SMA

Arthur VIGNE – Aurélien TARDY

Le code de ce projet est accessible sur Github : https://github.com/aurelien-tardy/TPSMA

TP1. Planification stochastique & MDP

Introduction

Dans ce TP nous allons étudier plusieurs méthodes afin qu'un agent puisse développer une stratégie dans un environnement spécifique. L'agent va donc changer d'état et ainsi changer ses actions en fonctions de ces dernières. Il existe également des récompenses lorsque l'agent atteint certains états, c'est ce que l'agent va essayer d'atteindre. La première étape de ce TP est la planification stochastique pour un agent. Ce dernier doit ainsi étudier le plateau afin de connaitre tous les états et transitions possibles.

5 Influence des paramètres

Question 1 :

Changez **un seul des deux paramètres**, soit soit le bruit, de sorte à ce que la politique optimale permette à l'agent de traverser le pont (cf. figure 1).

Le bruit doit être baissé du fait que lorsque le bruit est trop élevé, l'agent préfère rester sur place plutôt que de tenter un déplacement au risque de prendre une récompense de -100. Ceci est dû au fait que les récompenses négatives sont 10 fois plus impactantes que les récompenses positives

Question 2 :

En partant de valeurs initiales $= 0.9$, bruit=0.2, $r_{other} = 0.0$, vous devez obtenir une politique optimale qui suit un chemin sûr pour atteindre l'état absorbant de récompense +10 (cf. figure 2, image de droite).

1. qui suit un chemin risqué pour atteindre l'état absorbant de récompense +1

Afin de suivre un chemin risqué et atteindre l'état absorbant de récompense +1, il faut un gamma faible (0,1) afin que le seul chemin qui rapporte soit celui qui prend le chemin risqué.

2. qui suit un chemin risqué pour atteindre l'état absorbant de récompense +10

Afin de suivre un chemin risqué et atteindre l'état absorbant de récompense +10, il faut un bruit nul pour que le chemin risqué soit emprunté et que le chemin vers le +10 soit assez important pour que l'agent ne s'arrête pas au +1.

3. qui suit un chemin sûr pour atteindre l'état absorbant de récompense +1

Afin de suivre un chemin sûr et atteindre l'état absorbant de récompense +1, il faut mettre un gamma plus ou moins égale à 0,3 qui n'est ni trop élevé pour que l'agent prenne la direction du +10 ni trop faible pour qu'il prenne le chemin risqué.

4. qui évite les états absorbants

Afin de suivre un chemin sûr et atteindre l'état absorbant de récompense +10, il ne faut rien modifier afin qu'il prenne le chemin vers la plus grande récompense de manière sécurisée.

TP2. Apprentissage par renforcement

Introduction

Dans ce TP nous allons étudier l'apprentissage par renforcement, c'est à apprendre les actions que l'agent doit faire à partir de son expérience précédente. L'agent prend donc des décisions en fonction de son état actuel. L'agent cherche, par le biais de ses diverses expériences, un comportement décisionnel optimal afin de maximiser la somme des récompenses. Nous allons commencer par tester l'agent dans l'environnement gridworld du TP précédent puis nous utiliserons un autre environnement avec un agent qui apprend à avancer. Enfin, nous testerons cet apprentissage sur le jeu du Pacman.

Choix des états du Pacman

Nous avons choisi de donner au pacman les positions des fantômes ainsi que les positions des dots à récupérer. Pour réduire le nombre d'états, nous utilisons une distance de vision pour n'avoir les positions des dots et fantômes avec une distance en cases maximale.

On retrouve donc dans le hashcode (qui permet de faire la différence entre deux états) :

- positionsDots : coordonnées X et Y de chaque dot dans un carré de n cases autour du pacman.
- positionsFantomes : coordonnées X et Y de chaque fantôme présent dans un carré de n cases autour du pacman.

Q-Learning et généralisation pour le jeu Pacman

Le nombre d'états possibles dépend de la distance à laquelle le pacman « peut voir ». Cela forme un carré de n*n cases. On retire une case pour la case où se situe le pacman.

Pour chaque case, on peut donc avoir soit un dot, soit un fantôme, soit du vide. Le nombre d'états possibles est donc 3 exposant n-1.

L'utilisation des fonctions caractéristiques s'est bien déroulé et cela nous permet d'avoir de bon scores sur les différents labyrinthes.