CS 747 Assignment 3

Advait Padhye (200100014)

6th November 2022

1 Method Description

Firstly, this is a control problem since we need to estimate the policy and try to take optimal actions in order to maximise the long term reward. Secondly, this is an episodic task since in finite number of timesteps, the car will either reach the road, go out of the grid, crash into a mud pit or simply never reach the target and timeout. Since this is an episodic task, we will take the discount factor (γ) = 1. Finally, there are an infinite amount of states present in the underlying MDP for this problem since we can change either the position or direction of the car as well as the environmental parameters like mud pit positions change with episodes. Therefore, in order to solve the overall problem, we need to use function approximation for Control problem.
In control problem, we will try to approximate the action-value function $\hat{Q}$. Using this approximate $\hat{Q}$ function, we will take the optimal action using ϵ-greedy approaches. Now, to approximate the action-value function $\hat{Q}$, we will be using Linear Function Approximation.

$$\hat{Q}(\mathrm{S}, \mathrm{A}, \theta) = \theta^{\mathrm{T}}\mathbf{x}(\mathrm{S}, \mathrm{A})$$

Here, θ are the weights and $\mathbf{x}$(S,A) are the features associated with that state and action.
Now, to get the best approximation of Q, we will need to find the best weights (θ). For this, we will be using semi-gradient descent (the method which was described by professor in Lecture 17) in conjunction with SARSA control algorithm. This is an instance of on-line algorithm. The update step looks something like

$$\theta^{t+1} \leftarrow \theta^t + \alpha_{t+1}\{Q^\pi(s^t, a^t) - \hat{Q}(s^t, a^t, \theta^t)\}\nabla_\theta \hat{Q}(s^t, a^t, \theta^t)$$

Here, we will take $\alpha_t = \dfrac{1}{t}$. Since we will be using SARSA control algorithm,

$$Q^\pi(s^t, a^t) \approx r^t + \gamma\hat{Q}(s^{t+1}, a^{t+1}, \theta^t) = r^t + \hat{Q}(s^{t+1}, a^{t+1}, \theta^t)$$

The update step will now look like

$$\theta^{t+1} \leftarrow \theta^t + \alpha_{t+1}\{r^t + \hat{Q}(s^{t+1}, a^{t+1}, \theta^t) - \hat{Q}(s^t, a^t, \theta^t)\}\nabla_\theta \hat{Q}(s^t, a^t, \theta^t)$$

Since, we are using linear function approximation the gradient of action-value function will be given by

$$\nabla_\theta \hat{Q}(s^t, a^t, \theta^t) = \mathbf{x}(s^t, a^t)$$

The final update step will be given by

$$\theta^{t+1} \leftarrow \theta^t + \alpha_{t+1}\{r^t + \hat{Q}(s^{t+1}, a^{t+1}, \theta^t) - \hat{Q}(s^t, a^t, \theta^t)\}\mathbf{x}(s^t, a^t)$$

2 Next Action Function

As mentioned earlier, we will be using ϵ-greedy approach, to decide the next optimal action.

Algorithm 1 NextAction

```
n ← uniform random number between 0 and 1
if n < ε then
    A ← random action
else
    A ← max Q̂(S, .)
end if
```

3 Features

3.1 Task 1

We will be using four features for this task.

1. Distance of the car from the center of the road
2. The difference between the heading angle and the angle made with the horizontal by the line joining the car and the centre of the road

   Note: When the car is in the top right corner or the bottom right corner, instead of considering the above angle for the value of the second feature, we will take 90 if the car is in the top right corner and 270 if the car is in the bottom right corner. This is done because in certain experiments when the car started in the top right or bottom right corner, the car would simply go to the right and exit the grid without ever reaching the road. Hence, until the road starts, we must prevent the car from heading towards the centre of the road and instead make it run vertically towards y=0. Same goes for the feature in task 2.
3. Distance of the car from the left boundary of the grid

   Note: When the car is in the top right corner or the bottom right corner, instead of considering the horizontal distance from the left boundary, we will be considering the horizontal distance from the right boundary. This is done because in certain experiments when the car started in the top right or bottom right corner, the car would simply go to the right and exit the grid without ever reaching the road. Hence, until the road starts, we must prevent the car from going towards right which is done by taking the value of the third feature as the horizontal distance of the car from the right boundary of the grid. Same goes for the feature in task 2.
4. Distance of the car from the vertical boundaries of the grid
   - If the y-coordinate of the car is negative then we consider the distance of the car from the upper boundary of the grid for this feature
   - If the y-coordinate of the car is positive then we consider the distance of the car from the bottom boundary of the grid for this feature

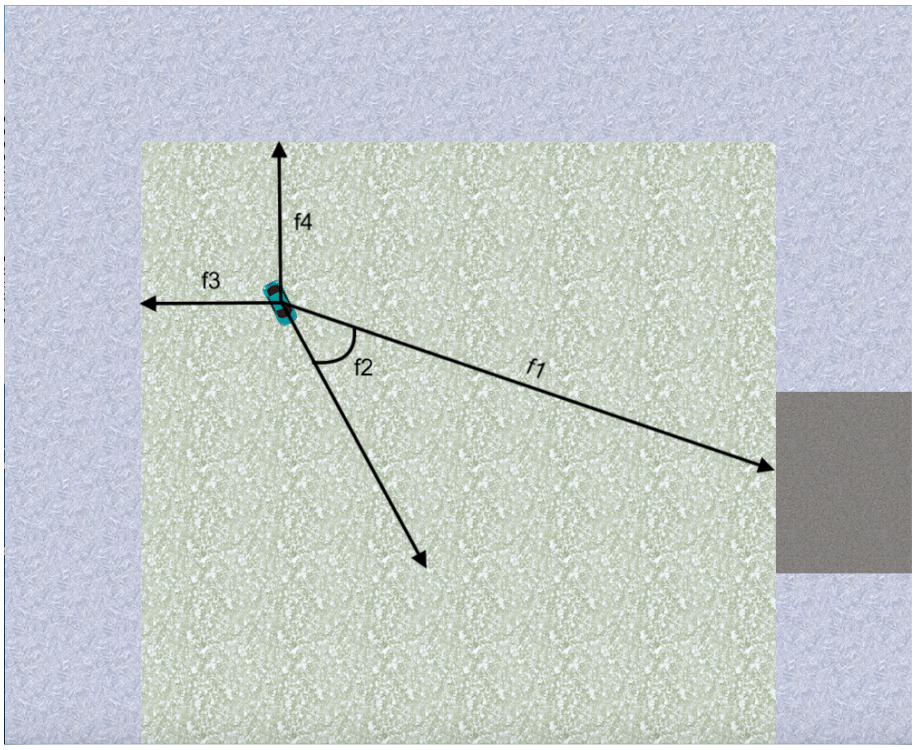

Figure 1: Features for Task 1

3.2 Task 2

We will be using six features for this task.

1. Distance of the car from the center of the road
2. The difference between the heading angle and the angle made by the line joining the car and the centre of the road with the horizontal
3. Distance of the car from the left boundary of the grid
4. Distance of the car from the vertical boundaries of the grid
   - If the y-coordinate of the car is negative then we consider the distance of the car from the upper boundary of the grid for this feature
   - If the y-coordinate of the car is positive then we consider the distance of the car from the bottom boundary of the grid for this feature
5. Difference between the heading angle and the angle made with the horizontal by the line joining the centre of the car and the closest point of the mud pit from the car
6. Distance of the car from the closest point of the mud pit

   Note 1: These two extra features will be considered only when the y-coordinate of the car is above 50 or below -50. This is done because there won't be any mud pits between y=50 and y=-50 and by incorporating these features in this case will result in driving the car slow thus making it inefficient.

   Note 2: These values depend on the closest point of the mud pit from the car and hence 8 cases are considered for calculating these two extra features based on the relative positioning of the car from the mud pit.

   Note 3: These features will be considered if the distance of the closest mud pit from the car is less than 50. Also, we will consider these features if the car is straight headed towards these pits.

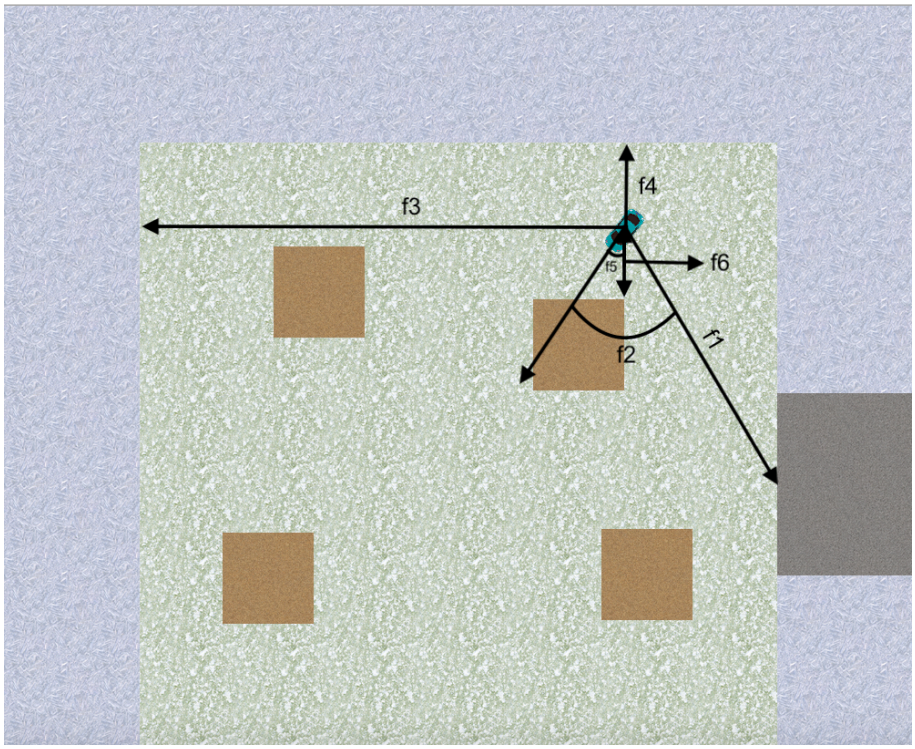

Figure 2: Features for Task 2

Here, f1 refers to the first feature and so on.

4 Parameter Tuning

The parameters for both these tasks include

1. The initial weights
2. ϵ which controls the amount of exploration
3. The initial action taken by the car

Since all the features are equally important for the car to reach its goal, I gave same weights to all the features in the first task. However, in the second task, after testing the car from several different starting positions, I inferred that the fifth feature is far more important than any other feature and hence must be given a larger initial weight. This is because if the car is headed towards the centre of the road and there is a mud pit in between the path then ideally it should change its trajectory away from the pit. However, when we give equal weights to both these features then the car will be in a state of dilemma over whether to change the direction of the car away from the centre of the road or to change its trajectory away from the mud pit. Most often times, in this case, the car remains in the same position resulting in a timeout.

I observed that the initial action that the car should take must be zero change in heading angle and zero acceleration. This is generally a safe option since we don't know where the car is currently positioned and where the surrounding mud pits are present.

The value of ϵ controls the amount of exploration done by the car. For high values of ϵ, the car doesn't obey the learned policy and goes in arbitrary directions resulting in crashing into mud pit or the boundary of the grid in some cases. Hence a small value like 0.2 and 0.3 was found to be favourable to allow for sufficient amount of exploration while being greedy with respect to the learned policy.

I even considered adding another feature which took into account the velocity of the car. By adding this feature, we can make the car to run at faster velocity. This could help in decreasing the amount of time it takes for the car to reach the road and thus make the car more efficient. However, in certain cases, the car with high velocity wouldn't be able to stop in time and dodge the obstacles in its way. This is especially concerning in case of mud pits in task 2. Hence, I have excluded this feature from the feature vector.

Also, I have scaled the values of all the features so that they lie in between 0 and 1. The greatest difficulty I faced during this assignment was to decide the features and how much weight should be given to each feature because these essentially decide the next action of the car. After several experiments and testing, I finalised on the above features which passed most of the testcases.