CS 7641 — Assignment 4: Reinforcement Learning

Yi Qin

# Introduction

# In this project, I will explore reinforcement learning experiments on two Grid world Markov Decision Process problems with three different algorithms which are value iteration, policy iteration and Q-learning. I will compare the results from these 3 algorithms based on converge speed, number of states and answers.

# MDP Problems

Grid world problems have been chosen for this project. Grid world problem is classic Markov Decision Process problem with less complexity of understanding problem itself. I will explore two Grid world problems, one with small size of grid (8 x 8) and another with large number of grid size (20 x 20), the agent will try to reach terminal sate in the least number of steps. At any given time, the agent can choose to move either up, down, left, or right. The catch is that there is a wind which occasionally blows the agent onto a space they didn’t choose.).

## Implementation methods

## The implementation of MDP problem are learned from public GitHub repository and used BURLAP package. The parameters of the Grid world have been set as below,

|  |  |
| --- | --- |
| Parameter | Values |
| **page1image23680Reward Function of goal** | page1image25440100 |
| **Cost Function** | -0.04 |
| **page1image28664Terminal State** | page1image29928Top Right |
| **page1image31232Transition Probability** | page1image329920.8 (correct), 0.067 (incorrect) |
| **Discount Factor** | 0.9 |
| **page1image36440Max Delta** | page1image38200-1 |
| **Max Iterations** | 100, 500 |

## I will receive a reward of 1 if I reach the goal, -1 for failing in a hole and a small negative reward for every step as 0.04.

## Why is this problem interesting?

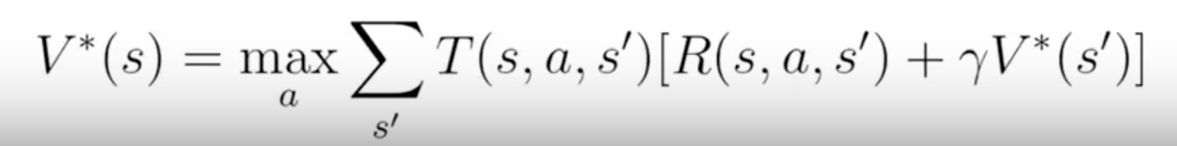
Grid world problems can be treated in real world as navigation problem in Google Map, or used for Uber and Lyft driver navigation. In real world, people want to travel from one staring point to a destination, there are the obstacles which can be buildings, construction areas or park that people or vehicles can’t access into, reinforcement learning can be used to find the best routes with least travel time or least distance. Also, Grid world problems is not complex question and can be used to learn reinforcement learning algorithms.

# Reinforcement Learning Algorithms

Reinforcement learning is one of machine learning techniques that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences. In this section, I will explore three different reinforcement learning algorithms.

## Value iteration

The essential idea is to use Bellman equation to evaluate reward of each state (s) and always choose the action that maximizes expected utility (V). The problem is that true utility (V\*(s)) of each state is unknow at beginning, we only know its immediate reward (R(s,a,a`)). However, the utility of nearby state can be calculated based on discounted(γ) utility of future state (V\*(s`)).



This algorithm is guaranteed to converge to the optimal solutions.

## Policy iteration

Value iteration has two limitations,

* it can take a long time to converge
* it indirectly helps us find the optimal policy

the algorithm had been simplied as:

1. Create arbitrary policy by selecting a random action for each state and calculate its utility under that policy

2. Update utilities from new policy if new policy utility increase

3. iterate until there is no utility increase

## Q-learning

## Unlike policy iteration and value iteration which are relied on domain knowledge, such as transition function and state reward, Q-learning is a model free learning algorithm.

## How does it work: each state is assigned with a Q-value. Q-value can be written as:

Q(s, a) = R(s, a) + γmaxa′ Q(s′, a′)

The value of taking action a in state s is the immediate reward (R(s, a)), plus the value of the best possible state-action pair for the successor state (γmaxa′ Q(s′, a′)).

We can use this to update the Q-value of a state-action pair as a reward r is observed:

Qt+1(s, a) = r + γmaxa′ Qt(s′, a′).

# Implementation

In this section, I will implement Value iteration, Policy iteration and Q-learning algorithm for easy and hard Grid world problems.

## Easy Grid world (8x8)

## Hard Grid world (20x20)

# 