Apprentissage par renforcement

Cours 4: méthodes de différences temporelles

Stéphane Airiau

Université Paris Dauphine

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.

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

- Méthode Monte Carlo "chaque visite"

Méthodes "Temporal-difference"pour la fonction de valeurs

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

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

- Méthode Monte Carlo "chaque visite"
  - estimation à l'aide du véritable gain G_t obtenu lors d'un épisode

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

Méthodes "Temporal-difference"pour la fonction de valeurs

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

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

- Méthode Monte Carlo "chaque visite"
  - estimation à l'aide du véritable gain G_t obtenu lors d'un épisode

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

  - G_t est accessible à la fin de l'épisode ➪ peut on éviter cette attente ?

Méthodes "Temporal-difference"pour la fonction de valeurs

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

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

- Méthode Monte Carlo "chaque visite"
  - estimation à l'aide du véritable gain G_t obtenu lors d'un épisode

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

  - G_t est accessible à la fin de l'épisode ➪ peut on éviter cette attente ?
- Autre méthode 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 $r_{t+1} + \gamma v(s_{t+1})$

Méthodes "Temporal-difference"pour la fonction de valeurs

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

Méthode de différences temporelles : on calcule une erreur entre l'estimation $v(s_t)$ et l'estimation $r + \gamma v(s_{t+1})$.

et on voit apparaître une différence entre deux temps $t+1$ et t.

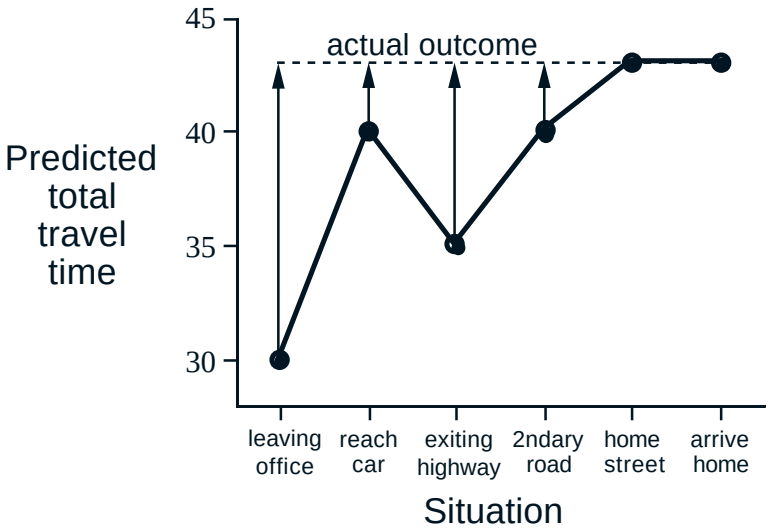
exemple du temps de trajet

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

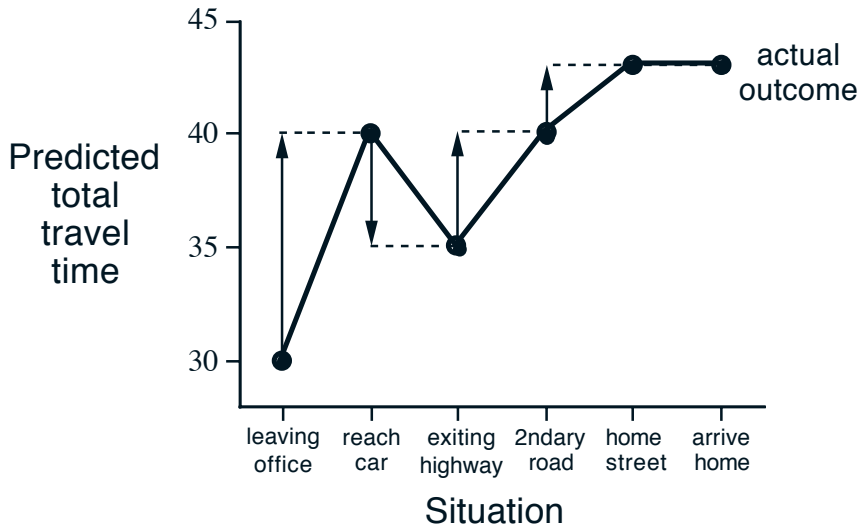
update avec méthode de Monte Carlo

$\alpha = 1$



update avec TD(0)

$\alpha = 1$



Avantages des méthodes TD

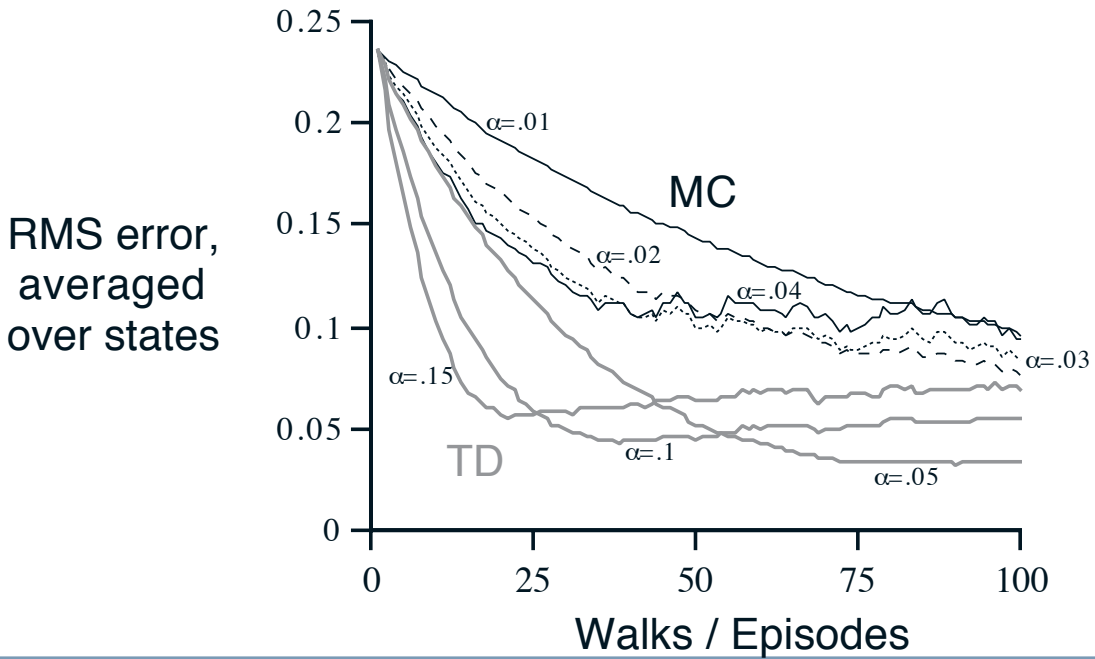
- pas besoin de connaissances des modèles
- méthode adaptée pour une utilisation online, et pas besoin d'attendre la fin de l'épisode
  les méthodes TD font une mise à jour après chaque itération
- il y a des garanties théoriques de convergence

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 :

$$\begin{array}{ll} B,0 & B,1 \\ B,1 & B,1 \\ B,1 & B,1 \\ B,1 & A,0,B,0 \end{array}$$

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 :

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

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

Initialise $q(s,a) \in \mathbb{R}$ arbitrairement (par exemple $q(s,a) = 0$)
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 $s' \in S$
        si s' est final
            $q(s,a) \leftarrow q(s,a) + \alpha\,[r - q(s,a)]$
        sinon
            choisir action $a' \in A$ pour s' à l'aide d'une politique dérivée de q
            $q(s,a) \leftarrow q(s,a) + \alpha\,[r + \gamma q(s',a') - q(s,a)]$
        $s \leftarrow s'$
        $a \leftarrow a'$
    jusqu'à ce que s soit terminal

Evaluation de la politique optimale : TD "on policy"

Théorème

SARSA converge vers la fonction optimale de valeur des actions sous les conditions suivantes :

- Glouton à la Limite avec Exploration Infinie (GLEI)
  - 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,a) ∈ ℝ arbitrairement (par exemple q(s,a) = 0)
2  Répète (éternellement) pour chaque épisode
3      aller à l'état initial s
4      choisir action a ∈ 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 ∈ ℝ et état suivant s' ∈ S
7          si s' est final
8              q(s,a) ← q(s,a) + α[r − q(s,a)]
9          sinon
8              q(s,a) ← q(s,a) + α[r + γ max_{a''∈A} q(s',a'') − q(s,a)]
7              choisir action a' ∈ A pour s' à l'aide d'une politique dérivée de q
9          s ← s'
10         a ← 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 GLEI, Q-learning converge

Théorème

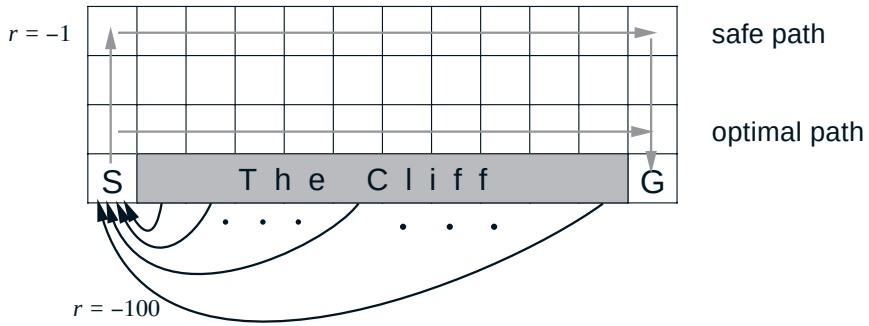
Q-learning converge vers la fonction optimale de valeur des actions sous les conditions suivantes :

- Glouton à la Limite avec Exploration Infinie (GLEI)
  - 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 $a = arg\max_{a' \in A} q(s,a')$
- $\sum_{t=1}^{\infty} \alpha_t = \infty$
- $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$

Comparaison SARSA/Q-learning

Sous les hypothèses GLEI, les deux algorithmes convergent vers une solution optimale. Cependant, ils vont peut-être passer par des "étapes" différentes lors de l'apprentissage.

Dans l'exemple, SARSA va souvent apprendre d'abord une politique sous-optimale avant de trouver la politique optimale. Q-learning va quant à lui trouver rapidement la politique optimale.



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

Résumé

Types d'algorithmes pour

- évaluer une politique donnée
- trouver une politique optimale

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 **mais** le nombre d'échantillons exponentiel est requis pour la convergence.
- d'autres algorithmes existent (double Q-learning, n-step SARSA, expected-SARSA, ...)
- Il existe des algorithmes qui fonctionnent en bâtissant un modèle
  ⇨ E^3, R_{max}
  mais qui nécessitent un nombre d'échantillons polynomial.

Ces méthodes sont des méthodes "tabulaires" : elles supposent que l'on peut stocker dans la table $Q(s,a)$ toutes les valeurs.