TP3 : Q-learning and Approximate Q-learning

Yan Chen & Dajing GU

October 2020

In this TP, the algorithms of Q-learning and approximate Q-learning are implemented in PacMan project of UC Berkeley. Code associated can be found in the GitHub repository: TP3 : Q-learning and Approximate Q-learning.

1 Q-learning

Q-learning is very similar to Value Iteration, however the agent doesn't know state transition probabilities or rewards in Q-learning. The update of Q-table is based on the best action from the next state.

$$Q(S,A) = (1-\alpha)Q(S,A) + \alpha\left[R(S,a) + \gamma \max_a Q\left(S',a\right)\right]$$

or

$$Q(S,A) = Q(S,A) + \alpha\left[R(S,a) + \gamma \max_a Q\left(S',a\right) - Q(S,A)\right]$$

After the implementation of Q-learning in PacMan project, the training result is shown here:

```
Beginning 10 episodes of Training
Pacman died! Score: -506
Pacman died! Score: -518
Pacman died! Score: -510
Pacman died! Score: -513
Pacman died! Score: -507
Pacman died! Score: -506
Pacman died! Score: -506
Pacman died! Score: -514
Pacman died! Score: -509
Pacman died! Score: -505
Training Done (turning off epsilon and alpha)
-----------------------------------------
Average Score: -509.4
Scores:        -506.0, -518.0, -510.0, -513.0, -507.0, -506.0, -506.0, -514.0, -509.0,
 -505.0
Win Rate:      0/10 (0.00)
Record:        Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss
```

Figure 1.1: Q Learning Result of the first ten training

It can be seen that in the fist ten training, the agent loses all the games. However, after 2000 training, teh agent can win nearly all the games in the smallGrid, which is shown in the figure below.

```
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 495
Pacman emerges victorious! Score: 495
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 503
Average Score: 499.4
Scores:        503.0, 499.0, 499.0, 495.0, 495.0, 499.0, 499.0, 499.0, 503.0, 50
3.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
```

Figure 1.2: Q Learning Result

2 Approximate Q-learning

The size of the state-action Q-value table is decided by the factors below:

1) PacMan's location (m possibilities)

2) Location of each ghost (m^2).

3) Locations still containing food (2^{m-2}).
   Not all feasible because PacMan can't jump.

4) Ghost remaining (2^2 possibilities).

5) Whether each ghost is scared (2^2 possibilities ... ignoring the timer)

There are too many factors that the agent should taken into consideration during the learning so some approximate methods to improve the efficiency of learning.

Key Idea: To learn a reward function as a linear combination of features. We can think of feature extraction as a change of basis. For each state encountered, determine its representation in terms of features. Perform a Q-learning update on each feature. Value estimate is a sum over the state's features.

$$Q(s,a) = \sum_{i}^{n} f_i(s,a) w_i$$

$$w_i \leftarrow w_i + \alpha[\text{correction}] f_i(s,a)$$

$$\text{correction} = (R(s,a) + \gamma V(s')) - Q(s,a)$$

After the implementation of Approximate Q-learning in PacMan project, the training result is shown here:

```
-----------------------------------------
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 495
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 503
Pacman emerges victorious! Score: 499
Pacman emerges victorious! Score: 499
Average Score: 500.2
Scores:        499.0, 503.0, 499.0, 503.0, 503.0, 495.0, 499.0, 503.0, 499.0, 499.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
(pacman) AppledeMacBook-Pro-2:pacman apple$
```

Figure 2.1: Approximate Q Learning Result

Compared with figure 1.2, the ApproximateQAgent has a relatively higher average score, which can be explained by the simpler update method.