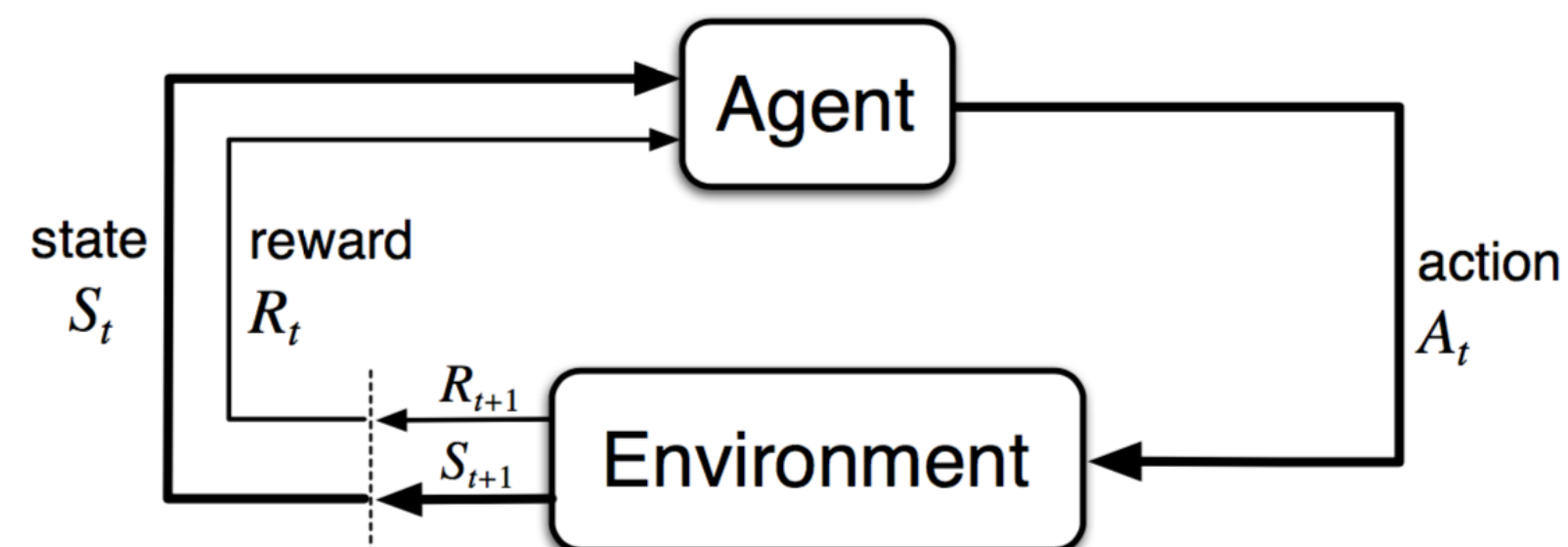


Apprentissage Automatique I MATH60629

Prise de décision séquentielle II
Sommaire
— Semaine #13

Bref rappel

- Markov Decision Processes (MDP)
- Un formalisme pour la prise de décision sous incertitude
 $\langle \mathbf{A}, \mathbf{S}, \mathbf{P}, \mathbf{R}, \gamma \rangle$
- But: trouver une politique optimale
- Programmation dynamique et deux algos (VI,PI)



Des MDPs au RL

- Dans les MDPs on connaît

Des MDPs au RL

- Dans les MDPs on connaît
 1. Probabilités de transition : $P(s' | s, a)$

Des MDPs au RL

- Dans les MDPs on connaît
 1. Probabilités de transition : $P(s' | s, a)$
 2. Fonction de récompense: $R(s)$

Des MDPs au RL

- Dans les MDPs on connaît
 1. Probabilités de transition : $P(s' | s, a)$
 2. Fonction de récompense: $R(s)$
- RL est plus général

Des MDPs au RL

- Dans les MDPs on connaît
 1. Probabilités de transition : $P(s' | s, a)$
 2. Fonction de récompense: $R(s)$
- RL est plus général
 - En RL les deux sont inconnus

Des MDPs au RL

- Dans les MDPs on connaît
 1. Probabilités de transition : $P(s' | s, a)$
 2. Fonction de récompense: $R(s)$
- RL est plus général
 - En RL les deux sont inconnus
 - Un agent de RL doit naviguer dans le monde pour obtenir ces informations

Expérience

A. Apprentissage supervisé:

- On nous donne un jeu de données
- But: maximiser l'objectif sur l'ensemble de test (population)

B. Apprentissage par renforcement (RL)

- Collecte des données en interagissant avec le monde
- But: maximiser la somme des récompenses

Algorithmes pour l'apprentissage par renforcement

Méthode Monte-Carlo

Méthode Monte-Carlo

- Model-free (sans modèle)

Méthode Monte-Carlo

- Model-free (sans modèle)
- Pour les environnements épisodiques
 - Par exemple un jeu de cartes (poker). Un épisode est une partie.
- Mise à jour de la politique après chaque épisode

Méthode Monte-Carlo

- Model-free (sans modèle)
- Pour les environnements épisodiques
 - Par exemple un jeu de cartes (poker). Un épisode est une partie.
 - Mise à jour de la politique après chaque épisode
- Intuition
 - Jouer plusieurs épisodes
 - Joue plusieurs parties (de poker)
 - Calculer la moyenne des récompenses obtenue après chaque état
 - La proportion de victoires à partir de chaque état

First-visit Monte Carlo

- Étant donné une politique
- Calcul la fonction de valeur $V(s)$ pour chaque état

First-visit MC prediction, for estimating $V \approx v_\pi$

Initialize:

$\pi \leftarrow$ policy to be evaluated
 $V \leftarrow$ an arbitrary state-value function
 $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:

Generate an episode using π
For each state s appearing in the episode:
 $G \leftarrow$ the return that follows the first occurrence of s
 Append G to $Returns(s)$
 $V(s) \leftarrow \text{average}(Returns(s))$

[Sutton & Barto,
RL Book, Ch 5]

- Converge à $V_\pi(s)$ quand le nombre de visites par état tend vers l'infini.

First-visit Monte Carlo

- Étant donné une politique
- Calcul la fonction de valeur $V(s)$ pour chaque état

First-visit MC prediction, for estimating $V \approx v_\pi$

Initialize:

$\pi \leftarrow$ policy to be evaluated
 $V \leftarrow$ an arbitrary state-value function
 $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:

Generate an episode using π

For each state s appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s

Append G to $Returns(s)$

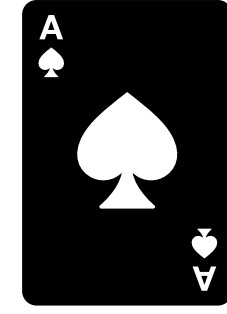
$V(s) \leftarrow \text{average}(Returns(s))$

[Sutton & Barto,
RL Book, Ch 5]

- Converge à $V_\pi(s)$ quand le nombre de visites par état tend vers l'infini.

$$V(s_t) = \max_{a_t} \left\{ R(s_t) + \gamma \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) V(s_{t+1}) \right\}$$

Exemple: Black Jack



- Épisode: une partie
- États: Sommes des cartes du joueur, la carte du croupier, As utile ou non
- Actions: {Rester, Carte}
- Récompense: {Victoire +1, Nulle 0, Défaite -1}
- Autres hypothèses: le paquet est infini

Fonction de valeur Q pour le contrôle

- On connaît déjà la fonction de valeur $V(s)$

Fonction de valeur Q pour le contrôle

- On connaît déjà la fonction de valeur $V(s)$
- Quand les probabilités de transition sont connues, on peut utiliser la fonction de valeur pour obtenir la politique optimale [VI]:

$$\pi^*(s) = \arg \max_a \left\{ R(s) + \gamma \sum_{s'} P(s' | s, a) V^*(s') \right\} \quad \forall s$$

Fonction de valeur Q pour le contrôle

- On connaît déjà la fonction de valeur $V(s)$
- Quand les probabilités de transition sont connues, on peut utiliser la fonction de valeur pour obtenir la politique optimale [VI]:

$$\pi^*(s) = \arg \max_a \left\{ R(s) + \gamma \sum_{s'} P(s' | s, a) V^*(s') \right\} \quad \forall s$$

- Que faire quand les transitions sont inconnues?

Fonction de valeur Q pour le contrôle

- On connaît déjà la fonction de valeur $V(s)$
- Quand les probabilités de transition sont connues, on peut utiliser la fonction de valeur pour obtenir la politique optimale [VI]:

$$\pi^*(s) = \arg \max_a \left\{ R(s) + \gamma \sum_{s'} P(s' | s, a) V^*(s') \right\} \quad \forall s$$

- Que faire quand les transitions sont inconnues?
- $Q(s,a)$ la fonction de valeur pour une paire (état, action) $\pi^*(s) = \arg \max_a \{ Q^*(s, a) \} \quad \forall s$

Monte-Carlo sans “exploring starts”

On-policy first-visit MC control (for ε -soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ε -soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each s in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

[Sutton & Barto,
RL Book, Ch.5]

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$\pi(s) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

Repeat forever:

Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0

Generate an episode starting from S_0, A_0 , following π

For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

For each s in the episode:

$\pi(s) \leftarrow \arg \max_a Q(s, a)$

Monte-Carlo sans “exploring starts”

On-policy first-visit MC control (for ϵ -soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ϵ -soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each s in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all $a \in \mathcal{A}(s)$:

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

[Sutton & Barto,
RL Book, Ch.5]

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$\pi(s) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

Repeat forever:

Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0

Generate an episode starting from S_0, A_0 , following π

For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

For each s in the episode:

$\pi(s) \leftarrow \arg \max_a Q(s, a)$

Monte-Carlo sans “exploring starts”

On-policy first-visit MC control (for ϵ -soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ϵ -soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each s in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all $a \in \mathcal{A}(s)$:

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

[Sutton & Barto,
RL Book, Ch.5]

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$\pi(s) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

Repeat forever:

Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0

Generate an episode starting from S_0, A_0 , following π

For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

For each s in the episode:

$\pi(s) \leftarrow \arg \max_a Q(s, a)$

Monte-Carlo sans “exploring starts”

On-policy first-visit MC control (for ε -soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ε -soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each s in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all $a \in \mathcal{A}(s)$:

$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$

[Sutton & Barto,
RL Book, Ch.5]

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$\pi(s) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

Repeat forever:

Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0

Generate an episode starting from S_0, A_0 , following π

For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

For each s in the episode:

$\pi(s) \leftarrow \arg \max_a Q(s, a)$

- Policy value cannot decrease

$$V_{\pi'}(s) \geq V_{\pi}(s), \forall s \in \mathcal{S}$$

π : policy at current step

π' : policy at next step

Sommaires des méthodes Monte-Carlo

- Permettent d'apprendre une politique à travers les interactions
- (N'apprennent pas les prob. de transition)
- Les états sont traités comme étant indépendants
- Épisodique (avec ou sans “exploring starts”)

TD pour le contrôle

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Choose A from S using policy derived from Q (e.g., ε -greedy)

 Loop for each step of episode:

 Take action A , observe R, S'

 Choose A' from S' using policy derived from Q (e.g., ε -greedy)

$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$

$S \leftarrow S'; A \leftarrow A';$

 until S is terminal

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated

Initialize $V(s)$ arbitrarily (e.g., $V(s) = 0$, for all $s \in \mathcal{S}^+$)

Repeat (for each episode):

 Initialize S

 Repeat (for each step of episode):

$A \leftarrow$ action given by π for S

 Take action A , observe R, S'

$V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$

$S \leftarrow S'$

 until S is terminal

TD vs. MC

- MC doit voir un épisode complet avant la mise à jour de la fonction de valeur.
- Converge à la solution optimale
- TD met à jour la fonction de valeur ($V(s)$, $Q(sua)$) après chaque transition. En ligne.
- Converge à la solution optimale (conditions sur le taux d'apprentissage α)
- Empiriquement les méthodes TD convergent plus rapidement