Rapport – Projet MN-IA

KIRLOS Youssef et DIRANI SAFAWI Hani

1. Introduction

Dans ce projet, nous avons implémenté et appliqué plusieurs algorithmes d'apprentissage par renforcement dans deux environnements distincts : **Simple** et **TicTacToe**. L'objectif principal était d'implémenter ces algorithmes pour optimiser la prise de décision de l'agent dans ces environnements.

Les algorithmes que nous avons implémentés incluent :

- Pour l'environnement **Simple** : **Epsilon-Greedy**, **Upper Confidence Bound (UCB)** et **Bandit Gradient**.
- Pour l'environnement **TicTacToe** : **Value Iteration**, **Policy Iteration**, et **Q-Learning**.

2. Environnement Simple

Problématique

L'environnement **Simple** représente un problème de **bandit manchot** où l'agent doit choisir parmi plusieurs actions, apprendre quelle action maximise la récompense et explorer les différentes options de manière optimale.

Algorithmes Utilisés

Nous avons implémenté trois algorithmes de bandit manchot dans cet environnement :

1. **Epsilon-Greedy** : Cet algorithme choisit une action de manière aléatoire avec une probabilité **epsilon**, et choisit l'action avec la meilleure récompense moyenne avec une probabilité de **1 - epsilon**. Ce mécanisme permet à l'agent d'explorer tout en exploitant les meilleures actions connues.
2. **Upper Confidence Bound (UCB)** : Cet algorithme choisit l'action en maximisant non seulement la récompense moyenne, mais aussi l'incertitude des estimations de récompenses, encourageant ainsi l'agent à explorer les actions moins explorées tout en exploitant les actions les plus prometteuses.
3. **Bandit Gradient** : Cet algorithme utilise la stratégie **Softmax** pour choisir des actions en fonction des préférences apprises au fil du temps. La probabilité de choisir une action est calculée en fonction des valeurs de préférence des actions.

Fonction de Récompense

La fonction de récompense que nous avons définie pour l'environnement **Simple** attribue une **récompense de +1 (pour les fims Aet B)** c'est les actions qui sont considérées comme des succès) et **0 (pour film C)**. Cette fonction permet à l'agent de différencier les bonnes actions des mauvaises, en fonction de l'environnement de bandit manchot.

3. Environnement TicTacToe

Problématique

L'environnement **TicTacToe** est un **Processus de Décision Markovien (MDP)** dans lequel l'agent doit apprendre à jouer au jeu du **Morpion**. L'objectif est de maximiser la récompense en jouant de manière optimale contre un adversaire.

Algorithmes Utilisés

Nous avons utilisé trois algorithmes classiques de MDP pour résoudre le problème **TicTacToe** :

1. **Value Iteration** : Ce modèle calcule les valeurs des états dans le jeu, et dérive la politique optimale en maximisant les récompenses futures.
2. **Policy Iteration** : Ce modèle alterne entre l'évaluation de la politique actuelle (calcul des valeurs des états) et l'amélioration de la politique (mise à jour des actions optimales pour chaque état).
3. **Q-Learning** : Cet algorithme apprend à partir de l'expérience en mettant à jour les **Q-valeurs** (fonction d'action-valeur) pour chaque état-action, afin de maximiser la récompense cumulative.

Fonction de Récompense

La fonction de récompense que nous avons utilisée pour **TicTacToe** est définie comme suit :

- **+1** si le joueur 0 gagne. (incite l'agent à atteindre un état gagnant)
- **-1** si le joueur 1 gagne.
- **0** en cas de match nul. (permet à l'agent de comprendre qu'il doit éviter de provoquer une égalité)
- **-0.25** pour encourager l'agent à éviter de se retrouver dans des positions défavorables.