Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №4
по дисциплине
«Методы машинного обучения»
на тему

«policy_iteration»

Выполнил:
студент группы ИУ5и-24М
Аунг Пьио Нанда

Москва — 2024 г.

1. Цель лабораторной работы

Ознакомление с базовыми методами обучения с подкреплением.

2. Задание

На основе рассмотренного на лекции примера реализуйте алгоритм Policy Iteration для любой среды обучения с подкреплением (кроме рассмотренной на лекции среды Toy Text / Frozen Lake) из библиотеки Gym (или аналогичной библиотеки).

3. Ход выполнения работы

```
[3] import gym
    import numpy as np
```

```
[4] # Создаем среду
    env = gym.make('Taxi-v3')
```

```
[5] # Инициализация параметров
    n_states = env.observation_space.n
    n_actions = env.action_space.n
    gamma = 0.99  # дисконтирующий фактор
    theta = 1e-6  # порог для оценки политики
```

```
[6] # Инициализация случайной политики
    policy = np.random.choice(n_actions, size=n_states)
```

```
[7] # Функция полезности
    V = np.zeros(n_states)
```

```
[8] # Оценка политики
    def policy_evaluation(policy, V, theta, gamma):
        while True:
            delta = 0
            for state in range(n_states):
                v = V[state]
                action = policy[state]
                new_v = sum([prob * (reward + gamma * V[next_state])
                             for prob, next_state, reward, done in env.P[state][action]])
                V[state] = new_v
                delta = max(delta, abs(v - new_v))
            if delta < theta:
                break
        return V
```

```
    # Улучшение политики
    def policy_improvement(V, gamma):
        policy_stable = True
        for state in range(n_states):
            old_action = policy[state]
            action_values = np.zeros(n_actions)
            for action in range(n_actions):
                action_values[action] = sum([prob * (reward + gamma * V[next_state])
                                             for prob, next_state, reward, done in env.P[state][action]])
            new_action = np.argmax(action_values)
            policy[state] = new_action
            if old_action != new_action:
                policy_stable = False
        return policy, policy_stable
```

```
[10] # Алгоритм Policy Iteration
     def policy_iteration(env, gamma, theta):
         policy = np.random.choice(n_actions, size=n_states)
         V = np.zeros(n_states)
         while True:
             V = policy_evaluation(policy, V, theta, gamma)
             policy, policy_stable = policy_improvement(V, gamma)
             if policy_stable:
                 break
         return policy, V
```

```
[11] # Выполнение алгоритма
     optimal_policy, optimal_value_function = policy_iteration(env, gamma, theta)
```

```
print("Optimal Policy:", optimal_policy)
```

```
Optimal Policy: [4 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 3
 0 0 0 0 0 0 0 2 2 2 2 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 2 2 2 2 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 4 4 4 4 0 0 0 0 0 0 0 0 0 0 5 0 0 1 1 1 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 1 1
 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 1 2 1 1 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 1
 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 2 2 2 2 0 0 0 0 2 2 2 2 1 2 0 2 1 1
 1 1 2 2 2 2 3 3 3 3 2 2 2 2 1 2 3 2 3 3 3 3 1 2 1 1 3 3 3 3 2 2 2 2 3 2 3
 2 3 3 3 3 1 2 1 1 3 3 3 3 0 0 0 0 3 2 3 0 3 3 3 3 1 1 1 1 3 3 3 3 0 0 0 0
 3 1 3 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 2 1 1 1 1 1 1 1
 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
 1 1 0 0 0 0 1 2 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1
 1 4 4 4 4 1 1 1 1 1 1 5 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 4 4 4 4 1 2 1 5 1
 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 1 1 1 3]
```

```
print("Optimal Value Function:", optimal_value_function)
```

```
Optimal Value Function: [944.72361809 864.01315934 903.55733909 873.75068256 789.53804327
 864.01315934 789.53804327 816.7669381  864.01317574 826.0271938
 903.55733909 835.3810204  807.59926872 826.0271938  807.59926872
 873.75068256 955.27638191 873.75066633 913.69428191 883.58654804
 934.27638191 854.37302775 893.5217657  864.01317574 798.52327603
 873.75066633 798.52327603 826.0272102  854.37304398 816.76692187
 893.5217657  826.0272102  816.7669381  835.38100417 816.7669381
 883.58654804 944.72361809 883.58653197 903.55733909 893.5217657
 883.58654804 807.59925265 844.82931354 816.7669381  844.82931354
 923.93360202 844.82931354 873.75068256 844.82931354 807.59925265
 883.58654804 816.7669381  826.0272102  844.82929747 826.0272102
 893.5217657  893.5217657  934.276366   893.5217657  903.55733909
 873.75068256 798.52326012 835.3810204  807.59926872 854.37304398
 934.276366   854.37304398 883.58654804 835.3810204  798.52326012
 873.75068256 807.59926872 835.3810204  854.37302807 835.3810204
 903.55733909 883.58654804 944.72360234 883.58654804 913.69428191
 864.01317574 789.53802752 826.0272102  798.52327603 864.01317574
 944.72360234 864.01317574 893.5217657  826.0272102  789.53802752
 864.01317574 798.52327603 826.0272102  844.82929779 826.0272102
 893.5217657  873.75068256 955.27636632 873.75068256 903.55733909
 934.27638191 854.37302775 893.5217657  864.01317574 798.52327603
 873.75066633 798.52327603 826.0272102  873.75068256 835.38100417
 913.69428191 844.82931354 816.7669381  835.38100417 816.7669381
 883.58654804 944.72361809 883.58653197 923.93361809 893.5217657
 923.93361809 844.82929747 883.58654804 854.37304398 807.59926872
 883.58653197 807.59926872 835.3810204  864.01317574 826.02719413
 903.55733909 835.3810204  826.0272102  844.82929747 826.0272102
 893.5217657  934.27638191 893.52174979 913.69428191 903.55733909
 893.5217657  816.76692219 854.37304398 826.0272102  835.3810204
 913.694266   835.3810204  864.01317574 854.37304398 816.76692219
 893.5217657  826.0272102  835.3810204  854.37302807 835.3810204
 903.55733909 903.55733909 923.93360234 903.55733909 913.69428191
 883.58654804 807.59925297 844.82931354 816.7669381  844.82931354
 923.93360234 844.82931354 873.75068256 844.82931354 807.59925297
 883.58654804 816.7669381  844.82931354 864.01315999 844.82931354
```

```
[14] # Применение оптимальной политики в среде
     def run_policy(env, policy, episodes=10):
         for episode in range(episodes):
             state = env.reset()
             done = False
             total_reward = 0
             while not done:
                 env.render()
                 action = policy[state]
                 state, reward, done, _ = env.step(action)
                 total_reward += reward
             print(f"Episode {episode + 1}: Total Reward: {total_reward}")
         env.close()
```

```
[15] # Запуск среды с оптимальной политикой
     run_policy(env, optimal_policy)
```

```
Episode 1: Total Reward: 4
Episode 2: Total Reward: 4
Episode 3: Total Reward: 9
Episode 4: Total Reward: 7
Episode 5: Total Reward: 4
Episode 6: Total Reward: 4
Episode 7: Total Reward: 9
Episode 8: Total Reward: 10
Episode 9: Total Reward: 11
Episode 10: Total Reward: 4
```

1) Инициализация:

Инициализация случайной политики и функции полезности.

2) Оценка политики (Policy Evaluation):

Обновление функции полезности для каждого состояния до сходимости.

3) Улучшение политики (Policy Improvement):

Обновление политики на основе текущих значений функции полезности.

4) Сходимость:

Повторение шагов оценки и улучшения политики до тех пор, пока политика не станет стабильной (не изменится).

5) Запуск среды:

Применение оптимальной политики в среде и выполнение нескольких эпизодов.

Это демонстрирует использование алгоритма Policy Iteration для среды "Taxi-v3" из библиотеки Gym, где доступна информация о вероятностях переходов и вознаграждениях для каждого состояния и действия.

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс] // GitHub. — 2024. — Режим доступа: https://github.com/ugapanyuk/courses_current/wiki/LAB_MMO___policy_iteration.