Reinforcement Learning

Generalized Policy Iteration

Dr. Alireza Aghamohammadi

Policy Iteration

- **Policy Evaluation:**
  - Evaluate a given policy by estimating its state-value function.
  - Use an iterative algorithm to approximate the state-value function of the policy, starting with an initial estimate, typically:

$$V_\pi^{(0)}(s) = 0 \quad \forall s$$

  - The iterative update is given by:

$$V_\pi^{(k+1)}(s) = \sum_a \pi(a \mid s) \sum_{s',r} P(s', r \mid s, a) \left[r + \gamma V_\pi^{(k)}(s') \right]$$

  - This process converges as $k \to \infty$, providing $V_\pi(s)$ for all states.
- **Policy Improvement:**
  - Use the evaluated state-value function to derive a better policy.
  - For each state, determine the action that maximizes the expected value:

$$\pi'(a \mid s) = \operatorname*{argmax}_a \sum_{s',r} P(s', r \mid s, a) \left[r + \gamma V_\pi(s') \right]$$

  - This results in a new policy π' that is at least as good as the previous one.
- **Policy Iteration:**
  - Start with an initial policy (e.g., random policy).
  - Alternate between:
    1. *Policy Evaluation:* Compute V_π for the current policy π.
    2. *Policy Improvement:* Generate a new policy π' based on V_π.
  - Repeat until the policy stabilizes (i.e., $\pi' = \pi$).

Value Iteration

- **Value Iteration (VI):**
  - Value iteration does not fully compute the state-value function for a policy before improving it.
  - Instead, it updates the value function for all states once and uses this intermediate result to improve the policy in the same step.
- The value iteration algorithm can be expressed as:

$$V^{(k+1)}(s) = \max_a \sum_{s',r} P(s', r \mid s, a) \left[r + \gamma V^{(k)}(s') \right]$$

- This process is repeated iteratively until $V(s)$ converges to the optimal value function $V^*(s)$.