MCPG

Monte Carlo Policy Gradient (MCPG) is a reinforcement learning algorithm that belongs to the family of policy gradient methods. It directly optimizes the policy function to maximize the expected cumulative reward.

Algorithm Steps

1. Initialization:

- Initialize the policy function `π(a|s, θ)`.
- Set the number of episodes to be collected `N`.

2. Collect Episodes:

- For `i` from 1 to `N`:
  - Initialize the current state `s`.
  - While the episode is not terminal:
    - Sample an action `a` from the policy `π(a|s, θ)`.
    - Take action `a` and observe the next state `s'` and reward `r`.
    - Store the transition `(s, a, r, s')` in a buffer.
    - Update the current state `s` to `s'`.

3. Calculate Returns:

- For each episode in the buffer:
  - Calculate the return `G(τ)` for that episode, where `τ` is the episode's trajectory.
  - The return is the sum of discounted rewards from that episode.

4. Update Policy:

- For each transition `(s, a, r, s')` in the buffer:
  - Calculate the policy gradient:

$$\nabla J(\theta) = \nabla log\pi(a|s,\theta)G(\tau)$$

  - Update the policy parameters `θ` using gradient ascent:

    where
    `α` is the learning rate.

$$\theta \leftarrow \theta + \alpha \nabla J(\theta)$$

5. Repeat:

- Repeat steps 2-4 until convergence or a desired number of iterations.

Advantages of MCPG

- **Simple to Implement:** MCPG is relatively straightforward to implement compared to other reinforcement learning algorithms.
- **Unbiased Estimates:** MCPG provides unbiased estimates of the policy gradient, which can lead to more stable learning.
- **Off-Policy Learning:** MCPG can be used for off-policy learning, where the policy used to collect data is different from the policy being evaluated.

Disadvantages of MCPG

- **High Variance:** MCPG can have high variance, especially when the returns are highly variable.
- **Inefficient Sampling:** MCPG can be inefficient in terms of sample complexity, as it requires many episodes to obtain accurate estimates of the policy gradient.