Dynamic Programming in Reinforcement Learning

September 25, 2023

1 Introduction

Dynamic Programming (DP) is a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov Decision Process (MDP) [Sutton and Barto(2018)]. It involves breaking down a problem into simpler subproblems and solving each subproblem only once, storing their solutions.

2 Principles of Dynamic Programming

Dynamic Programming in the context of Reinforcement Learning typically involves two main algorithms: Value Iteration and Policy Iteration. These algorithms are used to compute the value functions, which are essential for deriving optimal policies.

2.1 Value Iteration

Value Iteration is an algorithm that computes the optimal state-value function by iteratively improving the estimate of it [Bellman(1957)]. The update equation for Value Iteration is given by the Bellman optimality equation:

$$V^{k+1}(s) = \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^k(s') \right)$$

where $V^{k+1}(s)$ and $V^k(s)$ are the values of state s at the $k+1$ and k iteration, respectively, $R(s,a)$ is the immediate reward for taking action a in state s, $P(s'|s,a)$ is the transition probability from state s to state s' given action a, and γ is the discount factor.

2.2 Policy Iteration

Policy Iteration, on the other hand, consists of two steps: policy evaluation and policy improvement. The idea is to evaluate the current policy, and then improve it, until an optimal policy is found [Howard(1960)].

2.2.1 Policy Evaluation

Given a policy π, its value function is evaluated using the Bellman expectation equation:

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi}(s') \right)$$

2.2.2 Policy Improvement

Once the policy has been evaluated, it can be improved by acting greedily with respect to the value function:

$$\pi'(s) = \arg\max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi}(s') \right)$$

3 Conclusion

Dynamic Programming offers a systematic approach to solving Reinforcement Learning problems when a perfect model of the environment is available. While it has computational constraints for large state spaces, it lays the foundational principles for many approximation methods in RL.

References

[Bellman(1957)] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Howard(1960)] Ronald A Howard. Dynamic programming and markov processes. 1960.

[Sutton and Barto(2018)] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press Cambridge, 2018.