Cours MDP : Q-Learning et Renforcement

rhoumahaythem

Révision de la première partie - Apprentissage par renforcement,
09 Avril 2025

1. Introduction au Processus de Décision de Markov (MDP)

Un Processus de Décision de Markov (MDP) est un cadre mathématique utilisé pour modéliser la prise de décision dans des environnements stochastiques. Il est largement utilisé dans les domaines de l'intelligence artificielle, du contrôle automatique et de la recherche opérationnelle.

Les MDP permettent de modéliser un agent qui interagit avec un environnement dans lequel les résultats ne sont pas totalement déterministes. À chaque étape, l'agent perçoit un état, choisit une action, et reçoit une récompense en conséquence, tout en faisant une transition vers un nouvel état.

Les décisions sont prises de manière séquentielle, en tenant compte non seulement de la récompense immédiate, mais aussi des récompenses futures attendues. Ce concept est au cœur de l'apprentissage par renforcement (Reinforcement Learning).

Dans ce document, nous allons explorer les fondations des MDP, en posant les bases nécessaires pour comprendre les algorithmes d'apprentissage par renforcement les plus modernes, comme Q-learning ou Deep Q-Learning.

2. Composantes des MDP : États, Actions, Transitions, Récompenses et Politiques

Un MDP est défini formellement par un ensemble de composantes fondamentales qui permettent de modéliser les interactions entre un agent et son environnement. Ces composantes sont les suivantes :

- **États (S)** : Ensemble des situations possibles dans lesquelles peut se trouver l'agent. Chaque état représente une configuration unique de l'environnement.
- **Actions (A)** : Ensemble des choix possibles pour l'agent lorsqu'il se trouve dans un état donné. La décision d'action influence la transition vers un nouvel état.

- **Fonction de transition (T)** : Modélise la dynamique de l'environnement. Elle définit la probabilité $P(s'|s,a)$ de passer d'un état s à un état s' en prenant l'action a.
- **Récompense (R)** : Fonction qui attribue un score (positif ou négatif) à chaque transition (s,a,s'). Elle guide l'agent à privilégier certaines trajectoires.
- **Politique (π)** : Stratégie suivie par l'agent pour choisir une action dans chaque état. Elle peut être déterministe ($\pi(s) = a$) ou stochastique ($\pi(a|s)$).

La compréhension de ces composantes est essentielle pour pouvoir manipuler un MDP et concevoir des algorithmes d'apprentissage performants.

3. Équations de Bellman : Définition et Applications

Les équations de Bellman sont au cœur des Processus de Décision de Markov (MDP) et de l'apprentissage par renforcement. Elles formalisent la relation récursive entre la valeur d'un état (ou d'un état-action) et les valeurs des états suivants. Elles permettent d'évaluer la qualité d'une politique.

Valeur d'un état (fonction $V^\pi(s)$)

$$V^\pi(s) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right]$$

Équation de Bellman pour $V^\pi(s)$

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} P(s'|s,a) \left[R(s,a,s') + \gamma V^\pi(s') \right]$$

Valeur d'un couple état-action (fonction $Q^\pi(s,a)$)

$$Q^\pi(s,a) = \sum_{s'} P(s'|s,a) \left[R(s,a,s') + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s',a') \right]$$

Application

Les équations de Bellman sont utilisées pour :

- Évaluer une politique (policy evaluation)

- Améliorer une politique (policy improvement)
- Trouver une politique optimale π^*

4. Différence entre Apprentissage en Ligne (Online) et Hors-Ligne (Offline)

Dans le cadre des MDP et de l'apprentissage par renforcement, il est fondamental de distinguer deux approches d'apprentissage : en ligne (online) et hors-ligne (offline). Ces deux paradigmes se différencient par la manière dont les données sont collectées et utilisées.

...

Il est essentiel de comprendre cette distinction pour concevoir des systèmes efficaces selon les contraintes du problème ciblé.

5. Q-Learning : Principe, Formule et Importance du Taux d'Apprentissage (α)

Le Q-Learning est un algorithme d'apprentissage par renforcement sans modèle. Il apprend la fonction $Q(s, a)$, et la met à jour à chaque interaction :

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s,a) \right]$$

Le taux α contrôle la vitesse d'apprentissage. Un α trop grand rend l'agent instable, trop petit ralentit l'apprentissage.

Le Q-Learning est à la base de nombreuses extensions modernes telles que le DQN.

6. Exploration vs Exploitation : Trouver le Bon Équilibre

Trouver le juste milieu entre exploration (tester de nouvelles actions) et exploitation (utiliser les connaissances acquises) est fondamental.

Stratégie ε-greedy :

- $1 - \varepsilon$: exploitation.
- ε : exploration aléatoire.

Il est courant de faire décroître ε avec le temps pour améliorer la convergence.

7. Approches Value-Based vs Policy-Based : Comparaison et Applications

Value-Based : Apprennent une fonction Q ou V (ex: Q-learning). Adaptés aux espaces d'actions discrets.

Policy-Based : Apprennent directement $\pi(a|s)$ (ex: PPO). Mieux pour les actions continues.

Actor-Critic : Combine les deux approches pour bénéficier des avantages de chacune.

8. Démonstrations Pratiques : Application du Q-Learning en Environnement Simulé

Mise en œuvre du Q-Learning dans une grille 5×5 :

- Récompense : -1 par déplacement, $+10$ à l'arrivée, -10 en cas de collision.
- Mise à jour de la table $Q(s, a)$ via la formule classique.
- Visualisation de la politique apprise (flèches, score moyen, convergence).

9. Exercice Guidé : Résolution de Problèmes avec Q-Learning

Créer un environnement simple (grille), initialiser Q, fixer les hyperparamètres (α, γ, ε) et entraîner sur 1000 épisodes.

Comparer les résultats selon les variations de ε et analyser la stabilité de l'apprentissage.