ENSIAS

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

Markov Decision Process

Realized by :
Arbib Nada
Dahmani Zineb

Academic year 2021/2022

Abstract

The present report is a synthesis of the work carried out within the framework of the project of probabilistic learning , carried out within the ENSIAS, a big school of engineers specialized in Technologies of the Information and the Communication. Its missions are the training of state engineers and research for the technological and economic development of Morocco.

The mission we have been given is part of studying an example of markov decision processus and implementing its algorithms.

Contents

Introduction

Reinforcement Learning was originally developed for Markov Decision Processes (MDPs). It allows a single agent to learn a policy that maximizes a possibly delayed reward signal in a stochastic stationary environment. It guarantees convergence to the optimal policy, provided that the agent can sufficiently experiment and the environment in which it is operating is Markovian. However, when multiple agents apply reinforcement learning in a shared environment, this might be beyond the MDP model. In such systems, the optimal policy of an agent depends not only on the environment, but on the policies of the other agents as well. These situations arise naturally in a variety of domains, such as: robotics, telecommunications,economics, distributed control, auctions, traffic light control, etc. In these domains multi-agent learning is used, either because of the complexity of the domain or because control is inherently decentralized. In such systems it is important that agents are capable of discovering good solutions to the problem at hand either by coordinating with other learners or by competing with them.

1 What is MDP ?

A Markov decision process (MDP), by definition, is a sequential decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards. It consists of a set of states, a set of actions, a transition model, and a reward function. Here's an example :

2 Tic Tac Toe Example :

Let's take a simple example: Tic-Tac-Toe, a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game, with a forced draw assuming best play from both players.

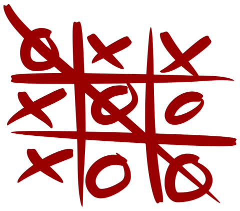


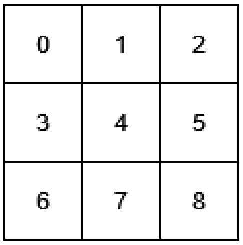
Figure 1 – Tic Tac Toe game

2.1 Modeling

The Markov decision process (MDP) is a mathematical model of sequential decisions and a dynamic optimization method. A MDP consists of the following five elements: {S,A,p,r} where :

1. S is a set of countable nonempty states, which is a set of all possible states of the system.
2. A is a set of all possible decision-making behaviors when the system is in a given state.
3. p indicates the probability of moving to state j when the system is in state i ∈ S and the decision-making behavior a ∈ A(i) is taken.
4. r=r (i, a) is called a reward function, which represents the expected reward obtained when the system is in a state i∈ S at any time and adopts a decision-making behavior a∈ A(i).

In our case ,The state space is a vector of length 9 that indicates the 9 possible position .



0	1	2
3	4	5
6	7	8

Figure 2 – States representation

The actions are on which of the 9 spot you can play (so there is 9 possible actions). Note that as the game evolves, some actions will become unavailable.

The transition function is dictated by your opponent's strategy.

For every move a player performs an action and state of tic tac toe board changes.Following figure shows sequence of actions taken and state of the board:

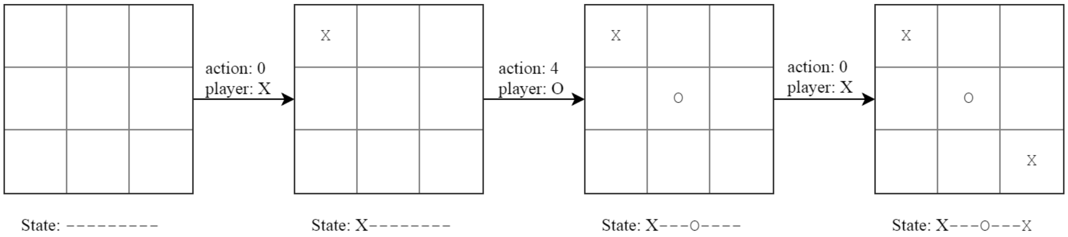


Figure 3 – States representation

Since there are two players we will have to learn separate policies for player 'X' and player 'O'.

Rewards R: Winning player receives a reward of 1 and losing player will receive a penalty of -1 and 0 otherwise this is for a goal reward .And for action penalty if the player wins ,he got 0 as reward,if the other player wins he will get -10 and -1 otherwise. We shall also reward the player who makes a move that prevents the opponent from winning in next step, a small positive reward.

2.2 Results after applying Value iteration algorithm

2.2.1 Random X VS Random O

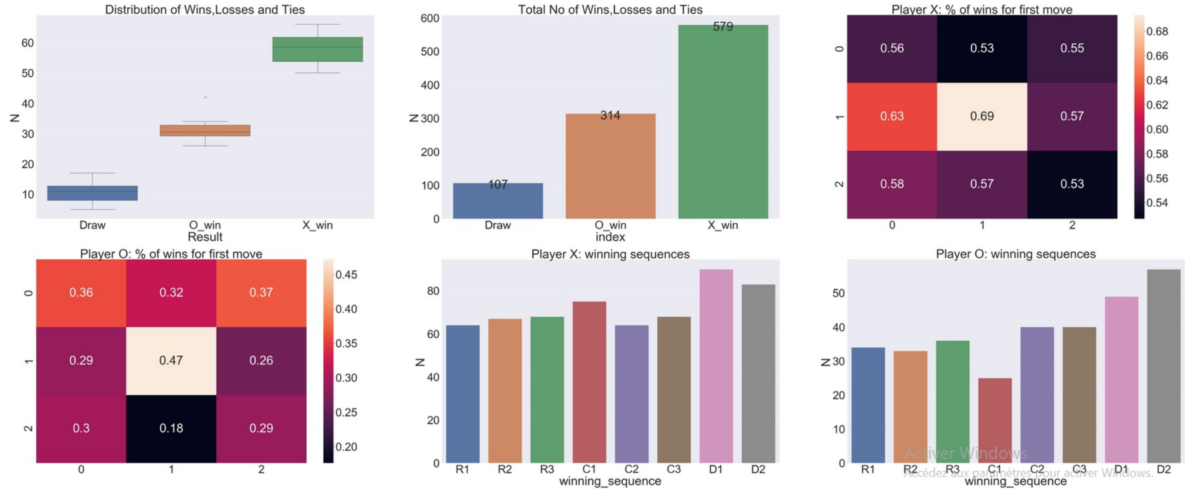


Figure 4 – States representation

These results indicates that when both players are using a random policy, X wins a majority of games due to first mover advantage.

For both players, occupying the central square in the first move maximizes the chances of winning.

Further, for both players, the winning sequence is most likely to be along the diagonal.

2.2.2 Trained X VS Random O

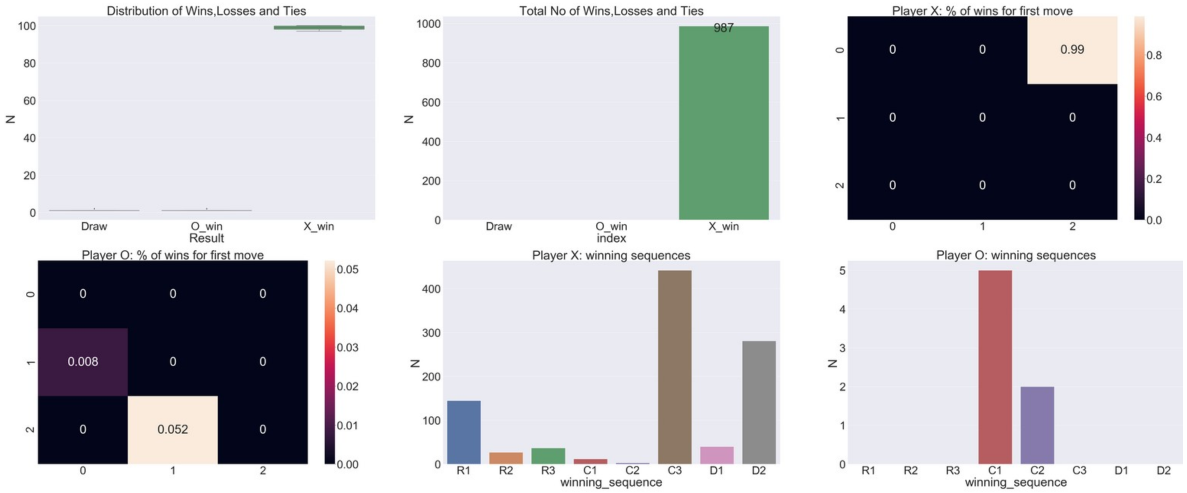


Figure 5 – States representation

These results indicate that player X has learned to easily beat a random player O. Player X consistently chooses the top right hand box and seem to win the majority of games through the right most column (C3) or the off diagonal(D2).

2.2.3 Trained X VS Trained O

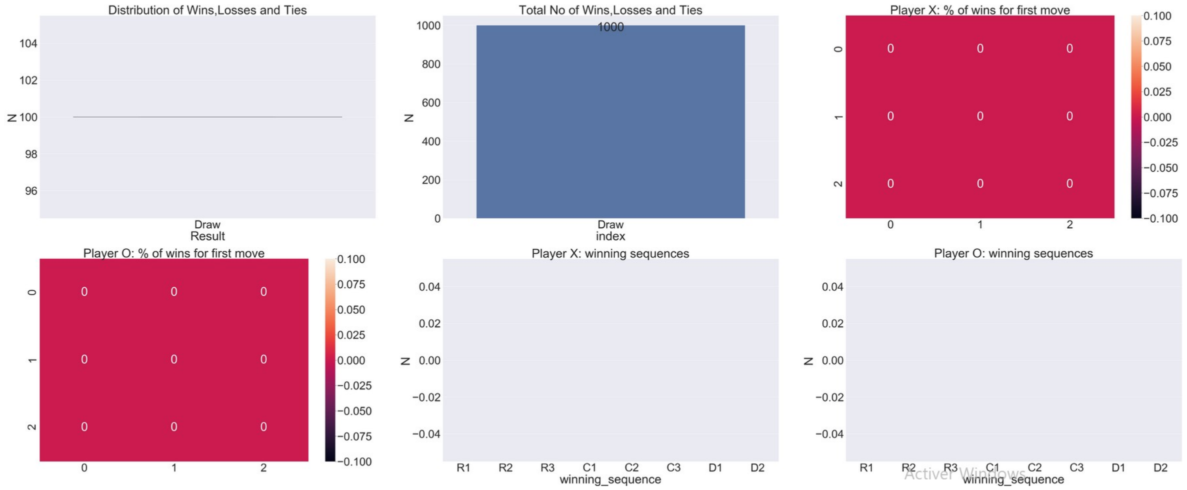


Figure 6 – States representation

When the two trained agents face off, all games end in ties.

Conclusion

Reinforcement learning is a powerful paradigm in AI that can potentially be the key to solving several real world problems. Although the early days of RL has seen an almost exclusively focus on games, there are several practical applications of RL outside of games that industry is working on. Here was an example to prove how it can affects in a real case : Winning strategy in Tic Tac Toe .