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

Лабораторная работа №4
по дисциплине
«Методы машинного обучения»

Выполнил:
студент группы ИУ5-22М
Лю Ченхао

Москва — 2024 г.

Цель лабораторной работы:

ознакомление с базовыми методами обучения с подкреплением.

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).

[1]:
```
import numpy as np
import gym
```

[2]:
```
# Инициализируем среду Gym
env = gym.make('FrozenLake8x8-v1') # или другую среду
```

[3]:
```
# Определяем функцию оценки стратегии
def policy_evaluation(policy, env, discount_factor=1.0, theta=0.00001):
    V = np.zeros(env.observation_space.n) # Вектор оценок состояний
    while True:
        delta = 0 # Дельта для проверки сходимости
        for s in range(env.observation_space.n): # Перебор всех состояний
            v = 0
            # Пересчет оценки состояния
            for a, action_prob in enumerate(policy[s]):
                for prob, next_state, reward, done in env.P[s][a]:
                    v += action_prob * prob * (reward + discount_factor * V[next_state])
            delta = max(delta, np.fabs(v - V[s]))
            V[s] = v
        # Прерываем цикл, если изменение функции ценности ниже порога
        if delta < theta:
            break
    return V
```

[4]:
```
# Определяем функцию улучшения стратегии
def policy_improvement(policy, env, V, discount_factor=1.0):
    policy_stable = True # Флаг стабильности стратегии
    for s in range(env.observation_space.n): # Перебор всех состояний
        # Выбор лучшего действия из текущей стратегии
        chosen_a = np.argmax(policy[s])
        action_values = np.zeros(env.action_space.n)
        # Вычисляем ценность каждого действия
        for a in range(env.action_space.n):
            for prob, next_state, reward, done in env.P[s][a]:
                action_values[a] += prob * (reward + discount_factor * V[next_state])
        best_a = np.argmax(action_values)

        # Обновляем стратегию жадным способом
        if chosen_a != best_a:
            policy_stable = False
        policy[s] = np.eye(env.action_space.n)[best_a]

    return policy, policy_stable
```

[5]:
```
# Определяем функцию итерации стратегии
def policy_iteration(env, discount_factor=1.0):
    policy = np.ones([env.observation_space.n, env.action_space.n]) / env.action_space.n # Равномерное распределение
    while True:
        V = policy_evaluation(policy, env, discount_factor) # Оценка стратегии
        policy, policy_stable = policy_improvement(policy, env, V, discount_factor) # Улучшение стратегии
        print(f"Оцененная функция ценности: {V}")
        print(f"Улучшенная стратегия: {policy}")
        # Если стратегия стабильна, завершаем процесс
        if policy_stable:
            return policy, V
```

[5]:
```python
# Определяем функцию итерации стратегии
def policy_iteration(env, discount_factor=1.0):
    policy = np.ones([env.observation_space.n, env.action_space.n]) / env.action_space.n # Равномерное распределение
    while True:
        V = policy_evaluation(policy, env, discount_factor) # Оценка стратегии
        policy, policy_stable = policy_improvement(policy, env, V, discount_factor) # Улучшение стратегии
        print(f"Оцененная функция ценности: {V}")
        print(f"Улучшенная стратегия: {policy}")
        # Если стратегия стабильна, завершаем процесс
        if policy_stable:
            return policy, V
```

[6]:
```python
policy, V = policy_iteration(env)
```

```
Оцененная функция ценности: [1.77291852e-03 2.05231760e-03 2.71200668e-03 4.02682148e-03
 6.45060006e-03 9.69730771e-03 1.33323211e-02 1.58487758e-02
 1.52729353e-03 1.69427642e-03 2.07670686e-03 2.93669579e-03
 5.64863740e-03 9.33260477e-03 1.44768170e-02 1.83922528e-02
 1.13470573e-03 1.13113238e-03 9.71534634e-04 0.00000000e+00
 3.88354185e-03 7.51853274e-03 1.68629320e-02 2.48647709e-02
 7.60085659e-04 7.30250376e-04 6.80298040e-04 7.62713639e-04
 2.37149567e-03 0.00000000e+00 2.05998575e-02 3.93483800e-02
 4.24537163e-04 3.53058351e-04 2.58339098e-04 0.00000000e+00
 4.84014067e-03 1.15878559e-02 2.61929417e-02 7.25859868e-02
 1.65174830e-04 0.00000000e+00 0.00000000e+00 1.44772835e-03
 5.40187309e-03 1.53198027e-02 0.00000000e+00 1.52219331e-01
 7.23056825e-05 0.00000000e+00 1.09152955e-04 3.89220327e-04
 0.00000000e+00 4.42895328e-02 0.00000000e+00 3.84073024e-01
 5.24361145e-05 3.32439149e-05 4.74511163e-05 0.00000000e+00
 5.39461066e-02 1.61838352e-01 3.87279448e-01 0.00000000e+00]
Улучшенная стратегия: [[0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
Оцененная функция ценности: [0.92160185 0.92162819 0.92167048 0.92495477 0.93669695 0.97707104
 0.99508152 0.99968545 0.92095657 0.92032848 0.91840835 0.91323302
 0.89634206 0.95907989 0.99049679 0.99970462 0.9025283  0.88411691
 0.82950132 0.         0.79326086 0.9096837  0.97671585 0.99973541
 0.85552657 0.80854073 0.68598482 0.41991578 0.5737664  0.
 0.93992554 0.99977633 0.7658227  0.67613281 0.45403921 0.
 0.50812548 0.59013103 0.84329282 0.9998255  0.76576504 0.
 0.         0.15434729 0.36048377 0.41898032 0.         0.99988077
 0.76572564 0.         0.15333566 0.10256098 0.         0.30632677
 0.         0.99993976 0.76570573 0.56157261 0.35745175 0.
 0.25       0.5        0.75       0.        ]
Улучшенная стратегия: [[0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]
Оцененная функция ценности: [0.98065675 0.98067557 0.98070579 0.98074595 0.98079422 0.98084847
 0.99545956 0.9998873  0.9804989  0.98035323 0.97989048 0.9786169
 0.97521776 0.96624452 0.99103871 0.99989417 0.96344763 0.94640838
 0.89542964 0.         0.83654378 0.92685096 0.97776657 0.9999052
 0.91934131 0.87524658 0.75999488 0.45585329 0.60756722 0.
 0.94235949 0.99991986 0.91926729 0.76794076 0.50931188 0.
 0.53030613 0.60588983 0.84939517 0.99993748 0.91920954 0.
 0.         0.16411542 0.37746373 0.43797057 0.         0.99995728
 0.91917009 0.         0.18053785 0.11488442 0.         0.33055856
 0.         0.99997842 0.91915016 0.67293633 0.42673472 0.
 0.27685248 0.55370538 0.77685262 0.        ]
Улучшенная стратегия: [[0. 0. 0. 1.]
```

```
Оцененная функция ценности: [0.99539991 0.99541278 0.99543346 0.99546096 0.99549405 0.99553128
 0.99557108 0.99995793 0.99538539 0.99537918 0.99534298 0.99521937
 0.99485657 0.99385794 0.9911868  0.9999605  0.97948584 0.96359473
 0.91592317 0.         0.85181105 0.94123288 0.97803041 0.99996461
 0.97939849 0.91765257 0.78883595 0.46939369 0.61934673 0.
 0.94294123 0.99997009 0.97932477 0.80996981 0.53293525 0.
 0.53683648 0.60955595 0.85082445 0.99997666 0.97926731 0.
 0.         0.16698067 0.38160824 0.44100814 0.         0.99998405
 0.97922807 0.         0.19102516 0.11933528 0.         0.33186047
 0.         0.99999194 0.97920826 0.71647092 0.4537457  0.
 0.27728659 0.55457345 0.77728668 0.        ]
Улучшенная стратегия: [[0. 0. 0. 1.]
 [1. 0. 0. 0.]]
Оцененная функция ценности: [0.99984132 0.99985021 0.99986454 0.99988366 0.99990679 0.99993298
 0.99996122 0.99999041 0.99984429 0.99985301 0.99986707 0.99988585
 0.99990856 0.99993431 0.99996211 0.99999099 0.99974537 0.97796872
 0.92623343 0.         0.85645018 0.9459487  0.98146201 0.99999193
 0.99965759 0.93431321 0.80086864 0.47478764 0.62349565 0.
 0.94443243 0.99999318 0.99958391 0.8253188  0.54206248 0.
 0.53924995 0.61110068 0.85184256 0.99999468 0.99952672 0.
 0.         0.16800404 0.38315461 0.44221027 0.         0.99999636
 0.99948775 0.         0.19457246 0.12085883 0.         0.33237571
 0.         0.99999816 0.99946809 0.73116003 0.46286394 0.
 0.2774584  0.55491699 0.77745847 0.        ]
Улучшенная стратегия: [[0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]]
Оцененная функция ценности: [0.99988922 0.99989544 0.99990248 0.99990975 0.99991688 0.99992359
 0.99992956 0.99993392 0.99988976 0.99989477 0.99990115 0.99990812
 0.99991522 0.99992229 0.9999295  0.99993794 0.99979189 0.97801257
 0.92627121 0.         0.85653934 0.94615668 0.98200981 0.99994441
 0.99970436 0.93435617 0.80090374 0.47481714 0.62354916 0.
 0.94462402 0.99995301 0.99963045 0.82535682 0.54208685 0.
 0.53929206 0.61114679 0.85191113 0.99996334 0.99957284 0.
 0.         0.16801534 0.38318183 0.44223873 0.         0.99997495
 0.9995335  0.         0.19458181 0.12086572 0.         0.33238785
 0.         0.99998734 0.99951362 0.73119335 0.46288523 0.
 0.27746237 0.55492502 0.77746246 0.        ]
Улучшенная стратегия: [[0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
```

[7]:
```
print("\nОптимальная стратегия:")
print(policy)
print("\nОптимальная функция ценности:")
print(V)
```

```
Оптимальная стратегия:
[[0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
```

```
Оптимальная функция ценности:
[0.99988922 0.99989544 0.99990248 0.99990975 0.99991688 0.99992359
 0.99992956 0.99993392 0.99988976 0.99989477 0.99990115 0.99990812
 0.99991522 0.99992229 0.9999295  0.99993794 0.99979189 0.97801257
 0.92627121 0.         0.85653934 0.94615668 0.98200981 0.99994441
 0.99970436 0.93435617 0.80090374 0.47481714 0.62354916 0.
 0.94462402 0.99995301 0.99963045 0.82535682 0.54208685 0.
 0.53929206 0.61114679 0.85191113 0.99996334 0.99957284 0.
 0.         0.16801534 0.38318183 0.44223873 0.         0.99997495
 0.9995335  0.         0.19458181 0.12086572 0.         0.33238785
 0.         0.99998734 0.99951362 0.73119335 0.46288523 0.
 0.27746237 0.55492502 0.77746246 0.        ]
```

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Подготовка обучающей и тестовой выборки,кросс-валидация и подбор гиперпараметров на примере метода ближайших соседей»[Электронный ресурс] // GitHub. Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_KNN

[2] Team The IPython Development. IPython 7.3.0 Documentation [Electronic resource] //
Read the Docs. — Access mode: https://ipython.readthedocs.io/en/
stable/

[3] Waskom M. seaborn 0.9.0 documentation [Electronic resource] // PyData. —

Access mode: https://seaborn.pydata.org/

[4] pandas 0.24.1 documentation [Electronic resource] // PyData. Access mode: http://pandas.pydata.org/pandas-docs/stable/

[5] dronio. Solar Radiation Prediction [Electronic resource] // Kaggle.— Access mode: https://www.kaggle.com/dronio/SolarEnergy (online; accessed:

[6] Chrétien M. Convert datetime.time to seconds [Electronic resource] // Stack Overflow.Access mode: https://stackoverflow.com/a/44823381

[7] scikit-learn 0.20.3 documentation [Electronic resource].— Access mode: https: //scikit-learn.org/