Monte Carlo R-L.

Planning by D.P. can be done when MDP is known.

It requires knowledge of transition probabilities $P(s, a, s')$

But when MDP is unknown, model free methods need to be used

Monte carlo is a model free method that learns directly from episodes.

MC learns from full episodes. It does not use bootstrapping.

All episodes must terminate

MC use empirical mean return instead of expected return.

Goal : Learn V_s from episodes of experiance under policy π

$$V_\pi(s) = E_\pi \left[G_t \mid S_t = s \right]$$

But in monte carlo, transition probability is not available.

Hence there is no weights of probability to multiply the rewards by.

Hence we cannot use E_π (expectation of reward of policy)

In monte carlo, empirical transition E is used

Value function for a policy is calculated by running the episodes and calculating reward for every state

$$\begin{array}{c} V \\ \circlearrowright \\ \pi \end{array} \longrightarrow V^*_{\pi^*}$$

Iteratively the policy is also improved.

First Visit Monte Carlo Policy Evaluation

To evaluate V of a state s

The first time that state s is visited in an episode (time t_i)

Increment counter $N(s) \leftarrow N(s) + 1$
Increment total return $S(s) \leftarrow S(s) + G_{t_i}$

At end of all episodes

$$V(s) = \frac{S(s)}{N(s)}$$

Here G_{t_i} is total discounted reward obtained from time t_i till end of episode

$$G_{t_i} = \sum_{n=i}^{end} \gamma^{n-i} R_n = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots R_T$$

Ignore subsequent visits to the same state

The visits are done w.r.t. policy

Every Visit MC

Do updation every time state is visited

First Visit MC	Every Visit MC
Unbiased estimate of value function	Biased because the returns for subsequent visits are not independant.
Higher variance	Lower variance since it utilizes all the data

Variance in Monte Carlo

In monte carlo, the calculations for all the state is not done unlike D.P.

Each episode is influenced by random events Stochastic environments or different consequences of actions can be random.

The total return G can vary significantly from one episode to next, even if the agent visits the same state multiple times.

In such cases, this causes a lot of variance in the model

Another source of variance is that MC waits until the episode is over to update values, so later rewards that are not directly connected to the current state are also considered.

Unusual rewards at any point in the episode can impact the value estimate for a state, resulting in higher variance.

For example episode ends with a rare large reward it can lead to a significant jump in the value update causing large swings in learning

V_π in Monte Carlo

V in M.C. is always with respect to π

This is because reward of all future states is considered in G_t

V is the goodness of states when agent follows π

the future states visited depends on π

$$V(s) \leftarrow V(s) + \frac{1}{N}\left(G_t - V(s)\right)$$

$$\Downarrow$$

$$\underbrace{r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \gamma^n r_{t+n}}$$

$$\Downarrow$$

depends on policy π
$\therefore V(s)$ also depends on π

V in MC depends on the policy taken π.

Changing π will change V, since π determines the actions that the agent will take from state S.

eg agent takes first move in chess as E4 as per its policy π

Case 1: agent is not trained properly and converges to non optimal π

V of E4 is less

Case 2: agent is trained properly and can win nearly every game

V of E4 is high

The actual value of $V(E_4)$ is not determined but the value functions determined are under policy π.

Episodes $\longrightarrow$ reward $\longrightarrow$ V $\longrightarrow$ π

run episodes w.r.t. π

r
some states are more rewarding than others

find goodness of states w.r.t. π from G_t

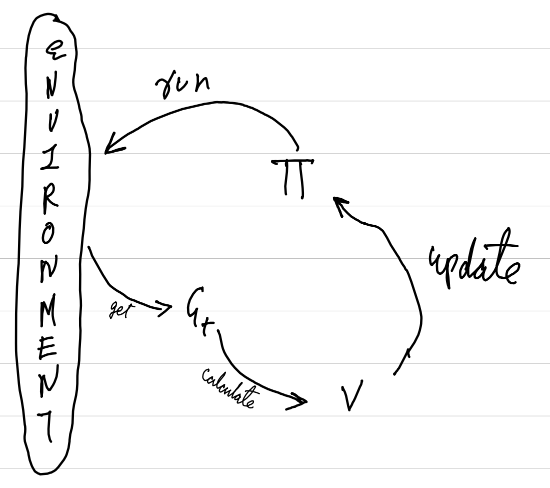
The policy for rewarding states are increased

r does not depend on π

r depends on s & a

choosing s & a depends on π

So G_t (total returns obtained) depends on π, but not r



Theoritically, G_t will be

G_t run episodes $\longrightarrow$ Probability of action w.r.t. π $\times$ reward obtained

eg
$$\begin{tikzcd} D & A \arrow[l, "d"', "R_D"] \arrow[r, "b", "R_B"'] \arrow[d, "c", "R_C"'] & B \\ & C & \end{tikzcd}$$

In iteration over many episodes

$$\therefore G_{tA} \simeq b \times R_B + c \times R_C + d \times R_D$$

[G_t is a measure how much rewards will be obtained on following policy π]

$$\pi = \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

$$V_A \leftarrow V_A + \alpha\left(G_{tA} - V_A\right)$$

$\therefore V_A$ depends on (b, c, d) ie. π

Then improve π greedly wrt V

Note $\rightarrow$ states explored BCD also depends upon π

Actions explored bcd also depends upon π

rewards r dont depend on anything

Incremental MC Updates (Prediction)

Update $V(s)$ incrementally after each episode instead of doing at the end

$$N(S_t) \leftarrow N(S_t) + 1$$

$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)} \left(G_t - V(S_t)\right)$$

This is exact same as the MC updates

However, we can now modify this equation to give importance to new episodes.

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t - V(S_t)\right)$$

This is done in order to forget old episodes.

Useful for modelling environments that are non stationary

ε- greedy Policy

In ε - greedy policy, we make choice between exploration & exploitation

While running the episode

Probability of $\varepsilon \rightarrow$ chose random action (explore)

$1-\varepsilon \rightarrow$ chose highest estimated value (exploit)

So agent will exploit with prob $(1-\varepsilon)$ & explore with probability ε

MC for deterministic vs Non deterministic

Deterministic

$$\pi(s) = a$$

That means episodes will choose only the best so far case while exploiting

$$\pi(s) \leftarrow \underset{a}{\text{Argmax}}\ Q(s,a)$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(G_t - Q(s,a)\right)$$

Stochastic policy

$$\pi(a|s) = P(a|s)$$

That means the agent is allowed to explore even in the exploitation phase

$$\pi(a|s) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}}$$

where τ is temperature parameter that controls the level of exploration

On Policy Monte Carlo

The methods till now were on policy, that is we used π to generate episodes

$$\begin{tikzcd} \pi \arrow[r] & \text{episodes} \arrow[l, bend left=60, "\text{update}"] \end{tikzcd}$$

Policy to be optimized π,
Same policy is used for generating episodes

But this is a compromise

It aims to learn action values V or Q that are dependant on subsequent optimal behaviour π^* but have to behave suboptimally to explore all actions and find optimal action

Learns V, Q not for π^*, but for a "near optimal" policy that still explores π^{ε}
eg epsillon greedy

Remember V & Q are always w.r.t. π

so we are actually finding $V^{\pi^{\varepsilon *}}$ & not V^{π^*}

Off policy MC is used to solve this

Off Policy Monte Carlo

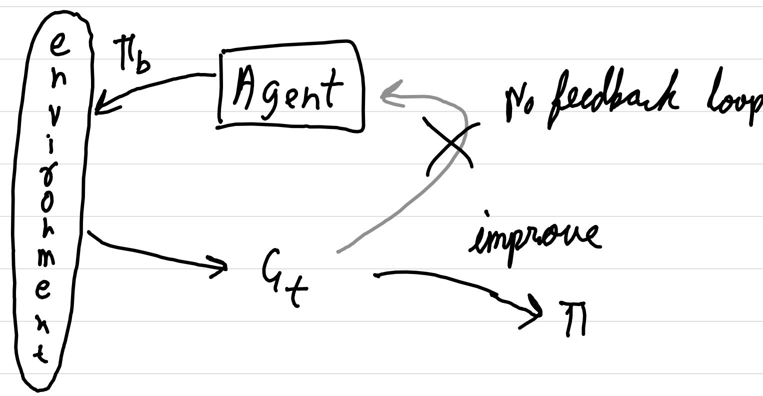
We use 2 seperate policies π & π_b

$\pi \rightarrow$ target policy

$\pi_b \rightarrow$ behavioural policy

Optimize π while generating behaviours from π_b

π_b can explore
π_b can even follow some huristic



But now how can we find V^{π}, Q^{π} & not V^{π_b} Q^{π_b} ?

for this, we need importance sampling

Importance Sampling

Importance sampling is used to evaluate properties of π when samples are generated by different distribution π_b

When we generate episodes from π_b & get return G_t, we need to normalize the return.

G_t : How much rewards are obtained when model follows π_b

G_t' : How much rewards would have had been obtained if model had followed π

We need G_t' & not G_t

Action that caused the reward

$$G_t' = \frac{\text{Likelyhood of } \pi \text{ taking action}}{\text{Likelyhood of } \pi_b \text{ taking action}} \times G_t$$

action taken by b at time t

state at t

$$= \rho_{0:T} \times G_t$$

where $\rho_{0:T} = \prod_{t=0}^{T-1} \frac{\pi(a_t \mid s_t)}{\pi_b(a_t \mid s_t)}$ (Importance sampling ratio)

$$\therefore \quad V(s) = \frac{\sum \rho_{t:T-1} \times G_t}{n}$$

t is states through time

no of times s is visited in all eps.

This simply is going through each episode where s is visited, then summing returns recieved after s was visited but weighting them by importance sampling ratio. finally taking the average } ordinary importance sampling

Here there is a variation (weighted importance sampling) that is preferred

$$V(s) = \frac{\sum \rho_{t:T-1} \times G_t}{\sum \rho_{t:T-1}}$$

The advantage is that we can learn from any historic data, given data G_t & π_b

How much rewards are obtained on following π_b

↓

How much rewards would have had been obtained on following π

↓

Update V

↓

Update π

Importance sampling is a variance reduction technique

Monte Carlo Control

In MC control, the goal is not only to evaluate a fixed policy but also to improve it wrt actions

Policy improvement is done greedly

$$\pi'(s) = \arg\max_a Q^\pi(s,a)$$

In cases where $P(s'|s,a)$ is known

$$\pi'(s) = \arg\max_a \sum_{s'} P(s'|s,a)\, V^\pi(s')$$

In MC predition, goodness of states is taken (state, value)

In MC control, goodness of state & actions are considered (state, action)

Predition	Control
Policy is supplied and goal is to check how well it performs	Control is where the policy is not fixed and goal is to find the optima policy

Prediction problems can be used for control when transition probabilities $P(s,a,s')$ are known

Useful in training games, navigation

Estimating Action value function

Similarly to estimate action value function $Q^{\pi}(s,a)$ Let $D(s,a)$ be set of all time steps t at which the state action pair (s,a) is visited.

Then the monte carlo estimate of $Q^{\pi}(s,a)$

$$Q^{\pi}(s,a) = \frac{1}{|D(s,a)|} \sum_{t \in D(s,a)} G_t$$

for Incremental Case

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha\left(G_t - Q(S_t, a_t)\right)$$

$$\pi(s) = \underset{a}{\operatorname{argmax}}\, Q(s,a)$$