CS 747 Assignment 3 Report

Shashank Roy - 180050097

Task 1

For this task we dicretize both position and velocities. Feature vector represents a boolean vector which has a single value as 1 pertaining to the current state the environment is in. Actions are chosen optimally as well as exploratory.

The task requires the perfect amount of discrete states through which environment can be represented adequately. However very large number of states can backfire since weights won't be updated appropriately owing to limit on number of episodes used to train. Therefore after a lot of trial and error following values are used.

Both position and velocity are discretized into 40 states. Hence 1600 states.
Effectively the shape of weight vector is 3 x 1600.

Parameters:
epsilon_T1 (ϵ) = 0.05
learning_rate_T1 (α) = 0.1

With the seeds already provided, using the specified parameters and the granularity of states we get test reward of: **-148.54**
Even on removing the seed (which causes random initializations of environment) the final average reward always comes greater than -160.

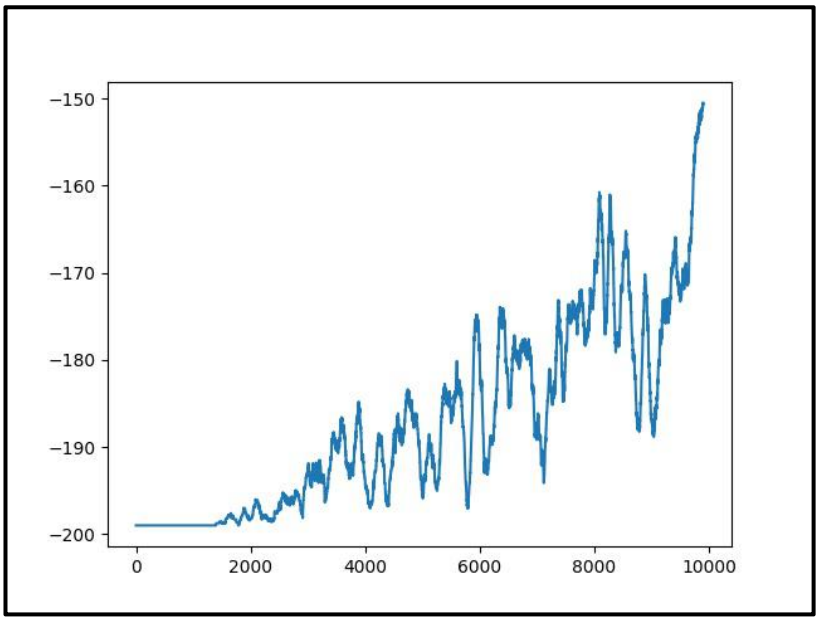


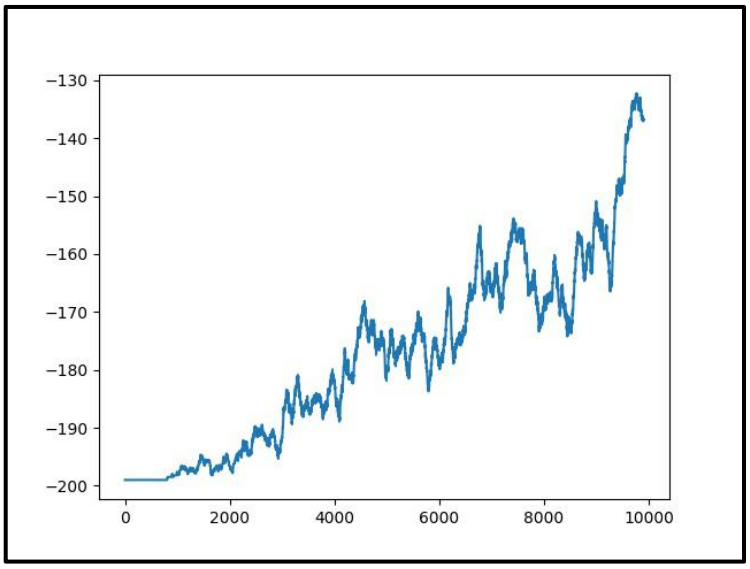
Figure obtained while training

As we can see the plot of reward is noisy (maybe due to the exploratory actions chosen) but it increases overall with subsequent episodes.

Further Experiments: (This has not been used in code and just reported for experimentation)

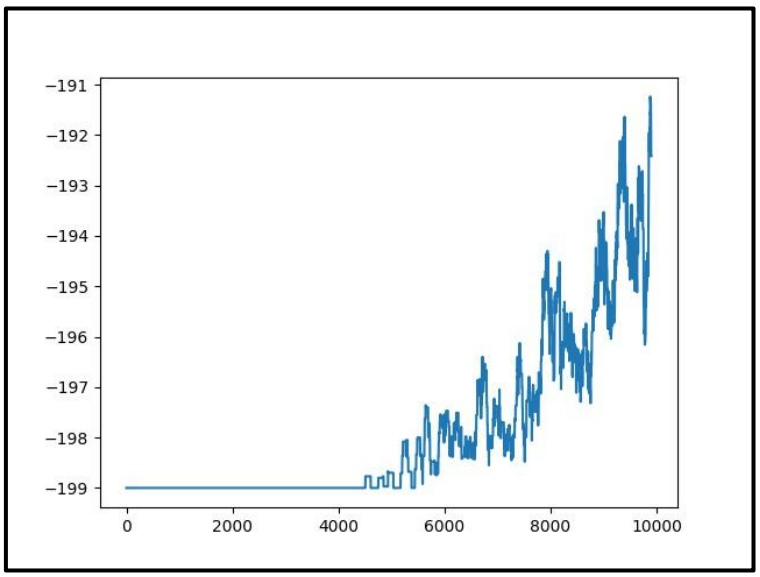
For this we split both position and velocity to 70 values giving a total of 4900 states. Hence it takes more time to train owing to calculation of bigger matrices.

For epsilon_T1 = 0, learning_rate_T1 = 0.85 we achieve very high test reward of **-133.22**.



As we can see rewards increase with episodes.It gives much better rewards than previously stated. However since epsilon is 0 it is not explorative.

With epsilon_T1 = 0.05 (very small) the situation dramatically worsens.



Rewards go down to -190 showing that with too many states we don't have the luxury to explore actions with limited number of episodes to train on.

Task 2

In this task we represent states through RBF kernel.

Process:
We choose 400 state **values** uniformly from both position and velocity ranges.
We create states from 20 position values and 20 velocity values.
Then these (position,velocity) 2D data points are standardized by subtracting mean and dividing by the standard deviation. Hence we have the centroids which will be used to get the final feature vector from the observed state values.

To featurize an observation, we standardize the observation data point by subtracting the mean and dividing by the standard deviation calculated from above. Then applying the standard RBF kernel function with each centroid:

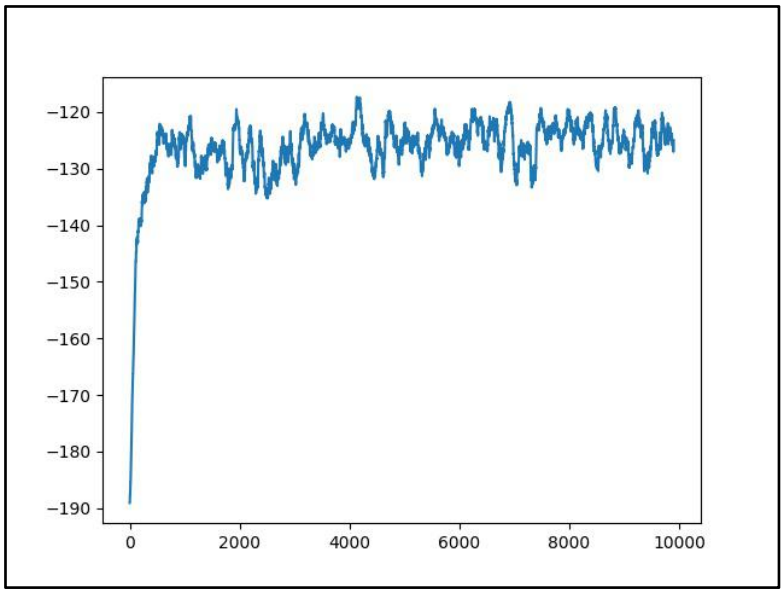
$$K(x, y) = \exp(-\gamma \| x - y \|^2)$$

Where **x** refers to standardized observation and **y** refers to a centroid data point. Hence we get the kernel values of the observation with each of the 400 centroid data points and return a vector of these values as the feature vector. Therefore weights are of shape: 3 x 400

Parameters:

epsilon_T2 (ϵ)	=	0.1	
learning_rate_T2 (α)	=	0.5	
gamma (γ)	=	50	(this refers to the value used in RBF kernel function)

On test run with the given seeds set, the trained weights give reward of **-124.83**.
Even on removing the seed the final average reward always comes more than -130.



As evident from the plot, reward increases dramatically in early episodes and then oscillates around 120-130 range. This shows the superiority of RBF kernels.