : $\pi_{k+1}(s) = \text{argmax}_a\, q_k(s,a)$

⇨ même garantie de convergence que pour itération des politiques

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 $q_\pi(s,a)$
2. plus extrème : utiliser seulement un épisode avant de faire une amélioration.

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

$q \in \mathbb{R}^{n \times m}$
$n \in \mathbb{N}^{n \times m}$
$Acc \in \mathbb{R}^{n \times m}$
initialise $v(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$
initialise $n(s,a) = 0$ 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 ocurrence de (s,a) est à l'instant t
            $Acc(s,a) \leftarrow Acc(s,a) + G_t$
            $n(s,a) \leftarrow n(s,a) + 1$
            $q(s,a) \leftarrow \frac{Acc(s,a)}{n(s,a)}$
        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 empiriquement, ça marche!

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 est en train d'apprendre "on policy"
- soit on va utiliser une politique pour explorer et apprendre une politique optimale (qu'on ne suit pas encore) "off policy"

"on-policy" Monte Carlo

- estime et améliorer la 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

ex : ϵ-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"

$q : S \times A \to \mathbb{R}$
$n : S \times A \to \mathbb{N}$
$Acc : S \times A \to \mathbb{R}$
$\pi : S \to \Delta(A)$
initialise $n(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$
initialise $q(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$
initialise $Acc(s,a) = 0$ pour chaque état $s \in S$ et action $a \in A$
initialise $\pi(s,a) > 0$ pour chaque état $s \in S$ et action $a \in A$

Répète éternellement
    Simule un épisode en suivant la politique π
    Pour **chaque paire** (s,a) qui est visitée dans l'épisode
        Si la première ocurrence de (s,a) est à l'instant t
            $Acc(s,a) \leftarrow Acc(s,a) + G_t$
            $n(s,a) \leftarrow n(s,a) + 1$
            $q(s,a) \leftarrow \frac{Acc(s,a)}{n(s,a)}$
        Pour chaque état s dans l'épisode
            $a^\star \leftarrow arg\max_{a \in A} q(s,a)$
            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...

on fera ça un jour...

Evaluer une politique tout en en suivant une autre

- Supposons qu'on suive une politique π'.
- 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'.
- Peut-on calculer v_π pour une autre politique π ?
- oui si $\pi(s,a) > 0 \Rightarrow \pi'(s,a) > 0$

Evaluer une politique tout en en suivant une autre

- Supposons qu'on suive une politique π'.
- Peut-on calculer v_π pour une autre politique π ?
- oui si $\pi(s,a) > 0 \Rightarrow \pi'(s,a) > 0$

- Soit t le moment où on visite pour la première fois l'état s
- Soit R_t la récompense à long terme observée
- Soit $p_t(s)$ et $p'_t(s)$ les probabilités que cette séquence soit effectivement la séquence d'états visités à partir de s en suivant π et π'.
- Soit n_s le nombre d'observation pour l'état s (i.e. le nombre d'épisode où s a été visité)

$$v_\pi(s) = \frac{\sum_{i=1}^{n_s} \frac{p_i(s)}{p'_i(s)} R_i(s)}{\sum_{i=1}^{n_s} \frac{p_i(s)}{p'_i(s)}}$$

Evaluer une politique tout en en suivant une autre

A priori, il faut connaître $p_i(s)$ et $p'_i(s)$.
En fait, il suffit de connaître les deux politiques π et π'.

En effet :
Soit $T_i(s)$ l'itération de fin du $i^{ième}$ épisode où s est visité.

$$\frac{p_i(s)}{p'_i(s)} = \frac{\prod_{t=1}^{T_i(s)-1} \pi(a_k|s_k) T^{a_k}_{s_k s_{k+1}}}{\prod_{t=1}^{T_i(s)-1} \pi'(a_k|s_k) T^{a_k}_{s_k s_{k+1}}} = \prod_{t=1}^{T_i(s)-1} \frac{\pi(a_k|s_k)}{\pi'(a_k|s_k)}$$

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

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 la technique précédente pour améliorer la politique estimée
- le choix de la politique courante va rendre la convergence plus ou moins rapide

Temporal-difference Learning

Combine des idées de
la programmation dynamique (DP)
avec
les méthodes de Monte Carlo (MC)

Méthodes "Temporal-difference"

- elles apprennent directement avec l'expérience (comme MC)
- elles sont sans modèle : pas besoin de connaître les modèles de transition ou de récompenses (comme MC)
- elles peuvent apprendre d'épisodes incomplets (comme PD)
- elles utilisent des estimations pour mettre à jour son estimation (comme DP)

Méthodes "Temporal-difference"pour la fonction de valeurs

On veut estimer la valeur des états pour une politique fixe π.

- Méthode Monte Carlo "chaque visite"
  - mise à jour à l'aide du véritable gain G_t

$$v(s_t) \leftarrow v(s_t) + \alpha(G_t - v(s_t))$$

  - G_t est accessible à la fin de l'épisode ⇨ peut on éviter cette atttente ?
- Méthode la plus simple : TD(0)

$$v(s_t) \leftarrow v(s_t) + \alpha\,[r_{t+1} + \gamma v(s_{t+1}) - v(s_t)]$$

mise à jour à l'aide de $r_{t+1} + \gamma v(s_{t+1})$

$$\begin{aligned}
\text{Rappel : } v_\pi(s) &= \mathbb{E}_\pi[G_t \mid s_t = s] \\
&= \mathbb{E}_\pi[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s] \\
&= \mathbb{E}_\pi[r_{t+1} + \gamma v_\pi(s_{t+1}) \mid s_t = s]
\end{aligned}$$

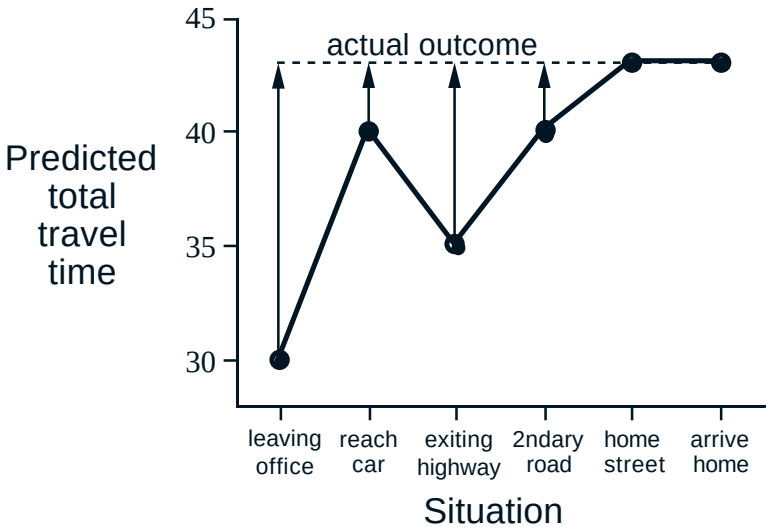
exemple du temps de trajet

Etat	temps écoulé	temps estimé	temps total prédit
départ du bureau	0	30	30
arrivée à la voiture, il pleut	5	35	40
sortie de l'autoroute	20	15	35
camion lent	30	10	40
arrivée dans le quartier	40	3	43
arrivée à la maison	43	0	43

exemple du temps de trajet

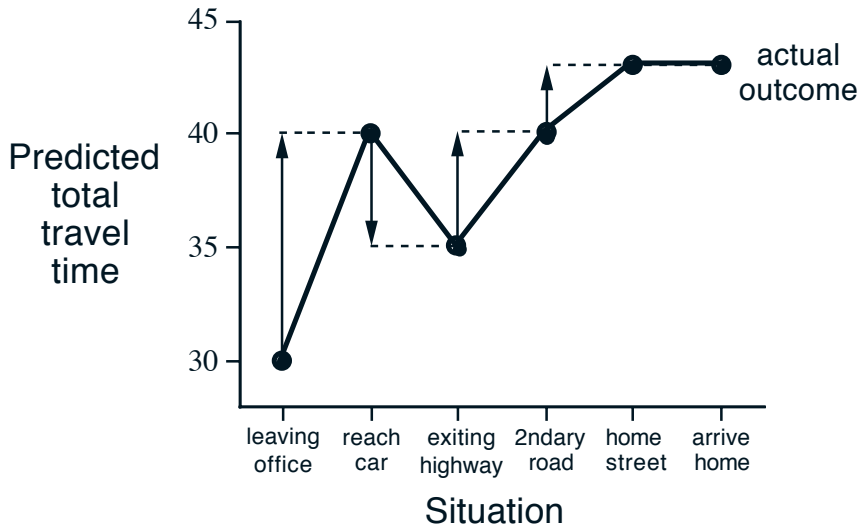
update avec méthode de Monte Carlo

$\alpha = 1$



update avec TD(0)

$\alpha = 1$



Avantages des méthodes TD

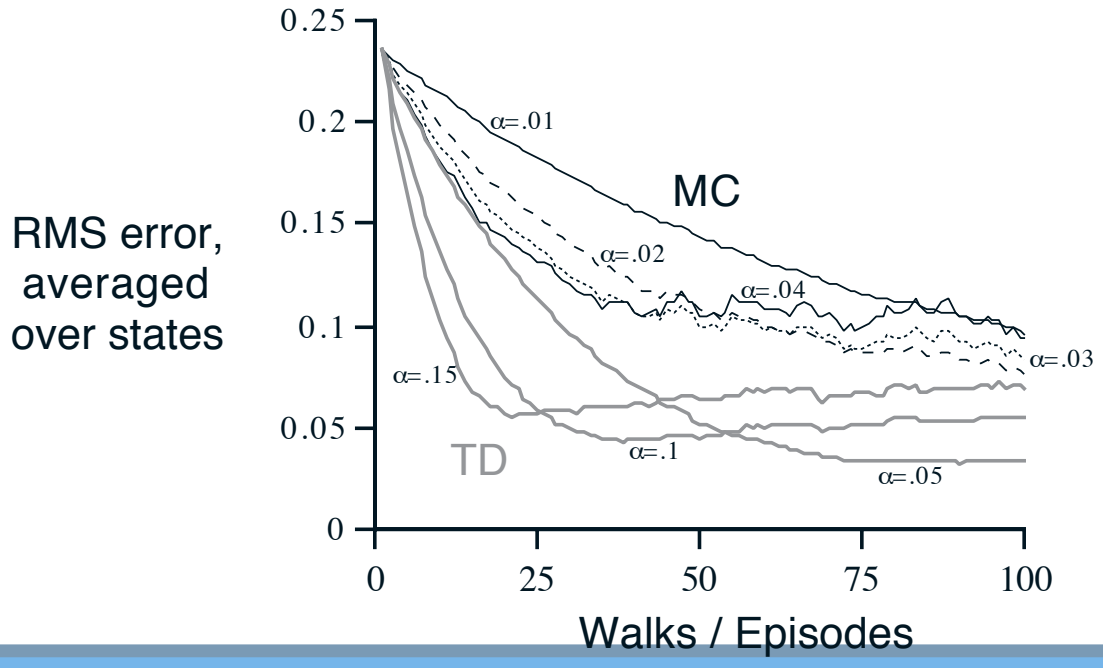
- pas besoin de connaissances des modèles
- méthode adaptée pour une utilisation online, et pas besoin d'attendre la fin de l'épisode
  les méthodes TD font une mise à jour après chaque itération
- il y a des garanties théoriques de convergence

Avantages des méthodes TD

- Pas de résultats théoriques comparant les performances des méthodes TD aux méthodes Monte Carlo.

⇨ en pratique, les méthodes TD sont plus rapides sur des problèmes stochastiques.

équiprobabilité d'aller à gauche ou à droite.



Exemple intuitif

On a un PDM a deux états A et B. Supposons qu'on observe les huit épisodes suivants :

$A,0,B,0$	$B,1$
$B,1$	$B,1$
$B,1$	$B,1$
$B,1$	$B,0$

Quel est votre évaluation pour $v(A)$ et $v(B)$?

Evaluation de la politique optimale : TD "on policy"

- On apprend la fonction de valeur des actions.
- Comme pour TD(0) pour la fonction de valeurs, on a :

$$q(s_t,a_t) \leftarrow q(s_t,a_t) + \alpha\left[r_{t+1} + \gamma q(s_{t+1},a_{t+1}) - q(s_t,a_t)\right]$$

State–action–reward–state–action (SARSA)

Initialise $q(s) \in \Delta(A)$ arbitrairement
Répète (éternellement) pour chaque épisode
    aller à l'état initial s
    choisir action $a \in A$ pour s à l'aide d'une politique dérivée de q (ex : ϵ-greedy)
    Répète pour chaque étape de l'épisode
        Exécute action a, observe $r \in \mathbb{R}$ et état suivant $s' \in S$
        choisir action $a' \in A$ pour s' à l'aide d'une politique dérivée de q
        $q(s,a) \leftarrow q(s,a) + \alpha\left[r + \gamma q(s',a') - q(s,a)\right]$
        $s \leftarrow s'$
        $a \leftarrow a'$
    jusqu'à ce que s soit terminal

Evaluation de la politique optimale : TD "on policy"

Théorème

L'algorithme converge vers la fonction optimale de valeur des actions sous les conditions suivantes :

- Glouton à la Limite avec Exploration Infinie
  - toutes les paires `(état,action)` sont explorées infiniement souvent $\lim_{k\to\infty} n_k(s,a) = \infty$
  - la politique converge vers une politique gloutonne $\lim_{k\to\infty} \pi_k(a|s) = 1$ pour $a = arg\max_{a' \in A} q(s,a')$
- $\sum_{t=1}^{\infty} \alpha_t = \infty$
- $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$

ϵ-greedy est GLEI si ϵ est une fonction décroissante (ex $e_k = \frac{1}{k}$)

Evaluation de la politique optimale : TD "off policy"

Q-learning (Watkins 1989)

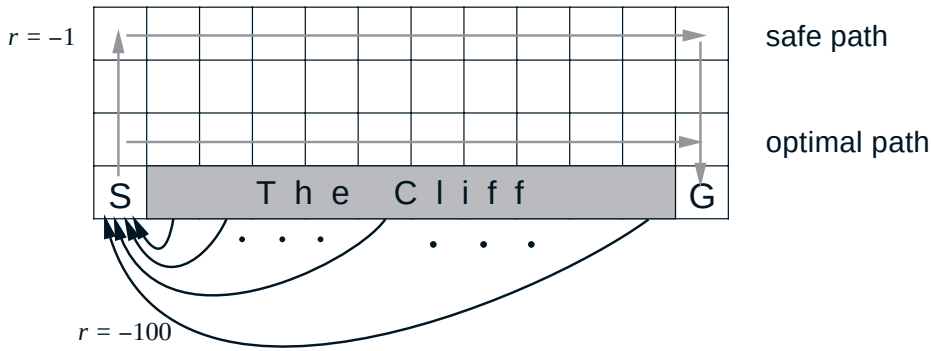
$$q(s_t,a_t) \leftarrow q(s_t,a_t) + \alpha \left[r_{t+1} + \gamma \max_{a \in A} q(s_{t+1},a) - q(s_t,a_t) \right]$$

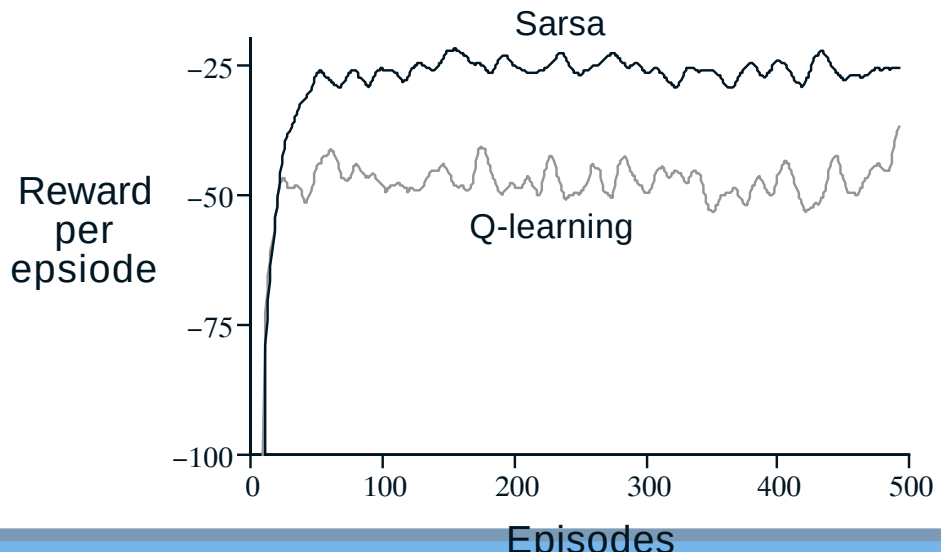
Apprend une estimation de $q^\star$ de façon indépendante à la politique suivie.

1. Initialise $q(s) \in \Delta(A)$ arbitrairement
2. Répète (éternellement) pour chaque épisode
3. aller à l'état initial s
4. choisir action $a \in A$ pour s à l'aide d'une politique dérivée de q (ex : ϵ-greedy)
5. Répète pour chaque étape de l'épisode
6. Exécute action a, observe $r \in \mathbb{R}$ et état suivant $s' \in S$
7. choisir action $a' \in A$ pour s' à l'aide d'une politique dérivée de q
8. $q(s,a) \leftarrow q(s,a) + \alpha [r + \gamma \max_{a" \in A} q(s',a") - q(s,a)]$
9. $s \leftarrow s'$
10. $a \leftarrow a'$
11. jusqu'à ce que s soit terminal

Evaluation de la politique optimale : TD "off policy"

- Pour assurer la convergence, il faut s'assurer de visiter suffisemment souvent les paires `(action,état)`.
- sous les hypothèses GLIE, Q-learning converge





Autres méthodes d'exploration

- soft max : choisir l'action a avec probabilité

$$\frac{e^{\frac{q_t(s,a)}{\tau}}}{\sum_{a' \in A} e^{\frac{q_t(s,a')}{\tau}}}$$

  - $\tau > 0$ est appelée la température
  - température haute ⇨ probabilité uniforme
  - température basse ⇨approche le comportement glouton
- initialisation optimiste : initialiser les valeurs de manière optimiste puis être glouton (Even-Dar & Mansour, NIPS 1994)
  ⇨ force l'exploration à regarder les états qui semblent prometteur

Dilemme central : explorer ou exploiter

- contrairement au cas supervisé, les données sur lesquelles on travaille dépendent du comportement de l'agent !
- exploration : le but est d'apprendre le mieux possible
- exploitaiton : le but est d'optimiser au mieux ses récompenses
- défi de l'exploration : quelles actions vont améliorer au plus vite la connaissance de l'agent pour obtenir de meilleures récompenses.

⇨ l'exploration est un trait d'intelligence

- quelle politique doit suivre l'agent pour ne pas manquer les états qui donnent les bonnes récompenses (et sans passer trop de temps dans les états qui donnent de mauvaises récompenses)
- exploitation : préfère des actions qui ont mené a de "bons états"
- exploration : prendre une action qui pourrait nous mener à de bons états.

Stratégie d'exploration

- ϵ-greedy
  - facile à implémenter et très utilisée
  - convergence garantie (avec un taux d'exploration qui décroit de bonne manière)
  - il faut un nombre exponentiel d'échantillons pour garantir convergence
- Boltzmann
  - même problème pour le nombre d'échantillons

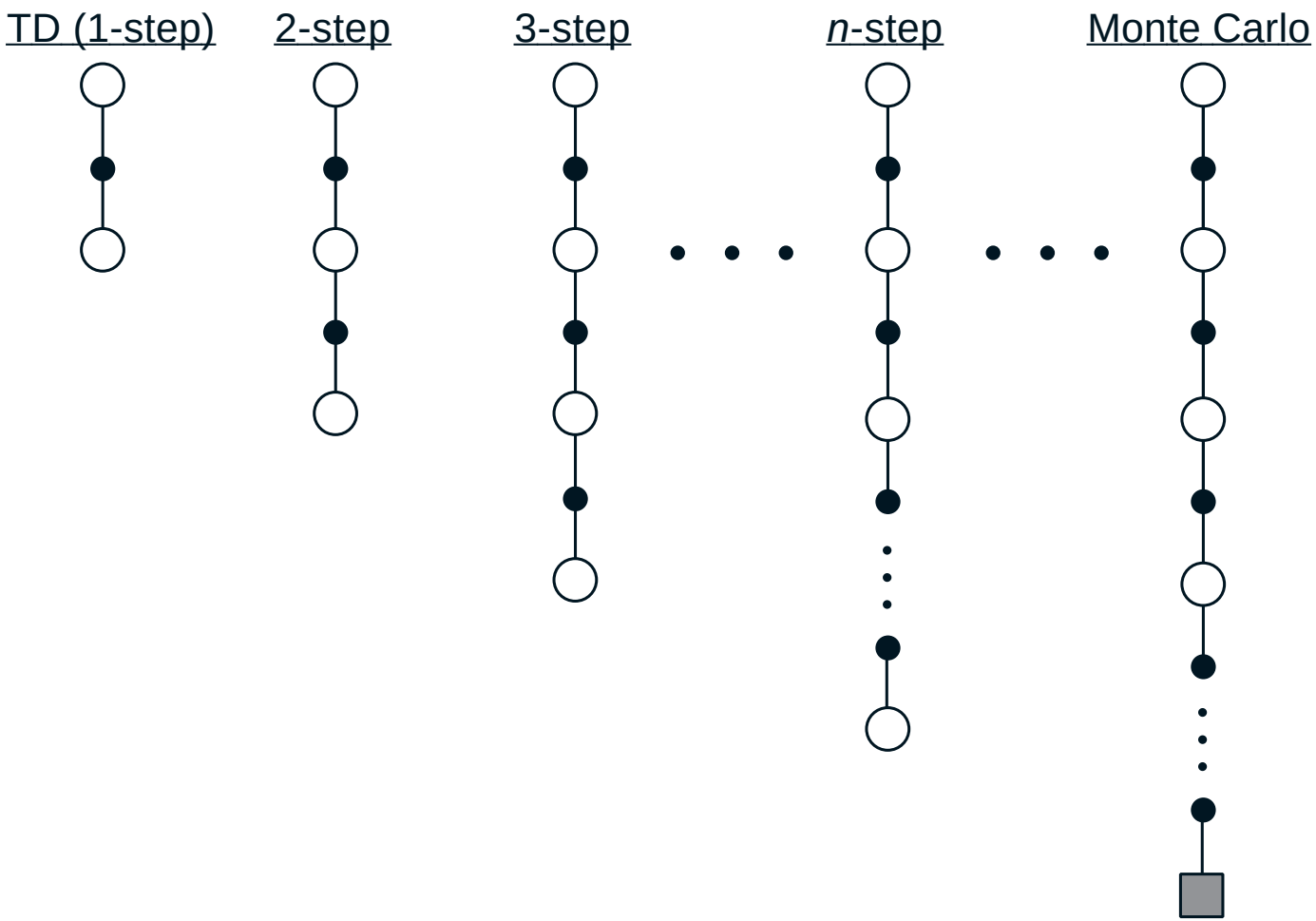
Explicit-Exploit-or-Explore E^3 (Kearns & singh 2002)

- Construit un modèle de PDM
  - optimiste
  - connait la récompense maximales
- compte le nombre de fois où chaque paire `(action,état)` a été visité pour estimer la qualité du modèle
  - ⇨ on connait l'état si on l'a visité un certain nombre de fois
  - Garantie statistique pour définir le nombre "suffisant" de viste
    $\mathcal{O}((NTG_{max}^{T}/\epsilon)^4 \mathrm{Var}_{max} log(\frac{1}{\delta}))$ ou Var_{max} est la variance maximale des récompenses sur tous les états
- E^3 gère deux modèles : le modèle avec les états connus, et celui avec les états par encore connu.
  connu ⇨ exploitation
  pas encore connu ⇨ exploration

R_{max} (Brafman & Tennenholtz 2002)

- utilise la même idée, mais avec un seul modèle
- initialement, tout est inconnu, on estime avec une valeur maximale R_{max} toutes les récompenses, le modèle de transition est estimé déterministe
- on compte aussi les transitions observées pour déterminer ensuite lesquelles sont connues
- on calcule une politique optimale pour ce qu'on connait jusqu'ici
  - si l'état était connu ⇨ exploitation
  - si l'état n'était pas connu ⇨ on explore l'état le plus prometteur

Entre TD(0) et Monte Carlo : TD(n)



baser la mise à jour après quelques récompenses
(entre 1 et la fin de l'épisode)

Entre TD(0) et Monte Carlo : TD(n)

$$
\begin{aligned}
R_t^{(1)} &= r_{t+1} + \gamma V_t(S_{t+1}) \\
R_t^{(2)} &= r_{t+1} + \gamma r_{t+1} + \gamma^2 V_t(S_{t+2}) \\
&\cdots \\
R_t^{(n)} &= r_{t+1} + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V_t(S_{t+n})
\end{aligned}
$$

- pour un algorithme, à t, on devrait mettre à jour la valeur de l'état visité à $t-n$

⇨ lors des $n-1$ premières étapes, pas de mise à jour

⇨ mise à jour à chaque étape de l'état visité il y a n étapes

⇨ lorsqu'on atteint la fin de l'épisode, on met à jour les n étapes précédant la fin

Entre TD(0) et Monte Carlo : TD(n)

⇨ Monte Carlo devient un cas spécial où n est suffisemment grand.

Quel n permet d'apprendre le plus vite ?

⇨ Méthodes intéressantes théoriquement (bonnes propriétés de réduction d'erreurs), mais plus difficiles à implémenter.

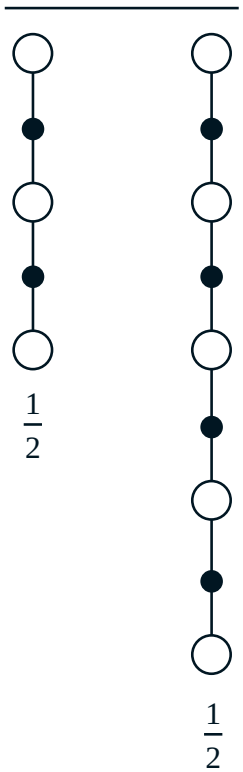
Sarsa à n étapes

On utilise cette idée pour la fonction q, en utilisant ϵ-greedy.

$$q_{t+n}(s_t, a_t) = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n \sum_{a \in A} \pi(s|S_{t+n}) q_{t+n-1}(s_{t+n}, a)$$

autre idée pour utiliser TD(0), TD(1), ..., TD(n)

On peut aussi penser à "mixer" les récompenses à plus ou moins long terme.

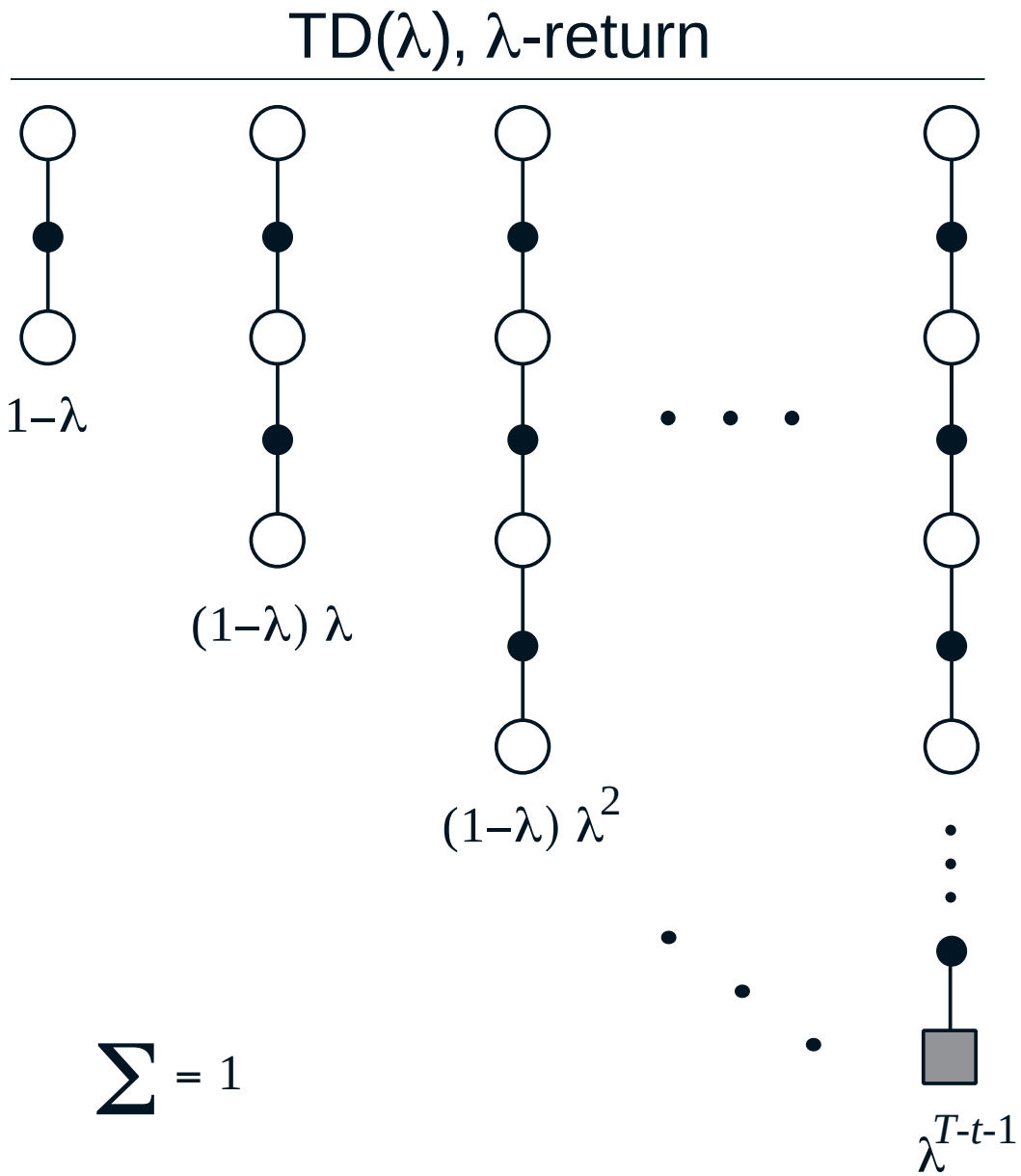


$$R_t^{moy} = \frac{1}{2} R_t^{(2)} + \frac{1}{2} R_t^{(4)}$$

autre idée pour utiliser TD(0), TD(1), …, TD(n)

- On peut donner des poids différents à chaque gain
  (ça marche même pour un ensemble infini)
- Tant que la somme des poids est 1.
- On a une propriété de réduction d'erreurs.

Un mixage particulier : TD(λ)



Un mixage particulier : TD(λ)

- $\lambda \in [0,1]$ similaire au paramètre de dévaluation.

⇨ A chaque étape suivante, le poids décroit avec le facteur λ

- le gain à n étapes possède un poids proportionnel à λ^{n-1}
- $\sum_{n=0}^{\infty} \lambda^n = \frac{1}{1-\lambda}$

⇨ facteur $1-\lambda$ pour que les poids somment à 1

- $R_t^\lambda = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)}$
- $\lambda = 1$ ⇨ Monte Carlo
- $\lambda = 0$ ⇨ TD(0)

On peut donc bâtir des algorithmes plus compliqués

mais c'est assez pour le moment !

Résumé

Types d'algorithmes pour

- évaluer une politique donnée
- trouver une politique optimale

Ce qu'on apprend

- $v_\pi : S \to \mathbb{R}$ la fonction de valeurs qui donne la valeur **à long terme** de chaque état en suivant une politique
- $\pi : A \times S \to \mathbb{R}$ la fonction qui donne la valeur **à long terme** de prendre une action puis de suivre une politique π depuis un état.

Algorithmes :

- modèle de transition et de récompenses connus ⇨ policy/value iteration
- Algorithmes pour domaines épisodique ⇨ méthodes de Monte Carlo
- Algorithme qui fonctione sans connaître ni bâtir un modèle Attention au dilemme Exploration Vs Exploitation ⇨ SARSA , Q-learning
- Algorithme qui fonctionne en bâtissant un modèle ⇨ E^3, R_{max}
- on policy : on améliore la politique que l'on suit / off policy : on utilise une stratégie pour apprendre une autre