RL methods in nlp.

① Суть

- окружение (среда) даёт наблюдение
- агент (оболочка над политикой) выбирает действие
- среда даёт обратную связь
- всё повторяется

②

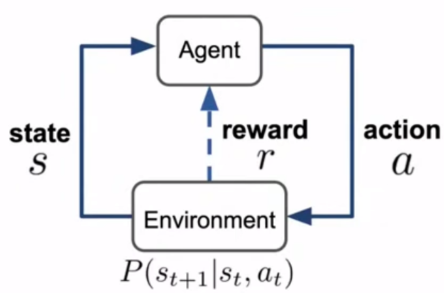
- про агента мы знаем всё, а про среду ничего
- есть дилемма:

что лучше — хорошо сейчас / хорошо потом

③ MDP

MDP formalism

- State: $s \in \mathcal{S}$
- Action: $a \in \mathcal{A}$
- Reward: $r \in \mathbb{R}$
- Dynamics: $P(s_{t+1}|s_t, a_t)$



Суть:

– текущее состояние зависит только от конечного числа предыдущих

Markov property:

$$P(s_{t+1}|s_t, a_t, \ldots, s_0, t_0) = P(s_{t+1}|s_t, a_t)$$

Total reward

- Total reward for session: $R = \sum_t r_t$
- Policy: $\pi(a|s) = P(\text{take action } a \text{ in state } s)$
- Goal: maximize reward; $\pi^*(a|s) = \arg\max_\pi \mathbb{E}_\pi[R]$

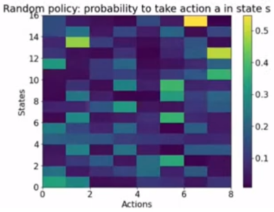
④ как оптимизироваться? (повышать общую награду за сессию)

- метод кросс-энтропии :

(конечный случай)

Cross-entropy method: tabular case

- Initialize policy (state-action matrix, every row sums up to 1)
- Sample N sessions
- Select M **elite** sessions with highest rewards
- Update policy using the elite session state-action sequences
- Repeat

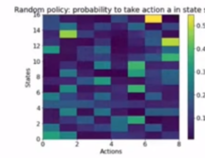


23

Cross-entropy method: tabular case

- Policy is a matrix

$$\pi(a|s) = A_{s,a}$$



- Sample N games with this policy
- Select M **elite** sessions with highest rewards

$$\text{Elite} = [(s_0, a_0), (s_1, a_1), \dots, (s_M, a_M)]$$

- Update policy: $$\pi_{\text{new}}(a|s) = \frac{\sum_{s_t, a_t \in \text{Elite}} [s_t = s][a_t = a]}{\sum_{s_t, a_t \in \text{Elite}} [s_t = s]}$$

35

Approximate cross-entropy method

- Model (e.g. parametric) predicts action probability given state:

$$\pi(a|s) = f_\theta(a, s)$$

`model = RandomForestClassifier()` Random Forest Classifier, Logistic Regression, NN etc.

- Sample N sessions, select M **elite** sessions

$$\text{Elite} = [(s_0, a_0), (s_1, a_1), \dots, (s_M, a_M)]$$

New training set; states are objects, actions are target values

- Maximize likelihood of actions in elite sessions:

$$\pi(a|s)_{\text{new}} = \arg\max_{\pi} \sum_{s_t, a_t \in \text{Elite}} \log \pi(a_i|s_i)$$

`model.fit(elite states, elite actions)`

38

Key differences

Supervised Learning	Reinforcement Learning
• Learn to approximate reference answers • Need reference answers • Model does not affect the input data	• Learn optimal strategy by trial and error • Need feedback on agent's actions • Agent actions affect the environment (so the observations)

43