Epoch RL Learning Phase

DP Methods

Indian Institute of Technology Hyderabad

Aakash Kamuju

A. Bellman Equation

$$V(s) = \max_a \left(\sum_{s'} P^a_{ss'} \left(R^a_{ss'} + \gamma V(s') \right) \right) \tag{1}$$

This has an optimal substructure(solution to the give can be constructed from solutions of the subproblems) and optimal overlapping sub-problems(The subproblems appear many times while solving, and their solution can be reused).
These types of problems can generally be solved by DP methods.

B. Value Iteration

The value iteration is a dynamic programming method that efficiently solves the Bellman equation.
It uses a one-step look ahead. That is, it updates the value of the given state by using the value of its next state.
It uses the iterative step given below

$$V_{k+1}(s) = \max_a \left(\sum_{s' \in S} P^a_{ss'} \left(R^a_{ss'} + \gamma V_k(s') \right) \right) \tag{2}$$

Here, the convergence indicates that the value of the states has reached the optimal value.
We create a new set of values for each state rather than updating the old ones simultaneously.
We can find the policy at each iteration greedily by the value of its states.

C. Policy Iteration

The policy iteration is a dynamic programming method that efficiently solves the Bellman equation. This algorithm picks an action greedily from the updated value function and updates the policy for each iteration.
It uses the iterative step(policy evaluation) given below

$$V^{\pi_i}_{k+1}(s) \leftarrow \sum_a \pi(a|s) \left(\sum_{s' \in S} P^a_{ss'} \left(R^a_{ss'} + \gamma V^{\pi_i}_k(s') \right) \right) \tag{3}$$

and greedily updates the policy(policy improvement)

$$\pi_{i+1} = greedy(V^{\pi_i}) \tag{4}$$

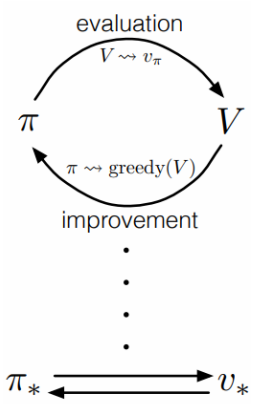
The main crux here is that even though it considers all the actions to update the value function, it greedily chooses the policy to solve the Bellman equation.
Here, the convergence indicates that the policy reached the optimal policy.

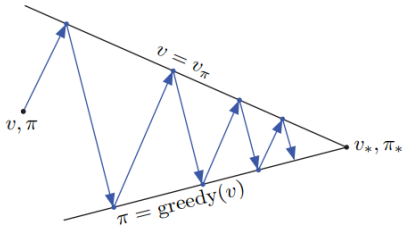
D. Generalized Policy Iteration

The *generalized policy iteration* is the idea of letting the policy evaluation and improvement interact, as shown

Making the policy greedy with respect to the value function typically makes the value function incorrect for the changed policy. Making the value function consistent with the policy typically causes that policy to no longer be greedy. In the long run, this makes the policy and value function optimal.

This gives a broad idea of what the policy iteration and value iteration are doing.





E. Advantages and Disadvantages of DP methods

The main advantage of the DP method is that it gives the optimal solution if it exists. It improves the efficiency of solving by avoiding redundant calculations.

The disadvantage of DP methods is that they require sweeps of the state set of the MDP. These methods can consume large amounts of memory due to the storage of immediate results. Asynchronous DP methods(These are in place iterative algorithms that are not organized in terms of sweeps that we do in the policy evaluation and improvement) or GPI(Generalized Policy Iteration) can be applied in such cases.