Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика, искусственный интеллект и системы управления»

Кафедра «Системы обработки информации и управления»



**Рубежный контроль № 4**

**по дисциплине «Методы машинного обучения»**

Реализация алгоритма Policy Iteration

ИСПОЛНИТЕЛЬ:

студентка ИУ5-23М

Морозевич М.А.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

\_\_\_ "\_\_\_\_\_\_\_\_" 2024 г.

Москва, 2024

# **Задание**

В рамках работы предполагается ознакомление с базовыми методами обучения с подкреплением.

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).

# **Описание среды и алгоритма**

Алгоритм Policy Iteration является методом решения задачи обучения с подкреплением в контексте марковских процессов принятия решений.

Основные шаги алгоритма Policy Iteration:

1. Оценка текущей стратегии;
2. Улучшение стратегии: путем выбора действий, которые могут привести к большему накопленному вознаграждению;
3. Проверка сходимости: Если стратегия сходится, то алгоритм завершается, иначе возврат к шагу 1.

Особенности алгоритма Policy Iteration:

* Гарантирует сходимость к оптимальной стратегии в конечном итоге, даже если начальная стратегия была произвольной.
* Эффективен для задач с небольшим количеством состояний и действий.
* Итерации позволяет последовательно улучшать стратегию.

В качестве среды была выбрана среда Taxi из библиотеки Gym. На карте есть четыре определенных места, обозначенных буквами R(ed), G(reen), Y(ellow) и B(lue). В рамках эпизода:

* Такси выезжает из случайного квадрата, а пассажир оказывается в случайном месте.
* Такси подъезжает к месту нахождения пассажира, забирает его, отвозит в пункт назначения (другое из четырех указанных мест), а затем высаживает пассажира.
* Как только пассажир высажен, эпизод заканчивается

Для такси определено 6 действий:

1. двигаться на юг
2. двигаться на север
3. двигаться на восток
4. двигаться на запад
5. забрать пассажира
6. высадить пассажира

Каждое пространство состояний представлено кортежем: (taxi\_town, taxi\_col, passenger\_location, destination). При этом наблюдение - это целое число, которое кодирует соответствующее состояние.

|  |  |
| --- | --- |
| **Расположение пассажиров** | **Места назначения** |
| 0: R(ed) | 0: R(ed) |
| 1: G(reen) | 1: G(reen) |
| 2: Y(ellow) | 2: Y(ellow) |
| 3: B(lue) | 3: B(lue) |
| 4: в такси | – |

Награды:

* -1 за каждый шаг, если не предусмотрено иное вознаграждение
* +20 за доставку пассажира.
* -10 за незаконное выполнение действий “погрузка” и “высадка”.

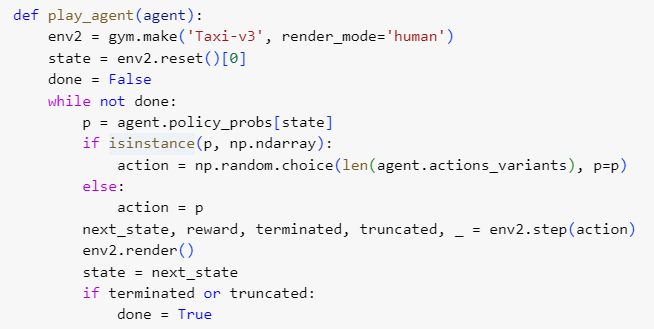
# **Реализация программы**

Определим класс PolicyIterationAgent, который будет эмулировать работу агента, с методами:

* print\_policy – выводит матрицы стратегии;
* policy\_evaluation – оценивает стратегии;
* policy\_improvement – улучшает стратегии;
* policy\_iteration – реализация итераций алгоритма.



Добавим функцию для рендера игры.



Запустим программу и выведем результаты обучения.

