CS 7641 Machine Learning
Assignment 4

Philip Bale
pbale3

Due Sunday April 22nd, 2018 11:59pm

Introduction

This assignment explores reinforcement learning. It begins by choosing two Markov Decision Procceses (MDPs). Both MDPs are solved using both value iteration and policy iteration, and the results are compared against each other. Afterwards, Q-learning is applied to both problems and those results are then compared to the original results.

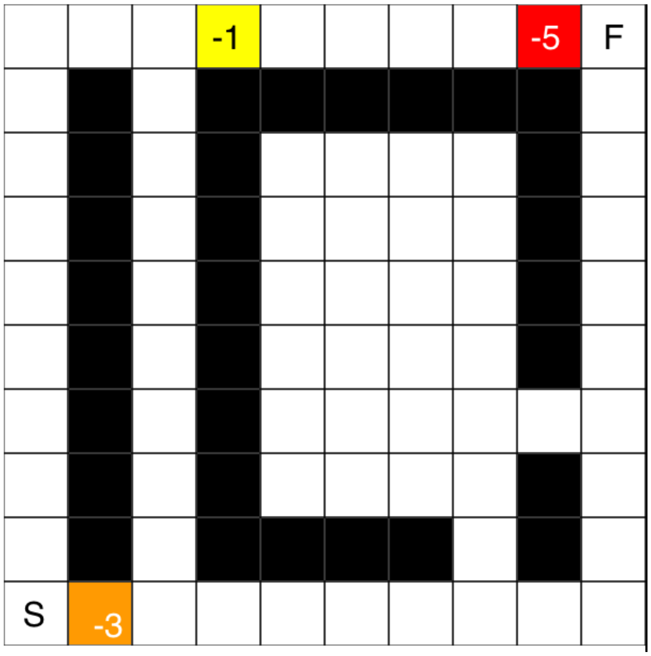
Problems Chosen

Gridworld path finding was chosen as the basis for the two Markov Decision Processes. This was done for a few different reasons. First, the ease in visual representation and understanding. Second, path finding is a common problem in real-life that has applications from walking to work, to driving cross-country, etc. Third, it is easy to scale the complexity of a grid as well as increase in the number of states. Fourth, for a solveable grid, one or more optimal solutions must exist. All-in-all, gridworld is great for demonstration reinforcement learning.

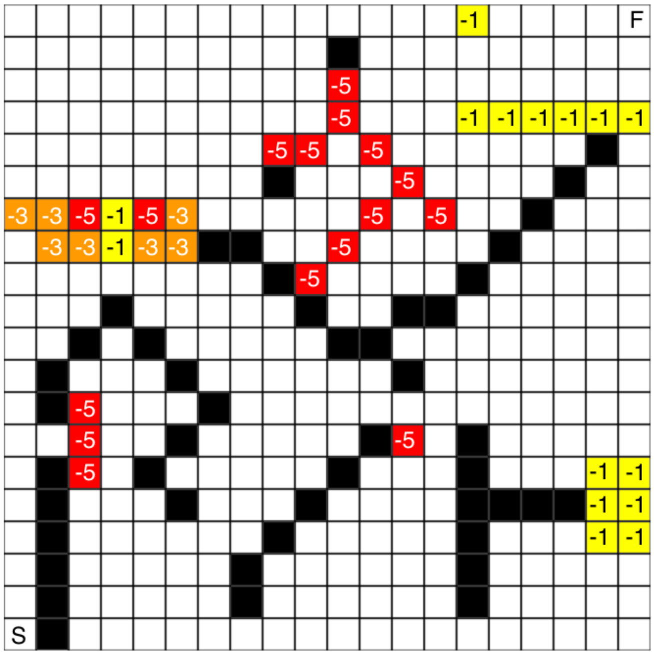
The first gridworld problem is an "easy" configuration. It has a 10x10 shape with 70 possible states.

The second gridworld problem is a "hard" configuration. It has a 20x20 shape with 353 possible states.

Each gridworld has its start state is in the bottom left corner and its end state is in the top right corner. Various "walls" exist that force the solution to optimize around. Various states have negative rewards assosciated with them and are color-coated in the state graphs below.



10x10 Easy Gridworld w/ 70 States



20x20 Hard Gridworld w/ 353 States

Value Iteration

Introduction

The first reinforcement learning algorithm used to optimally solve the MDPs is Value Iteration. Value iteration works by using the Bellman equation while moving from state to state in an attempt to reach convergence. Since the utility is not initially known, the algorithm begins with the reward at the final state and works backwards calculating utility for nearest states. This continues until all states are evaluated. After a number of iterations, the algorithm will eventually converge on an optimal solution.

Policy Iteration

Introduction

The second reinforcement learning algorithm used to optimally solve the MDPs is Policy Iteration. Policy iteration works by beginning with a random policy and then attemping to modify it state by state by taking different actions. This continues throughout each state of the policy in an attempt to find a better solution. When no further policy changes are found, the algorithm is finished. Policy Iteration tends to take longer than Value Iteration due to the additional number of explorations and calculations being done.

Q-learning

Introduction

The third reinforcement algorithm used to optimally solve the MDPs is Q-learning. Q-learning works by assigning q values to every state. At each state, the learner calculates new q values based on both immediate and future rewards. Initially, Q-learning will spend time exploring and learning, whereas it will eventually optimize based on knowledge learned and converge. While the first two algorithms have domain knowledge about the problem, Q-learning has no such information and learns as it goes.