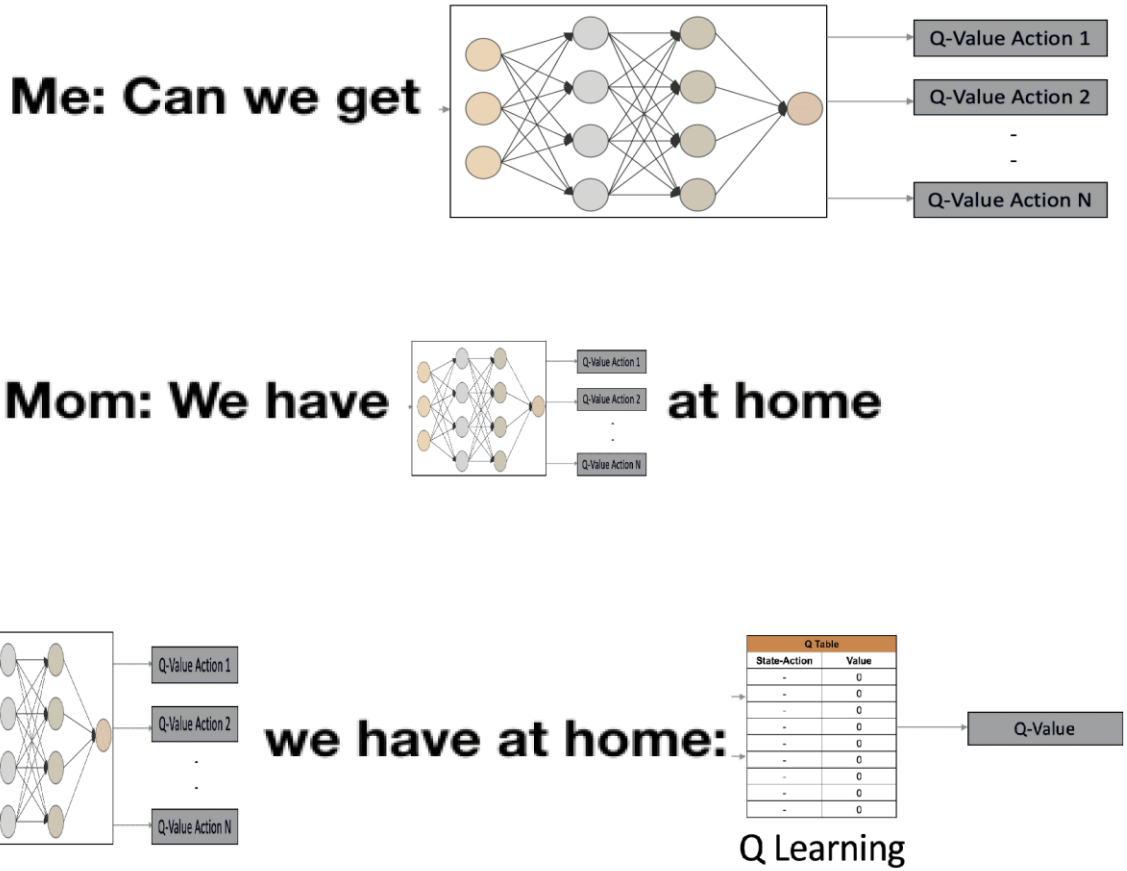
Value-based методы
Часть 1. Табличные методы

Разворотнев Иван 18.06.2019

Tinkoff.ru

Табличные методы обучения с подкреплением



Оценка вознаграждения

$$R_t = \sum_{k=0}^{T} r_{t+k+1}$$ *- ожидаемая выгода (expected reward)*

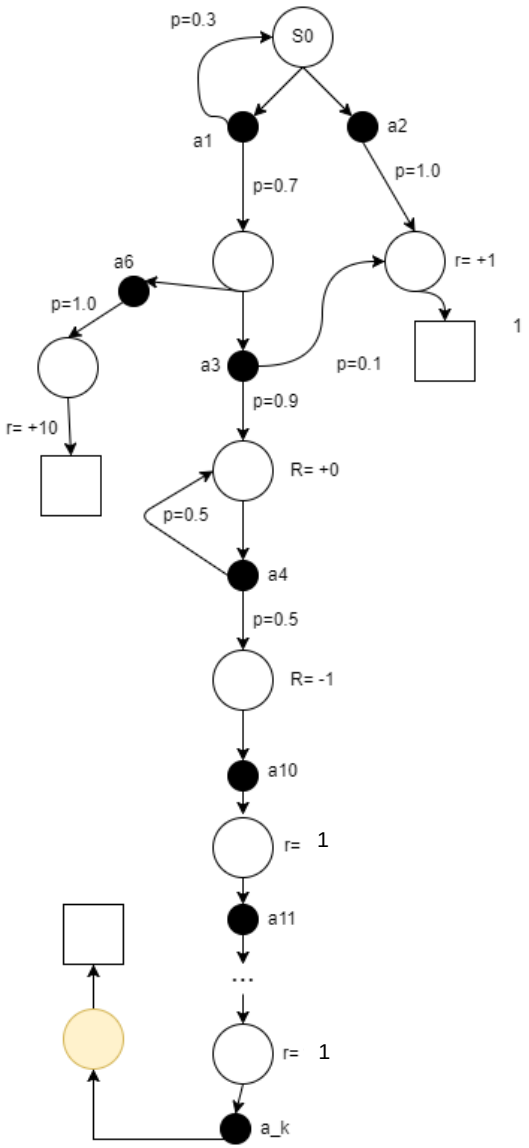
$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$$ *- дисконтированная выгода (discounted reward)*

$\gamma \in [0,1]$

Упражнение. Подбери гамму

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$$

$\gamma - ?$



Ценность состояния и действия

Функция ценности состояния s при стратегии π:

$$V_\pi(s) = \mathbb{E}_\pi(R_t|s_t = s) = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}\,\middle|\,s_t = s\right)$$

Функция ценности действия a в состоянии s при стратегии π:

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

Уравнение Беллмана

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left(r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+2} \,\middle|\, s_t = s\right) = \mathbb{E}_{\pi}\left(r_{t+1} + \gamma \mathbb{E}_{p(s' \mid s,a)} V^{\pi}(s') \,\middle|\, s_t = s\right)$$

$$= \sum_{a} \pi(a|s) \sum_{s'} \sum_{r} p(s', r|s, a)[r + \gamma V^{\pi}(s')] \quad \textit{- уравнение Беллмана для V}$$

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}\left(r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+2} \,\middle|\, s_t = s, a_t = a\right)$$

$$= \sum_{s'} \sum_{r} p(s', r|s, a)[r + \gamma V^{\pi}(s')] \quad \textit{- уравнение Беллмана для Q}$$

$$V_{\pi}(s) = \sum_{a} \pi(a|s) Q_{\pi}(s, a) \quad \textit{- Связь V и Q}$$

Оптимальность уравнения Беллмана

$\pi > \pi'$ когда $v_\pi > v_{\pi'}, \forall s \in S$

π_* -*оптимальная политика*, если $\pi_* = argmax_\pi V^\pi(s)$

$$V^*(s) = \max_a \left[r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} V^*(s') \right] \quad - \text{условие оптимальности } V^*$$

$$Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \max_{a'} Q^*(s',a') \quad - \text{условие оптимальности для } Q^*$$

$$\pi^*(a \mid s) = \operatorname*{argmax}_a Q^*(s,a)$$

Методы оценки Q и V функций

В табличных методах храним V и Q функции в виде таблиц

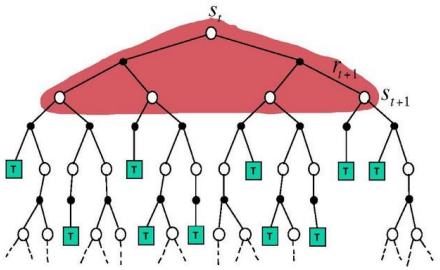
$$Q = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \text{Action} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

Табличные методы применимы когда S и A конечно и не велико

Или схемы методов аппроксимации V

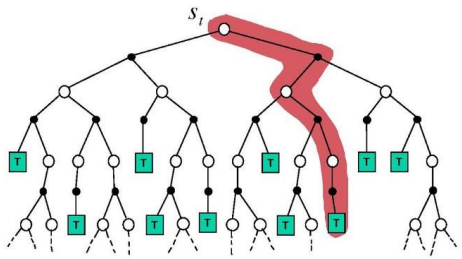
Dynamic Programming Backup

$$V(S_t) \leftarrow \mathbb{E}_\pi [R_{t+1} + \gamma V(S_{t+1})]$$



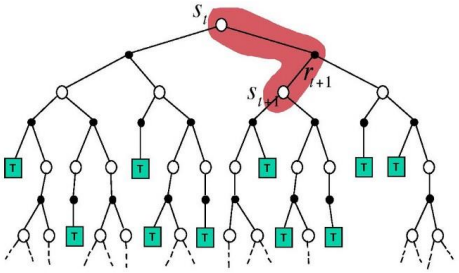
Monte-Carlo Backup

$$V(S_t) \leftarrow V(S_t) + \alpha (G_t - V(S_t))$$



Temporal-Difference Backup

$$V(S_t) \leftarrow V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$



Динамическое программирование

Принцип

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma V_\pi(s')\right]$$

На каждом шаге берем значение V(s') из таблицы V:

$$V_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma V_k(s')\right]$$

Метод итерации по ценностям:

$$V_{k+1}(s) = \max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma V_k(s')\right]$$

Выводим стратегию:

$$\pi(s) = arg\max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma V_k(s')\right]$$

Динамическое программирование

Алгоритм:

Инициализировать:

$V(s) \leftarrow 0$ для всех s

$\pi \leftarrow$ оцениваемая стратегия

Повторять:

$\triangle \leftarrow 0$

Для каждого $s \in S$:

$v \leftarrow V(s)$

$$V(s) \leftarrow \sum_{a} \pi(s,a) \sum_{a'} p(s'|s,a)[r(s,a) + \gamma V(s')]$$

$\triangle \leftarrow \max(\triangle, |v - V(s)|$

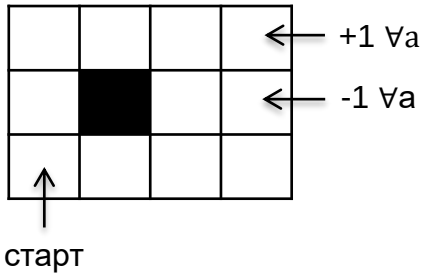
Пока $\triangle > \theta$ (малое положительное число)

Выход $V \approx V^{\pi}$

Рассмотрим тривиальный пример

Найти V

$$V_{k+1}(s) = \max_a [r(s,a) + \gamma V_k(s')]$$



V_0:

0	0	0	0
0		0	0
0	0	0	0

V_1:

0	0	0	1
0		0	-1
0	0	0	0

V_2:

0	0	γ	1
0		0	-1
0	0	0	0

V_3:

0	γ^2	γ	1
0		γ^2	-1
0	0	0	0

...

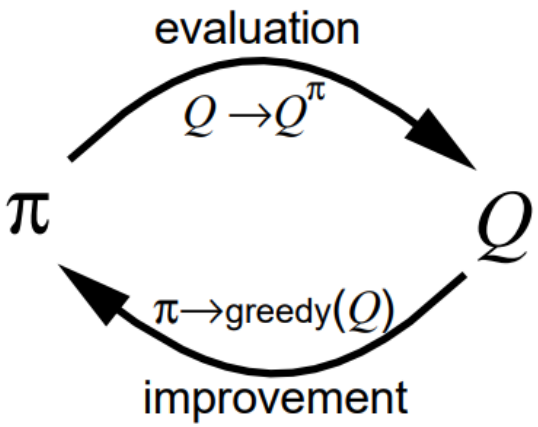
γ^3	γ^2	γ	1
γ^4		γ^2	-1
γ^5	γ^4	γ^3	γ^4

Недостатки динамического программирования

- Требуется знание динамики среды $p(s', r|s, a)$
- On-policy – обучение только на собственных траекториях
- Требует большого числа проходов по всем состояниям

Монте-Карло обучение

Принцип

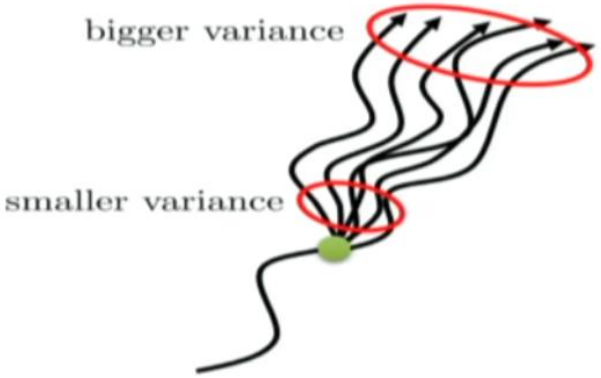


$$\pi(s) = argmax_a\, q(s,a)$$

Достоинства

- Оценка V и Q функции напрямую
- Нет смещения

Недостатки



Монте-Карло обучение

Алгоритм:

Инициализировать:

$V \leftarrow$ произвольная функция ценности состояний

$\pi \leftarrow$ оцениваемая стратегия

Returns(s) $\leftarrow$ оцениваемая стратегия

Повторять циклически:

(а) Сформулировать эпизод, используя π

(б) Для каждого состояния s, появляющегося в эпизоде

R $\leftarrow$выгода, следующая за первым посещением s

Добавить R к $Returns(s)$

$V(s) \leftarrow$ среднее $(Returns(s))$

Монте-Карло Метод первого посещения для V

TD-методы

TD(Temporal-Difference) – обучение на основе временных различий

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma V_\pi(s')\right]$$

Рассмотрим эпизод $s_t \to a_t \to r_t \to s_{t+1}$

Можем посчитать $V(s_t)$ и $r_t + \gamma V(s_{t+1})$

Ошибка $\left(r_t + \gamma V(s_{t+1}) - V(s_t)\right)^2$

$V(s_t) = V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$ - градиентный спуск

α – шаг обучения

TD-методы

Алгоритм:

Инициализируем таблицу $V(s)$

Для каждого эпизода:

Для каждого шага эпизода:

a *– действие согласно* π *для* s

Выполнить a*, получить* r, s'

$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$

$s \leftarrow s'$

Преимущества:

- Не требует завершения эпизода для оценки
- Требует меньше проходов, чем ДП и Монте-Карло
- Может быть Off-policy(Q-learning)
- Не требует знания динамики среды

TD-методы. SARSA

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$$

Алгоритм:

Инициализируем таблицу $V(s)$
Для каждого эпизода:
    Для каждого шага эпизода:
        найти a по s используя π, полученную из Q для s
        Выполнить a, получить r, s'
        Найти a' по s', используя π, полученную из Q
        $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$
        $s \leftarrow s', a \leftarrow a'$

On-policy метод(учится только на своих траекториях)

TD-методы. Q-learning

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

Алгоритм:

Инициализируем таблицу $Q(s,a)$
Для каждого эпизода:
    Для каждого шага эпизода:
        найти a по s используя π, полученную из Q для s
        Выполнить a, получить r, s'
        $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
        $s \leftarrow s', a \leftarrow a'$

Off-policy метод(учится не только на своих траекториях)

Exploration-exploitation dilemma

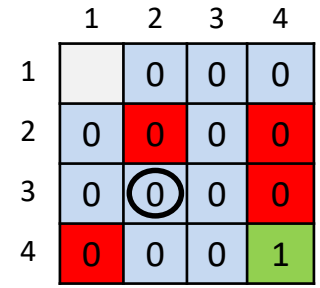
$\boldsymbol{\varepsilon}$ – жадная стратегия:

$$\pi(a \mid s) = \begin{cases} \underset{a}{\operatorname{argmax}} Q(s,a) & \text{с вероятностью } 1-\varepsilon \\ \forall a \neq \underset{a}{\operatorname{argmax}} Q(s,a) & \text{с вероятностью } \dfrac{\varepsilon}{|A|-1} \end{cases}$$

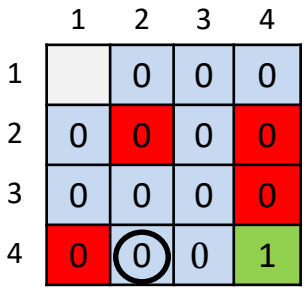
Пример. Замерзшее озеро

Цель: Дойти от белой до зеленой клетки, минуя красные $\alpha = 0.5,\ \varepsilon = 0.0, \gamma = 1$

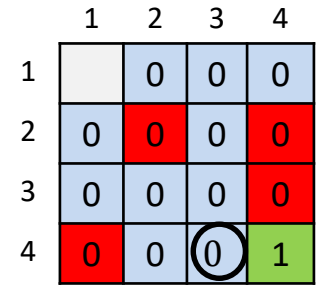
1.



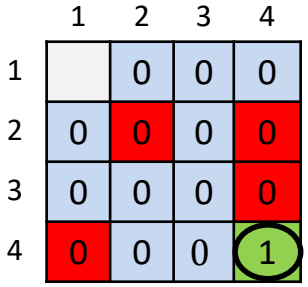
2.



3.



4.



s	a	V
(3,2)	L	0.2
(3,2)	R	0.4
(3,2)	D	0.65
(3,2)	T	0.2
(4,2)	U	0.5
(4,2)	L	0.0
(4,2)	R	0.8
(4,3)	U	0.3
(4,3)	L	0.5
(4,3)	R	0.95

1. Q((3,2), D) = 0.6 + 0.5(0 + 1 max[0, 0.5, 0.7] - 0.6) = 0.65

2. Q((4,2), R) = 0.7 + 0.5(0 + 1 max[0.3, 0.5, 0.9] - 0.7) = 0.8

3. Q((4,3), R) = 0.9 + 0.5(1.0 - 0.9) = 0.95

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

Пример. Прогулка у пропасти

Q-learning:

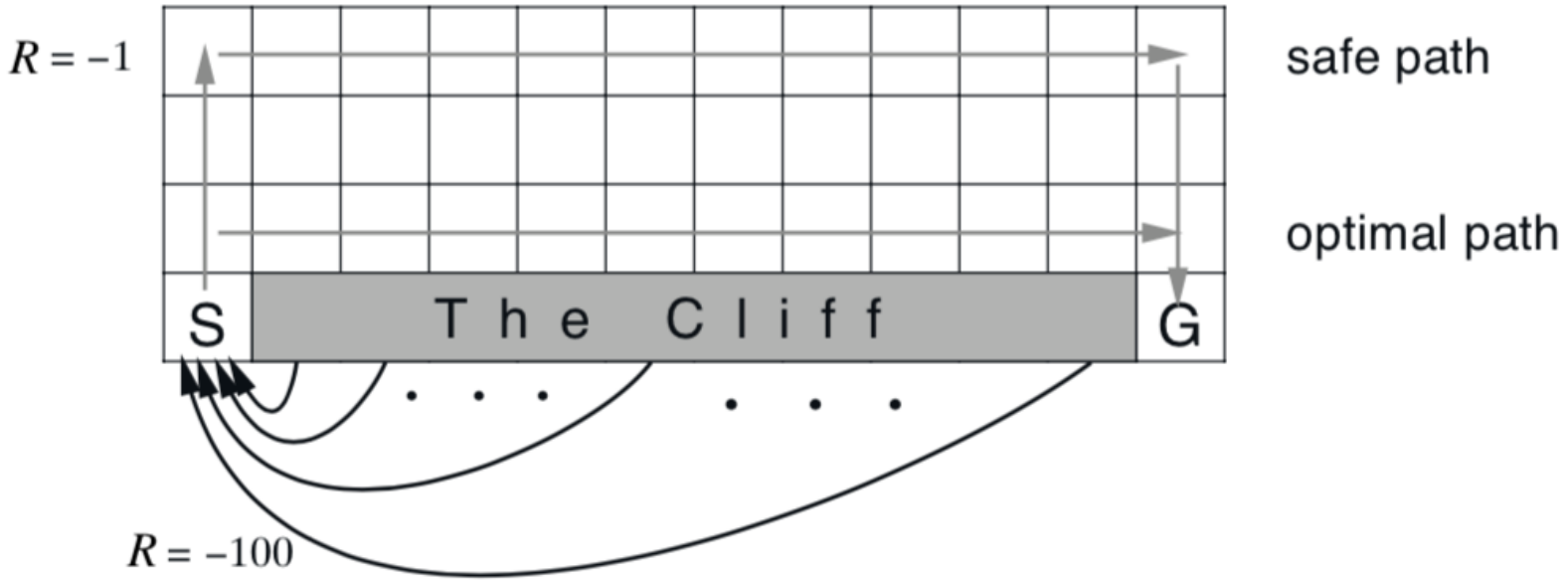
$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

SARSA:

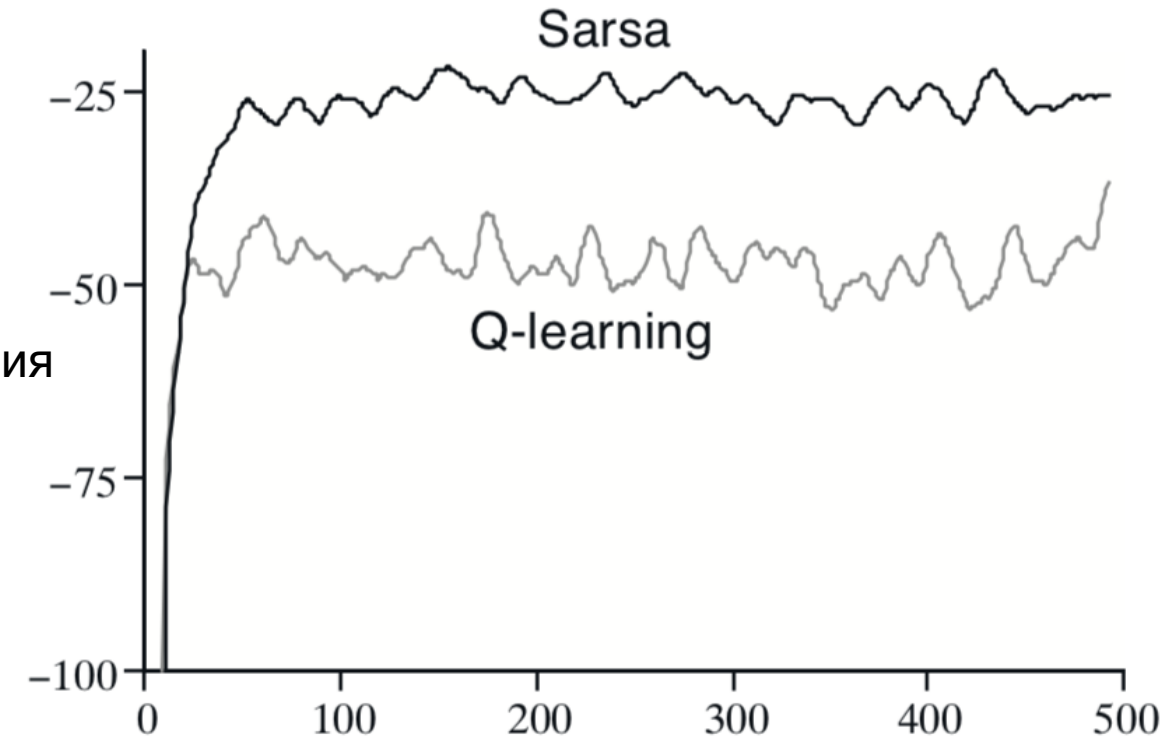
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$$

Цель:
Дойти от S до G.

Используем $\varepsilon -$ жадную стратегию $\varepsilon = 0.1$

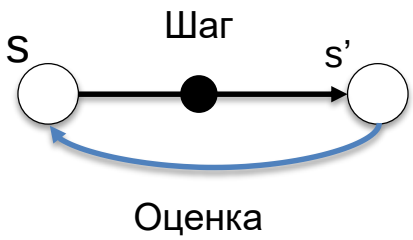


Пример. Прогулка у пропасти

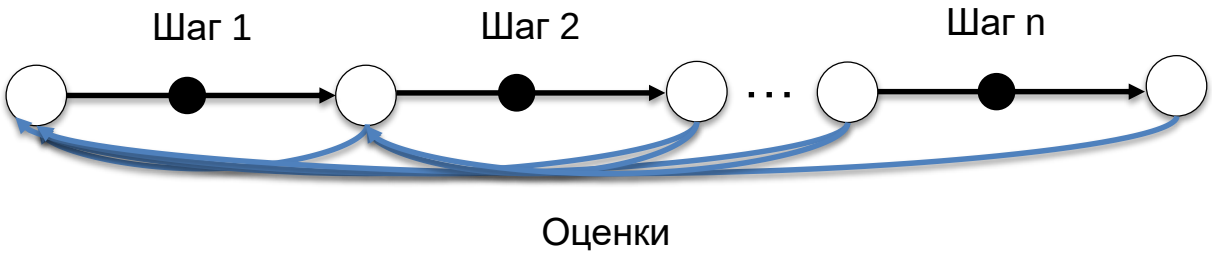


N-шаговые методы

TD:



n-шаговое TD:



λ – коэффицент приемлемости.
Это γ, для оценок в следующие моменты времени

Если хочется подробностей про RL

- основы в книге Sutton & Barto Book: Reinforcement Learning: An Introduction
  - complete draft нового издания, на английском, есть pdf
  - первое издание книги: html версия, pdf на английском, pdf на русском, можно купить бумажную книгу на русском
- online-курсы и лекции
  - лекции Дэвида Сильвера (DeepMind)
  - лекции курса CS294 (UC Berkley)
  - материалы курса ШАД по RL

Конец

Спасибо за внимание