RLNotes

Norah Jones

2024-07-11

Table of contents

Preface

Reinforced Learning is learning what to do -how to map situations to actions- so as maximize a numerical reard signal.

We formalize the problem of reinforcement learning using ideas from dynamical systems theory, specially, as the optimal control of incompletely-known Markov decission processes.

One of the challenges that arise in reinforcement learning, and not in other kinds of learning, is the trade-off betweeen exploration and exploitation. The agent has.

1 Elements of Reinforcement Learning.

Beyond the agent and the enviroment, one can identify four main subelements od a reinforcement learning system:

- A Policy
- A Reward Signal
- A Value Function

and optionally, a model of the enviroment.

A *policy* defines the learning agent's way of behaving at a given time. In general, policies may be stochastic, specifying probabilities for each action.

A *reward signal* defines the goal of a reinforcement learning problem. The agent's role objective is to maximize the total reward it receives over the long run.

Whereas the reward signal indicates what is good in immediate sense, a *value function* specifies what is good in the long run.

The fourth and final element of some reinforcement learning systems is a model od th eenviroemnt. This is something that mimics the behavior of the envirometn, or more generally, that allows inferences to be made about how the enviroment will behave.

2 An Extended Example: Tic-Tac-Toe

Consider the familiar child's game of tic-tac-toe. Two players take turns playing on a three-by-three board. One player plays Xs and the other Os until one player wins by placing three marks in a row, horizontally, vertically, or diagonally, as the X player has in the game shown to the right. If the board fills up with neither player getting three in a row, then the game is a draw.

Because a skilled player can play so as never to lose, let us assume that we are playing against an imperfect player, one whose play is sometimes incorrect and allows us to win. For the moment, in fact, let us consider draws and losses to be equally bad for us.

How might we construct a player that will find the imperfections in its opponent's play and learn to maximize its chances of winning?

2.1 Construct Decision Making Model

Here is how the tic-tac-toe problem would be approached with a method making use of a value function. First we would set up a table of numbers, one for each possible state of the game. Each number will be the latest estimate of the probability of our winning from that state. We treat this estimate as the state's value, and the whole table is the learned value function. State A has higher value than state B, or is considered "better" than state B, if the current estimate of the probability of our winning from A is higher than it is from B. Assuming we always play Xs, then for all states with three Xs in a row the probability of winning is 1, because we have already won. Similarly, for all states with three Os in a row, or that are filled up, the correct probability is 0, as we cannot win from them. We set the initial values of all the other states to 0.5, representing a guess that we have a 50% chance of winning.

2.2 Exploratory vs Greedy

We then play many games against the opponent. To select our moves we examine the states that would result from each of our possible moves (one for each blank space on the board) and look up their current values in the table. Most of the time we move greedily, selecting the move that leads to the state with greatest value, that is, with the highest estimated probability of winning. Occasionally, however, we select randomly from among the other moves instead.

These are called exploratory moves because they cause us to experience states that we might otherwise never see.

3 Excersice Code: Tic Tac Toe.

3.1 Objective

In the first excersice code going to make a Tic Tac Toe and check the probabilites of win.

3.2 Construct Tic Tac Toe Model.

The Tic Tac Toe Board is a game over a grid of 3 × 3 squares

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

In each square (a_{ij}) can write a **X** or **O**. A player use the **X** and the other player use the **O**. In the game by turns both players place us simbols $\{\mathbf{X}, \mathbf{O}\}$ in the board. The winner is the first player in obtain a three row, column or principal diagonal.

For example, if the first player is **X** and

$$s : a_{11} = \mathbf{X}, a_{12} = \mathbf{O}, ... \begin{bmatrix} \mathbf{X} & \mathbf{O} & \mathbf{O} \\ \mathbf{X} & \mathbf{X} & \\ \mathbf{X} & & \mathbf{O} \end{bmatrix}.$$

In this case, the $\mathbf{X}'s$ player win. To construct the Decision Model be consider the States Set $(\mathcal{S})$ as the Tic Tac Toe diagrams

$$\mathcal{S} = \{(k, (i, j), a)\}_k, 1 \leq k \leq 9, 1 \leq i, j \leq 3, a \in \{\mathbf{X}, \mathbf{O}\}\}.$$

Then $s \in \mathcal{S}$

$$\begin{aligned} s =& \Bigg\{ \bigg(1, \Big((1,1), \mathbf{X}\Big)\bigg), \bigg(2, \Big((1,2), \mathbf{O}\Big)\bigg), \bigg(3, \Big((2,2), \mathbf{X}\Big)\bigg), \dots \\ & \bigg(4, \Big((3,3), \mathbf{O}\Big)\bigg), \bigg(5, \Big((3,1), \mathbf{X}\Big)\bigg), \bigg(6, \Big((1,3), \mathbf{O}\Big)\bigg), \bigg(7, \Big((2,1), \mathbf{X}\Big)\bigg) \Bigg\} \\ s \equiv & \begin{bmatrix} \mathbf{X} & \mathbf{O} & \mathbf{O} \\ \mathbf{X} & \mathbf{X} & \\ \mathbf{X} & & \mathbf{O} \end{bmatrix}. \end{aligned}$$

Note that exists states $u_1, u_2, \dots, u_6$ such that

$$u_1 \to u_2 \to u_3 \to \dots \to u_6 \to s.$$

In other words, exists a function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ such that

$$S_{t+1} = f(S_t, a_t),$$

where f represent the model dynamics. In this case f was the response of the rival. Consider the initial condition S_0

$$S_0 = \{\} \equiv \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

when you are the first player, if you are the second (with **X** or **O**)

$$S_0((i,j), p) = (1, ((i,j), p))\,, p \in \{\mathbf{X}, \mathbf{O}\}.$$

In both cases, with S_0, the agent continue choose a action a_0 and obtain the following state

$$S_1 = f(S_0, a_0).$$

Later defines the history H_t as the sucession of the following shape

$$H_t = \{S_0, a_0, S_1, a_1, \dots, a_{t-1}, S_t\}$$

4 The k-bandits

P

References