Севастопольский государственный университет
Институт информационных технологий

Методы и системы искусственного интеллекта

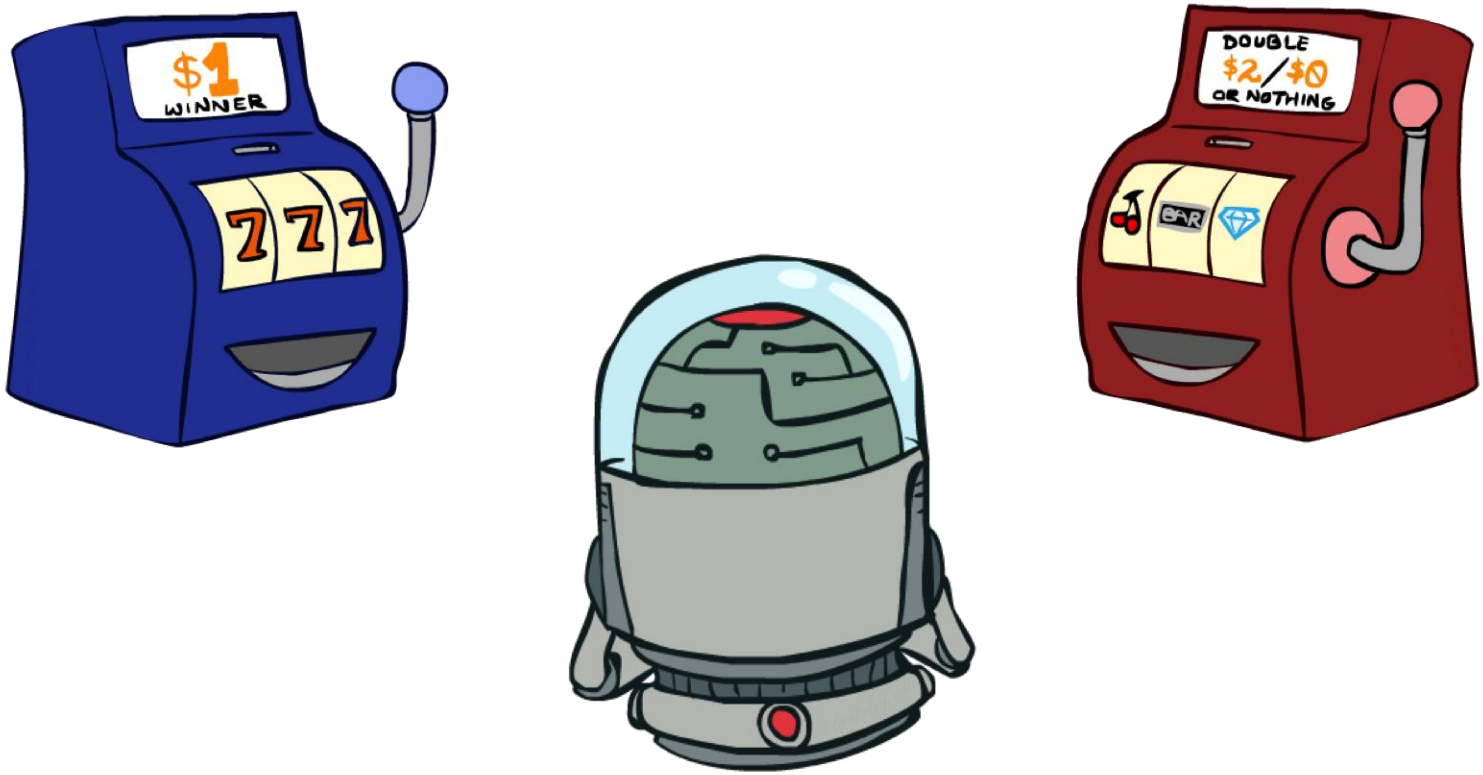
Бондарев Владимир Николаевич

Севастопольский государственный университет
Институт информационных технологий

Лекция
Обучение с подкреплением (reinforcement learning, RL)

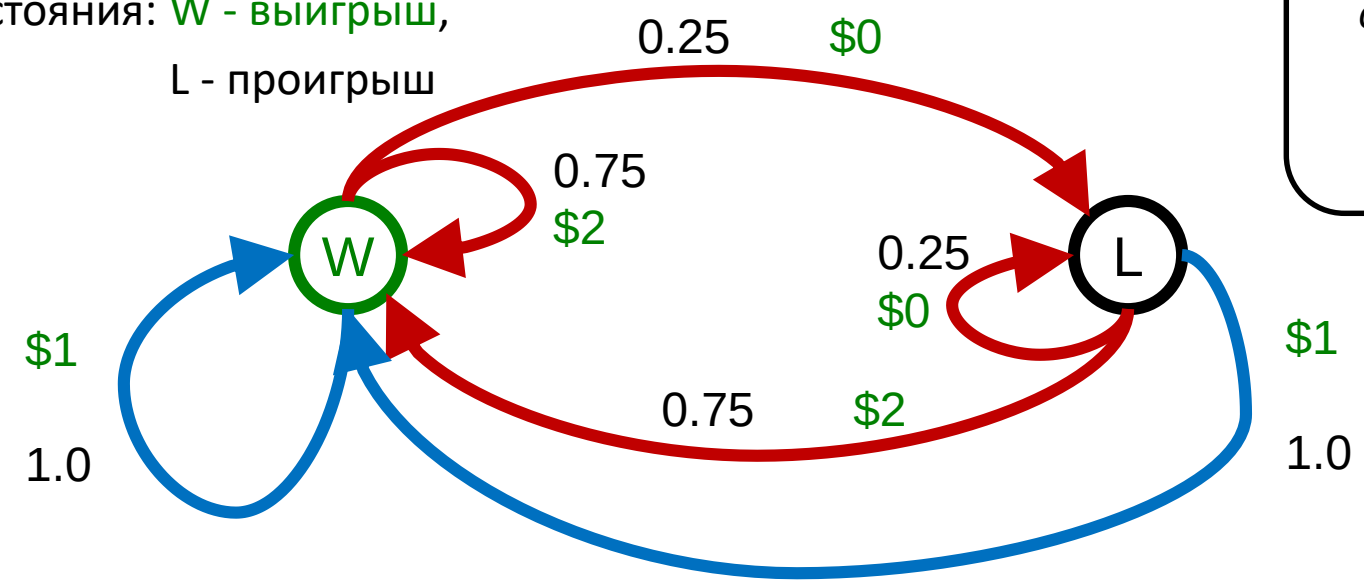
Бондарев Владимир Николаевич

Два одноруких бандита



MDP двойного бандита

- Действия: *Синий*, *Красный*
- Состояния: W - выигрыш,
  L - проигрыш



Нет дисконта, 10 шагов.. Оба состояния имеют равные значения

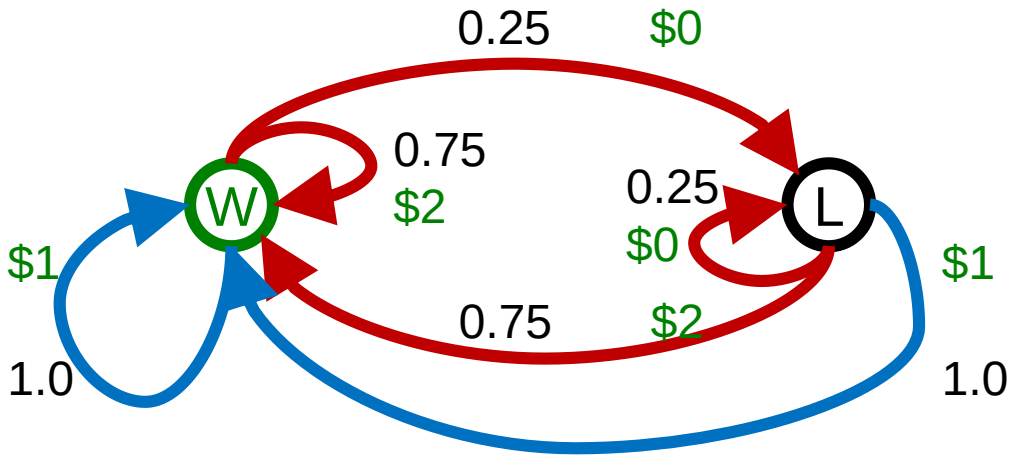
Off-лайн планирование

- Решение MDP – это off-лайн планирование
  - Мы определяем все значения путем вычислений
  - Необходимо знать детали MDP
  - В действительности мы не играли реально в игру!

    Мы только провели моделирование.

Нет дисконта, 10 шагов. Оба состояния имеют равные значения

	Значение
Красный	15
Синий	10



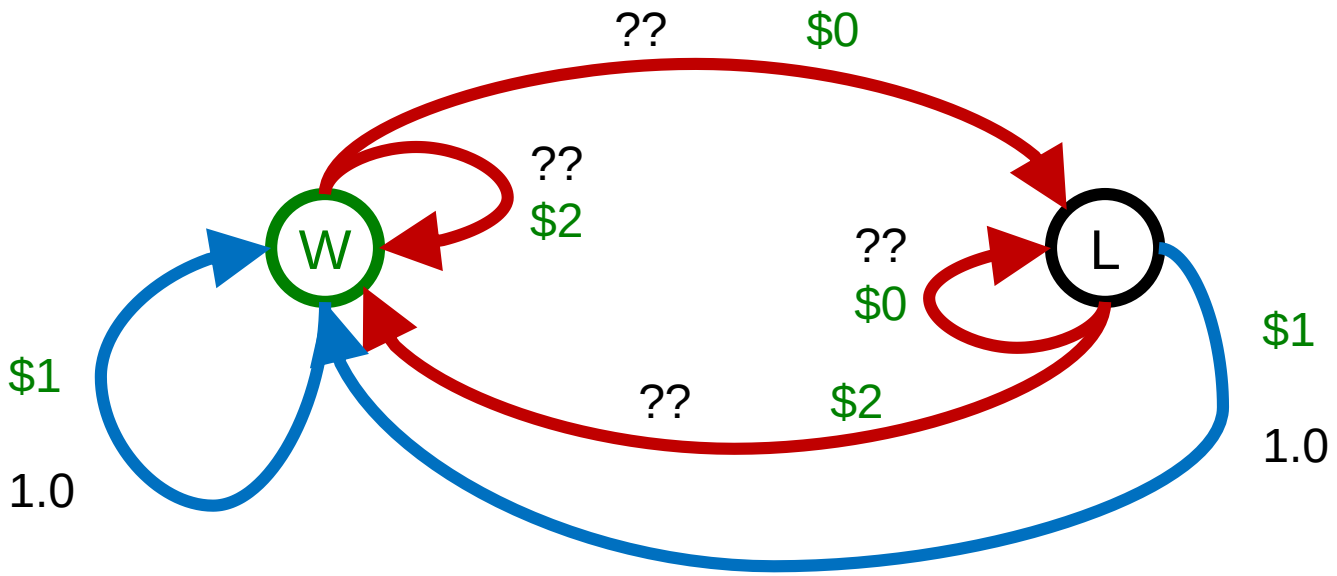
Давайте поиграем!

$2 $2 $0 $2 $2

$2 $2 $0 $0 $0

On-лайн планирование

- Правила изменились! Шансы красного автомата на победу другие.



Какова оптимальная политика? Мы не знаем MDP, поэтому здесь нет off-лайн планирования.

Давайте поиграем!

$1 $1 $1

$0 $0 $0 $2

Нужно потратить деньги, чтобы выяснить оптимальную стратегию

Что произошло?

- Это уже не планирование, это обучение!
  - В частности, обучение с подкреплением;
  - Это также MDP, но мы не можем найти решение простыми вычислениями;
  - Необходимо в действительности выполнять действия, что понять работу автомата .

- Важные идеи обучения с подкреплением:
  - Исследование: вы должны проверять новые действия, чтобы получать информацию о их полезности;
  - Эксплуатация: иногда, вы должны использовать уже известные действия;
  - Недостатки: даже если обучение успешно, возможны ошибки;
  - Необходимо осуществлять выборки: поскольку рассматриваются шансы, то необходимо повторять действия для накопления статистик;
  - Трудности: обучение может быть намного сложнее , чем решение известного MDP

Обучение с подкреплением

Обучение с подкреплением (**RL**, reinforcement learning) — область машинного обучения, в которой обучение осуществляется посредством взаимодействия с окружающей средой.

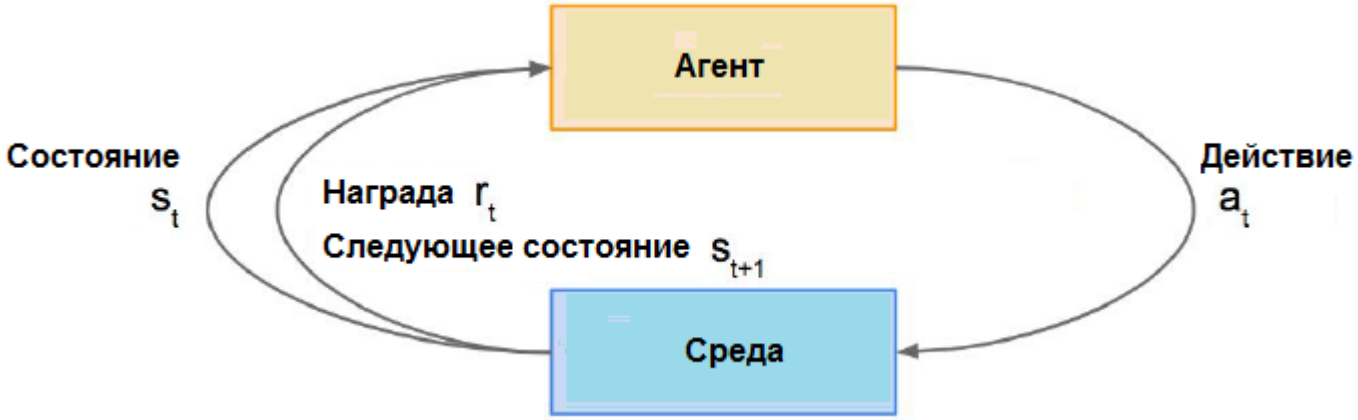
Это целенаправленное обучение, в котором обучаемый не получает информации о том, какие действия следует выполнять, вместо этого он узнает о последствиях своих действий.

Сейчас область обучения с подкреплением стремительно развивается, в ней появляется множество разнообразных алгоритмов, она является одной из самых активных областей исследования в сфере **искусственного интеллекта.**

Обучение с подкреплением

Имеется агент, взаимодействующий с окружающей средой, которая в ответ на действия агента, обеспечивает его вознаграждением.

Цель: научиться действовать, так чтобы максимизировать вознаграждение



В среде RL вы не учите агента, что и как он должен делать, вместо этого **вы даете агенту награду за каждое выполненное действие**. Награда может быть положительной или отрицательной. Агент выбирает действия, обеспечивающие максимальную награду. Таким образом, обучение превращается в процесс проб и ошибок.

Обучение с подкреплением

Предполагается наличие среды, описываемой Марковским процессом принятия решений (MDP), который определяется:

1. **Множеством состояний $s \in S$**;
2. **Множеством действий $a \in A$**;
3. **Моделью переходов $T(s,a,s')$;**
4. **Функцией вознаграждения $R(s,a,s')$.**

Цель решения также заключается в поиске политики $\pi(s)$;

Особенности задачи обучения с подкреплением:

1. Если в методах итерации по значениям и итерации по политикам нам были известны функции переходов T и вознаграждений R, то при обучении с подкреплением эти функции неизвестны;
2. При обучении с подкреплением агент выполняет он-лайн планирование, предпринимая попытки **исследования** среды, совершая действия и получая обратную связь в форме новых состояний и наград. Агент использует эту информацию для определения оптимальной политики в ходе процесса, называемого RL, прежде, чем **эксплуатировать** полученную политику.

Элементы RL

Последовательность одного шага взаимодействия агента со средой (s, a, s', r) называют **выборкой.** Коллекция выборок, которая приводит к терминальному состоянию, называется **эпизодом.**

Агент RL может **исследовать** (explore) новые выборки, которые обеспечивают различные награды, или же **эксплуатировать** (exploit) предыдущие выборки, которые привели к положительной награде.

Если агент RL исследует различные новые выборки, существует высокая вероятность того, что агент получит отрицательную награду, так как не все действия будут лучшими. Если агент RL эксплуатирует только известные лучшие выборки, существует высокая вероятность упустить лучшее действие, которое может принести более высокую награду. Всегда существует компромисс между **исследованием** и **эксплуатацией**. Невозможно заниматься исследованием и эксплуатацией одновременно.

Агент обычно выполняет много эпизодов в ходе исследования для того, чтобы собрать достаточно данных для обучения.

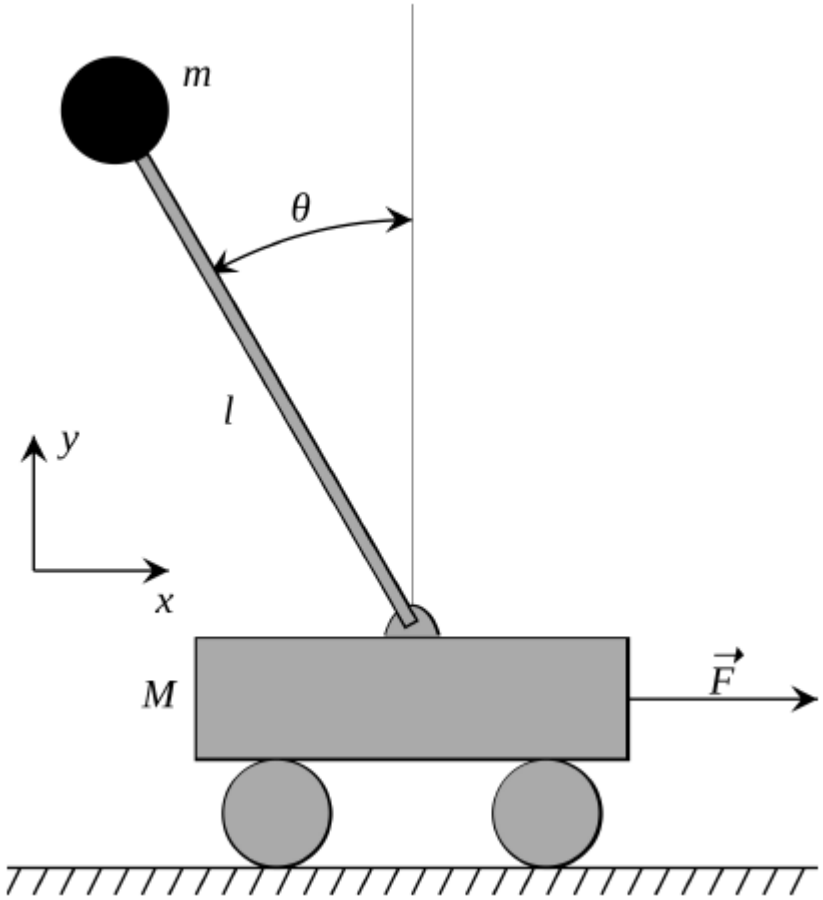
Элементы RL

Модель (model) является представлением среды с точки зрения агента. Различают два типа обучения с подкреплением: **основанное на модели и без модели**.

В процессе обучения, основанного на модели, агент предпринимает попытки оценивания вероятностей переходов **T** и вознаграждений **R** по выборкам, получаемым во время исследования, перед тем как использовать эти оценки для нахождения MDP решения, обычно путем итерации по политикам.

При обучении без модели агент осуществляет непосредственную оценку ценности состояний и q-ценностей без восстановления модели распределения вероятностей переходов и наград. При обучении без модели агент просто полагается на метод проб и ошибок для выбора оптимального действия.

Задача обратного маятника



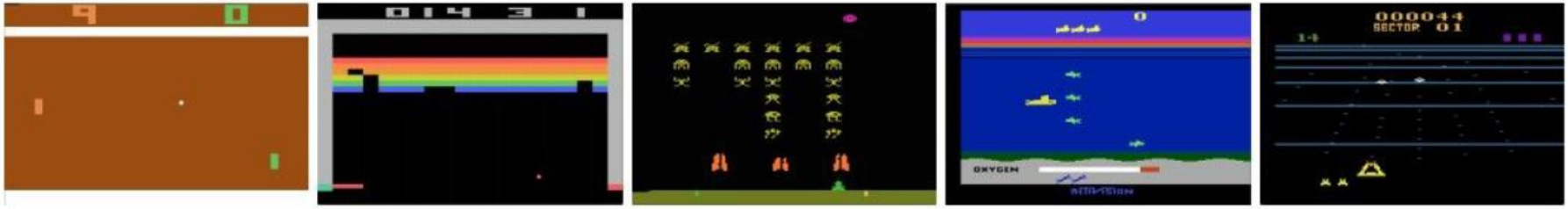
Цель: обеспечить равновесие маятника на подвижной тележке

Состояние: угол, угловая скорость, положение, горизонтальная скорость

Действие: изменение горизонтальной силы, приложенной к тележке

Награда: 1 на каждом временном шаге, если маятник находится в вертикальном положении

Atari игры



Цель: завершить игру с наибольшим количеством очков

Состояние: значения пикселей экранных форм игры

Действие: управление игрой, например: влево, вправо, вверх, вниз

Награда: оценка увеличивается / уменьшается на каждом временном шаге

Volodymyr Mnih и др., 2013

Пример: Обучение хождению

Начало

Одна попытка обучения

После обучения
[1K попыток]

Пример: Обучение хождению

Начало

Пример: Обучение хождению

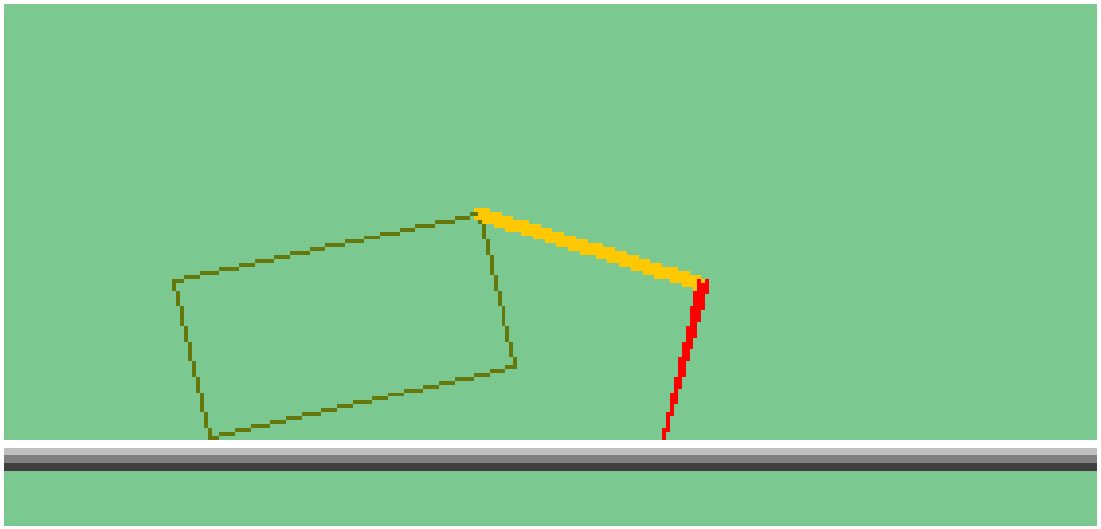
Обучение

Пример: Обучение хождению

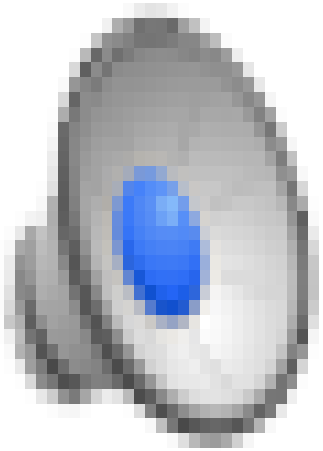
После обучения

Пример: Боковое движение (змея)

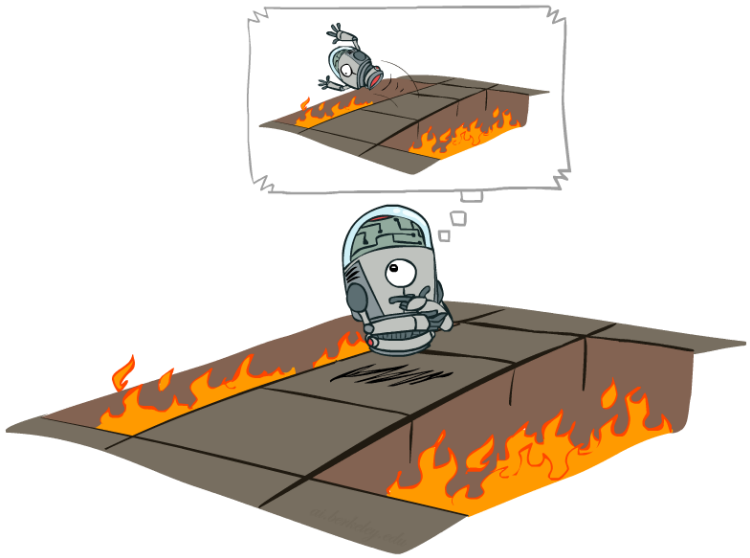
Ползунок!



Демо: ползунок



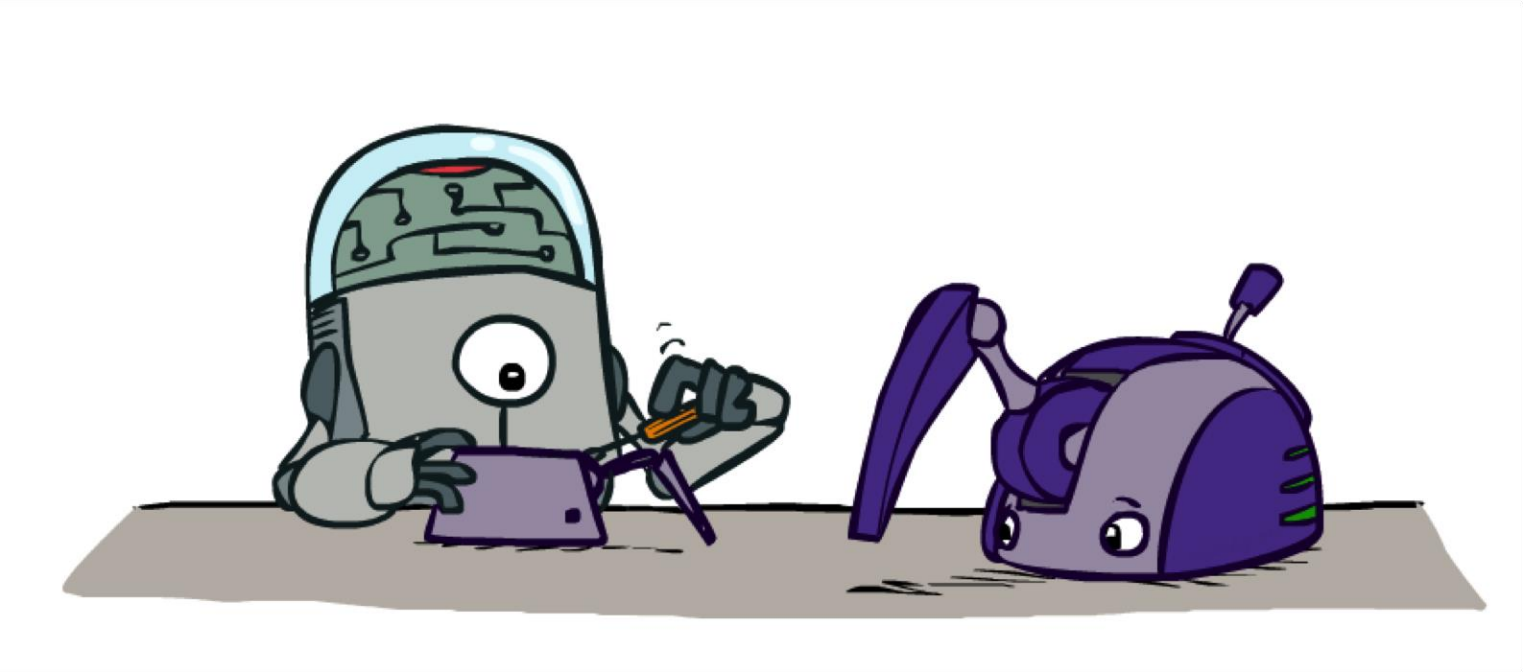
Офф-лайн MDP и он-лайн RL



Офф-лайн
решение

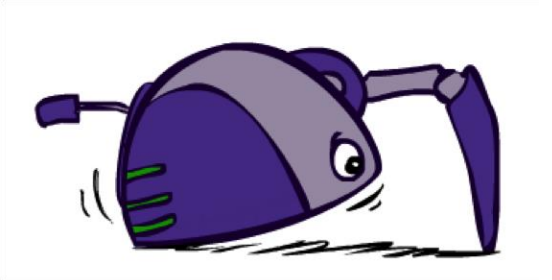
Он-лайн
обучение

Обучение на основе модели



Обучение на основе модели

- Идея обучения на основе модели:
  - Обучиться аппроксимации модели на основе выборок;
  - Выполнить вычисление значений, предполагая, что обученная модель корректна

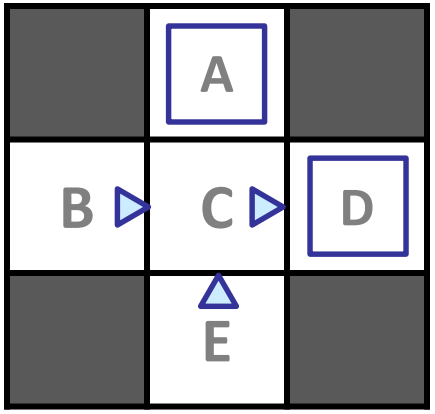


- Шаг 1: **Эмпирическое обучение MDP модели**
  - Подсчёт исходов *s'* для каждой пары (*s*, *a*)
  - Нормализация для получения оценки $\hat{T}(s, a, s')$
  - Оценка наград $\hat{R}(s, a, s')$ для каждого перехода (*s, a, s'*)

- Шаг 2: **Получение решения на основе MDP**
  - Например, использование итераций по значениям

Пример: Обучение, основанное на модели

Исходная политика π



Примем: $\gamma = 1$

Эпизоды (Обучение)

Эпизод 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Обучение модели

Статистика выборок

s	a	s’	N
A	Exit	x	1
B	East	C	2
C	East	A	1
C	East	D	3
D	Exit	x	3
E	North	C	2

Функция переходов $\hat{T}(s, a, s')$

T(A,exit,x)=N(A,exit,x)/N(A,exit)=1/1=1

T(B,East,C)=N(B,East,C)/N(B,East)=2/2=1

T(C,East,A)=N(C,East,A)/N(C,East)=1/4=0.25

T(C,East,D)=N(C,East,D)/N(C,East)=3/4=0.75

T(D,Exit,x)=N(D,Exit,x)/N(D,Exit)=3/3=1

T(E,North,C)=N(E,North,C)/N(E,North)=2/2=1

Функция наград $\hat{R}(s, a, s')$

R(A,exit,x)=-10

R(B,East,C)=-1

R(C,East,A)=-1

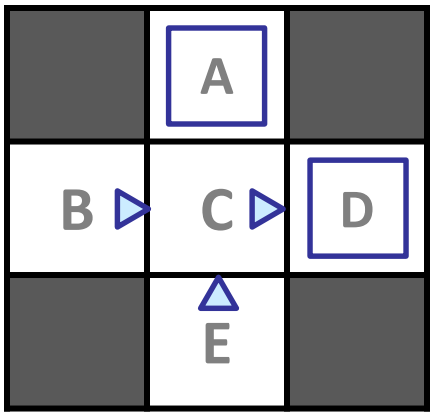
R(C,East,D)=-1

R(D,Exit,x)=+10

R(E,North,C)=-1

Пример: Обучение, основанное на модели

Исходная политика π



Примем: $\gamma = 1$

Эпизоды (Обучение)

Эпизод 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Обученная модель

$\hat{T}(s, a, s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
…

$\hat{R}(s, a, s')$

R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
…

После схождения оценок T обучение завершается генерацией политики $\pi(s)$ на основе алгоритмов итерации по значениям или политикам. Обучение на основе модели интуитивно простое, но характеризуется большой пространственной сложностью (из-за необх. хранения счетчиков (s,a,s’))

Аналогия: Средний возраст

Цель: Вычислить средний возраст группы

Известная P(A)

$$E[A] = \sum_a P(a) \cdot a \qquad = 0.35 \times 20 + \ldots$$

Без P(A), просто накапливаем выборки [a_1, a_2, … a_N]

Неизв. P(A): “На основе модели”

$$\hat{P}(a) = \frac{\text{num}(a)}{N}$$

$$E[A] \approx \sum_a \hat{P}(a) \cdot a$$

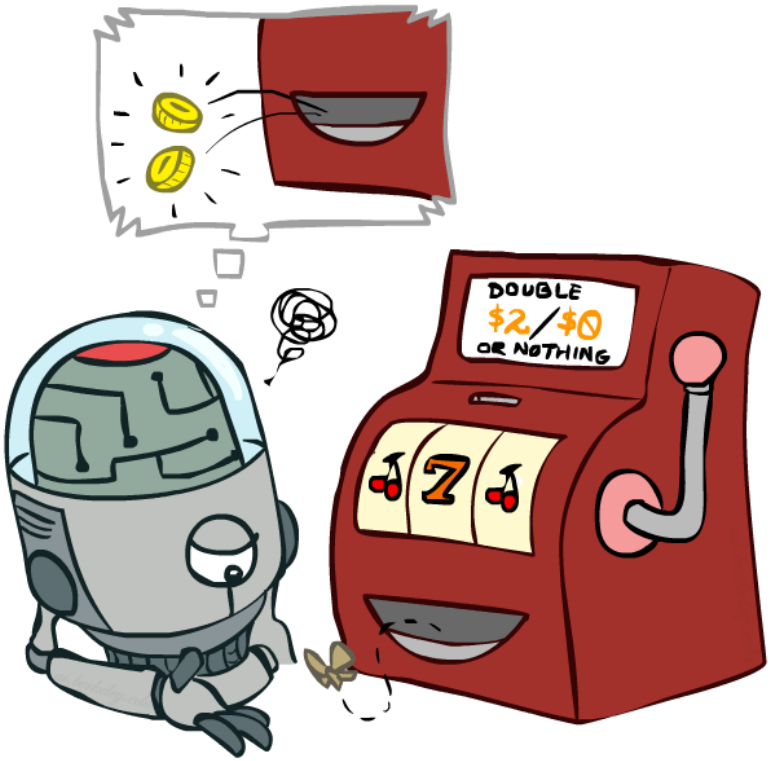
Почему работает? Т.к. в итоге обучаемся правильной модели.

Неизв. P(A): “Без модели”

$$E[A] \approx \frac{1}{N} \sum_i a_i$$

Почему работает? Т.к. выборки появляются с определенной частотой.

RL обучение без модели



Пассивное и активное RL

При **обучении с подкреплением без модели** используют три алгоритма, которые разделяются на две группы:

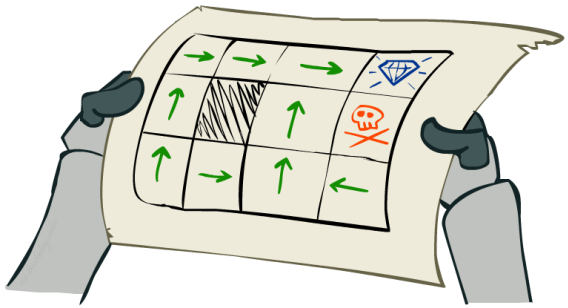
1. **Алгоритмы пассивного обучения:**
   - Алгоритм прямого оценивания;
   - Обучение на основе временных различий;
2. **Алгоритмы активного обучения:**
   - Q-обучение.

В случае пассивного обучения агент следует заданной политике и обучается ценностям состояний на основе накопления выборочных значений из эпизодов, что в общем, соответствует оцениванию политики при решении задачи MDP, когда $\boldsymbol{T}$ и $\boldsymbol{R}$ известны.

В случае активного обучения агент использует обратную связь для итеративного обновления его политики, пока не построит оптимальную политику после достаточного объема исследований.

Пассивное RL

- Упрощенная задача:
  - Фиксируется политика $\pi(s)$;
  - Переходная функция $T(s,a,s')$ неизвестна
  - Функция наград $R(s,a,s')$ неизвестна
  - Цель: обучиться оцениванию ценности состояний



- В этом случае:
  - Обучение « в пути»
  - Все равно, какие действия предпринимать
  - Просто следуйте политике и учитесь на собственном опыте
  - Это НЕ офлайн-планирование! Агент реально действует в мире среды.

Алгоритм прямого оценивания (обучение без модели)

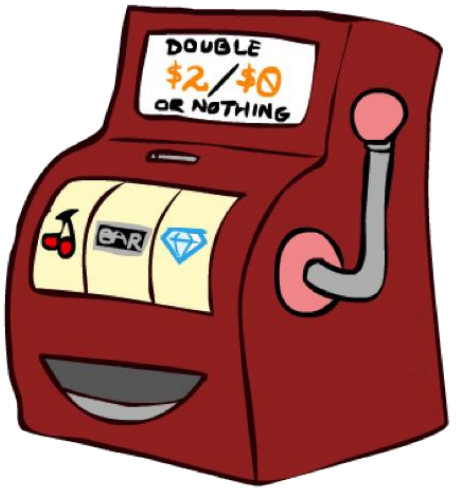
Цель: Вычисление ценности каждого состояния при политике π

Идея: Усреднять наблюдаемые выборки значений ценности

Алгоритм:

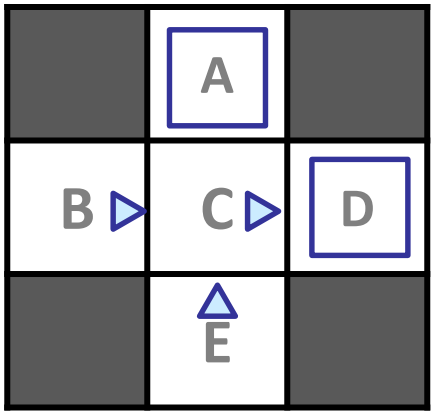
- Действовать в соответствии с π:
  - Каждый раз при посещении состояния, подсчитывать и аккумулировать ценности состояния и число посещений состояния;
  - Найти среднюю ценность состояния

- Это называется **прямым оцениванием**



Пример: алгоритм прямого оценивания

Входная политика π



Примем: $\gamma = 1$

Наблюдаемые эпизоды (Обучение)

Эпизод 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

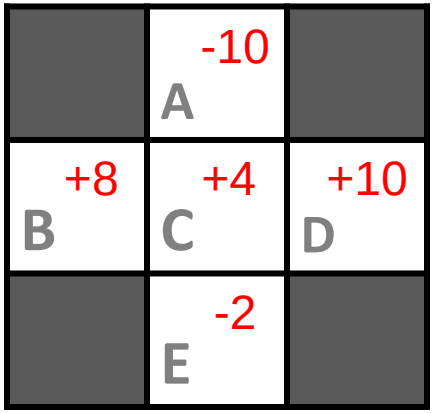
Эпизод 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Эпизод 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Ценности состояний



S	Сумм.награда	Число посещений	$V^{\pi}(s)$
A	-10	1	-10
B	16	2	8
C	16	4	4
D	30	3	10
E	-4	2	-2

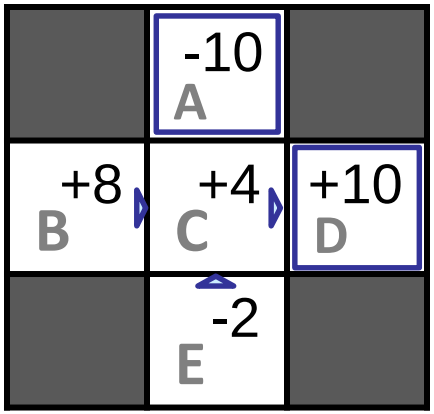
V(B)= (-1)+(-1)+10=8

Проблемы с прямым оцениванием

- Положительные свойства прямого оценивания:
  - Простота;
  - Не требует никаких знаний T, R;
  - В итоге вычисляет средние значения ценностей, используя просто выборки переходов.

- Недостатки:
  - Утрачивает информацию о связи состояний;
  - Каждое состояние, должно обучаться отдельно;
  - Таким образом, требует большого времени обучения

Ценности



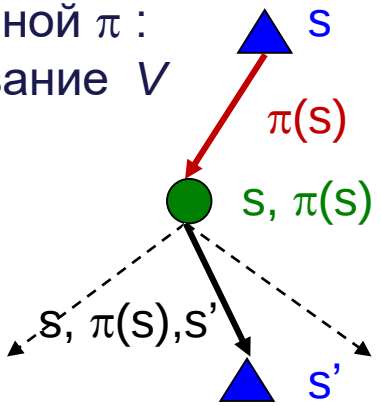
Если оба B и E приводят к C при фикс.политике, как могут их ценности быть различными?

Почему бы не использовать метод оценивания политики?

- Упрощенное обновление Беллмана вычисляет *V* при заданной π :
  - На каждом шаге V заменяется на одношаговое предсказание *V*

$$V_0^\pi(s) = 0$$

$$V_{k+1}^\pi(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_k^\pi(s')]$$



  - Этот подход полностью эксплуатирует связь между состояниями
  - К сожалению, необходимы *T* и *R* , чтобы вычислить *V*

- **Ключевой вопрос: как выполнить обновление *V* без знания *T* и *R*?**
  - Другими словами, как вычислить взвешенное среднее без информации о весах?

Оценивание политики на основе выборок

- Мы хотим улучшить оценку *V*, вычислив среднее:

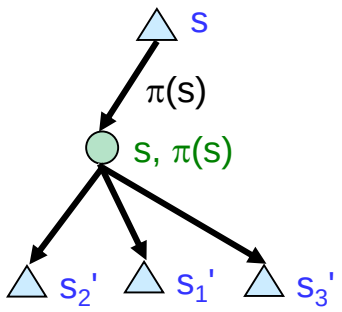
$$V^{\pi}_{k+1}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^{\pi}_{k}(s')]$$

- **Идея**: Взять выборки результатов переходов и усреднить

$$sample_1 = R(s, \pi(s), s'_1) + \gamma V^{\pi}_{k}(s'_1)$$
$$sample_2 = R(s, \pi(s), s'_2) + \gamma V^{\pi}_{k}(s'_2)$$
$$\cdots$$
$$sample_n = R(s, \pi(s), s'_n) + \gamma V^{\pi}_{k}(s'_n)$$

$$V^{\pi}_{k+1}(s) \leftarrow \frac{1}{n} \sum_{i} sample_i$$

- Таким образом мы смогли бы выполнить шаг обновления значения *V*



Не работает, т.к. мы не можем обратить время: чтобы получить новую выборку после уже совершенного перехода из s и s'

Экспоненциальное скользящее среднее

- Экспоненциальное скользящее среднее :

$$\bar{Y}(n)=1/n\,(Y1)+Y(2)+\ldots+Y(n))=1/n\,(Y(1)+Y(2)+\ldots+Y(n-1))+1/n\,Y(n)=$$

$$=(n-1)/n\;1/(n-1)\,(Y(1)+Y(2)+\ldots+Y(n-1))+1/n\,Y(n)=(1-1/n)\,\bar{Y}(n-1)+1/n\,Y(n)$$

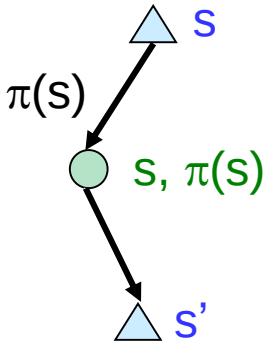
- Обозначим $\alpha= 1/n$, тогда

$$\bar{Y}(n) = (1-\alpha)\,)\,\bar{Y}(n-1) + \alpha\, Y(n)$$

  - Забывает предысторию ;
  - Уменьшающаяся скорость обучения (альфа) создает условия схождения среднего

Обучение на основе временных различий (temporal difference -TD)

- Основная идея: обучаться на основе каждого опыта (попытки)!
  - Обновлять V(s) каждый раз, при испытании перехода (s, a, s', r)
  - Более вероятные исходы будут вносить вклад в обновления более часто
- TD – обучение
  - Политика фиксируется, пока выполняется оценивание!
  - Выполнять скользящее усреднение выборок



Выборка V(s): $sample = R(s, \pi(s), s') + \gamma V^{\pi}(s')$

Обновление V(s): $V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + (\alpha)sample$

Или: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$

Метод временных различий TD (temporal difference)

В TD-обучении для обновления ценности состояния используется так называемое **правило *обновления на основе временных различий***, основанное на аппроксимации матожидания в уравнении Беллмана *экспоненциальным скользящим средним*

$$V^{\pi}(s) := V^{\pi}(s) + \alpha\,(r + \gamma\, V^{\pi}(s') - V^{\pi}(s)).$$

Что фактически означает это выражение?

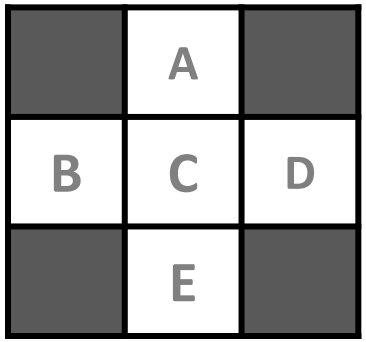
Интуиция подсказывает, что результат будет равен разности между выборочной наградой ($r + \gamma\, V^{\pi}(s')$) и ожидаемой наградой $V^{\pi}(s)$, умноженной на скорость обучения α. Фактически эта разность есть погрешность, и мы будем называть ее *TD-погрешностью*.

Последовательность действий **алгоритма TD-прогнозирования** выглядит так:

1. Инициализировать $V(s)$ нулями или произвольными значениями.
2. Запустить эпизод, для каждого шага в эпизоде выполнить действие a в состоянии s, получить награду r и перейти в следующее состояние (s').
3. Обновить предыдущую ценность состояния по правилу TD-обновления.
4. Повторять шаги 2 и 3, пока не будет достигнуто завершающее состояние.

Пример: **TD-обучение**

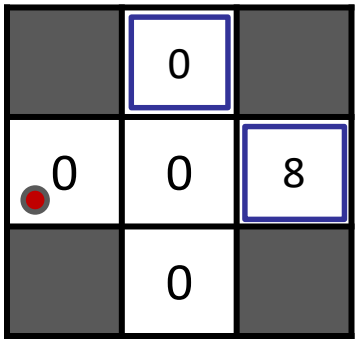
Состояния



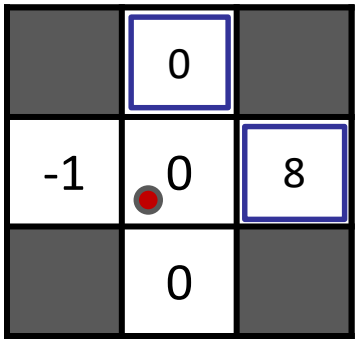
Примем: $\gamma = 1$, $\alpha = 1/2$

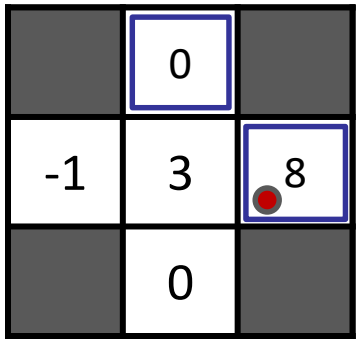
Наблюдаемые переходы

B, east, C, -2



C, east, D, -2





$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

0+1/2(-2+0)=-1

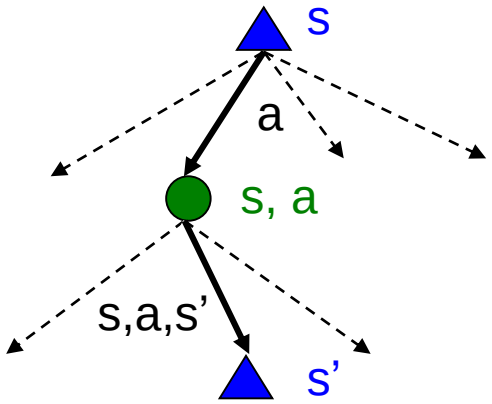
0+1/2(-2+8)=3

Проблемы TD-обучения

- TD обучение – подход к оцениванию политики без модели, моделирующий обновление Беллмана с помощью он-лайн усреднения выборок
- Однако, если необходимо будет преобразовать значения V в новую политику возникнут сложности:

$$\pi(s) = \arg\max_a Q(s,a)$$

$$Q(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s') \right]$$



- Вывод: необходимо обучаться Q-ценностям, а не ценностям состояний
- **Тогда, возможен ли выбор действия также без модели**(без T) **?**

Напоминание: итерации по Q-ценностям

- При итерации по значениям: находим значения с ограничением по глубине
  - Начинаем с $V_0(s) = 0$
  - По заданному V_k, вычисляем значения на шаге k+1 для всех состояний:

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

- Q-ценности более полезны, поэтому лучше вычислять их
  - Начинаем $Q_0(s,a) = 0$
  - По заданному Q_k, вычисляем q-ценности на шаге k+1 для всех q-состояний

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

Q-обучение

- **Q-обучение - итерации по Q-ценностям с использованием выборок**

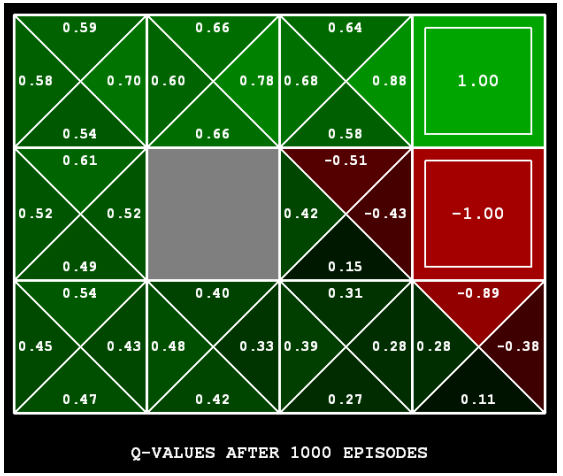
$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

- **Шаги Q-обучения**:
  - Получить выборку *(s,a,s',r)*
  - Определить предыдущую оценку q-состояния *(s,a)*: *Q(s,a)*
  - Получить новую выборочную оценку:

$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

  - Выполнить обновление оценки q-состояния на основе скользящего усреднения выборки:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\,[sample]$$



Избавляет от оценивания политики

Q-обучение

Q-обучение — это очень простой TD-алгоритм, широко применяемый на практике.
В *Q*-обучении интерес представляет ценность пар «состояние-действие», т.е. эффект от выполнения действия *a* в состоянии *s*.

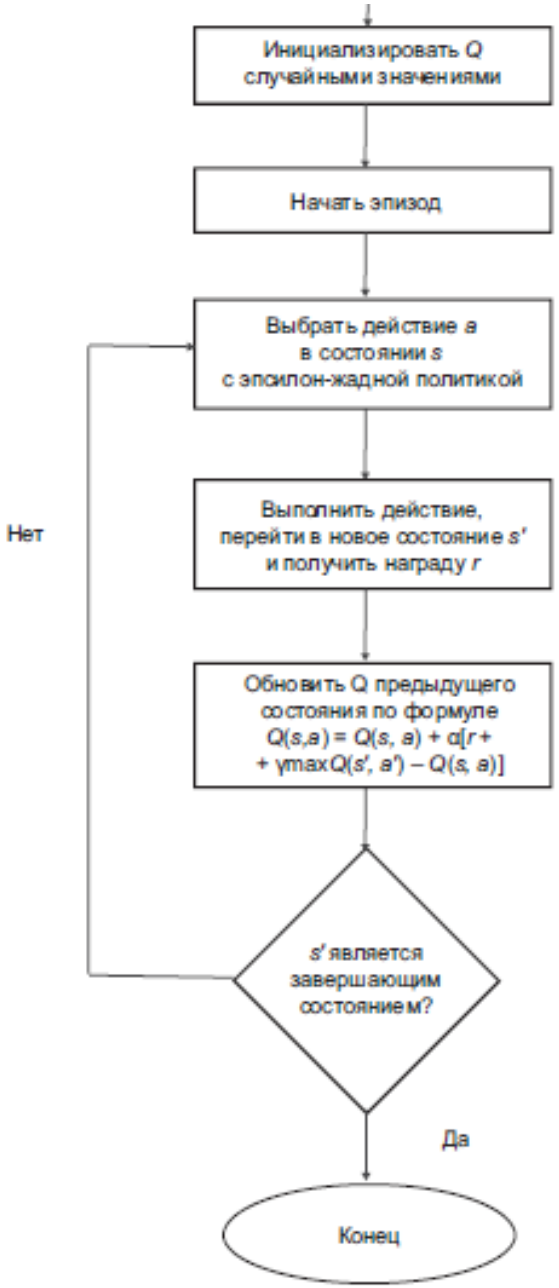
Значение *Q* обновляется в соответствии с правилом, которое связано аппроксимацией матожидания в уравнении Беллмана для Q-функции скользящим средним:

$$Q(s, a) := Q(s, a) + \alpha(r + \gamma \max Q(s', a') - Q(s, a)). \qquad (1)$$

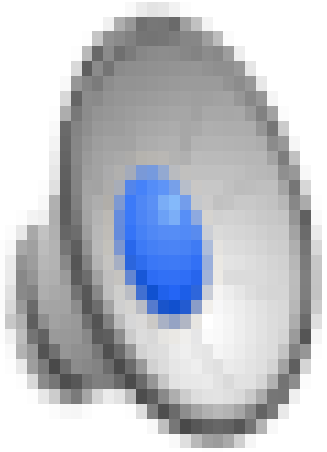
Алгоритм Q-обучения:

1. Инициализация *Q*-ценности произвольными значениями.
2. Выбрать действие в состоянии *s* с использованием эпсилон-жадной стратегии ($\varepsilon > 0$), получить награду r и перейти в новое состояние *s'*.
3. Обновить предыдущую ценность *q*-состояния по правилу (1).
4. Повторять шаги 2 и 3 до достижения завершающего состояния.

Q-обучение


Инициализировать Q
случайными значениями
Начать эпизод
Выбрать действие a
в состоянии s
с эпсилон-жадной политикой
Выполнить действие,
перейти в новое состояние s'
и получить награду r
Обновить Q предыдущего
состояния по формуле
Q(s,a) = Q(s, a) + α[r +
+ γmaxQ(s', a') – Q(s, a)]
s' является
завершающим
состоянием?
Нет
Да
Конец

Q-обучение в мире Gridworld



Свойства Q-обучения

- **Впечатляющий результат**: Q-обучение сходится к оптимальным значениям, т.е. обучается оптимальным q-значениям – даже если агент действует не всегда оптимально!
Это делает Q-обучение таким революционным !

- Q-обучение относится **к off-policy обучению, т.е. к обучению без привязки к политике** (в то время как TD-обучение и прямая оценка изучают ценности состояний в рамках политики, следуя политике, прежде чем определять оптимальность политики с помощью других методов).

- Предостережения:
  - Необходимо, чтобы объем исследований был достаточным
  - Скорость обучения должна стать достаточно малой
  - … но она не должна снижаться слишком быстро

  По сути, в пределе, не имеет значения как выбираются действия!