COSC 6368

ARTIFICIAL INTELLIGENCE

PROJECT-2

Q LEARNING PD WORLD

PRISCILLA IMANDI

PSID : 1619570

## Prof. CHRISTOPH F. EICK – Fall 2017

University Of Houston

**Introduction:**

Reinforcement Learning is a branch of machine learning which deals with how agents should react to an environment and which actions need to be taken to maximize the cumulative reward.

The basic reinforcement is modeled as a [Markov decision process](https://en.wikipedia.org/wiki/Markov_decision_process):

1. a set of environment and agent states, S;
2. a set of actions, A, of the agent;
3. {\displaystyle P\_{a}(s,s')=Pr(s\_{t+1}=s'|s\_{t}=s,a\_{t}=a)}is the probability of transition from state s to state s' under action a.
4. {\displaystyle R\_{a}(s,s')} is the immediate reward after transition from s to {\displaystyle s'} with action a.
5. rules that describe what the agent observes

In this project we use Q Learning and SARSA Q learning techniques to obtain attractive paths for an agent to navigate through the World.

**Q Learning:**

The algorithm is a model of the Markov decision process but uses the following formula for every iteration to update values of the Utilities called the Q-Values.

where the transition is being done from state s to s’ using action a with a Reward R(s,a).

In our experiments we have

**Operators:**

Operators are Actions that can be performed on a state S to reach the next state S‘. Here in our world we have 6 different operators : North, South, East, West, Pick-Up and Drop-Off that can be performed on any state.

Policies:

These are the basis for selecting an operator to be applied on a state to reach the next state. We are required to use the following 3 policies.

**PRandom:** Choosing the PickUp and DropOff operators if applicable otherwise choosing any operator randomly.

**PGreedy:** Choosing the PickUp and DropOff operators if applicable otherwise choosing an operator with the maximum utility.

**PExploit:** Choosing the PickUp and DropOff operators if applicable with 0.85 probability and any other operator with 0.15 probability.

**Experiment : 1**

In experiment 1, we use PRandom for 3000 steps and PGreedy for 3000 steps with Q-Learning technique.

The Q tables are generated after each policy. It can be seen that better results are obtained after using PGreedy approach.

The termination state can be reached multiple times before the Experiment is completed.

**Experiment : 2**

In experiment 2, we use PRandom for 200 steps and PExploit for 5800 steps with Q-Learning technique.

It can be seen that PExploit policy works differently from the PGreedy approach. It chooses Exploration method where sometimes even non-promising paths are chosen with some probability so that Rewards can be obtained in the future. Trying to obtain immediate rewards is not as profitable as trying to achieve long-term rewards.

**Experiment : 3**

In experiment 3, we use PRandom for 200 steps and PExploit for 5800 steps with SARSA Q-Learning technique.

The SARSA variant of Q Learning provides better results because it not only computes the Q values for the current state using the policy but also goes a step ahead and does so for the next state as well and uses these values to generate the Q values.

**Performance Measures & Results**

Bank Balance: This is the cumulative reward that we get out of this activity.

Below is given a Bank Balance Table that is obtained for each of the Experiments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Runs** | **Experiment 1** | **Experiment 2** | **Experiment 3** |
| Run 1 | 3854 | 3404 | 5471 |
| Run 2 | 3958 | 3612 | 5733 |

Attractive Path : These paths can be obtained from the Q table by choosing operator at each state based on the Q values of each state. This helps us to reach the termination state using the minimal number of operators.

One of the attractive paths :

(0,4) , (1,4), (2,4),(3,4),(4,4),(4,3),(3,3),(3,4),(4,4),(4,3),(3,3), (3,4),(4,4),(4,3),(3,3),(3,4),(4,4),(4,3),(3,3),(2,3),(2,2)(3,2),(3,3),(2,3),(2,2),(3,2),(3,3),(2,3),(2,2),(3,2),(3,3),(2,3),(2,2),(3,2),(3,3),(3,2),(3,1),(3,0),(4,0),(3,0),(4,0),(3,0),(4,0),(3,0),(4,0),(3,0),(2,0),(1,0),(0,0),(1,0),(2,0),(3,0),(4,0),(3,0),(2,0),(1,0),(0,0),(1,0),(2,0),(3,0),(4,0),(3,0),(2,0),(1,0),(0,0),(1,0),(2,0),(3,0),(4,0),(3,0),(2,0),(1,0),(0,0),(1,0),(2,0),(3,0),(4,0).

An example of the QTable that is obtained.

0   4   0   1   South   -1.489593

1   4   0   0   North   -1.4234779

1   4   0   1   South   -1.2577268

2   4   0   0   North   -1.6204063

2   4   0   1   South   -0.7799443

3   4   0   0   North   -1.6905961

3   4   0   1   South   4.503888

4   4   0   4   Pick-Up   11.010316

4   4   1   0   North   -1.9820759

3   4   1   1   South   -1.9101487

3   4   1   0   North   -1.976473

2   4   1   0   North   -1.9813684

1   4   1   0   North   -1.9837327

1   4   1   1   South   -1.9488711

2   4   1   1   South   -1.921286

0   4   1   1   South   -1.9821585

0   4   1   3   West   -0.3889875

2   4   1   3   West   -1.0362514

**Interpretation Of Results**

1. It can be seen that Randomly choosing an operator doesn’t add as much value to the bank Balance as using some intelligent policy.
2. The PGreedy policy is better than the PRandom policy in that it tries to get immediate gain and tries to get to the state that is promising
3. The PExploit follows the Exploration policy rather than exploitation policy which helps in getting rewards in the future rather than getting immediate rewards. This helps in choosing goods paths even though they don’t appear appealing in the beginning.
4. The SARSA Q learning is much better than Q learning in that it even considers the policies for the next state, i.e, it even considers exploration policy for the next state as well.