In policy and value iteration, to obtain the optimal policy, the state value function is fed into the policy improvement equation (1) which is then looped for all states until the policy converges and can be extracted.

(1)

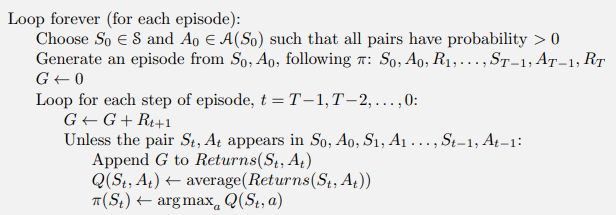
However, this equation uses the state transition probability , which is not known in model-free learning since not knowing about the environment means we cannot look ahead for future states.

Therefore, the policy improvement step is instead done by estimating action values, in particular the action value function which returns the expected value of the return (overall reward) for using action in a certain state , and so incorporates future states similar to the transition probability. The policy can then be extracted using equation (2) which shows that for each state the action that maximises the action value function is used as the policy.

(2)

In MC prediction, this step is used to refine the policy as shown in figure 1.0 where at the end of each episode, the new policy is updated with line .

Figure 1.0 Pseudo code that extracts and updates new policy every episode



However, this is not typically used in practice because this is a greedy policy and so can result in an effect called policy freezing where, if all actions are not understood or explored completely, it may continually execute an incorrect action. As a result, the policy will be stuck in a suboptimal result.

In policy and value iteration, to obtain the optimal policy, the state value function is fed into the policy improvement equation (1) which is then looped for all states until the policy converges and can be extracted.

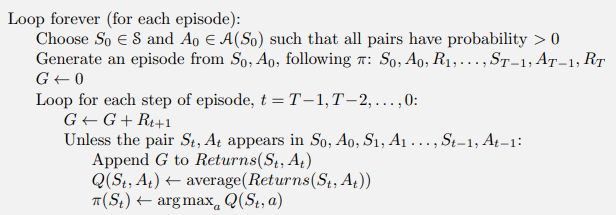
(1)

However, this equation uses the state transition probability , which requires knowledge of the environment and therefore is not known in model-free learning. Instead, the Monte Carlo approach uses an empirical approach to predict the optimal policy. This can be done by estimating action values, in particular the action value function which returns the expected value of the return (overall reward) for using action in a certain state , and also incorporates future states similar to the transition probability. The policy can then be extracted using equation (2) which shows that for each state the action that maximises the action value function is used as the policy.

(2)

In MC prediction, this step is used to refine the policy as shown in figure 1.0 where at the end of each episode, the new policy is updated with line .

Figure 1.0 Pseudo code that extracts and updates new policy every episode



By using the data gathered over many episodes, MC is able to accurately predict the result of actions and use this information to estimate the optimal policy. However, this is not typically used in practice because this is a greedy policy and so can result in an effect called policy freezing. Once the optimal actions have been determined by the last step in Figure 1.0, this is stored into policy which is then used to generate future episodes. This results in the inability to change the policy once the choice has been fixed and could result in the policy being “stuck” in a suboptimal position.