E0123030 RL EXPT 4 OUTPUT

import numpy as np

import matplotlib.pyplot as plt

class MultiArmedBanditEnv:

def \_\_init\_\_(self, n\_arms=10):

self.n\_arms = n\_arms

self.true = np.random.normal(0, 1, n\_arms)

self.best\_action = np.argmax(self.true)

self.reset()

def reset(self):

return 0

def step(self, action):

reward = np.random.normal(self.true[action], 1)

return 0, reward, False, {}

def run\_epsilon\_greedy(env, episodes=500, epsilon=0.1):

n = env.n\_arms

Q = np.zeros(n)

N = np.zeros(n)

rewards = []

for \_ in range(episodes):

env.reset()

if np.random.rand() < epsilon:

action = np.random.choice(n)

else:

action = np.argmax(Q)

\_, reward, \_, \_ = env.step(action)

N[action] += 1

Q[action] += (reward - Q[action]) / N[action]

rewards.append(reward)

return Q, rewards

def run\_softmax(env, episodes=500, temperature=0.5):

n = env.n\_arms

Q = np.zeros(n)

N = np.zeros(n)

rewards = []

for \_ in range(episodes):

env.reset()

logits = Q / temperature

logits -= np.max(logits)

exp\_logits = np.exp(logits)

probabilities = exp\_logits / np.sum(exp\_logits)

action = np.random.choice(n, p=probabilities)

\_, reward, \_, \_ = env.step(action)

N[action] += 1

Q[action] += (reward - Q[action]) / N[action]

rewards.append(reward)

return Q, rewards

env = MultiArmedBanditEnv()

Q\_eps, rewards\_eps = run\_epsilon\_greedy(env, episodes=500, epsilon=0.1)

Q\_soft, rewards\_soft = run\_softmax(env, episodes=500, temperature=0.5)

print("Epsilon-Greedy Q-values:", Q\_eps)

print("Total reward (Epsilon-Greedy):", sum(rewards\_eps))

print()

print("Softmax Q-values:", Q\_soft)

print("Total reward (Softmax):", sum(rewards\_soft))

avg\_rewards\_eps = np.cumsum(rewards\_eps) / (np.arange(len(rewards\_eps)) + 1)

avg\_rewards\_soft = np.cumsum(rewards\_soft) / (np.arange(len(rewards\_soft)) + 1)

plt.plot(avg\_rewards\_eps, label="Epsilon-Greedy")

plt.plot(avg\_rewards\_soft, label="Softmax")

plt.xlabel("Episode")

plt.ylabel("Average Reward")

plt.title("Average Reward Comparison")

plt.legend()

plt.grid(True)

plt.show()

