# Mini-Project 1, DP

## Reinfiorcement Learning Spring 2023

**Sara Ghavampour**

9812762781

**Abstract – The goal of this project is to implement and underastand DP methods in one of the OpenAI gym environment called FrozenLake.**

Index terms – DP, Value Iteration, Policy Iteration, Reward

Introduction
DP methods are feasible ways to iteratively compute best estimations of value functions and optimal policy. However, DP methods need complete knowledge of model of the environment or MDP.

Policy evaluation
The goal of this method is to update It's estimation of value function according to the current policy $\pi$ iteratively. This done by by using bellman equation as an update rule to update previous estimation of value function following policy $\pi$ repeatedly.

Policy improvement
The goal is to find a strictly better policy by using policy improvement theorem. It improves the original policy by making it greedy with respect to value function of the original policy. The formula is the same as the bellmen optimality equation.

Policy Iteration
This algorithm combines policy evaluation and policy Iteration to find the optimal policy. Policy Iteration is a dance of evaluation and improvement. The Best estimation of value function is computed and then policy is gridified with respect to last estimation of value function.

Value Iteration
Value Iteration is an algorithm of generalized policy iteration. The formula is really close to policy evaluation except that It evaluates and improves at the same time meaning that policy evaluation is executed for one sweep before gridifying.

## Experiments

| Iteration mode | is_slippery | gamma | Step reward | Hole reward | Time steps | Cumulative reward |
|---|---|---|---|---|---|---|
| Policy | False | 0 | 0 | 0 | 6 | 1 |
| Policy | False | 0 | 0 | -2 | 6 | 1 |
| Policy | False | 0 | -0.05 | 0 | 6 | 0.75 |
| Policy | False | 0 | -0.05 | -2 | 6 | 0.75 |
| Policy | False | 0.9 | 0 | 0 | 6 | 1 |
| Policy | False | 0.9 | 0 | -2 | 6 | 1 |
| Policy | False | 0.9 | -0.05 | 0 | 6 | 0.75 |
| Policy | False | 0.9 | -0.05 | -2 | 6 | 0.75 |
| Policy | False | 1 | 0 | 0 | 6 | 1 |
| Policy | False | 1 | 0 | -2 | 6 | 1 |
| Policy | False | 1 | -0.05 | 0 | 6 | 0.75 |
| Policy | False | 1 | -0.05 | -2 | 6 | 0.75 |
| Policy | True | 0 | 0 | 0 | 13 | 1 |
| Policy | True | 0 | 0 | -2 | 40 | 1 |
| Policy | True | 0 | -0.05 | 0 | 17 | 0.19 |
| Policy | True | 0 | -0.05 | -2 | 60 | -4.94 |
| Policy | True | 0.9 | 0 | 0 | 47 | 1 |
| Policy | True | 0.9 | 0 | -2 | 47 | 1 |
| Policy | True | 0.9 | -0.05 | 0 | 7 | 0.7 |
| Policy | True | 0.9 | -0.05 | -2 | 77 | -2.79 |
| Policy | True | 1 | 0 | 0 | 37 | 1 |
| Policy | True | 1 | 0 | -2 | 17 | 1 |
| Policy | True | 1 | -0.05 | 0 | 47 | -2.3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Policy | True | 1 | -0.05 | -2 | 35 | -0.7 |
| Policy | False | 0 | 0 | 0 | 6 | 1 |
| Value | False | 0 | 0 | -2 | 6 | 1 |
| Value | False | 0 | -0.05 | 0 | 6 | 0.75 |
| Value | False | 0 | -0.05 | -2 | 6 | 0.75 |
| Value | False | 0.9 | 0 | 0 | 6 | 1 |
| Value | False | 0.9 | 0 | -2 | 6 | 1 |
| Value | False | 0.9 | -0.05 | 0 | 6 | 0.75 |
| Value | False | 0.9 | -0.05 | -2 | 6 | 0.75 |
| Value | False | 1 | 0 | 0 | 6 | 1 |
| Value | False | 1 | 0 | -2 | 6 | 1 |
| Value | False | 1 | -0.05 | 0 | 6 | 0.75 |
| Value | False | 1 | -0.05 | -2 | 6 | 0.75 |
| Value | True | 0 | 0 | 0 | 31 | 0 |
| Value | True | 0 | 0 | -2 | 42 | 1 |
| Value | True | 0 | -0.05 | 0 | 86 | -3.25 |
| Value | True | 0 | -0.05 | -2 | 87 | -3.29 |
| Value | True | 0.9 | 0 | 0 | 24 | 1 |
| Value | True | 0.9 | 0 | -2 | 46 | 1 |
| Value | True | 0.9 | -0.05 | 0 | 18 | 0.15 |
| Value | True | 0.9 | -0.05 | -2 | 84 | -3.14 |
| Value | True | 1 | 0 | 0 | 6 | 1 |
| Value | True | 1 | 0 | -2 | 27 | 1 |
| Value | True | 1 | -0.05 | 0 | 32 | -0.55 |
| Value | True | 1 | -0.05 | -2 | 58 | -1.84 |

When gamma is close to 0, state value functions only depend on immediate reward, thus these values may not reach the optimum. On the other hand, gamma=1 results in values take long-term reward into account.

When  is_slippery = True, agent acts differently in each episode and takes more time steps to reach the terminal state because of randomness. In contrast, when  is_slippery = False,all the episodes end in 6 steps.