Dynamic Programming

Aayush Adlakha

August 31, 2023

1 Overview

Dynamic Programming (DP) is used to compute optimal policies given a *perfect model* of the environment as a Markov Decision Process (MDP).

2 Equations

$$v_*(s) = \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \tag{1}$$

$$= \max_a \sum_{s^{'},r} p(s^{'}, r|s, a)[r + \gamma v_*(s^{'})] \tag{2}$$

Similar equations can be written in terms of value function as well.

3 Policy Evaluation

- Used for computing value function for any policy π.
- If the environment's dynamics are completely known, then (2) is a system of $|S|$ equations in $|S|$ variables. simultaneous linear equations in $|S|$ unknowns.
- We use the set of equations (2) as assignments update rules iteratively.
- We also maintain the maximum change in a state value, when it is as low as required we break.

4 Policy Iteration

- Policy Iteration has 2 steps, namely, Policy Evaluation and Improvement.
- We initialize the policy randomly and evaluate it.

- For Policy Improvement, we check if the action prescribed by the policy always has the maximum value of all the possible actions. If yes, we have found a stable policy. Hence, we break. Otherwise, we construct a new policy based on the maximum action value possible at every step.

This process is repeated until we find a stable policy.

5 Value Iteration

- Value Iteration is much simpler to write as it combines the 2 steps of policy iteration into one single loop.
- For each state, we consider all the possible actions and assign the state the value based on the maximum action path.
- We break when this update is small enough.

Value iteration effectively combines, in each of its sweeps, one sweep of policy evaluation and one sweep of policy improvement.

6 Generalized Policy Iteration

Generalized Policy Iteration is a term we use to describe the process of evaluating a policy and then improving it. Almost all RL Algorithms can be described as GPIs.

If both the evaluation process and the improvement process stabilize, that it no longer produce changes, then the value function and policy must be optimal. The value function stabilizes only when it is consistent with the current policy, and the policy stabilizes only when it is greedy with respect to the current value function.

The evaluation and improvement processes in GPI can be viewed as both competing and cooperating. They compete in the sense that they pull in opposing directions. Making the policy greedy with respect to the value function typically makes the value function incorrect for the changed policy, and making the value function consistent with the policy typically causes that policy to no longer be greedy. In the long run, however, these two processes interact to find a single joint solution: the optimal value function and an optimal policy.

7 Conclusion

- DP methods are often simple,
- Converge quickly to correct answers for smaller problems.
- DP may not be practical for very large problems.

- DP required complete knowledge of the dynamics of the environment, which is often unavailable.
- DP suffers from the *Curse of Dimensionality* that is, the number of states often grows exponentially with the number of state variables.