

Comparaison de solveurs pour les Processus de Decisions Markovien

Calic Petar, Muyang Shi
Sorbonne Universite, Paris
Encadré par Emmanuel Hyon

petarcalic99@gmail.com
muyangforwork@gmail.com

04/04/21

Abstract

Nous avons pour objectif dans ce projet, de faire une comparaison numérique de solveurs pour MDP. Afin de faire cela, nous allons d'abord faire des recherches sur les fondements des MDP pour les maîtriser. Ceci est essentiel puisque nous devons ensuite prendre en main chacun des 3 solveurs que nous allons étudier. Nous allons ensuite préparer un benchmark qui doit être appliqué à tous les solveurs. La dernière phase est celle où on doit faire tourner les solveurs sur nos exemples et analyser les résultats obtenus.

Keywords: MDP; solveurs; benchmark

1 Introduction

En théorie des probabilités, un processus de décision markovien (MDP) est un modèle stochastique (de fonction aléatoire) où un agent prend des décisions et où les résultats de ses actions sont aléatoires. Les MDPs sont une extension des chaînes de Markov avec plusieurs actions à choisir par état et où des récompenses sont gagnées par l'agent. Les MDPs sont utilisés pour étudier des problèmes d'optimisation à l'aide d'algorithmes de programmation dynamique ou d'apprentissage par renforcement, que nous allons étudier et comparer.

1.1 Objectifs du Projet:

1.1.1 Maîtriser les MDP

Avant de pouvoir manipuler les solveurs, il est crucial de comprendre les fondements théoriques des MDP. Tout d'abord il faut savoir reconnaître les situations dans lesquelles un problème peut être modélisé par un MDP. Des hypothèses telles que l'hypothèse de Markov doivent être vérifiées. Ensuite il va falloir savoir modéliser un problème sous forme de MDP, puis adapter le modèle au solveur qu'on souhaite utiliser.

1.1.2 Prendre en main les Solveurs

L'objectif principal de ce projet est de comparer 3 solveurs de MDP. Nous allons observer les performances de Gurobi(programme mathématique) MarmotMDP et ToolboxMDP. Avant de faire cela il va falloir bien s'appropriier les 3 solveurs et savoir les manipuler. Trois grandes familles de méthodes existent pour résoudre de tels MDP en cherchant la politique optimale:

- Value Iteration: L'approche la plus classique qui se base aussi sur la résolution directe de l'équation d'optimalité de Bellman V. On fait des Itérations sur les états jusqu'à convergence.
- Policy Iteration: Iteration sur la politique; On propose une politique de départ et chaque itération fait une évaluation puis amélioration de la politique jusqu'à convergence.
- Programmation linéaire: Transformation de l'MDP en un problème linéaire et son Dual pour le résoudre.

Selon les critères, les algos ne sont pas les mêmes à utiliser. Les types de résultats(politiques) ne sont pas les mêmes aussi.

1.1.3 Modéliser un problème MDP pour solveur

Pour pouvoir comparer les solveurs, nous allons prendre un exemple de problème modélisable comme un MDP et le résoudre. Il y a une étape intermédiaire importante et c'est la modélisation du problème et donc son adaptation ou traduction pour un solveur spécifique. En effet, chaque solveur requiert une modélisation particulière d'un problème pour pouvoir le résoudre.

1.1.4 Analyse des solveurs

C'est l'étape finale et clé du projet. Nous allons faire tourner les 3 solveurs sur un même problème et observer les résultats. Nous allons construire un benchmark (Un ensemble d'exemples pour tester les caractéristiques) afin de comparer leurs caractéristiques comme la vitesse d'exécution, l'optimalité de la solution, la précision, la complexité d'utilisation.

Nous allons ensuite faire une synthèse des caractéristiques de chaque solveur et noter leurs avantages et inconvénients. Nous allons déterminer aussi pour chaque type de problème, quel solveur serait le plus adapté et performant.

1.1.5 Planning estimé:

- Recherche et maîtrise des MDP: 01 février - 30 février
- Prise en main des solveurs: 01 mars - 21 mars
- Préparation du benchmark: 21 mars - 14 avril
- Analyse des résultats: 14 avril - 25 avril
- Rédaction du rapport : 25 avril - 10 mai

Cela correspond à 3 mois et demi de travail. Ceci est notre estimation de temps nécessaire pour faire un travail de qualité sans grande pression du temps.

Sous demande du professeur encadrant, l'ensemble du projet sera codé sous Python à l'aide d'une machine virtuelle sous Linux. Le rapport aussi devra être écrit à l'aide de Overleaf en Latex.

2 Présentation des MDP

2.1 Définition formelle

Un MDP est un quadruplet S, A, T, R définissant :

- un ensemble d'états S , qui peut être fini, dénombrable ou continu; cet ensemble définit l'environnement tel que perçu par l'agent (dans le cas d'un robot, on peut voir cela comme l'ensemble produit des valeurs de ses différents capteurs);
- un ensemble d'actions A , qui peut être fini, dénombrable ou continu et dans lequel l'agent choisit les interactions qu'il effectue avec l'environnement (dans le cas d'un robot on peut voir cela comme l'ensemble produit des paramètres de ses différentes commandes);
- une fonction de transition $T : S \times A \times S \rightarrow [0;1]$; cette fonction définit l'effet des actions de l'agent sur l'environnement: $T(s, a, s')$ représente la probabilité de se retrouver dans l'état s' en effectuant l'action a , sachant que l'on était à l'instant d'avant dans l'état s ;
- une fonction de récompense $R : S \times A \times S \times \mathcal{R} \rightarrow [0;1]$; elle définit la récompense (positive ou négative) reçue par l'agent: $R(s, a, s', v)$ est la probabilité d'obtenir une récompense v pour être passé de l'état s à s' en ayant effectué l'action a . Ici encore cette définition est très générale, bien souvent on se contentera par exemple des cas particuliers suivants :
 - $R : S \times A \times S \rightarrow \mathcal{R}$ (récompense déterministe, c'est le choix que nous adopterons dans la suite) ;
 - $R : S \times A \rightarrow \mathcal{R}$ (récompense déterministe rattachée à l'action en ignorant son résultat) ;
 - $R : S \rightarrow \mathcal{R}$ (récompense déterministe rattachée à un état donné).

2.2 Les politiques d'actions

Soit l'état initial d'un agent s_0 , on choisit une action a_0 dans A à exécuter. Après la exécution, l'agent passe aléatoirement à l'état prochain s_1 en fonction de la proba de T , $s_1 \subseteq T0a_0$. Et puis l'action a_1 permet de passer à $s_2...$ On peut représenter cette processus comme la figure ci-dessous :

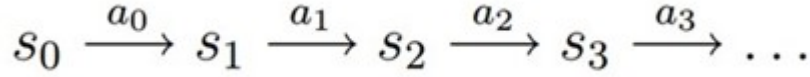


Figure 1: Exemple d'un MDP simple.

Si la récompense a obtenu par rapport aux états s et aux actions a effectué, le MDP peut être représenté ainsi :

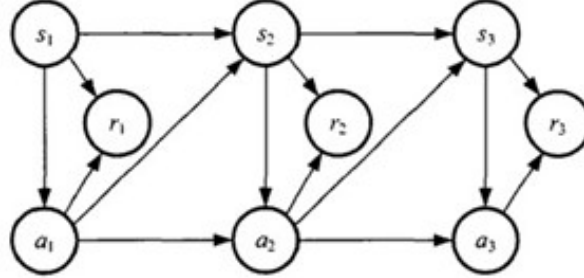


Figure 2: MDP et récompenses.

Le domaine T des étapes de décision est un ensemble discret, qui peut être fini ou infini (on parle d'horizon fini ou d'horizon infini). Par ailleurs, comme pour S et A , les fonctions de transition et de récompense peuvent elles-mêmes varier au cours du temps, auquel cas on devrait prendre en compte une nouvelle variable de temps. Lorsque ces fonctions de varient pas, on parle de processus stationnaires.

Les processus décisionnels de Markov permettent de modéliser la dynamique de l'état d'un système soumis au contrôle d'un agent, au sein d'un environnement stochastique. On peut ainsi définir la notion de politique (noté π). Appelé aussi stratégie, c'est la procédure suivie par l'agent pour choisir à chaque instant l'action à exécuter. On peut faire deux distinctions ici. Une politique peut déterminer précisément l'action à effectuer selon l'état où l'on se trouve, ou simplement définir une distribution de probabilité selon laquelle cette action doit être sélectionnée. Dans notre projet on se limitera au premier cas. Il s'agit de MDP à politique déterministe. C'est le modèle le plus simple et essentiel de stratégie décisionnelle.

2.3 Critères de performance

On doit en effet choisir un critère afin de pouvoir juger la performance de notre politique. Cette étape est importante puisque il faut savoir choisir un critère adapté au modèle notre objectif étant de rechercher la politique qui l'optimise. En termes mathématiques, cela revient à évaluer une politique sur la base d'une mesure du cumul espéré des récompenses instantanées le long de l'exécution. Voici les critères les plus étudiés au sein de la théorie des MDP:

- Le critère fini : $E[r_0 + r_1 + r_2 + \dots + r_{N-1} | s_0]$
- le critère -pondéré : $E[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots + \gamma^t r_t + \dots | s_0]$
- le critère total : $E[r_0 + r_1 + r_2 + \dots + r_t + \dots | s_0]$
- le critère moyen : $\lim_{n \rightarrow \infty} \frac{1}{n} E[r_0 + r_1 + r_2 + \dots + r_{n-1} | s_0]$

Avec E l'espérance et r les récompenses instantanées.

Ces formules sont à la base du principe d'optimalité de Bellman [BEL 57] (« les sous-politiques de la politique optimale sont des sous-politiques optimales »). Il est à l'origine d'algorithmes de programmation dynamique qu'on verra permettant de résoudre efficacement les MDP.

2.4 Fonctions de valuation

Lorsque nous avons déterminé un critère et une politique, nous pouvons déterminer une fonction de valeur. Cette fonction associe à tout état initial $s \in S$ la valeur du critère considéré en suivant π à partir de s :

$$\forall \pi \ V^\pi : S \mapsto \mathbb{R}$$

$V^\pi(s)$ représente le gain reçu par l'agent s'il démarre à l'état s et applique ensuite la politique π à l'infini. Les problèmes décisionnels de Markov peuvent alors être traduits en terme d'équations d'optimalité portant sur les

fonctions de valeur, dont la résolution est de complexité beaucoup moins importante que le parcours exhaustif de l'espace global des politiques. Nous pouvons aussi définir une autre fonction de valuation dite états-actions : $Q^\pi(s, a)$:

$$\forall \pi \ Q^\pi : S \times A \mapsto \mathbb{R}$$

elle représente le gain de l'agent, s'il démarre à l'état s et commence par effectuer l'action a , avant d'appliquer ensuite la politique π à l'infini. Ceci correspondrais en pratique à une matrice qui, pour chaque état possible, associe les actions possible a effectuer et leur gain.

On note que les deux fonctions sont très liées. On a en effet $V^\pi(s) = Q^\pi(s, \pi(s))$.

2.5 Équation de Belmann

Sans doute le coeur du fonctionnement des algorithmes de résolutions de MDP. La fonction de valuation vérifie une relation de récurrence:

$$V^\pi(s) = \sum_{s' \in S} [R(s, \pi(s), s') + \gamma V^\pi(s')] T(s, \pi(s), s')$$

Avec R la matrice de récompenses et T la matrice de probabilités de transitions.

Le but de l'agent est de trouver la politique optimale π^* qui lui permet de maximiser le gain selon le critère qu'on a choisit, c'est-à-dire la politique qui vérifie, pour tout état $s \in S$, $V^{\pi^*}(s) \geq V^\pi(s)$ pour toute autre politique π . On peut montrer que la fonction de valeurs optimale V^* vérifie l'équation d'optimalité de Bellman:

$$V^* = \max_{a \in A} \sum_{s' \in S} [R(s, a, s') + \gamma V^*] T(s, a, s')$$

La fonction de valeurs optimale peut donc être approché par un algorithme itératif.

2.6 Algorithmes

2.6.1 Présentation

2.6.2 Critères et méthodes de résolution possible

3 Solveurs