

# Вводная лекция

## Обучение с подкреплением

Никита Юдин, iudin.ne@phystech.edu

Московский физико-технический институт  
Физтех-школа прикладной математики и информатики

7 февраля 2024



# Команда курса

- лектор: Никита Юдин
- ассистенты:
  - Александр Моложавенко
  - Варвара Руденко
  - Юрий Сапронов
  - Андрей Семёнов
  - Георгий Акиндинов

# Telegram беседа



Ссылка на беседу: [https://t.me/+6p7Z\\_KQqxp8yMjEy](https://t.me/+6p7Z_KQqxp8yMjEy)

## Правила игры

В курсе предусмотрено пять лабораторных работ (с жёсткими дедлайнами) в формате ноутбуков и устный экзамен. Итоговая оценка по курсу в 10-балльной шкале рассчитывается по формуле:

$$\text{Итоговая оценка} = \text{Округл.вверх} (0.3 * \text{Экз} + 0.7 * \text{Лаб})$$

Оценке «отлично» соответствует оценка 8 и выше, оценке «хорошо» — оценка [5, 8), оценке «удовлетворительно» — промежуток [3, 5).

Помимо баллов необходимо также выполнить следующие условия:

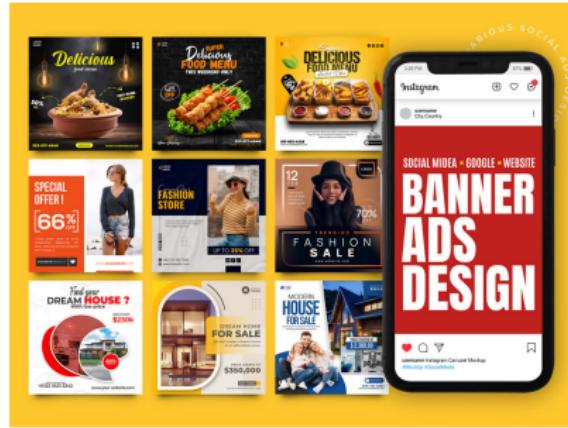
- на «отлично» — сдать 5 лабораторных работ, оценка за экзамен не меньше 5;
- на «хорошо» — сдать 4 лабораторные работы, оценка за экзамен не меньше 3;
- на «удовлетворительно» — сдать 3 лабораторные работы, оценка за экзамен не меньше 3.

## Обучение с учителем

- дано объекты и ответы  $(x, y)$
- семейство алгоритмов  $a_\theta(x) \mapsto y$
- функция потерь  $L(y, a_\theta(x))$
- найти  $\hat{\theta} \in \operatorname{Arg} \min_{\theta} \{L(y, a_\theta(x))\}$

Как раз «ответов» в обучении с подкреплением (reinforcement learning) в чистом виде нет. Зато у нас есть **среда-симулятор (environment)**, её **состояния (states)**, **действия (actions)**, влияющие на состояния среды и **награды (rewards)** за совершённые действия. Основной принцип обучения **агента (стратегии)** для оптимального выбора действий — «метод проб и ошибок».

# Примеры предметных областей: онлайн реклама



Дано:

- сервис видеоХостинга;
- метаданные потоковой трансляции (баннеры, видео, клики);
- признаки, применимые для задачи обучения по прецедентам.

Хотим показывать релевантную рекламу (на странице, пользователю онлайн).

# Примеры предметных областей: робототехника



Дано:

- робот;
- неограниченное количество запчастей;
- показания датчиков.

**Хотим научить робота ходить.**

# Примеры предметных областей: видеоигровые среды



Дано:

- дешёвый симулятор;
- неограниченное количество попыток;
- фрейм игры и сопутствующая ему метаинформация.

Хотим научить бота выигрывать.

# Примеры предметных областей

- диалоговые системы
- количественные финансы (оптимизация портфеля)
- глубокое обучение: оптимизация негладких функций, поиск оптимальной архитектуры нейронной сети
- персонализированная терапия

# Возникающие проблемы

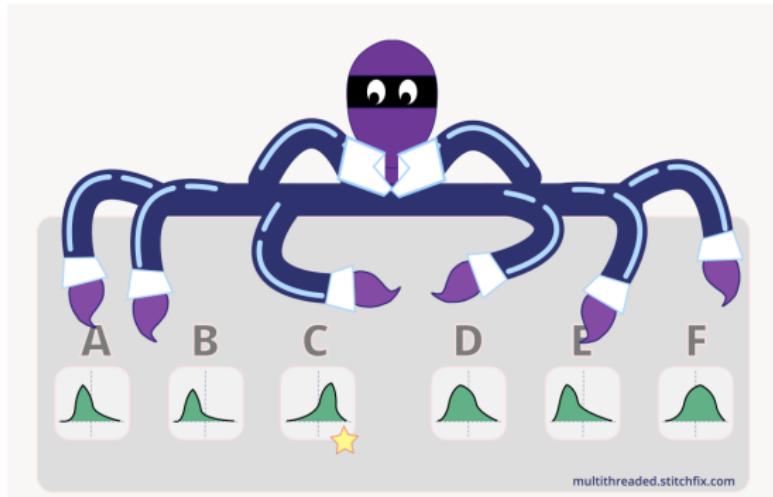
## Что значит «оптимальное действие»

- повысить доход с одного пользователя или
- сделать пользователя счастливым, чтобы он пришёл снова

# Возникающие проблемы

- следуя исключительно оптимальной в «данный момент» стратегией есть шанс никогда не найти стратегию лучше (при условии её наличия)
- например, невозможно оценить влияние других баннеров, показывая один и тот же баннер

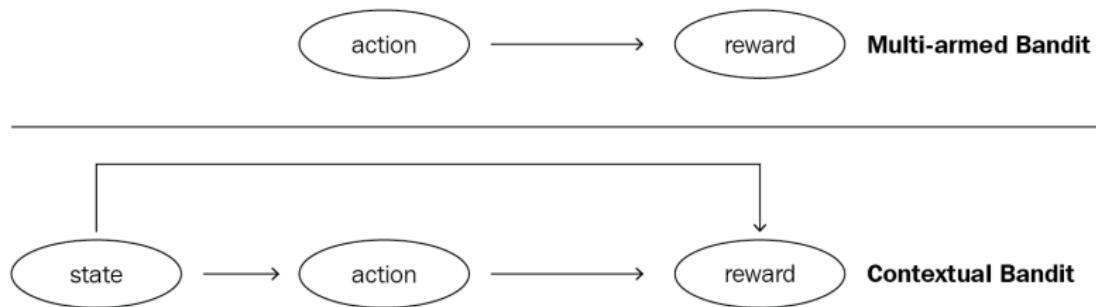
# Пример модели: многорукий бандит (multi-armed bandit)



## примеры применения

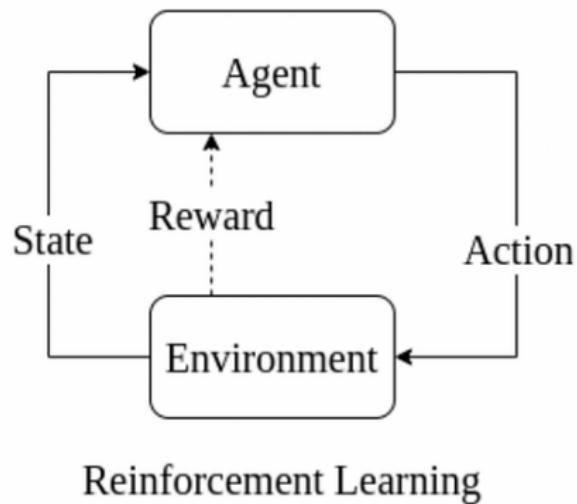
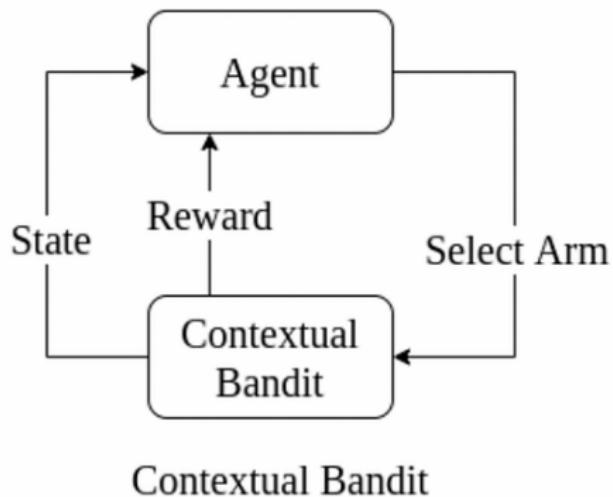
- рекламные баннеры
- системы рекомендаций
- разработка лекарств

# Пример модели: многорукий бандит (multi-armed bandit)



Основное действие: выбор «рычага» и «нажатие» на него. В контекстном бандите (contextual bandit) используется состояние бандита для хранения дополнительной информации, которая влияет на награду за действие, что приближает взаимодействие с ним к обучению с подкреплением.

# Пример модели: многорукий бандит (multi-armed bandit)



**Замечание:** над агентом у нас полный контроль, среда/бандит для нас – «чёрный ящик».

# Обучение

## с учителем

- обучаемся аппроксимировать данные ответы
- требуются корректные ответы
- настраиваемая модель не влияет на входные данные

## с подкреплением

- обучаемся оптимальной стратегии по принципу «проб и ошибок»
- требуется отклик на собственные действия агента
- агент может влиять на входные наблюдения

# Обучение

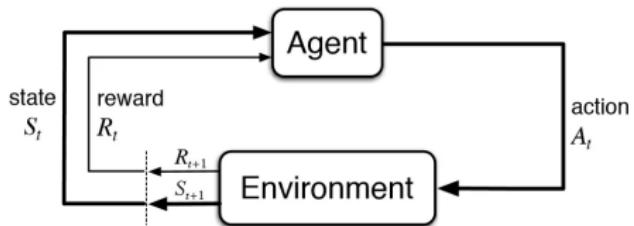
## без учителя

- учим структуру данных
- отклик не требуется
- настраиваемая модель не влияет на входные данные

## с подкреплением

- обучаемся оптимальной стратегии по принципу «проб и ошибок»
- требуется отклик на собственные действия агента
- агент может влиять на входные наблюдения

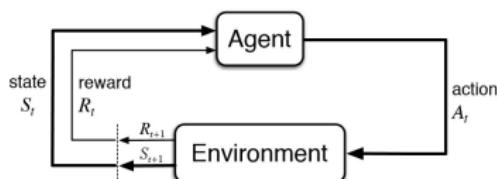
# Постановка задачи



- $s_t$  — состояние среды в момент  $t$
- $a_t$  — действие, выбранное в момент  $t$
- $r_t$  — награда, полученная в момент  $t$

- $\pi(a|s, \theta)$  — политика — вероятность совершить  $a$  из  $s$  при параметрах политики  $\theta$
- $\tau = \{(s_0, a_0, r_0), \dots, (s_T, a_T, r_T)\}$  — траектория агента согласно политике
- $\gamma \in [0, 1]$  — дисконтирующий фактор
- $p(s'|s, a)$  — вероятность попасть в состояние  $s'$  из состояния  $s$  с помощью действия  $a$

# Постановка задачи



- $p(\tau|\pi) = p(s_0) \prod_{t=0}^{\infty} \pi(a_t|s_t)p(s_{t+1}|s_t, a_t)$  — вероятность реализации траектории  $\tau$  при фиксированной политике  $\pi$
- $R := \sum_{t=0}^T \gamma^t r_t$  — суммарная награда на момент времени  $T$ , с учетом дисконтирования, на произвольной траектории
- $\mathbb{E}_{p(\tau|\theta)} R \rightarrow \max_{\theta}$  — общая оптимизационная задача,  $\theta$  — параметры  $\pi$

# Марковский процесс принятия решений

## Markov Decision Process (MDP)

Марковский процесс принятия решений — кортеж из  $(S, A, R, T, \gamma)$ , где:

$A$  — множество действий,

$S$  — множество состояний,

$R : S \times A \rightarrow R$  — функция наград,

$T : S \times A \times S \rightarrow [0, 1]$  — функция вероятности перехода

Для которого  $T(s, a, s') = \mathbb{P}(s_{t+1} = s' | S_t = s, A_t = a)$ , то есть вероятность оказаться в следующем состоянии зависит только от предыдущего состояния и действия из него.

Метод кросс-энтропии [1], [ссылка на конспект](#), стр. 32-35

Идея: сэмплированием оценить вероятность редкого события

$$\mathbb{E}_{p(x)}[s(x) \geq \hat{\gamma}] \approx \frac{1}{M} \sum_{j=1}^M [s(x_j) \geq \hat{\gamma}] \cdot \frac{p(x_j)}{q(x_j | \lambda_k)},$$

$M$  — кол-во сэмплов,  $x_j \sim q_{opt}(x | \lambda_k)$ ,  $\lambda_k$  — фиксированный параметр, такой что  $q_{opt}(x | \lambda_k) \propto [s(x) \geq \hat{\gamma}] p(x)$  (его нужно найти)

### Замечание

Полученная задача есть задача максимума правдоподобия на параметр  $\lambda_k$

## Схема метода кросс-энтропии [2]

- 1) Иниц.  $\lambda_0, M, \rho \in [0, 1]$
- 2) Сэмплирование  $x_1, \dots, x_M \in q(x|\lambda_t)$
- 3) Вычисляем редкое событие  $s_j = s(x_j)$  и сортируем их по возрастанию  $s_{(1)} \leq \dots \leq s_{(M)}$
- 4)  $\hat{\gamma}_t := \min(s_{(\lfloor \rho M \rfloor)}, \hat{\gamma})$
- 5)  $\lambda_{t+1} := \arg \max_{\lambda} \frac{1}{M} \sum_{j=1}^M [s(x_j) \geq \hat{\gamma}_t] \cdot \frac{p(x_j)}{q(x_j|\lambda_t)} \log q(x_j|\lambda)$
- 6) Критерий останова  $\hat{\gamma}_t = \hat{\gamma}$

### Получение итоговой оценки

- 1) Сэмплируем  $x_1, \dots, x_M \in q(x|\lambda_t)$
- 2) Вычисляем  $\frac{1}{M} \sum_{j=1}^M [s(x_j) \geq \hat{\gamma}_t] \cdot \frac{p(x_j)}{q(x_j|\lambda_t)}$

# Применение метода кросс-энтропии

## Вывод

Полученный метод ищет то распределение  $q(x|\lambda)$ , на котором в среднем  $s(x)$  больше некого порога чаще всего. Что является безградиентным методом решения задачи оптимизации

$$s(x) \rightarrow \max \Leftrightarrow \mathbb{E}_{p(x)}[s(x) \geq \hat{\gamma}], \hat{\gamma} \approx \max_x s(x), p(x) := q(x|\lambda)$$

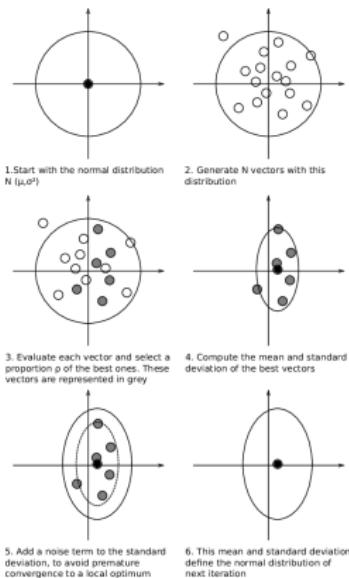
## Оптимизационный метод кросс-энтропии

1) - 3) Как и раньше

$$4) \quad \hat{\gamma}_t := s_{(\lfloor \rho M \rfloor)}$$

$$5) \quad \lambda_{t+1} := \arg \max_{\lambda} \frac{1}{M} \sum_{j=1}^M [s(x_j) \geq \hat{\gamma}_t] \log q(x_j|\lambda)$$

# Оптимизационный метод кросс-энтропии



Динамика кросс-энтропийного метода при оптимизации функции, точки-кандидаты на экстремум генерируются из нормального распределения.

# Метод кросс-энтропии в терминах RL

- 1) Иниц.  $\theta, M, \rho \in [0, 1]$
- 2) Сэмплирование  $\tau_1, \dots, \tau_M \in p(\tau|\theta)$  — получаем  $M$  траекторий по политике  $\pi(a|s, \theta)$
- 3) Вычисляем  $R_j = R(\tau_j)$  и сортируем их по возрастанию  
 $R_{(1)} \leq \dots \leq R_{(M)}$
- 4)  $\hat{\gamma}_t := R_{(\lfloor \rho M \rfloor)}$
- 5)  $\theta_{t+1} := \arg \max_{\theta} \frac{1}{M} \sum_{j=1}^M [R(\tau_j) \geq \hat{\gamma}_t] \log p(\tau_j|\theta)$ , где  
 $\log p(\tau_j|\theta) = \sum_{a,s \in \tau_j} \log \pi(a|s, \theta)$

# Метод кросс-энтропии в терминах RL

## Интерпретация алгоритма

На каждом шаге алгоритм смотрит на самые успешные, согласно награде, траектории и повышает вероятность совершения действий из этих траекторий. То есть алгоритм учится совершать действия из лучших траекторий.

## Замечание

Алгоритм почти ничего не использует из структуры задачи: как и когда начисляется награда и проч.

## Источники I

- [1] S. Ivanov, "Reinforcement learning textbook," [arXiv preprint arXiv:2201.09746](#), настольная книга по данному курсу, 2022.
- [2] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L'Ecuyer, "The cross-entropy method for optimization," in [Handbook of statistics](#), vol. 31, pp. 35–59, Elsevier, 2013.