Apprentissage par renforcement
TD

26 janvier 2022

Objet du TD : Apprentissage par renforcement, processus de Markov, Q-learning.

---

1 Modèles d'optimalité (d'après la référence [3])

On considère le problème défini comme suit :

1. États : $S = \{s_0, s_{11}, s_{12}, s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{31}, s_{32}, s_{33}, s_{34}, s_{35}, s_{36}\}$.
2. Actions :
   (a) À partir de s_0 : trois actions possibles conduisant à s_{11}, s_{21} et s_{31}.
   (b) À partir de s_{12}, s_{25} et s_{36} : une seule action possible qui fait boucler sur le même état.
   (c) Pour les autres états : une seule action possible $s_{ij} \rightarrow s_{ij+1}$.
3. Reward :
   (a) Pour $s_{12} \rightarrow s_{12}$, $r = 2$.
   (b) Pour $s_{25} \rightarrow s_{25}$, $r = 10$.
   (c) Pour $s_{36} \rightarrow s_{36}$, $r = 11$.
   (d) r=0 pour toutes les autres actions.

1. Faites un schéma récapitulatif de ce problème.
2. Comparer les 3 modèles d'optimalité en prenant les valeurs suivantes pour les paramètres h et γ :
   (a) $h = 5$ et $\gamma = 0.9$.
   (b) $h = 1000$ et $\gamma = 0.2$.

2 Processus de Markov (d'après la référence [1])

On considère un jeu à un seul joueur dans lequel à chaque tour il n'y a que deux possibilités :

— Gagner un point → probabilité = p.
— Perdre un point → probabilité = $1 - p$.

Le jeu s'arrête dès que le nombre de points atteint 0, M.

1. Vérifiez que ce jeux peut être modélisé par une chaîne de Markov et donnez sa matrice de transition.
2. En prenant p=0.25 et M=7 et en supposant que le nombre initial de points x_0 est obtenu en jettant un dé, calculer les probabilités des suites de nombre de points suivantes :
   (a) $1, 2, 3, 4, 5, 6, 7$.
   (b) $6, 7$.
   (c) $6, 5, 4, 3, 2, 1, 0$.

3 Processus de Markov 2 (d'après la référence [2])

On considère un graphe décrivant la soirée de révision d'un étudiant sous la forme d'un ensemble d'activités et de transitions (probabilistes) entre ces dernières.

— Les activités sont les suivante :
  $S = \{Cours1, Cours2, Cours3, Pause, Dormir, Sortir, Facebook\}$.
— Les probabilités des transitions sont les suivantes :
  — À partir de *Cours1* → *Cours2* ou *Facebook* avec une probabilité de 0.5.
  — À partir de *Cours2* → *Cours3* (0.8) ou *Dormir* (0.2).
  — À partir de *Cours3* → *Pause* (0.4) ou *Sortir* (0.6).
  — À partir de *Pause* → *Cours1* (0.2), *Cours2* (0.4) ou *Cours3* (0.4).
  — À partir de *Dormir* → Aucune transition.
  — À partir de *Sortir* → *Dormir* (1.0).
  — À partir de *Facebook* → *Cours1*(0.1) ou *Facebook* (0.9).

1. Donnez la matrice de transisition de la chaîne de Markov associée à ce graphe.
2. Donnez quelques suites d'états possibles en partant de l'état *Cours1* et évaluez leurs probabilités.

3. Complétez le problème en associant une récompense à chaque état et expliquez comment on peut calculer la valeur de chaque état.
4. Écrivez un programme python permettant de calculer ces valeurs pour différentes valeurs du facteur d'actualisation γ.

4 Un premier 'Gridworld'

On considère un agent qui se déplace à gauche, à droite, vers le haut et vers le bas sur le gridworld de la figure 1. Sur cette figure nous avons indiqué les états, les principales actions possibles et les récompenses correspondantes. Précisons pour compléter cette figure que :

— le déplacement diagonal $s_4 \rightarrow s_1$ signifie que quelque soit le mouvement effectué par l'agent en s_4, il se retrouve en s_1.
— L'agent peut 'tenter' de sortir du gridwold et que dans ce cas il est remis à son état de départ (il se heurte à un mur !) et reçoit une punition de -1.

Il vous est demandé de :

1. Donner le processus de décision markovien décrivant ce problème.
2. Trouver une stratégie non optimale π et calculer les valeurs V^π des états pour cette stratégie.
3. Faites tourner 'à la main' l'algorithme Value Iteration pour quelques étapes.
4. Ecrire un programme Python pour calculer la valeur optimale des états et déduire une stratégie optimale (on essaiera plusieurs valeurs de γ).

5 Q-learning, cas déterministe

On considère un agent qui se déplace sur le gridworld de la figure figure 2. La case rouge est une case interdite et l'atteindre provoque une punition de -1. La case verte est l'objectif et l'atteindre permet à l'agent d'obtenir une récompense de +1. L'agent peut se déplacer dans les 4 directions. S'il tente de sortir du gridwold il est tout simplement remis dans sa case de départ.

1. Donnez le processus de décision markovien décrivant ce problème.
2. Écrire un programme Python permettant d'appliquer le Q-learning pour trouver une stratégie optimale. On prendra γ=0.9, α=0.2 et on essaiera plusieurs valeurs de ϵ.
3. Quel sont les conséquences des changements de la valeur de ϵ ?

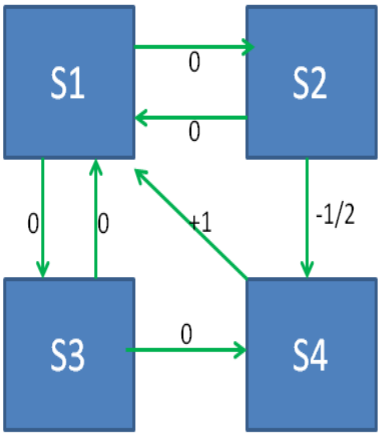

FIGURE 1 –

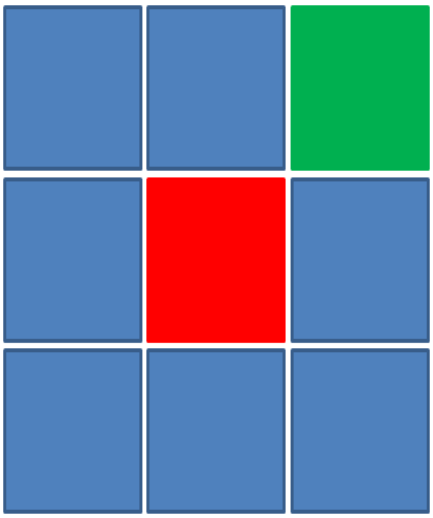

FIGURE 2 –

6 Q-learning, cas non déterministe, (d'après la référence [4])

On considère un agent qui se déplace sur le gridworld de la figure figure 3. La case rouge est une case interdite et l'atteindre provoque une punition de -1. La case verte est l'objectif et l'atteindre permet à l'agent d'obtenir une récompense de +1. La case noire est inaccessible. L'agent peut se déplacer dans les 4 directions. S'il tente de sortir du gridwold ou accéder à la case noire, il est remis dans sa case de départ.

Chaque déplacement coûte -0.04.

Lorsque l'agent prend une direction d, il y réussit avec une probabilité 0.8 et prend la direction située à sa droite ou celle située à gauche avec une probabilité de 0.1.

1. Donnez le processus de décision markovien décrivant ce problème.
2. Écrire un programme Python permettant d'appliquer le Q-learning pour trouver une stratégie optimale. On prendra =0.9, α=0.2 et on essaiera plusieurs valeurs de ϵ et de γ.
3. Étudiez les conséquences des changements de la valeur de ϵ et de γ.

Références

[1] S.M. Ross. Initiation aux probabilités.Presses polytechniques et universitaires romandes, 1996.

[2] D. Silver. UCL Course on RL. Lecture 2. `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html`. 2015.

[3] L. P. Kaelbling,M. L. Littman and A. W. Moore.Reinforcement Learning : a Survey. J. Artif. Intell. Res. Volume 4. pages 237–285. 1996

[4] Jacob Schrum. Reinforcement learning. `https://www.youtube.com/watch?v=XrxgdpduWOU`.

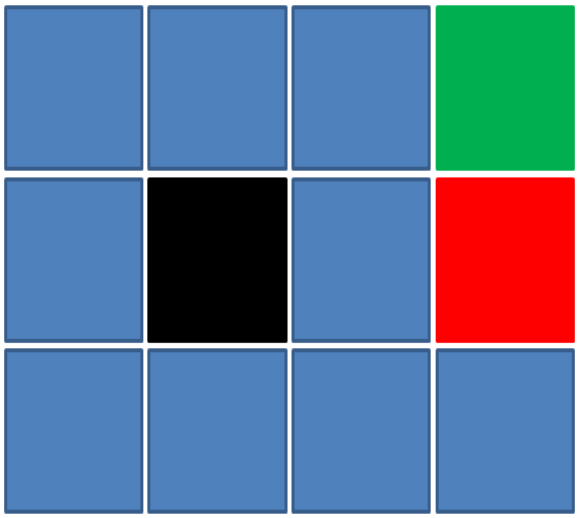

FIGURE 3 –