Reinforcment Learning Recap

DOROZHKO Anton

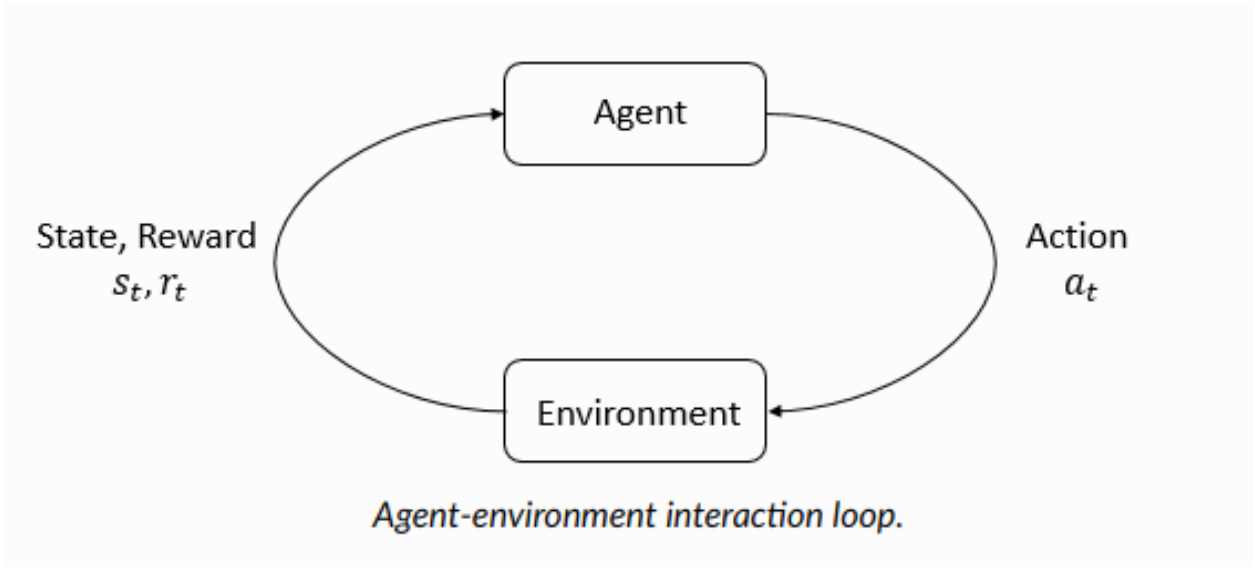
Novosibirsk State University

May 29, 2019

Outline

Novosibirsk State University
*THE REAL SCIENCE

Agent and Env



Agent-environment interaction loop.

Novosibirsk State University
*THE REAL SCIENCE

Markov Decision Process

Definition

A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a transition probability matix

$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

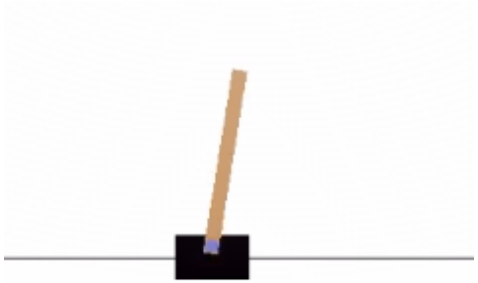
- $\mathcal{R}$ is a reward function :

$$\mathcal{R}^a_s = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

- γ is a discount factor, $\gamma \in [0, 1]$

States space

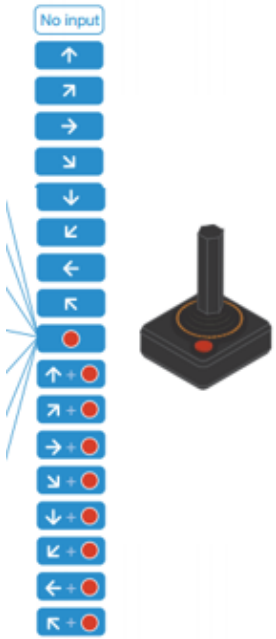
- Taxi-v2 : number from 0 to 499
- CartPoleEnv : [cart position, cart velocity, pole angle, pole velocity] Example: $[-1, 1.2, 10, -2.4]$



- ATARI Breakout : MxNx3 image

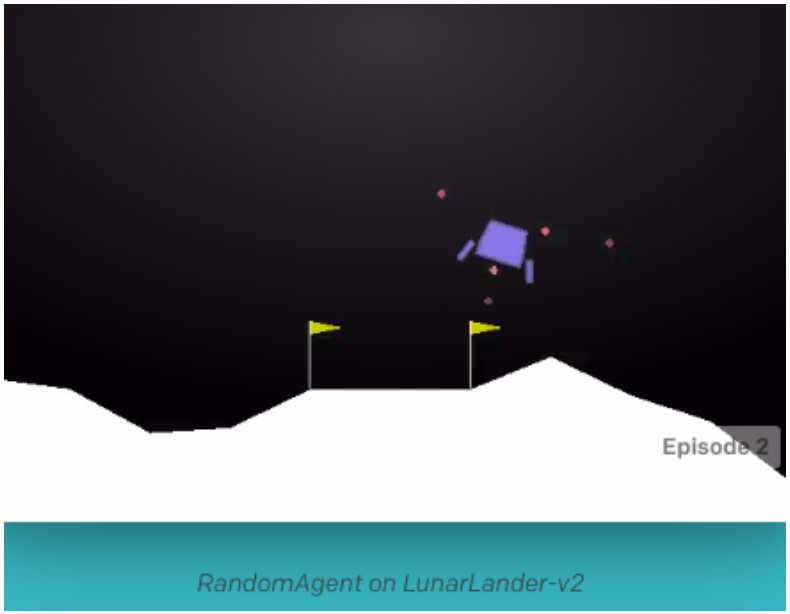
Novosibirsk State University
*THE REAL SCIENCE

Discrete Actions space



Continuous Action Space

Action is two floats
[main engine, left-right engines].



Torque force for each motor

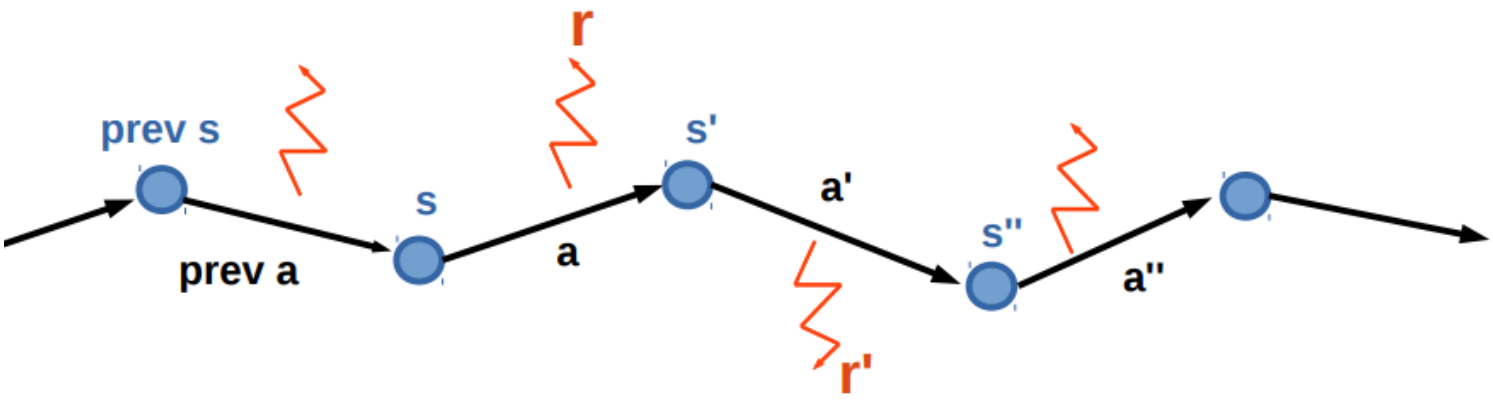
Novosibirsk State University
*THE REAL SCIENCE

Policy

- **Deterministic** $a_t = \mu(s)$
- policy[s1] $=$ a1
- **Probabilistic** $a_t \sim \pi(\cdot|s_t)$
- policy[s1] $=$ a1 : 0.9, a2 : 0.1

We want to learn policy

Trajectories



$$(s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

How to find a policy π ?

Value Function

State-value function

The state value function $V_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$$

Action-value function

The action-value function $Q_\pi(s, a)$ is expected return starting from state s, taking action a, and then following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$

Optimal Value Functions

Definition

The optimal state-value function $V_*(s)$ is the maximum value function over all policies

$$V_*(s) = \max_\pi V_\pi(s)$$

The optimal action-value function $Q_*(s, a)$ is the maximum action-value function over all policies

$$Q_*(s, a) = \max_\pi Q_\pi(s, a)$$

- If you have $Q^*(s, a)$, how you will find a policy π ?
- If you have $V^*(s, a)$, how you will find a policy π ?

Prediction / Evaluation

$$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

Monte-Carlo MC

$$V(S_t) = V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t))$$

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

Temporal Difference TD(0)

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

Novosibirsk State University
*THE REAL SCIENCE

Control

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

Monte-Carlo

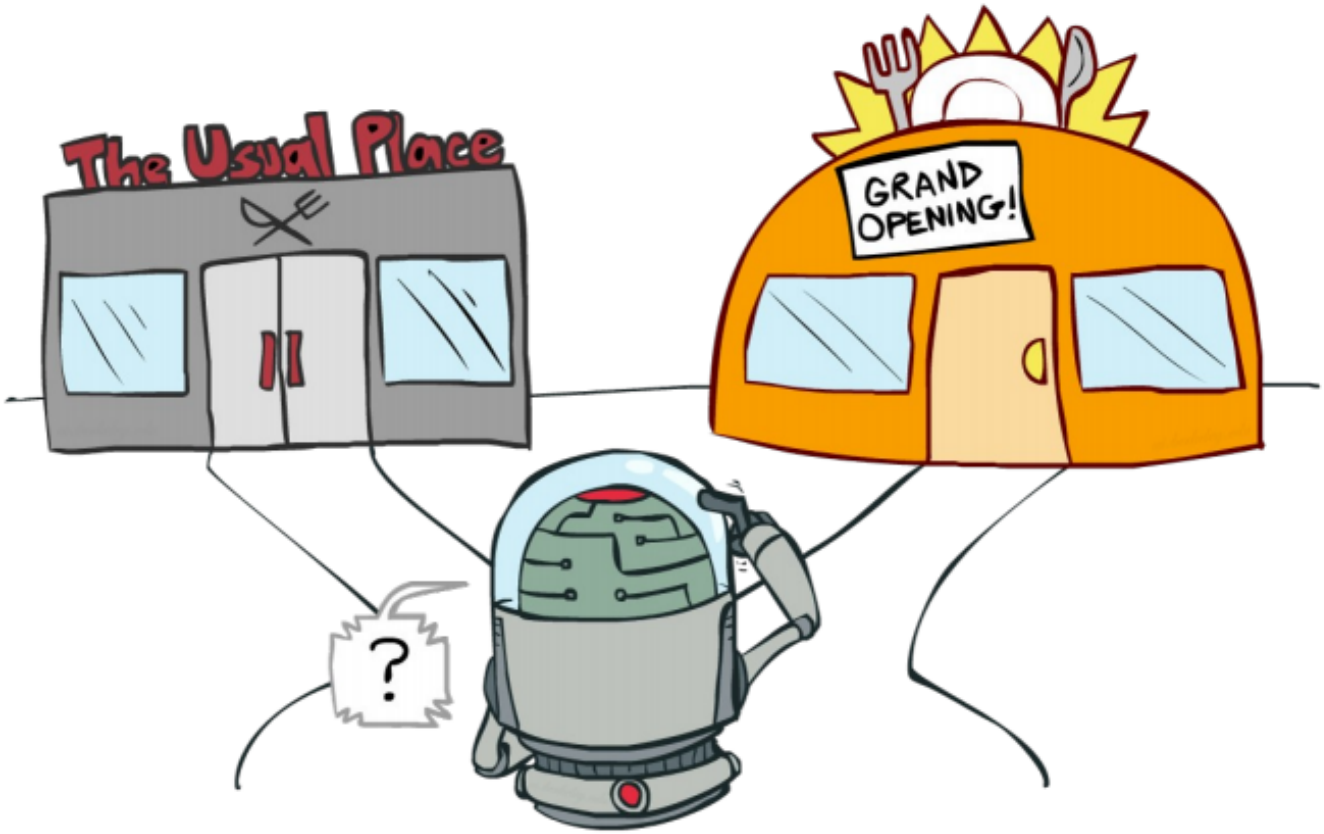
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)}(G_t - Q(S_t, A_t))$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(G_t - Q(S_t, A_t))$$

TD(0)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R + \gamma Q(S_{t+1}, A') - Q(S_t, A_t))$$

Exploration vs Exploitation



[1]CS188 http://ai.berkeley.edu/home.html

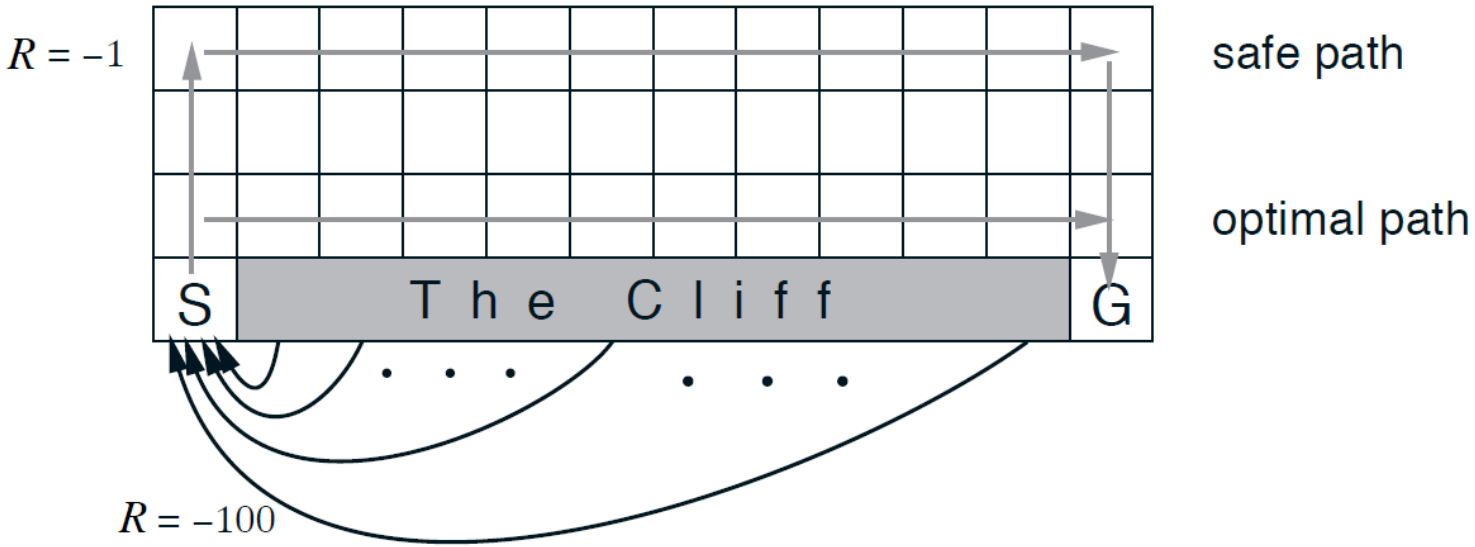
ϵ-Greedy Exploration

- Simple idea for continual exploration
- All actions are tried with $p > 0$

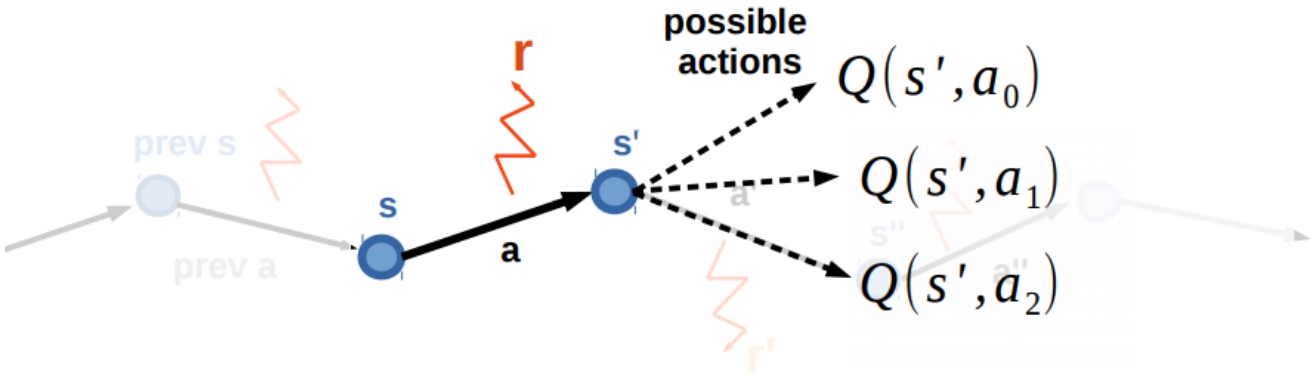
$$\pi(a|s) = \begin{cases} \epsilon/|A| + 1 - \epsilon & a^* = arg\max_{a \in A} Q(s,a) \\ \epsilon/|A| & otherwise \end{cases}$$

Cliff World

What will happen if we use ϵ-Greedy policy ?



Recap Q-learning [2]



$$\forall s \in S, \forall a \in A, Q(s,a) \leftarrow 0$$

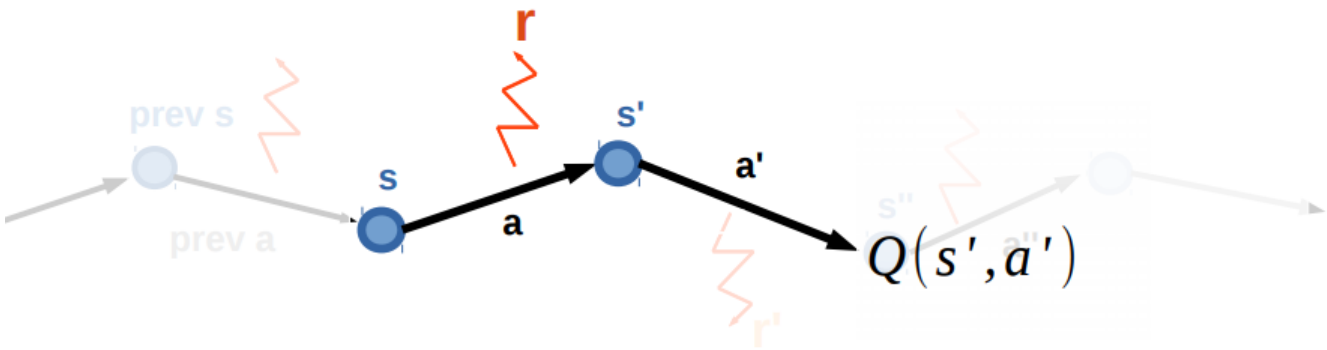
Loop:

- Sample <s,a,r,s'> from env
- Compute $\hat{Q}(s,a) = r(s,a) + \gamma \max_{a_i} Q(s',a_i)$
- Update $Q(s,a) \leftarrow \alpha \cdot \hat{Q}(s,a) + (1-\alpha) Q(s,a)$

[2] Yandex Practical RL week 3

Novosibirsk State University
THE REAL SCIENCE

Recap SARSA [3]



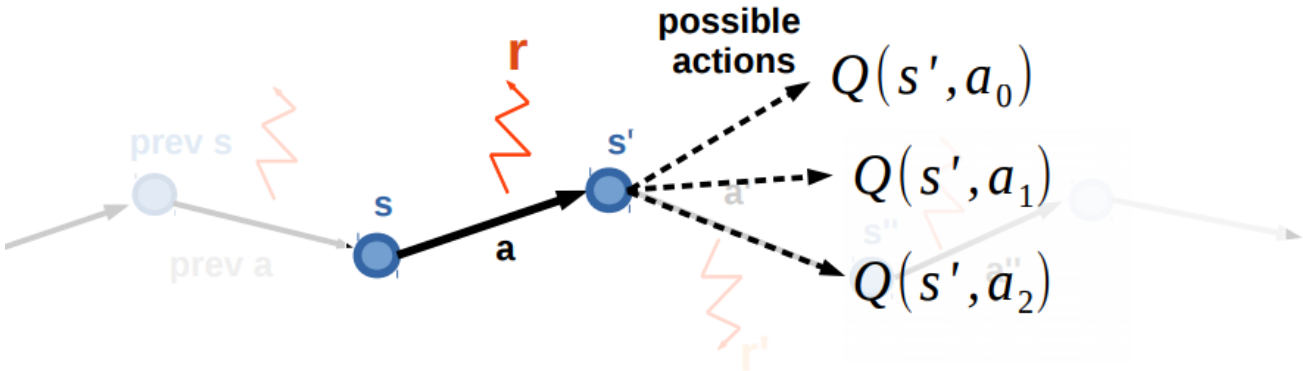
$\forall s \in S, \forall a \in A, Q(s,a) \leftarrow 0$

Loop:

- Sample <**s**,**a**,**r**,**s'**,**a'**> from env
- Compute $\hat{Q}(s,a) = r(s,a) + \gamma Q(s',a')$
- Update $Q(s,a) \leftarrow \alpha \cdot \hat{Q}(s,a) + (1-\alpha) Q(s,a)$

[3] Yandex Practical RL week 3

Recap Expected Value SARSA [4]



$\forall s \in S, \forall a \in A, Q(s,a) \leftarrow 0$

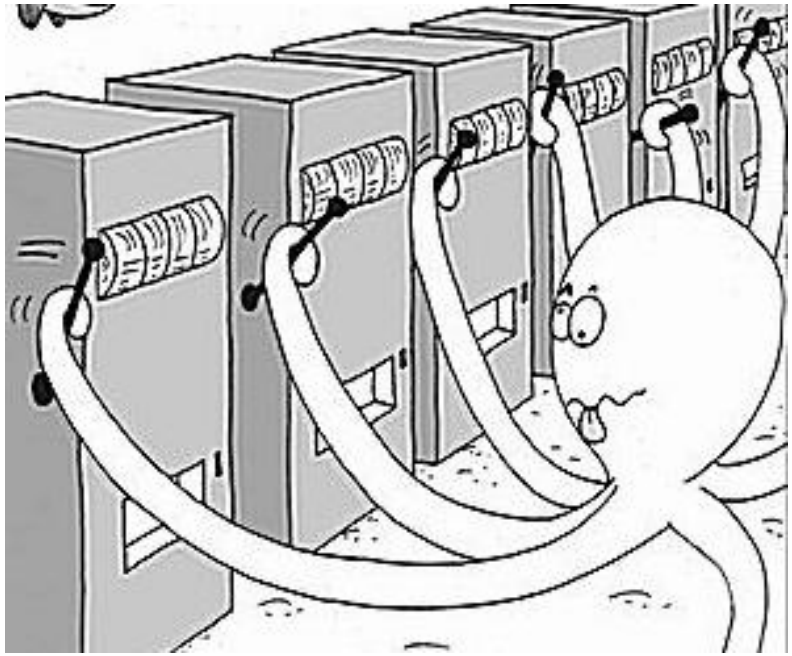
Loop:

- Sample <s,a,r,s'> from env
- Compute $\hat{Q}(s,a) = r(s,a) + \gamma \underset{a_i \sim \pi(a|s')}{E} Q(s',a_i)$
- Update $Q(s,a) \leftarrow \alpha \cdot \hat{Q}(s,a) + (1-\alpha) Q(s,a)$

[4] Yandex Practical RL week 3

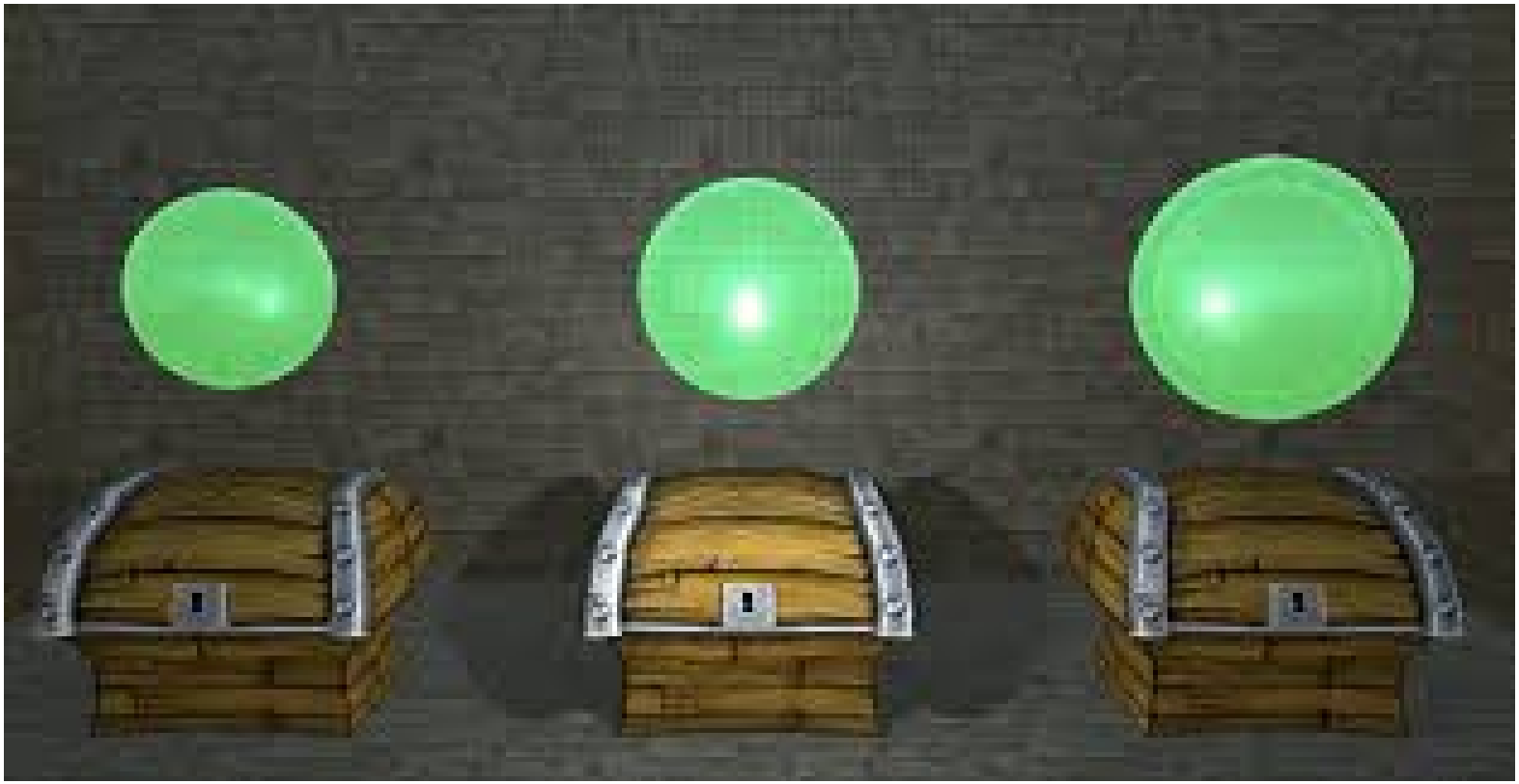
Novosibirsk State University
THE REAL SCIENCE

Multi-Armed Bandit



Novosibirsk State University
*THE REAL SCIENCE

Multi-Armed Bandit



Novosibirsk State University
*THE REAL SCIENCE

Regret minimization

Bad idea: by the sound of the name
Good idea: by $$$ it brought/lost you

Regret of policy π(a|s):
Consider an optimal policy, π*(a|s)

Regret = sum over training time [optimal – yours]

$$\eta = \sum_t \underset{s,a\sim\pi^*}{E} r(s,a) - \underset{s,a\sim\pi_t}{E} r(s,a)$$

Finite horizon: t < max_t Infinite horizon: t → inf[8]

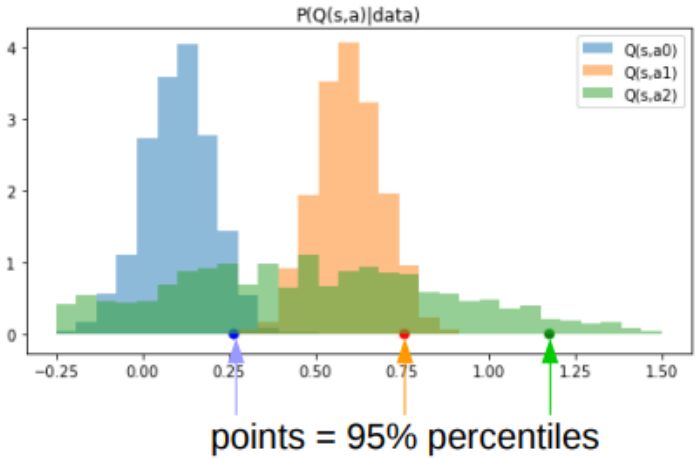
vosibirsk
State University
*THE REAL SCIENCE

- ϵ-greedy regret grows linearly
- UCB and Thompson sampling grows with $log(T)$

Novosibirsk State University
*THE REAL SCIENCE

Optimism in face of uncertainty

- Policy:
  - Compute 95% upper confidence bound *for each a*
  - Take action with highest confidence bound
  - Adjust: change 95% to more/less



Novosibirsk State University
*THE REAL SCIENCE