

```
In [1]: 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 None

    def step(self, action):
        reward = np.random.normal(self.true[action], 1)
        return None, reward, False, False, {}

def run_epsilon_greedy(env, episodes=500, epsilon=0.1):
    k = env.n_arms
    Q = np.zeros(k)
    N = np.zeros(k)
    rewards = []

    for _ in range(episodes):
        env.reset()

        action = np.random.choice(k) if np.random.rand() < epsilon else 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):
    k = env.n_arms
    Q = np.zeros(k)
    N = np.zeros(k)
    rewards = []

    for _ in range(episodes):
        env.reset()

        exp_Q = np.exp((Q - np.max(Q)) / temperature)
        probabilities = exp_Q / np.sum(exp_Q)
        action = np.random.choice(k, p=probabilities)

        _, reward, _, _, _ = env.step(action)
        N[action] += 1
        Q[action] += (reward - Q[action]) / N[action]
        rewards.append(reward)

    return Q, rewards

env = MultiArmedBanditEnv(n_arms=10)

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):", round(sum(rewards_eps), 2))
print("Softmax Q-values:", Q_soft)
print("Total reward (Softmax):", round(sum(rewards_soft), 2))

avg_reward_eps = np.cumsum(rewards_eps) / (np.arange(len(rewards_eps)) + 1)
avg_reward_soft = np.cumsum(rewards_soft) / (np.arange(len(rewards_soft)) + 1)

plt.plot(avg_reward_eps, label='Epsilon-Greedy')
plt.plot(avg_reward_soft, label='Softmax')

plt.xlabel('Episode')
plt.ylabel('Average Reward')
plt.title('Average Reward Comparison')
plt.legend()
plt.grid(True)
plt.show()

Epsilon-Greedy Q-values: [ 0.09635669 -0.9491209  -0.04241926 -0.10452734  0.62579554  1.05757926
 -0.50404691  1.63039388 -0.85900464 -0.86837205]
Total reward (Epsilon-Greedy): 684.43
Softmax Q-values: [ 0.14276542 -1.10507395  0.30840896 -0.8725776   0.45392201  1.04016079
 -0.81556474  1.67883924 -1.61065682 -0.41890186]
Total reward (Softmax): 591.67
```



In []: