3I005 : Statistique et Informatique

Projet :

Exploration/Exploitation

Par :
Angelo ORTIZ
Celina HANOUTI

Licence d'Informatique
Année 2018/2019

Table des matières

Table des figures

Liste des tableaux

Introduction

En intelligence artificielle, plus précisément en Machine Learning, on est souvent amené, pour un ensemble de choix possibles, à trouver un bon compromis entre l'exploration et l'exploitation. L'exploration consiste a obtenir d'avantage d'informations dans le but de prendre de meilleures decisions dans le future. L'exploitation, quant à elle, consiste à agir de manière optimale en fonction de ce qui est déjà connu. Ce dilemme constitue un problème largement étudié dans le domaine de l'apprentissage par renforcement. En effet, Des algorithmes ont été proposés pour résoudre un certain nombre de problèmes qui illustrent des applications de ce dilemme.

Dans ce projet, nous allons étudier et analyser un certain nombre d'algorithmes basés sur le dilemme de *l'exploration vs exploitation* à travers des IAs de jeu. Dans un premier temps, nous allons effectuer une evaluation experimentale d'algorithmes classiques dans un cadre simplifié, la deuxèime partie est consacrée à l'implementation d'un algorithme de Monte-Carlo et enfin, dans les deux dernieres partie, nous allons étudier des algorithmes plus avancés ainsi qu'un jeu de stratégie combinatoire.

Première partie

Bandits-manchots

1 Description du problème

On condière le problème d'apprentissage suivant : un agent est confronté à plusieurs reprises à un choix parmi N différentes actions, chacune des N actions procure une récompense moyenne μ^i inconnue par l'agent et nous désignons l'action sélectionnée sur le pas de temps t par a_t, la récompense correspondante par r_t et $N_T(a)$ le nombre de fois ou l'action a a été choisi jusqu'au temps T. L'objectif est de maximiser la somme des récompense obtenue au bout des T premieres parties, pour celà, l'agent doit identifier l'action au rendement le plus elevé i^*= $\text{argmax}_{i\in 1,\dots,N}\,\mu^i$. Les valeurs μ^i etant inconnues, il est donc necessaire de faire des estimations qui doivent être le plus représentatives possible des valeurs μ^i, notons $\hat{\mu}^i$ ces estimations. L'estimation associée a une action est la récompense moyenne lorsque cette action est sélectionnée.

$$\hat{\mu}^i = \frac{1}{N_T(a)} \sum_{t=1}^{T} r_t \mathbb{1}_{a_t=a}$$

2 Algorithmes

Si de nombreux algorithmes ont été proposés pour résoudre ce dilemme, on se contentera ici d'en citer que quelques uns :

2.1 Algorithme Aléatoire

l'algorithme aléatoire se contente de choisir l'action uniformément au hasard. On peut donc déjà présager que cet algorithme sera forcément le moins optimal. **En effet, il effectue purement de l'exploitation et est ainsi utilisé comme baseline pour les tests présentés par la suite**

2.2 Algorithme Greedy

L'algorithme Greedy est basé sur une politique de selection très simple qui consiste à sélectionner l'une des actions ayant la valeur estimée la plus élevée : a_t = $\text{argmax}_{i\in 1,\dots,N}\,\hat{\mu}_t^i$. Autrement dit, L'algorithme Greedy ne fait qu'éxploiter les connaissances dont on dispose afin de maximiser la recompense à un instant t.

2.3 Algorithme ε-Greedy

Une approche courante pour trouver un compromis exploitation/exploration est l'algorithme ε-Greedy qui consiste a choisir a l'instant t, avec une probabilité 1 - ε, l'action qui maximise le rendement moyen sur les estimations faites jusque la et avec une probabilite ε, une action uniformement au hasard. L'avantage de cet algorithme repose sur le fait que dans la mesure où le nombre d'étape d'apprentissage augmente, chaque action sera choisie un nombre infini de fois assurant ainsi que tous les $\hat{\mu}^i$ convergeront vers les μ^i. On peut conjécturer la puissance de cet algorithme et sa capacité a explorer toutes les actions possibles et de prendre des décisions quasi proches de l'optimum. **Je sais pas si c'est vraiment un nombre infini, mais c'est sûr que les estimateurs se rapprocheront davantage aux gains moyens réels. On pourra aussi dire que la valeur de ε a un fort impact sur l'apprentissage, c'est pourquoi on en a testé différentes valeurs (constantes) et même une forme en fonction (décroissante) du temps ? ? ?**

2.4 Algorithme UCB

L'algorithme UCB est une autre stratégie qui permet de trouver une balance entre l'exploration et l'exploitation. L'idée de cet algorithme consiste à se fier à une borne supérieure de confiance, en effet, une incertitude existera toujours sur l'optimalité des estimations faites jusque là. Les deux algorithmes définis précedemment choisissent toujours l'action qui semble meileure à un instant t ou se contentent de choisir des actions aléatoirement sans distinction. L'algorithme UCB prend en compte l'optimalité des estimations mais egalement l'incertitude sur celles-ci. Il selectionne l'action : $a_t = \text{argmax}_{i \in 1,\dots,N}(\hat{\mu}_t^i + \sqrt{\frac{2\log(t)}{N_T(i)}})$. **Ici, cette incertitude est mesurée par le deuxième terme : plus grand est le déséquilibre entre le(s) (nombre d') actions effectuées, plus on se méfie au résultats. On priviliégiera ainsi les actions le plus défavorisées**