Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Machine Learning (CS60050)
Instructor : Aritra Hazra

Vaibhav Saxena : 20CS60R57 || Suprajit Sardar : 20CS60R53

01-April-2021

Drawbacks of existing Q learning algorithm:

The existing Q learning algorithm discussed so far has some drawbacks, In this section we will discuss the solutions to these drawbacks.

Exploration vs. Exploitation

There are two competing objectives in Q learning model of reinforcement learning which can be thought of as below :

- Always try to find a new action to do or an action which have not been done very often hoping that the action may lead to discovery of something never discovered before i.e a large reward state, a transition that leads to a good reward state etc.
- Always try to take the greedy action with the information in hand about the world encapsulated by the Q function in order to maximize the reward.

Now with the above two choices in hand we have competing objective of what should be done to increase the chance of getting an optimised result.

Exploration:

Must take actions that may be sub optimal but help discover new rewards and in the long run increase utility.

Exploitation:

Must take actions that are known to be good (and seem currently optimal) to optimize the overall utility.

Fun examples:

Some real life examples from the general behaviour of people towards life.

- **Explorers-** They always have some new stuff in the box to learn or in other words they keep on switching their interests time to time.
- **Exploiters-** Perfectionists in a single domain (lets say maths) ask any question related to mathematics and bang on they have an answer to it but if you ask them about any other stuff in the world then they may fumble.

Now as we can easily infer that both these approaches are not intelligent enough and at some point of time we may get stuck to the **local maxima** so their is a fundamental trade off between exploration and exploitation in reinforcement learning in order to improve the chances of getting an optimal outcome.

Solution:

Slowly move from exploration to exploitation by enhancing the formula of probability of choosing an action **a** given a state **S**. Select an action a with the below probability:

$$P(a|S) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{a' \in A} e^{\frac{Q(s,a')}{T}}}$$

The above formula can be considered as an analogy of simulated annealing and the effect of increasing and decreasing T is as below:

- If T is large lots of exploring.
- If T is small, follow current policy.

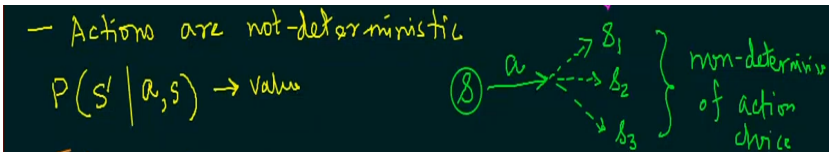
T can be decreased over time in order to slowly move from exploration to exploitation.

Non determinism in outcome of actions:

Setup of MDP in the Q learning algorithm has the actions to be deterministic but in various scenarios their is a non determinism in outcome of actions example in the game of backgammon.

Pictorially:

If we choose an action **a** then due to non determinism in the system we could land up in the environment to potentially different states as shown below:



Solution:

We can incorporate the expectations in terms of probability in order to tackle the non determinism.

Value function:

Value function of a policy in a given state S is given by the expected sum of the cumulative rewards given as below equation:

$$V^{\pi}(S_t) = E[\sum_{i=0}^{\infty} \gamma^i r_{t+1}]$$

Q equation:

Q equation of a policy can be modified as taking the expectation over the existing bellman function given as below equation:

$$Q(s,a) = E[r(s,a) + \gamma V^*(\delta(s,a)]$$

Simplifying the above equation we get:

$$Q(s,a) = E[r(s,a)] + \gamma E[V^*(\delta(s,a)]$$

$$Q(s,a) = E[r(s,a)] + \gamma \sum_{s'} P(s^{'}|s,a) V^*(s^{'})$$

Therefore final recursive value of the $\hat{Q}$ function is given by below equation:

$$\hat{Q}(s,a) = E[r(s,a)] + \gamma \sum_{s'} P(s^{'}|s,a) max_{a'} \hat{Q}(s^{'}, a^{'})$$

Probability of $s^{'}$ given the present state s and action a can be computed by MLE as Number of epochs such that it reaches $s^{'}$ from state s and action a divided by Total number of epochs with combination s,a as shown below:

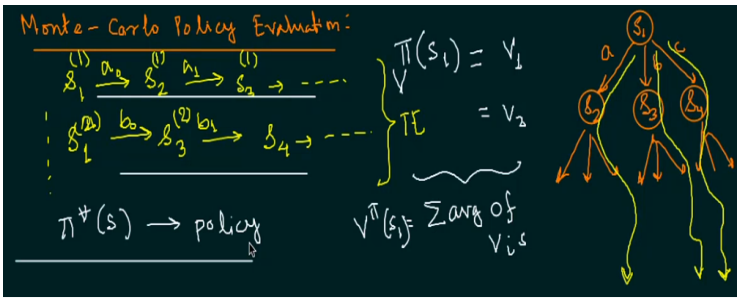
$$P(s^{'}|s,a) = \frac{Count(s^{'} \leftarrow s, a)}{Count(s,a)}$$

Ramifications:

One of the problems with the environment is that rewards usually are not immediately observable. We only know the rewards on the final move (terminal state). All other moves will have 0 immediate rewards. It is to predict a variable's expected value in a sequence of states. It uses a mathematical trick to replace complex reasoning about the future with a simple learning procedure that can produce the same results. Instead of calculating the total future reward, It tries to predict the combination of immediate reward and its own reward prediction at the next moment in time.

Monte Carlo Policy Evaluation:

Some times for experimentation we go for Monte Carlo simulation.



Here, we can see that from S_1 we can reach to S_2, S_3, S_4, by applying input a,b,c respectively. In this Monte Carlo Simulation we use the average of all this paths.

Control:

One of practical domains where the Reinforcement Learning is applied. There is a plant, sensor, actuator and an agent which senses the plant and actuates the plant dynamic.

- **Curse of Dimensionality:** Let there are n dimension and if we discretize the n into k values, then the state phase will be K^n. So, here the problem is the number of attributes or the dimensions increases, the number of training samples required to generalize a model also increase phenomenally, but in reality the available training samples may not have observed targets for all combinations of the attributes. This is because some combination occurs more often than others. Due to this, the training samples available for building the model may not capture all possible combinations.

Contribution:

Vaibhav Saxena: 20CS60R57 −− > Till Non determinism in outcome of actions i.e Page (1-3).
Suprajit Sardar: 20CS60R53 −− > Ramifications to till end of the lecture.