

Projet apprentissage par renforcement - Pacman

1ere partie

Olivier Goudet

February 4, 2022

Tout d'abord, télécharger le dossier *PacmanQlearning* sur Moodle et créer un projet contenant les fichiers java contenus dans ce dossier. Gérer le *build path* du projet en ajoutant la librairie *commons-lang3-3.12.0.jar* qui est contenue dans le dossier en tant qu'*external jar* pour votre projet.

Implémentation de premières stratégies d'apprentissage pour le Pacman

Pour chaque méthode d'apprentissage implémentée, il est conseillé de la tester tout d'abord en pas à pas avec le lancement du programme de debug via l'interface graphique (*main_debugMode*), puis de faire les évaluations avec le programme *main_standardMode*.

1. Compléter la classe *TabuLarQLearning* qui correspond à l'algorithme de TabularQlearning présenté dans le cours 2.
2. Tester votre méthode pour les très petits niveaux *very_very_smallMaze.lay* et *very_smallMaze.lay*. Quels résultats obtenez vous ? Comment évolue l'apprentissage au fil du temps ?
3. Peut-on appliquer cet algorithme pour le niveau *originalClassic.lay* ?
4. Implémenter l'algorithme *ApproximateQlearning* vu dans le cours 3.
5. Quels features proposez vous d'implémenter ?
6. Tester votre méthode pour les très petits niveaux. Quels résultats obtenez vous ? Comment évolue l'apprentissage au fil du temps ?
7. Tester pour un plus grand niveau comme *originalClassic.lay* en mode *normal* et *nightmare*.
8. Quels résultats obtenez vous ?
9. Quels sont les limites de cet algorithme ?

Fonctions du jeu pacman utiles pour la réalisation des algorithmes

- Dans la classe **PacmanGame** vous avez accès à l'agent pacman qui est caractérisé par une position de type **PositionAgent** avec ses coordonnées en X et en Y.
- Dans la classe **PacmanGame** vous avez accès à la liste des agents fantômes avec leurs positions.
- Dans la classe **PacmanGame** vous avez accès à l'attribut labyrinthe de type **Maze** qui dispose des méthodes *isCapsule()* et *isFood()* qui permettent de savoir si une capsule ou bien une pacgomme se trouve en position (x,y).

- Dans la classe **PacmanGame** vous avez accès à une méthode *isLegalMove* qui permet de savoir si un agent peut réaliser une action donnée. Pour améliorer la performance de l'algorithme, il est en effet conseillé de considérer uniquement des actions légales pour le pacman dans vos algorithmes.
- Dans vos stratégies d'apprentissage, il est préférable d'adapter les choix des actions au mode *train* ou *test*. Notamment, lorsqu'on est en mode *test*, on réglera la valeur d'épsilon à 0 (sans choix aléatoire). Ceci permettra de mieux évaluer et comparer vos différentes stratégies.