Week 1

Sunday, August 25, 2019 8:51 AM

Reinforcement Learning is a subfield under machine learning. In RL, the agent collects it's own data by trial and error. Every time, an agent is asked to make a decision, it is given a few options. The agent gets a reward for selecting an option. The agent's job is to maximize the reward for it's actions.

If the reward for actions was known, the problem of RL would be trivial. The agent can greedily select the actions with the highest reward. However, the reward most often is not known. What we know are the estimates of these rewards. The estimates are known as action values.

Theoratically, the expected reward is given as:-

$$q^*(a) \doteq E[R_t \mid A_t = a] \quad \forall a \in \{1, \ldots, k\}$$

ie the expected reward is the summation of expected rewards received if an action a is taken at times 1 .... k.

This can be found by

$$q^*(a) = p(r \mid a) \cdot r$$

Again the goal is to maximize this expected reward which can also be represented as

$\arg\max_a q_*(a)$ ie the selection of actions yielding max expected reward.

In practical terms, we can find this by doing an experiment multiple times and plotting the results or using other stastical methods to find the kind of distribution. Then just find the expected value for that distribution.

mean of the distribution

This problem is called the problem of bandits. It is a smaller problem under RL and is a good intro to the field of RL.

Method of Estimating $q^*(a)$ [Sample Average Method]

One method of estimating $q^*(a)$ is called the sample average method. In the sample average method, we estimate the action value using previous values.

$$q^*(a)_n = \frac{a_1 + a_2 + \cdots a_{n-1}}{n-1}$$

$$= \frac{\sum_{i=1}^{n-1} a_i}{n-1}$$

It's $n-1$ because the value is based on previous

It's n-1 because the value is based on previous values.

This method has a problem that in the above form, the time and space complexity grows by O(n). We can make this better using previous values as follows :-

$$q^*(a)_{n+1} = \frac{\sum_{i=1}^{n} a_i}{n}$$

$$= \frac{1}{n}\left[a_n + \sum_{i=1}^{n-1} a_i\right]$$

$$= \frac{1}{n}\left[a_n + \frac{(n-1)}{(n-1)} \sum_{i=1}^{n-1} a_i\right] \rightarrow q^*(a)_n$$

$$= \frac{1}{n}\left[a_n + (n-1)\, q^*(a)_n\right]$$

$$= \frac{1}{n}\left[a_n + n\, q^*(a)_n - q^*(a)_n\right]$$

$$\boxed{q^*(a)_{n+1} = q^*(a)_n + \frac{1}{n}\left[a_n - q^*(a)_n\right]}$$

This is a very common form in RL. It can also be written as

New Estimate = Old Estimate + StepSize [Target − Old Estimate]
or New Estimate = Old Estimate + Step Size × Error

The $\frac{1}{n}$ term is called the step size.
The choice of $\frac{1}{n}$ is varied according to the problem.

If we do not have a stationary RL problem, ie the distribution of action values varies with time, we use a constant step size b/w $(0,1]$.
What this does is that it weighs the newer values more in the calculation of action value estimates.
This can be seen as follows :-

constant step size

$$q^*(a)_{n+1} = q^*(a)_n + \alpha [a_n - q^*(a)_n]$$

$$= q^*(a)_n + \alpha a_n - \alpha q^*(a)_n$$
$$= \alpha a_n + (1-\alpha) q^* a_n \quad \text{—— } ①$$
$$= \alpha a_n + (1-\alpha)[\alpha a_{n-1} + (1-\alpha) q^* a_{n-1}] \quad [\text{Using } ①]$$
$$= \alpha a_n + (1-\alpha)\alpha a_{n-1} + (1-\alpha)^2 q^* a_{n-1}$$
$$= \alpha a_n + (1-\alpha)\alpha a_{n-1} + (1-\alpha)^2 [\alpha a_{n-2} + (1-\alpha) q^* a_{n-2}]$$
$$= \alpha a_n + (1-\alpha)\alpha a_{n-1} + (1-\alpha)^2 \alpha a_{n-2} + (1-\alpha)^3 q^* a_{n-2}$$

$$= \alpha a_n + (1-\alpha)\alpha a_{n-1} + (1-\alpha)^2 \alpha a_{n-2} + \cdots + \cdots + (1-\alpha)^{n-2}\alpha a_2 + (1-\alpha)^{n-1}\alpha a_1 + (1-\alpha)^n q^*(a)_1$$

$$= \left[(1-\alpha)^n q^*_1(a) + \sum^{n} \left[\alpha (1-\alpha)^{n-i} a_i \right] \right]$$

$$= \boxed{(1-\alpha)^n q_1^*(a) + \sum_{i=1}^{n} \left[\alpha (1-\alpha)^{n-i} a_i \right]}$$

From this =n, we can see that as we go closer to the current value, the weightage goes higher. This is because $(1-\alpha) < 1$ and greater the value of i, smaller is n-1 and higher will be $(1-\alpha)^{n-1}$.

We also see that $q^*(a)$, ie the initial estimate influences the estimate hugely.
This can be used to influence exploration or exploitation as will be seen later.

Exploration - Exploitation Dilemma

When initially an experiment is started, the agent has no knowledge of the estimates for actions. The agent can explore initially to get some estimates. After some time, it has to now decide if it should explore more or exploit the knowledge it has acquired. The agent can exploit it's current knowledge by greedily selecting the estimates with the highest expected values. It may learn better actions if it explores more or it may not.

One way to approach this problem is to behave greedily most of the time and sometimes decide

One way to approach this problem is to behave greedily most of the time and sometimes decide to explore. The agent can explore with a probability ε and exploit with a probability $1-\varepsilon$. This method is known as ε greedy method of exploration.

The textbook shows the following results for this strategy as compared to the strategy with $\varepsilon = 0$:-