OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Обучение с подкреплением.

Он всегда был не прочь подкрепиться.
Кроме того, он было поэт.

Артур Кадурин
Преподаватель

План на сегодня

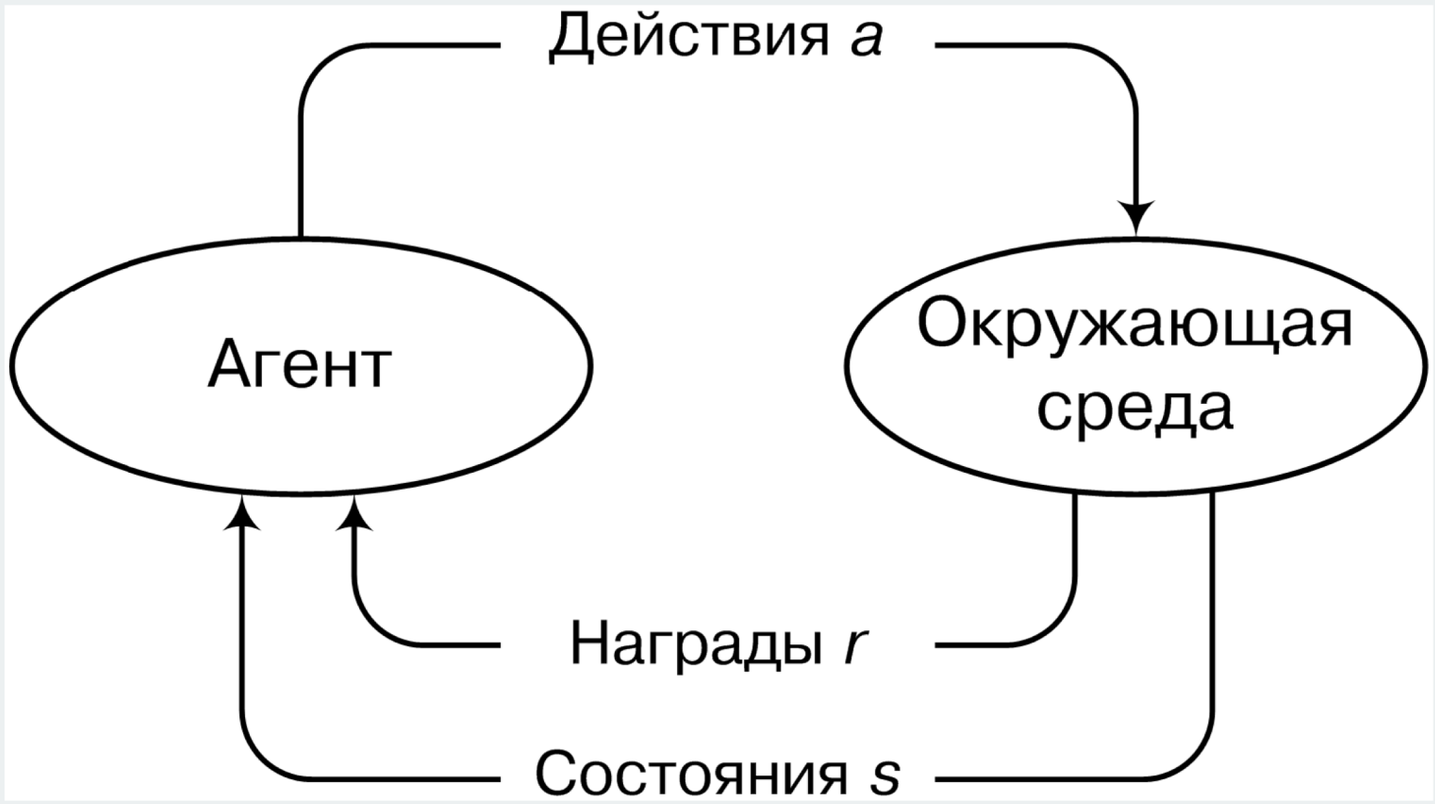
1. **Обучение с подкреплением**
2. Бандиты
3. Стратегии

Обучение с подкреплением

Обучение с учителем: Для заранее заданного набора объектов известны «правильные» ответы. Нужно научиться предсказывать ответы для новых объектов.
Обучение с подкреплением: Можно совершить действие и узнать ответ. Нужно научиться совершать действия приводящие к «лучшим» ответам.

Обучение с подкреплением

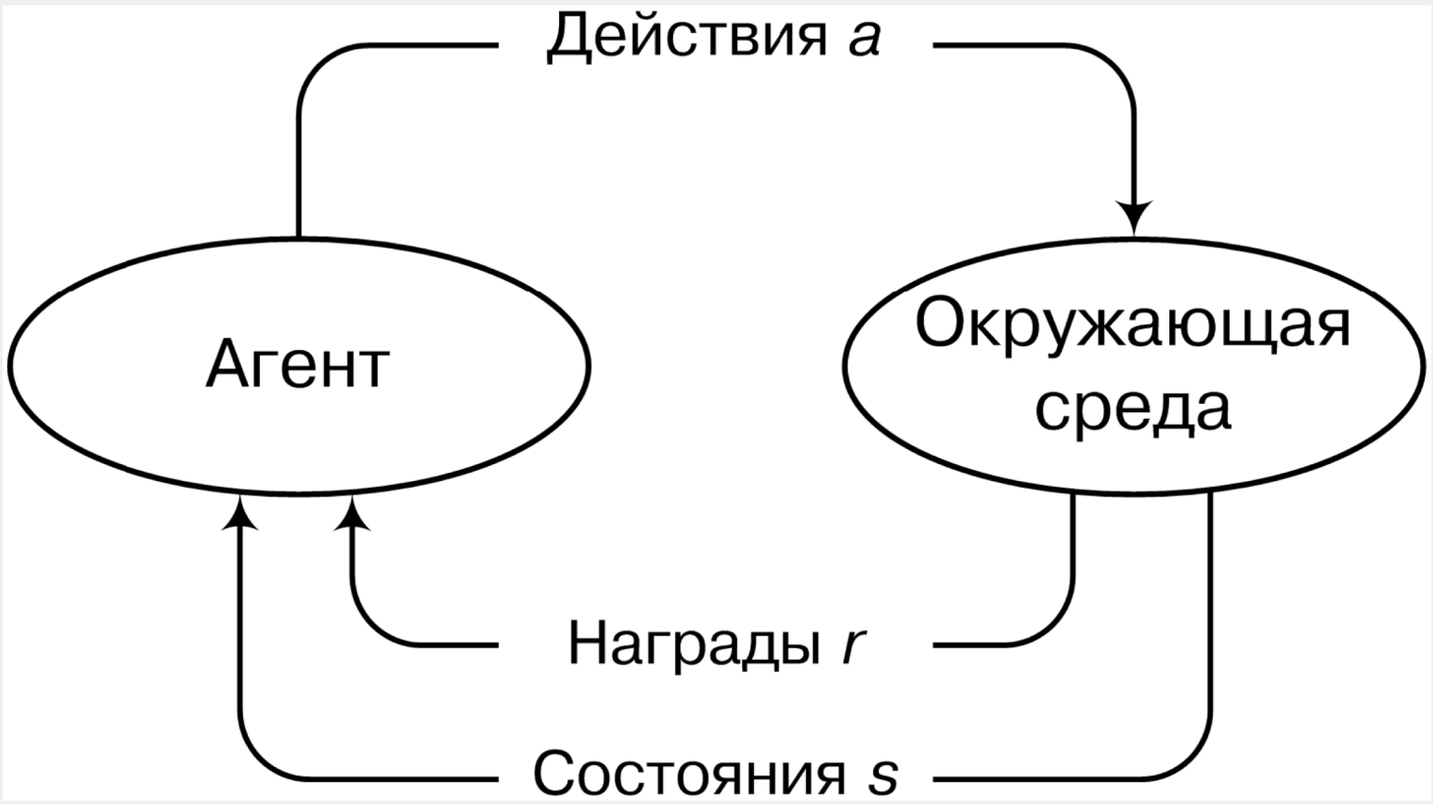
Обучение с учителем: Для заранее заданного набора объектов известны «правильные» ответы. Нужно научиться предсказывать ответы для новых объектов.
Обучение с подкреплением: Можно совершить действие и узнать ответ. Нужно научиться совершать действия приводящие к «лучшим» ответам.



Обучение с подкреплением

Марковский процесс принятия решений состоит из:

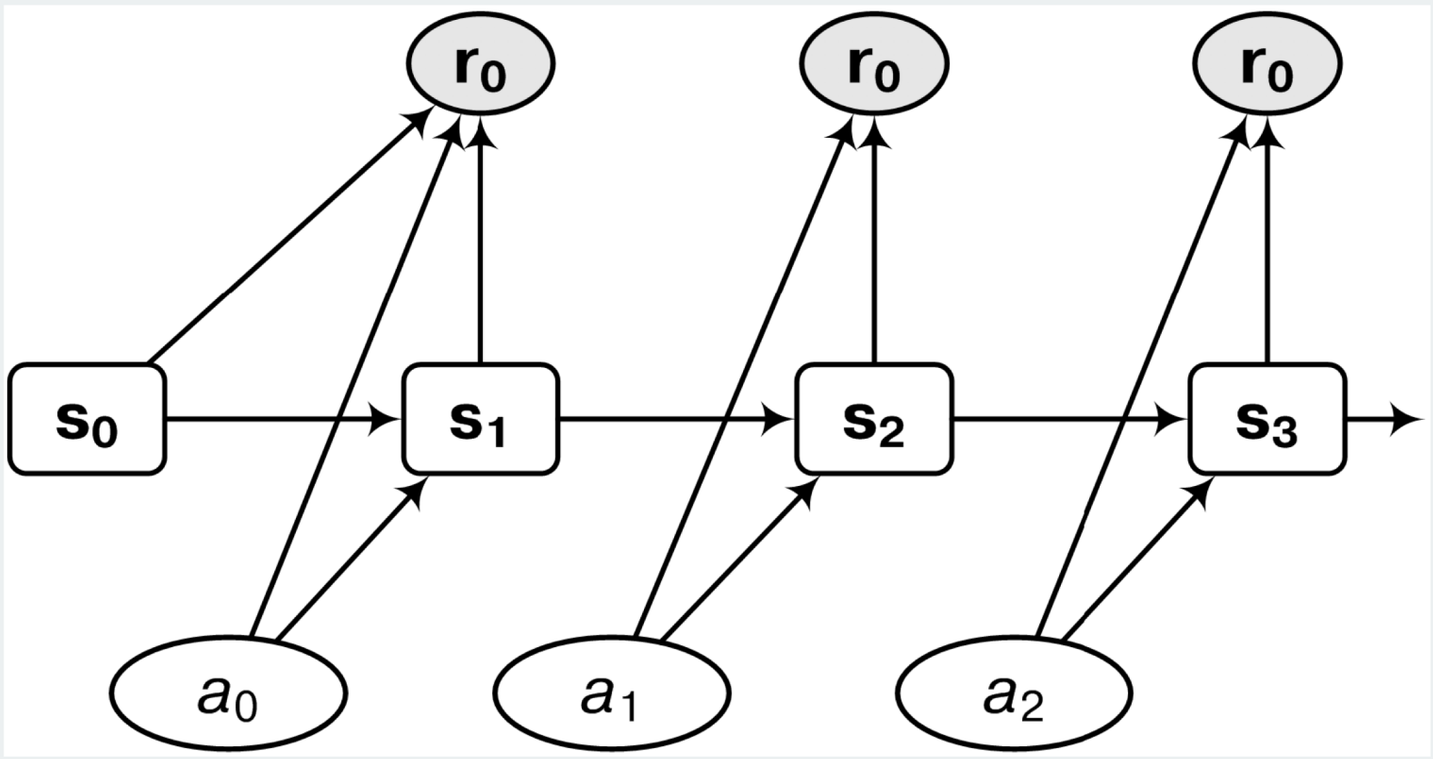
- Множества состояний S
- Множества действий A
- Функции вознаграждения $R: S \times A \to \mathbb{R}$, которая задает $R_{ss'}^{a}$
- Функции перехода между состояниями $p_s^a: S \times A \to \Pi(S)$, которая задает $P_{ss'}^{a}$



Артур Кадурин, преподаватель

Обучение с подкреплением

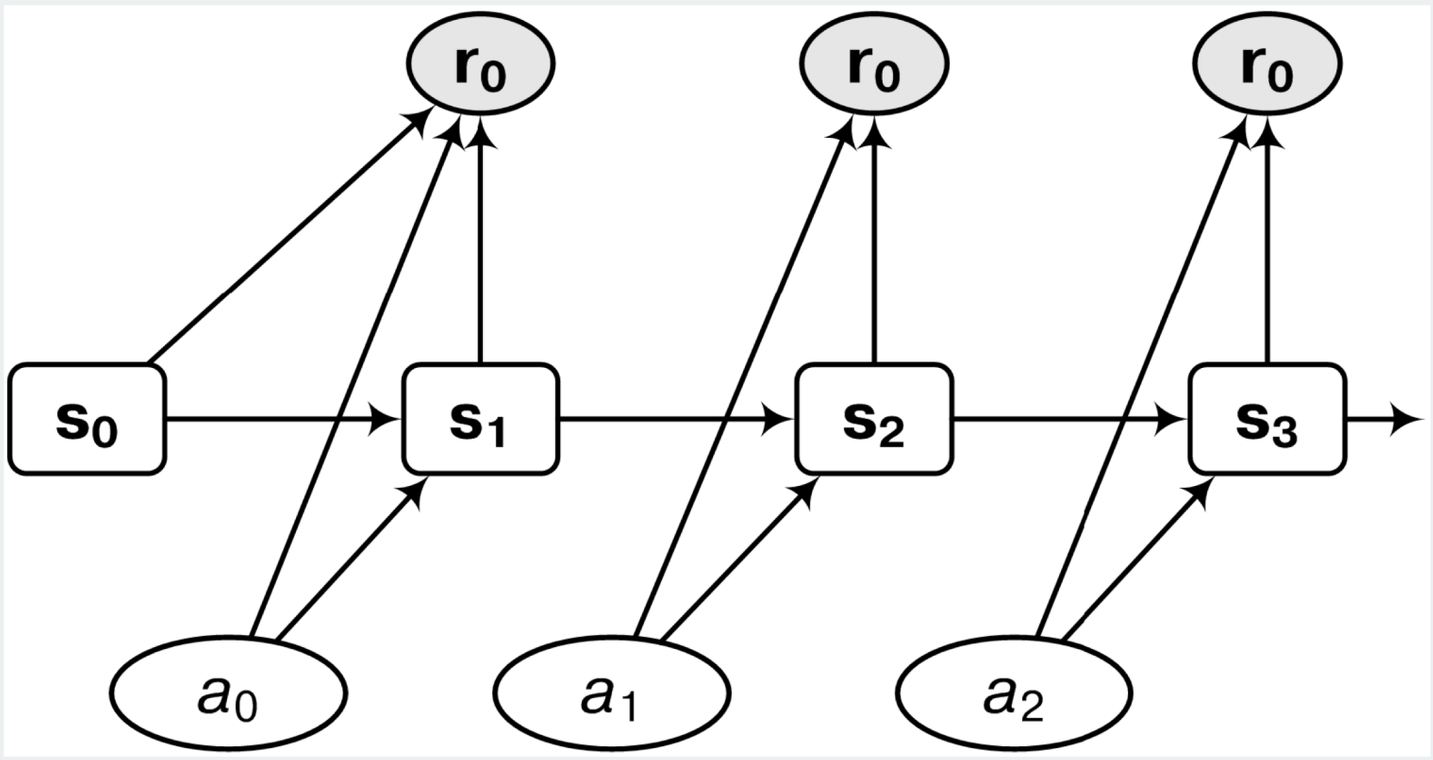
Марковское свойство заключается в том, что вероятности переходов между состояниями при выбранных действиях не зависят от истории предыдущих переходов.



Артур Кадурин, преподаватель

Обучение с подкреплением

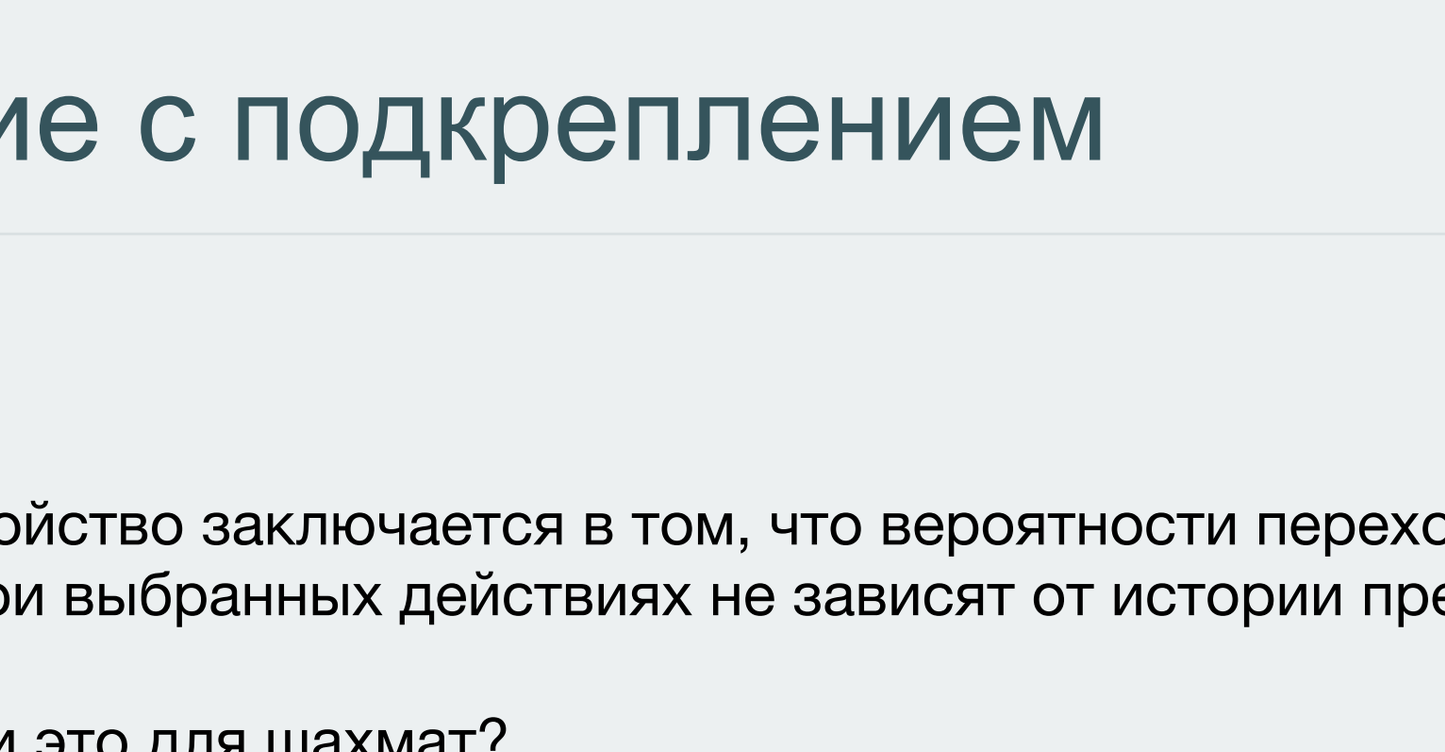
Марковское свойство заключается в том, что вероятности переходов между состояниями при выбранных действиях не зависят от истории предыдущих переходов.
Выполняется ли это для шахмат?



Артур Кадурин, преподаватель

Обучение с подкреплением

Марковское свойство заключается в том, что вероятности переходов между состояниями при выбранных действиях не зависят от истории предыдущих переходов.
Выполняется ли это для шахмат?
А для покера?

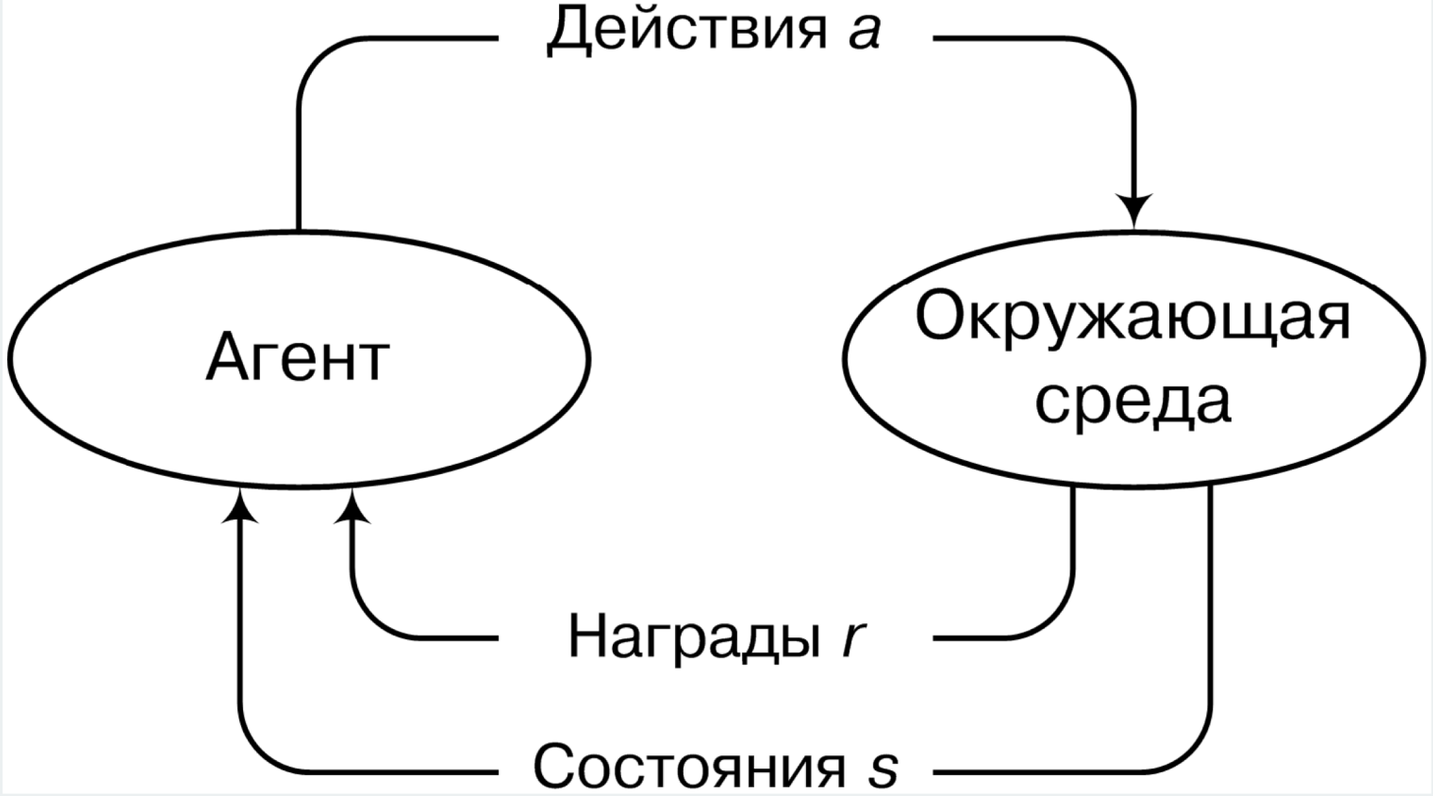


Артур Кадурин, преподаватель

Обучение с подкреплением

В реальных задачах вознаграждение совсем не обязательно следует сразу же за совершенным действием, поэтому награда обычно оценивается:

- Либо за отведенное время h: $R = \mathbb{E}\left[\sum_{t=0}^{h} r_t\right]$
- Либо за бесконечное время с дисконтом: $R = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$
- Либо в среднем за 1 шаг: $R = \lim_{h\to\infty} \mathbb{E}\left[\frac{1}{h}\sum_{t=0}^{h} r_t\right]$

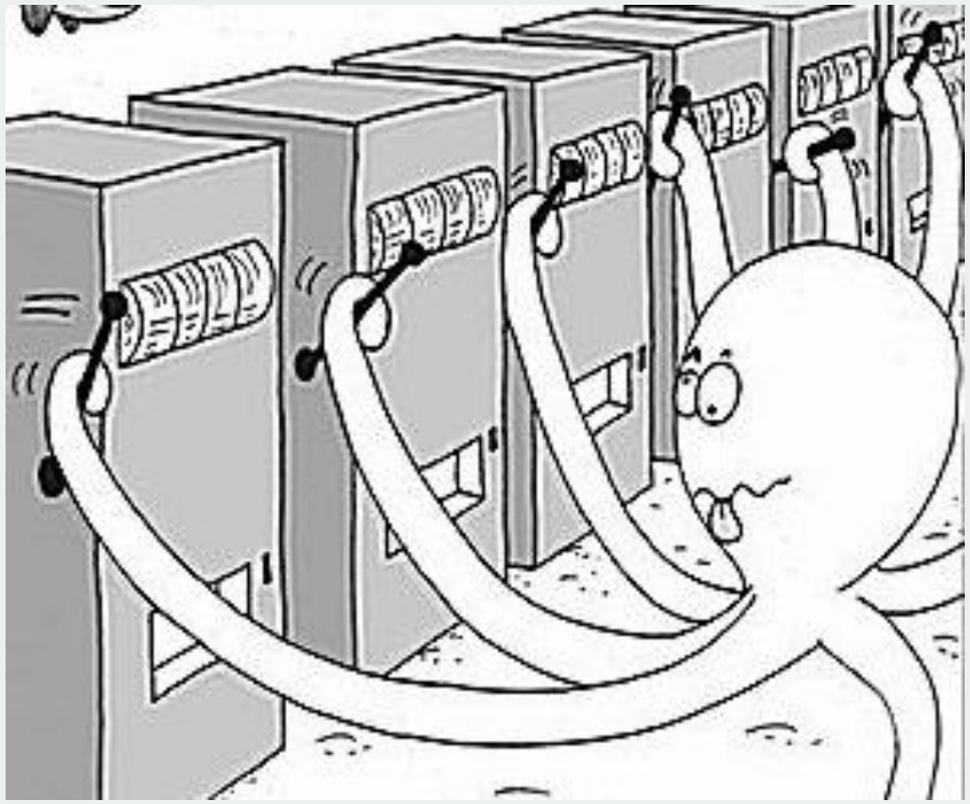


План на сегодня

1. Обучение с подкреплением
2. **Бандиты**
3. Стратегии

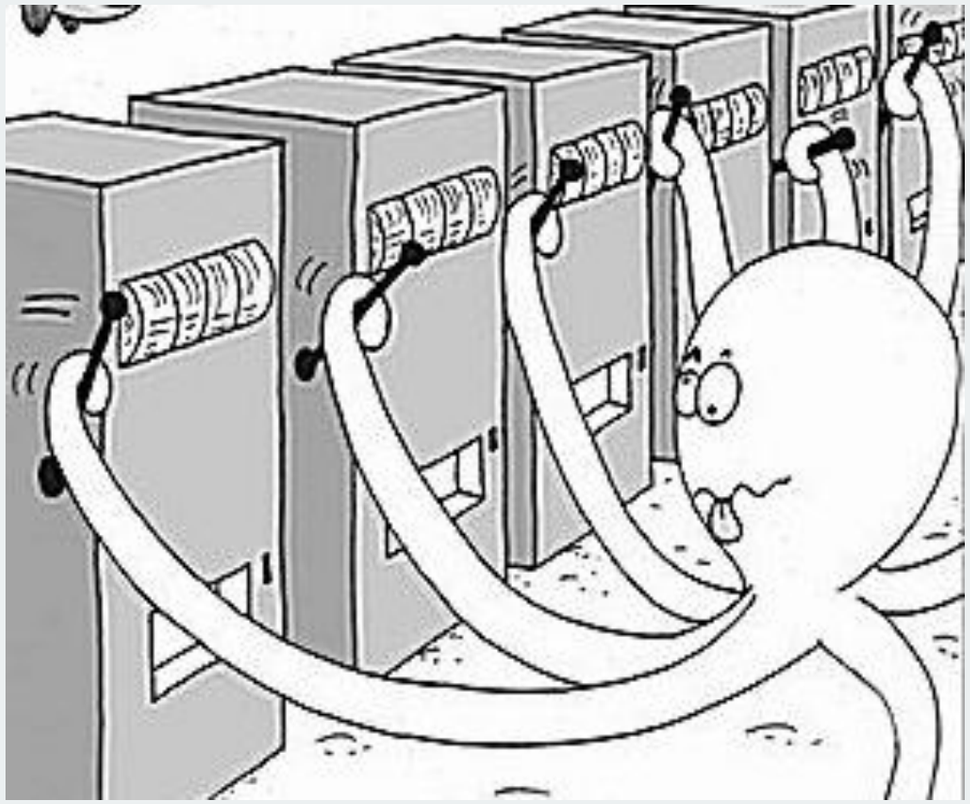
Многорукие бандиты

Допустим вы попали в комнату с кучей одноруких бандитов, каждый из которых выдает выигрыш со своей собственной вероятностью, которую вы не знаете. Ваша задача выиграть как можно больше (проиграть как можно меньше) денег за ночь.
Как вы будете это делать и почему это обучение с подкреплением?



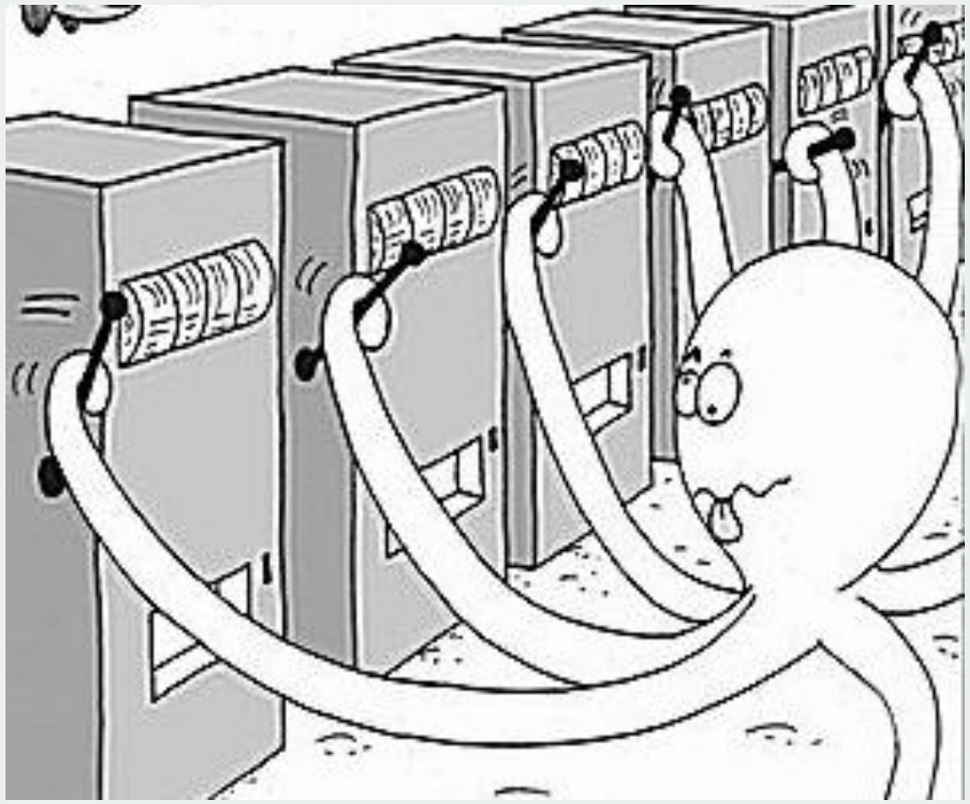
Многорукие бандиты

Жадный алгоритм:
Дернем каждую ручку n раз, а потом будем дергать только ту, у которой средний выигрыш после очередного шага является наибольшим.
Что с этой стратегией не так?



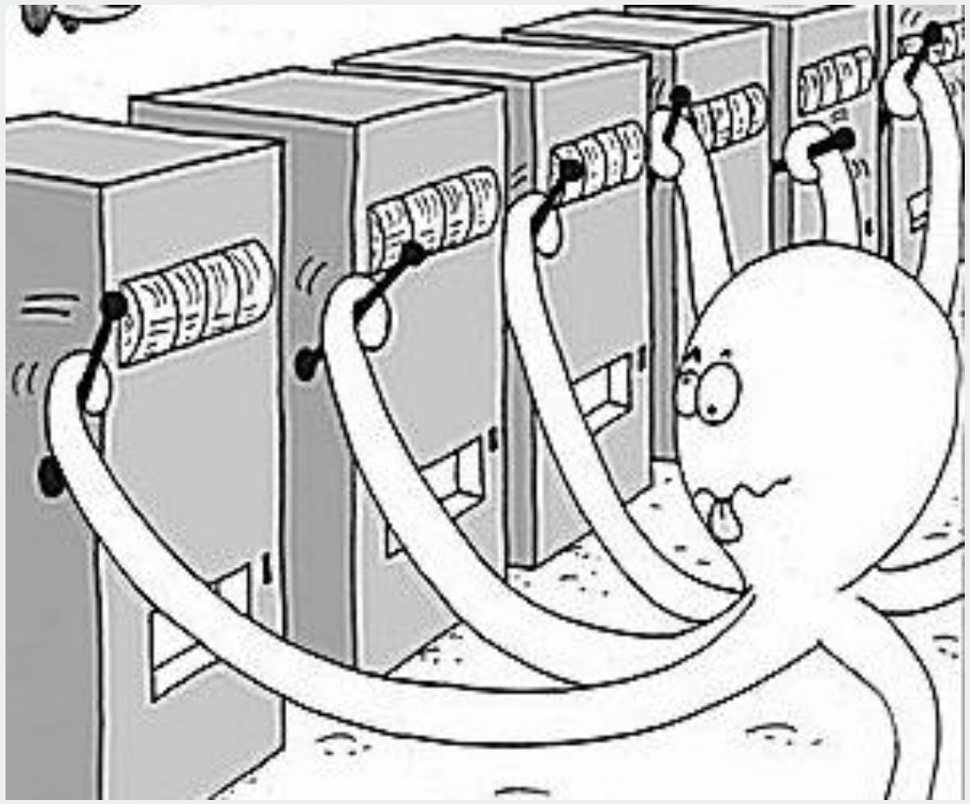
Многорукие бандиты

Эпсилон-жадный алгоритм:
Давайте теперь закладывать определенный уровень «неуверенности» в нашу стратегию и, скажем, с вероятностью ε будем совершать случайное действие. Мы гарантируем некоторую вероятность дернуть каждую ручку, и, значит, даже промахнувшись вначале, в конце концов найдем лучшую



Многорукие бандиты

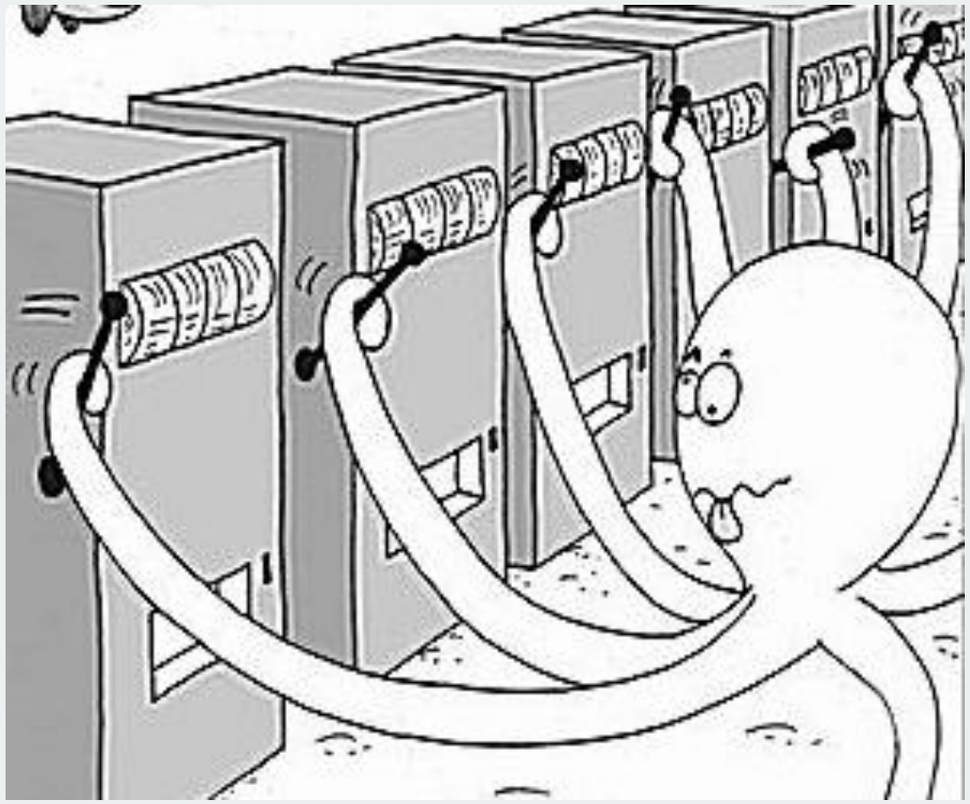
Оптимистично-жадный алгоритм:
Давайте оценивать не только среднее вознаграждение, но и доверительный интервал для каждой ручки, а выбирать ручки в соответствии с верхней границей доверительного интервала. Тогда, изначально мы оптимистично оцениваем все ручки и постепенно улучшаем оценку.



Многорукие бандиты

Мы не будем вдаваться в подробности алгоритмов, но в большинстве своем разные алгоритмы сводятся к разным оптимистичным способам оценки ручек.
Можно посмотреть:

- UCB1 и аналоги
- Thompson sampling

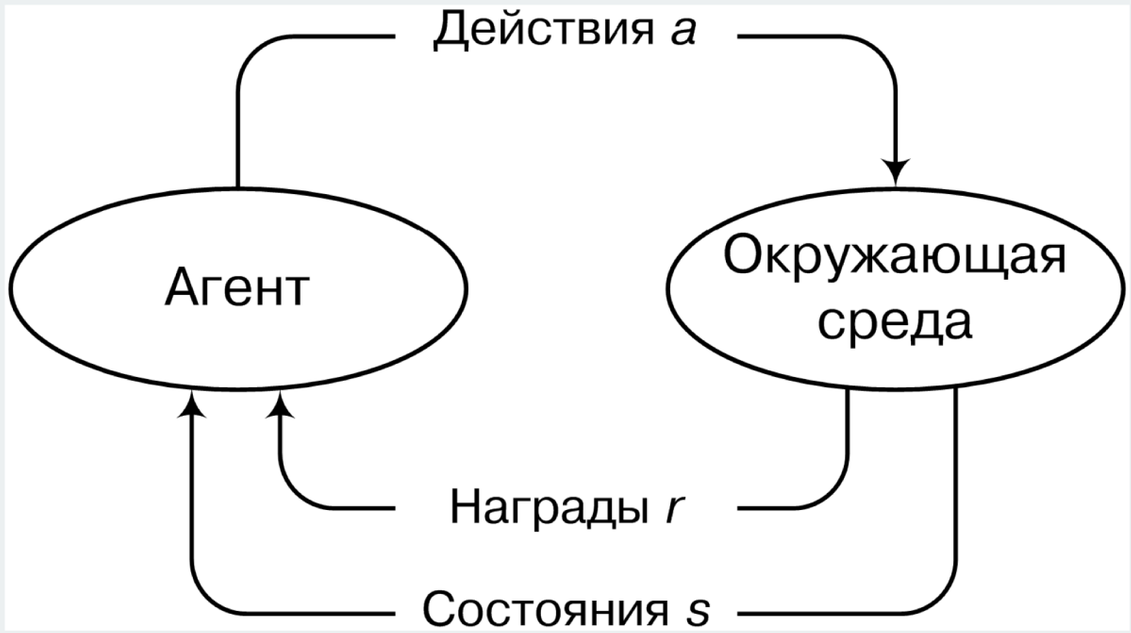


План на сегодня

1. Обучение с подкреплением
2. Бандиты
3. **Стратегии**

Поиск стратегии

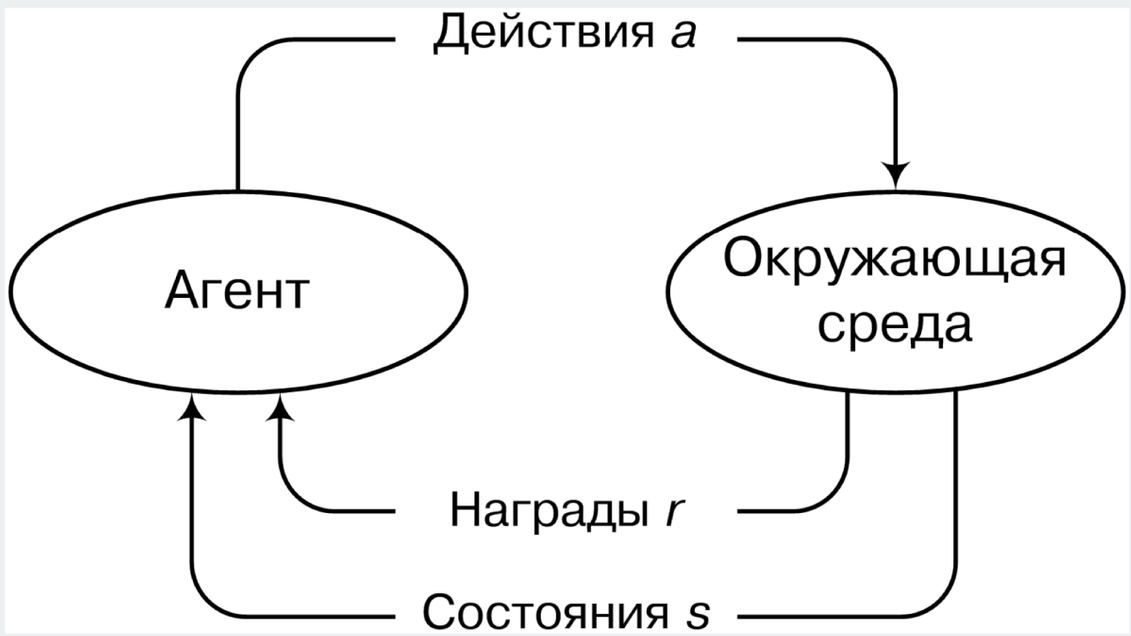
В более общем случае мы хотим уметь оценивать состояния, действия в зависимости от состояния или и то и другое одновременно.
Как можно оценить текущее состояние?



Поиск стратегии

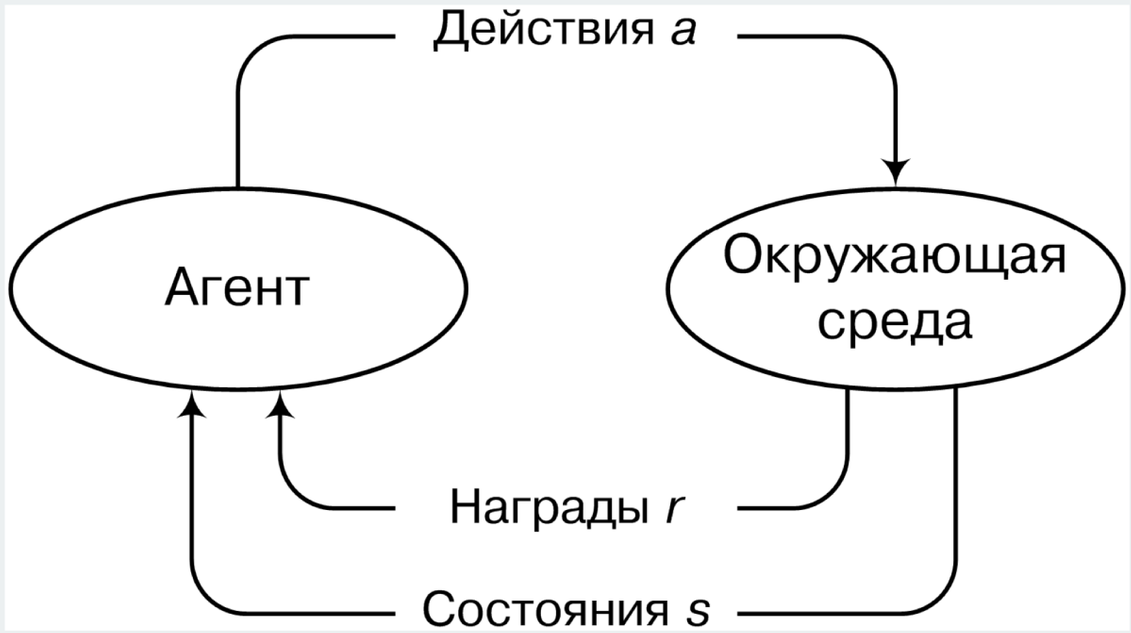
В более общем случае мы хотим уметь оценивать состояния, действия в зависимости от состояния или и то и другое одновременно.
Как можно оценить текущее состояние?

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s\right]$$



Поиск стратегии

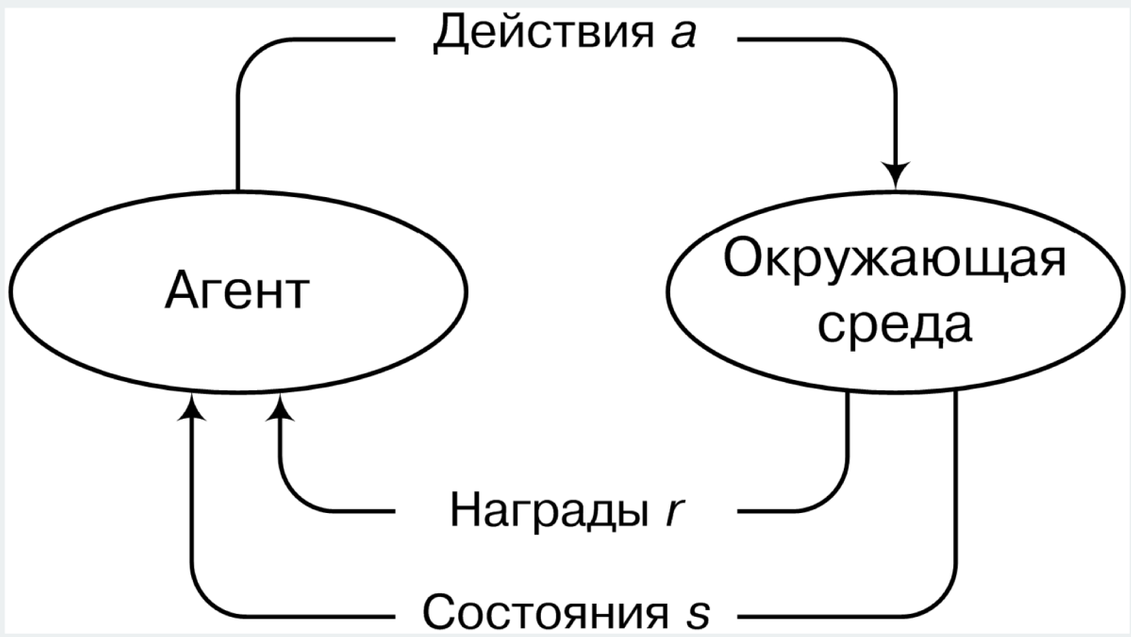
В более общем случае мы хотим уметь оценивать состояния, действия в зависимости от состояния или и то и другое одновременно.
Как можно оценить действие в заданном состоянии?



Поиск стратегии

В более общем случае мы хотим уметь оценивать состояния, действия в зависимости от состояния или и то и другое одновременно.
Как можно оценить действие в заданном состоянии?

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s, a_t = a\right]$$



Поиск стратегии

Вообще, стратегия π — это функция которая задает распределение вероятностей на множестве действий в зависимости от состояния. Все три функции — функцию оценки состояния V, функцию оценки действия Q и стратегию можно связать простым уравнением:

$$V^{\pi}(s) = \sum_{a \in A} \pi(s, a) * Q^{\pi}(s, a)$$

Поиск стратегии

Вообще, стратегия π — это функция которая задает распределение вероятностей на множестве действий в зависимости от состояния. Все три функции — функцию оценки состояния V, функцию оценки действия Q и стратегию можно связать простым уравнением:

$$V^{\pi}(s) = \sum_{a \in A} \pi(s,a) * Q^{\pi}(s,a)$$

При этом, Q мы тоже можем оценить с помощью V:

$$Q^{\pi}(s,a) = \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma V^{\pi}(s') \right)$$

Поиск стратегии

Вообще, стратегия π — это функция которая задает распределение вероятностей на множестве действий в зависимости от состояния. Все три функции — функцию оценки состояния V, функцию оценки действия Q и стратегию можно связать простым уравнением:

$$V^{\pi}(s) = \sum_{a \in A} \pi(s,a) * Q^{\pi}(s,a)$$

При этом, Q мы тоже можем оценить с помощью V:

$$Q^{\pi}(s,a) = \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma V^{\pi}(s') \right)$$

Понятно, что подставляя первое уравнение во второе или наоборот, мы можем избавиться от V или от Q соответственно:

$$V^{\pi}(s) = \sum_{a \in A} \pi(s,a) * \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma V^{\pi}(s') \right)$$

$$Q^{\pi}(s,a) = \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma \sum_{a' \in A} \pi(s',a') * Q^{\pi}(s',a') \right)$$

Поиск стратегии

$$V^{\pi}(s) = \sum_{a \in A} \pi(s,a) * \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma V^{\pi}(s') \right)$$

$$Q^{\pi}(s,a) = \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma \sum_{a' \in A} \pi(s',a') * Q^{\pi}(s',a') \right)$$

В реальности мы хотим оценивать все эти функций только с целью найти оптимальную стратегию, поэтому, нам совершенно не обязательно уметь делать оценки для совсем любой стратегии. Тогда, для оптимальной стратегии эти формулы превратятся в:

$$V^{*}(s) = \max_{a} \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma V^{*}(s') \right)$$

$$Q^{*}(s,a) = \sum_{s' \in S} P^{a}_{ss'} \left(R^{a}_{ss'} + \gamma \max_{a'} Q^{*}(s',a') \right)$$

Осталось только разобраться в том, как же научиться делать такие оценки

Поиск стратегии

Оказывается, есть очень простые итеративные способы. Один из основных современных подходов основан на принципе TD-learning (temporal difference). Суть которого заключается в том, чтобы при каждом очередном переходе немножко улучшать функцию оценки для предыдущего состояния основываясь на полученном вознаграждении и новом состоянии.

Поиск стратегии

Оказывается, есть очень простые итеративные способы. Один из основных современных подходов основан на принципе TD-learning (temporal difference). Суть которого заключается в том, чтобы при каждом очередном переходе немножко улучшать функцию оценки для предыдущего состояния основываясь на полученном вознаграждении и новом состоянии.

Простейший алгоритм TD-обучения можно записать так:

- Инициализировать s;
- Для каждого шага в эпизоде:
  - выбрать a по стратегии π или по формуле для V^*
  - сделать a и получить награду r и новое состояние s'
  - обновить $V(s) \coloneqq V(s) + \alpha\big(r + \gamma V(s') - V(s)\big)$
  - перейти к новому состоянию s'

Поиск стратегии

Оказывается, есть очень простые итеративные способы. Один из основных современных подходов основан на принципе TD-learning (temporal difference). Суть которого заключается в том, чтобы при каждом очередном переходе немножко улучшать функцию оценки для предыдущего состояния основываясь на полученном вознаграждении и новом состоянии.

Простейший алгоритм TD-обучения можно записать так:

- Инициализировать s;
- Для каждого шага в эпизоде:
  - выбрать a по стратегии π или по формуле для V^*
  - сделать a и получить награду r и новое состояние s'
  - обновить $V(s) \coloneqq V(s) + \alpha\big(r + \gamma V(s') - V(s)\big)$
  - перейти к новому состоянию s'

Для функции Q мы можем сделать абсолютно то же самое. Существенное ограничение этого метода заключается в том, что для того чтобы получить математические гарантии сходимости нужно обеспечить возможность встретить любую пару (s, a) бесконечное количество раз.
Как это можно сделать?

Заключение

1. Обучение с подкреплением основано на возможности и необходимости взаимодействовать с окружающей средой
2. В простейшем случае состояние окружающей среды не меняется
3. В задачах с известным поведением среды можно использовать простейшие итеративные методы.

Спасибо
за внимание!