|  |
| --- |
| **Reinforcement Learning-Project 1 Report** |

**Gautam Suryawanshi**

Department of Engineering Science

University at Buffalo

[*gautamja@buffalo.edu*](mailto:gautamja@buffalo.edu)

**Abstract**

The goal of the project is to explore and get an experience of building reinforcement learning environments, following the OpenAI Gym standards. The project consists of building deterministic and stochastic environments that are based on Markov decision process, and applying a tabular method to solve them.

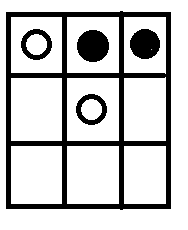
* 1. **Deterministic Environment**

Given the current state of the environment and an action of an AI agent, the AI can know with certainty the next state of the environment.

f(state i, action i) = state f

So the resulting state does not change given the same state and action at all times. This is applicable for both single and multi-agent environments (with only one, or more than one AIs interacting with the environment)

For example, in the case of tic-tac-toe having this environment state and 2 AI agents (black and white) and it is currently the white’s turn.



Choosing the rightmost bottom square always guarantee a win for the white AI agent. There are no instances where that action in this situation does not guarantee a win for said AI agent.Otherwise, the environment is stochastic. For an AI agent to deliberate the next action in this type of environment, it uses probabilities to maximize the performance for a task.

For example, if we tweak the game of tic tac toe and add a mechanism where the location of the current stones change randomly, it can now be called a stochastic environment because predictability of an outcome is not always guaranteed 100% due to the random behavior of the environment

* 1. **Stochastic Environment**

An environment is called Stochastic is where your agent’s actions don’t uniquely determine the outcome. For example in games with dice, you can determine your dice throwing action but not the outcome of the dice.

* 1. **Difference between Stochastic and Deterministic Environment**

While being in a state, taking an action forms a (state, action) tuple. Given a (state, action) tuple say (s1, a1) as an input, a deterministic function will always output the same state s2. For (s1, a2), it may output s3. This deterministic function will always output s2 for (s1,a1) and will always output s3 for (s1,a2).

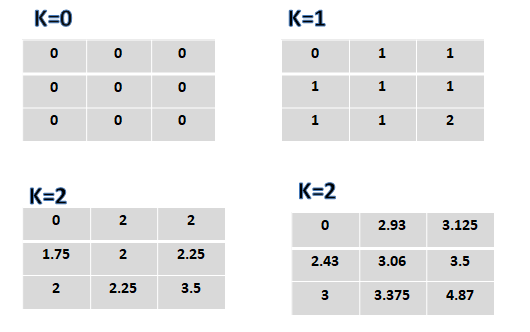
A stochastic function on the other hand for the input (s1,a1) may output s2 will 90% probability but s3 with a 10% probability.So, being in the same state, taking the same action, the behaviour is well-defined in the former, while probabilistic in the later.

* 1. **Rewards ,States, Actions in our Environment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actions** | **Rewards** | **States** | |
| UP | 0 | S1 | S2 |
| RIGHT | 1 | S3 | S4 |
| DOWN | 2 | S5 | S6 |
| LEFT |  | S7 | S8 |
|  |  | S9 |  |

**1.5 Transition Probablity Matrix for Stochastic Environment**

Here we are able to observe that a our policy converges in 3 iterations only. The agent is able to find the optimal path and reaches the state with maximum reward.



**1.6 Main Components of Reinforcement Learning**

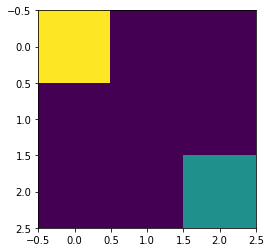
**1) Policy**: Agent’s behavior function

**2)Value Function:** How good is each state and/or action

**3)Model:** Agent’s representation of the environment

* 1. **Results of Deterministic Environment**

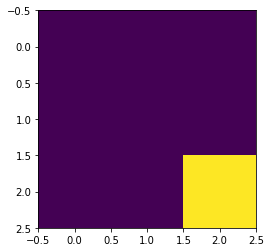
1) Start Position 2) Action: Right

3)Action: Down 4)Action: Right

5)Action: Down



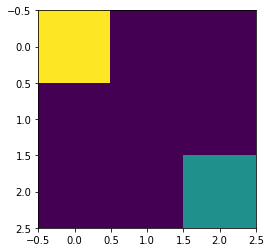
|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **State** | **Action** | **Reward** |
| 1 | S1 | Right | 1 |
| 2 | S2 | Down | 1 |
| 3 | S5 | Right | 1 |
| 4 | S6 | Down | 2 |

* 1. **Results of Stochastic Environment using Dyanamic Programming**

Our agent takes the optimal path using the following transition matrix. It checks where it will get the maximum reward and according to that takes the action.The transition matric is shown below-

|  |  |  |
| --- | --- | --- |
| 0 | 2.93 | 3.125 |
| 2.43 | 3.06 | 3.5 |
| 3 | 3.375 | 4.87 |

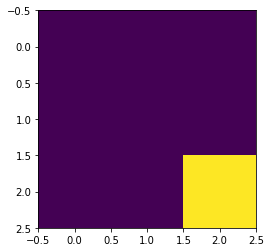
1)Start position 2)Action: Right

3)Action:Right 4)Action:Down

5)Action:Down

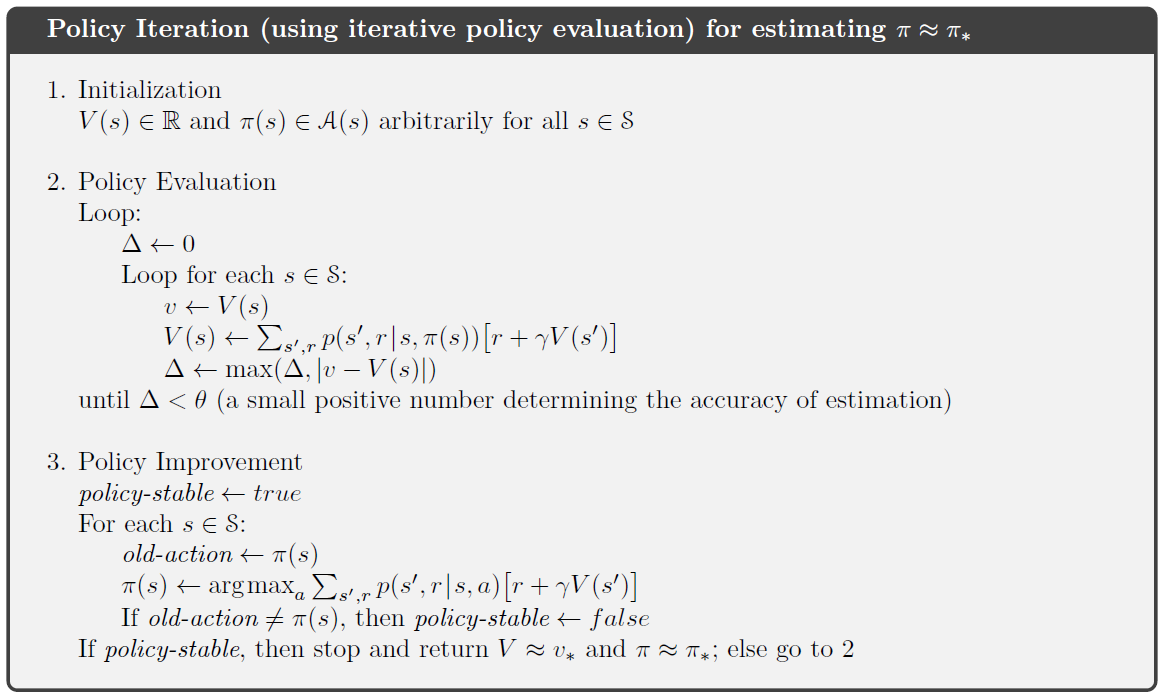


|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Action** | **Reward** | **Total Reward** |
| S1 | Right | 2.93 | 2.93 |
| S2 | Right | 3.125 | 6.055 |
| S3 | Down | 3.5 | 9.555 |
| S6 | Down | 4.87 | 14.425 |

**1.9 Tabular Method Used-Dynamic Programming**

Dynamic programming is a method in which each value at a state is computed by taking as input the values of surrounding states (disregarding if those values are accurate or not). Once a value of one state is computed, we move to another state and we repeat the same process (taking into account any new value computed in previous states).

This process is iterated enough times until changes in every state is less than a certain limit that we have defined.



References

[1] <https://towardsdatascience.com/summary-of-tabular-methods-in-reinforcement-learning-39d653e904af>

[2] <https://www.quora.com/What-s-the-difference-between-a-deterministic-environment-and-a-stochastic-environment-in-AI>

[3] Reinforcement Learning:An Introduction by Richard S. Sutton and Andrew G. Barto