

Обучение с подкреплением #1: Intro, Behavior Clonning, CEM

Павел Темирчев

@cydoroga

Наша команда



Артем Сорокин
@supergriver



Павел Темирчев
@cydoroga



**По всем вопросам пишите в
телеграм чат или в ЛС**

Критерии оценивания

Критерии оценивания

Домашние задания - 4 штуки:

- 2 простые по 2 балла
- 2 сложные по 4 балла

Критерии оценивания

Домашние задания - 4 штуки:

- 2 простые по 2 балла
- 2 сложные по 4 балла

Мягкий дедлайн (за неделю до жесткого):

- 1 неделя после выдачи
- 2 недели после выдачи

Каждый день после **мягкого дедлайна** снижает оценку на **0.1 \ 0.2** балла для **простых и сложных** заданий соответственно.

Критерии оценивания

Домашние задания - 4 штуки:

- 2 простые по 2 балла
- 2 сложные по 4 балла

Мягкий дедлайн (за неделю до жесткого):

- 1 неделя после выдачи
- 2 недели после выдачи

Каждый день после **мягкого дедлайна** снижает оценку на **0.1 \ 0.2** балла для **простых и сложных** заданий соответственно.

Еще будет **экзамен** на 4 балла.

Критерии оценивания

Домашние задания - 4 штуки:

- 2 простые по 2 балла
- 2 сложные по 4 балла

Мягкий дедлайн (за неделю до жесткого):

- 1 неделя после выдачи
- 2 недели после выдачи

Каждый день после **мягкого дедлайна** снижает оценку на **0.1 \ 0.2** балла для **простых и сложных** заданий соответственно.

Еще будет **экзамен** на 4 балла.

"Максимум" баллов за домашки - 8 баллов.

Все, что выше, засчитывается в сумму за экзамен - можно получить **автомат**.

Критерии оценивания

Домашние задания - 4 штуки:

- 2 простые по 2 балла
- 2 сложные по 4 балла

Мягкий дедлайн (за неделю до жесткого):

- 1 неделя после выдачи
- 2 недели после выдачи

Каждый день после **мягкого дедлайна** снижает оценку на **0.1 \ 0.2** балла для **простых и сложных** заданий соответственно.

Еще будет **экзамен** на 4 балла.

"Максимум" баллов за домашки - 8 баллов.

Все, что выше, засчитывается в сумму за экзамен - можно получить **автомат**.

$$\text{GRADE} = 10 * (0.6 * HW/8 + 0.4 * EX/4), \text{округляется до целого}$$

Критерии оценивания

Домашние задания - 4 штуки:

- 2 простые по 2 балла
- 2 сложные по 4 балла

Мягкий дедлайн (за неделю до жесткого):

- 1 неделя после выдачи
- 2 недели после выдачи

Каждый день после **мягкого дедлайна** снижает оценку на **0.1 \ 0.2** балла для **простых и сложных** заданий соответственно.

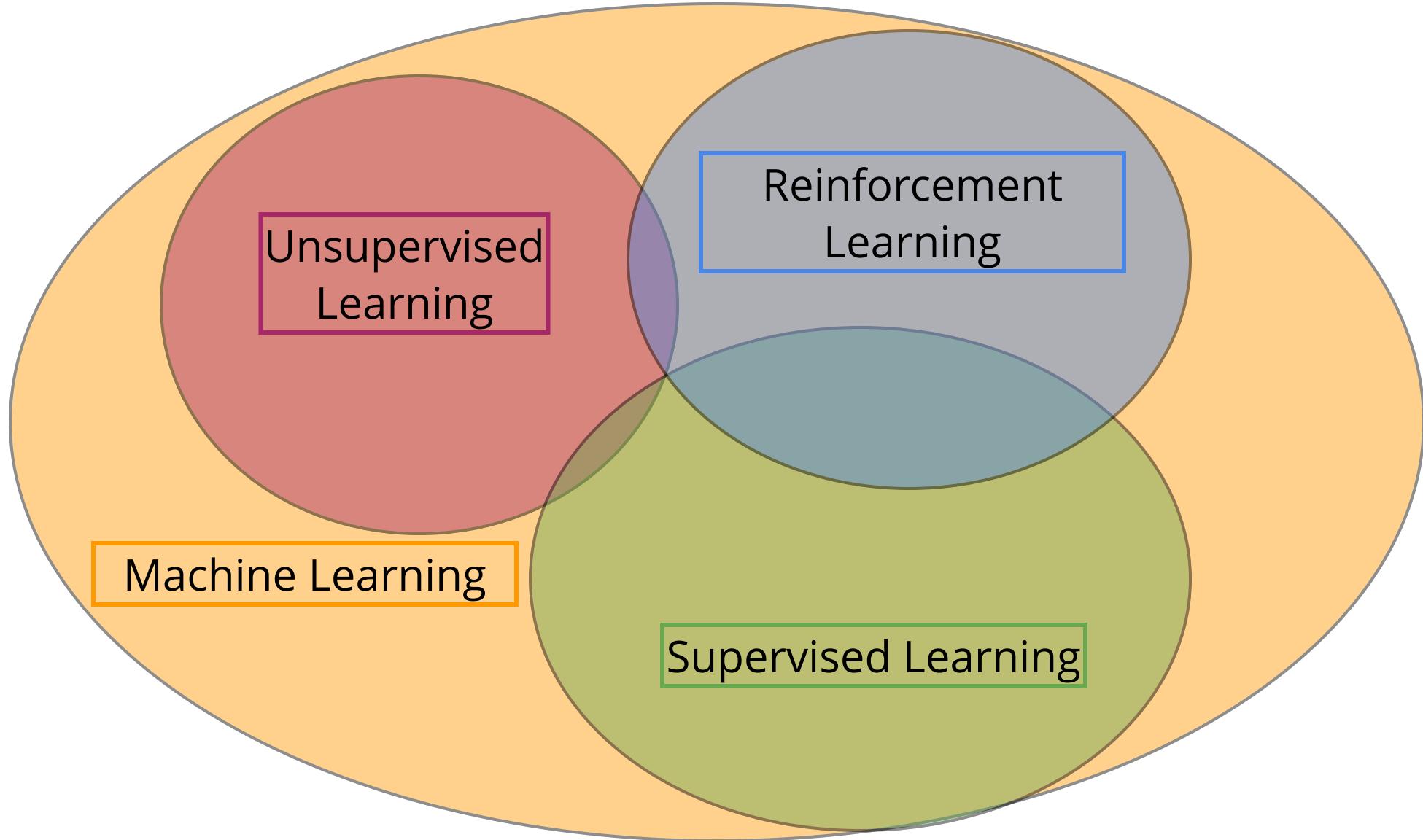
Еще будет **экзамен** на 4 балла.

Списывать домашки
строго нельзя!

"Максимум" баллов за домашки - 8 баллов.

Все, что выше, зчитывается в сумму за экзамен - можно получить **автомат**.

$$\text{GRADE} = 10 * (0.6 * HW/8 + 0.4 * EX/4), \text{округляется до целого}$$



Curricula

- Напоминалка: обучение с учителем
- Процессы принятия решений
- Принятие решений с учителем!
- Обучение с подкреплением (reinforcement learning)
- Примеры решения задач RL
- Выводы

Напоминалка: обучение с учителем

Supervised learning

x_i - изображение



y_i - метка класса



СОБАКА



КОШКА

Напоминалка: обучение с учителем

Supervised learning

x_i - изображение



y_i - метка класса



СОБАКА

Стандартное предположение:

- Правильные ответы y_i известны



КОШКА

Учитель

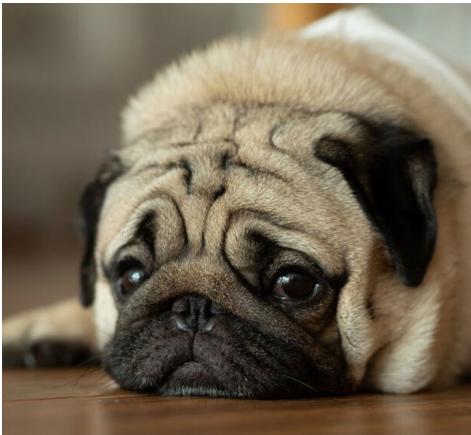
Напоминалка: обучение с учителем

Supervised learning

- Имеется выборка:

$$D := \{(x_i, y_i)\}$$

x_i - изображение



y_i - метка класса



СОБАКА

Стандартное предположение:

- Правильные ответы y_i известны



КОШКА

Учитель

Напоминалка: обучение с учителем

Supervised learning

- Имеется выборка:

$$D := \{(x_i, y_i)\}$$

- Нужно выучить отображение:

$$\hat{y}_i = f(x_i)$$

x_i - изображение



y_i - метка класса



СОБАКА

Стандартное предположение:

- Правильные ответы y_i известны



КОШКА

Учитель

Напоминалка: обучение с учителем

Supervised learning

- Имеется выборка:

$$D := \{(x_i, y_i)\}$$

- Нужно выучить отображение:

$$\hat{y}_i = f(x_i)$$

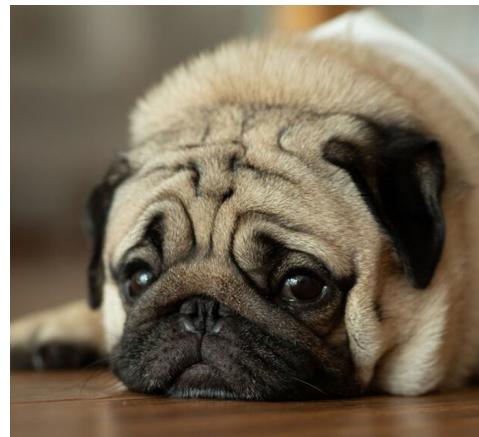
- Такое, что:

$$\hat{y}_i \approx y_i$$

Стандартное предположение:

- Правильные ответы y_i известны

x_i - изображение



y_i - метка класса



СОБАКА



КОШКА

Учитель

Процесс принятия решений

Decision Process (a self-driving car)



Процесс принятия решений

Decision Process

Наблюдения:

- Изображения с видеокамеры
- Данные с датчиков

s_i - изображение



Процесс принятия решений

Decision Process

Наблюдения:

- Изображения с видеокамеры
- Данные с датчиков

Действия:

- Газ \ тормоз
- Поворот руля

s_i - изображение



a_i - действие

ДЕЙСТВИЕ 1



ДЕЙСТВИЕ 2

Процесс принятия решений

Decision Process

Наблюдения:

- Изображения с видеокамеры
- Данные с датчиков

Действия:

- Газ \ тормоз
- Поворот руля

Цель:

- ехать по маршруту

s_i - изображение



a_i - действие

ДЕЙСТВИЕ 1



ДЕЙСТВИЕ 2

Процесс принятия решений

Decision Process

Наблюдения:

- Изображения с видеокамеры
- Данные с датчиков

Действия:

- Газ \ тормоз
- Поворот руля

Цель:

- ехать по маршруту

? кто разметит выборку ?

s_i - изображение



a_i - действие

??

ДЕЙСТВИЕ 1



??

ДЕЙСТВИЕ 2

Процесс принятия решений

Decision Process

Наблюдения:

- Изображения с видеокамеры
- Данные с датчиков

Действия:

- Газ \ тормоз
- Поворот руля

Цель:

- ехать по маршруту

? кто разметит выборку ?

!! действия влияют на наблюдения !!

s_i - изображение



a_i - действие

??

ДЕЙСТВИЕ 1



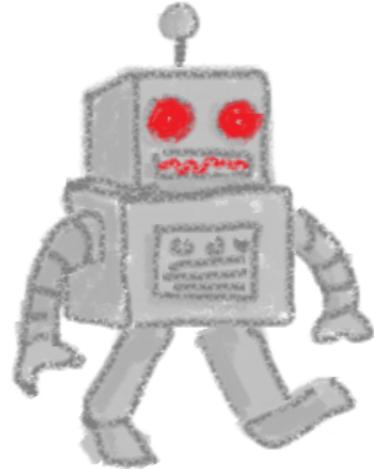
??

ДЕЙСТВИЕ 2

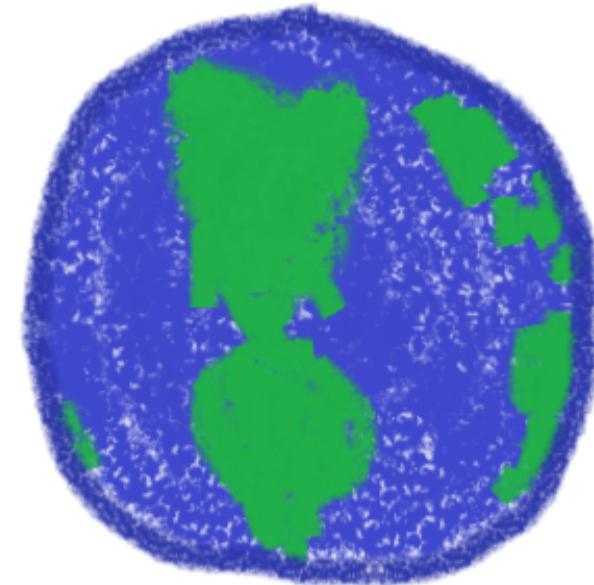
Процесс принятия решений

Decision Process

агент



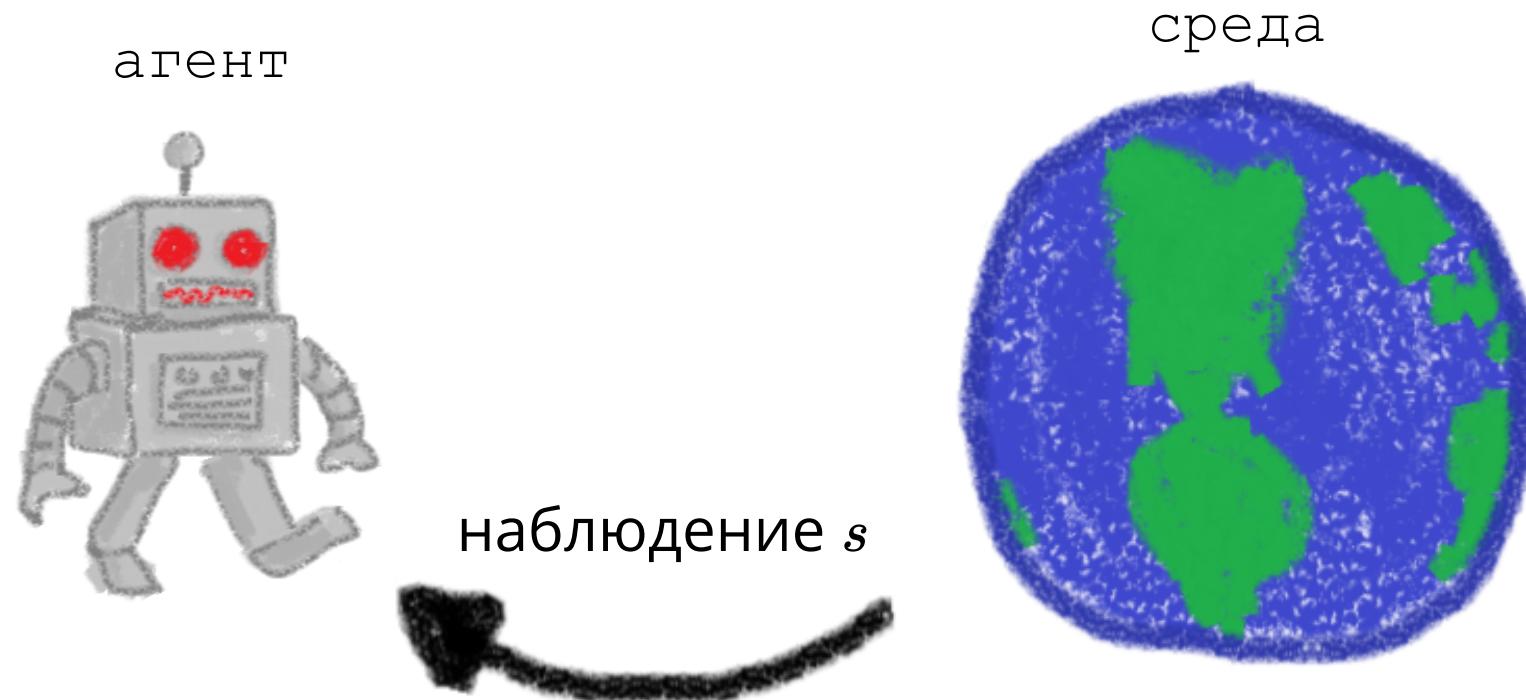
среда



Decision Process - выбор действий по наблюдениям

Процесс принятия решений

Decision Process



Decision Process - выбор действий по наблюдениям

Процесс принятия решений

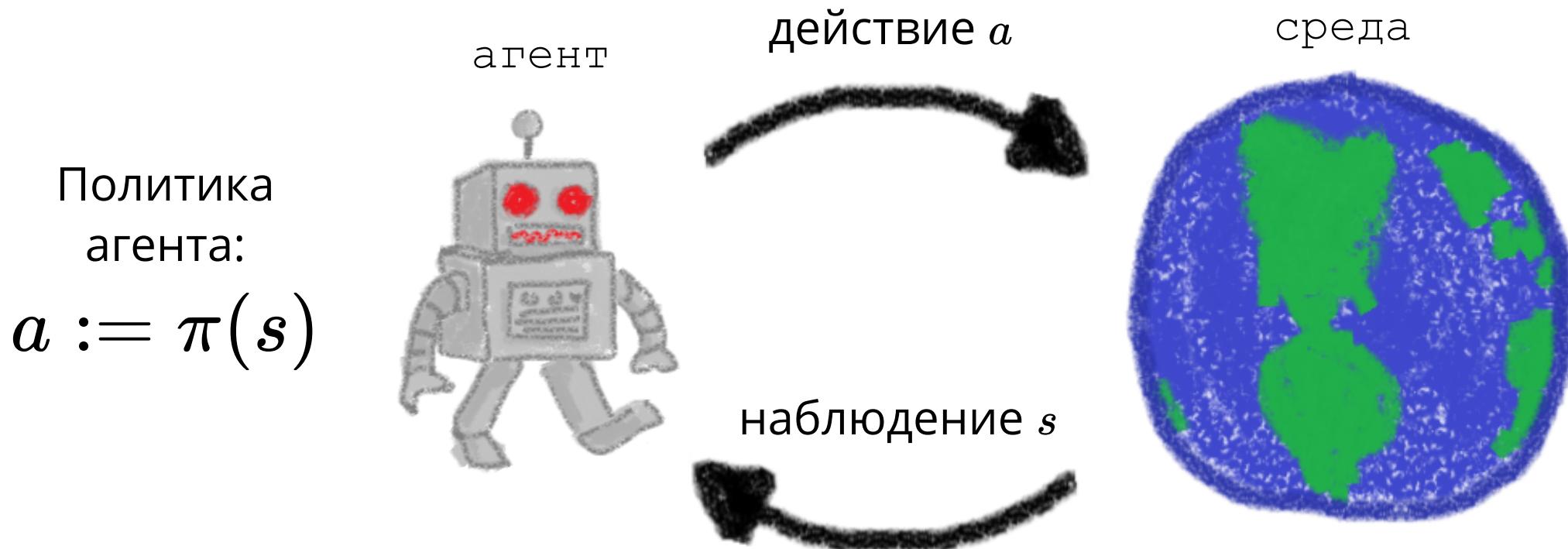
Decision Process



Decision Process - выбор действий по наблюдениям

Процесс принятия решений

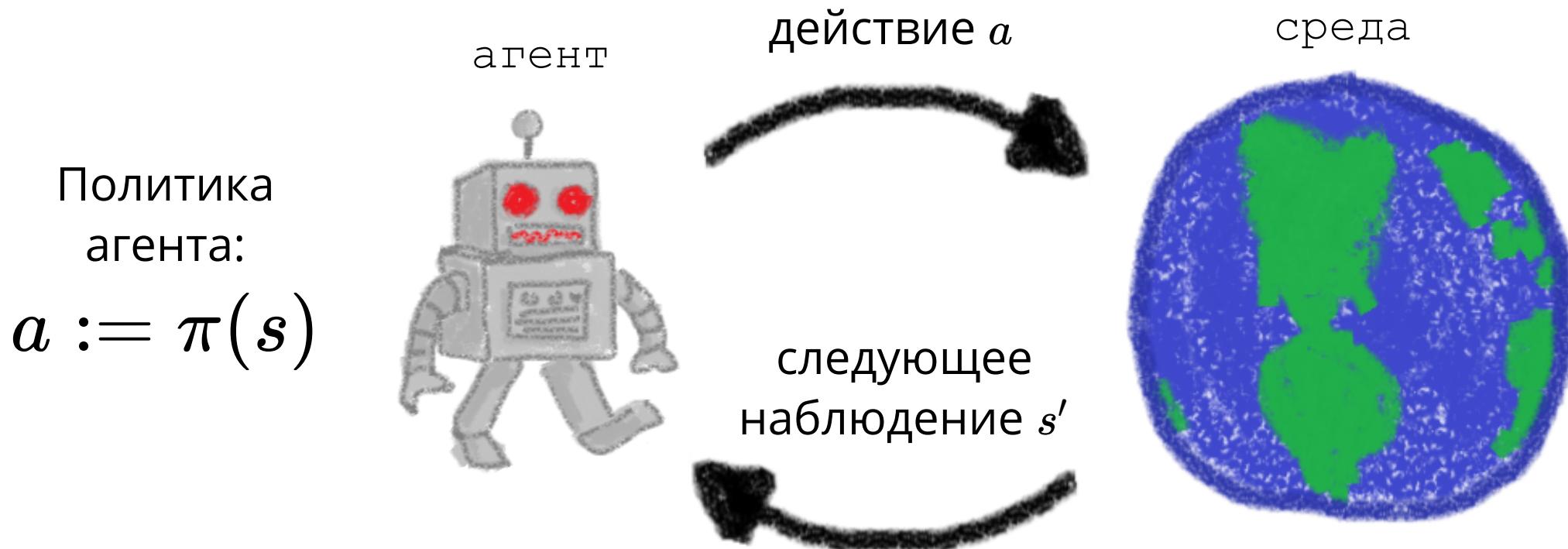
Decision Process



Decision Process - выбор действий по наблюдениям

Процесс принятия решений

Decision Process



Decision Process - выбор действий по наблюдениям

Behavior Cloning

Что, если попросить эксперта сказать, какие действия хорошие?

А затем применить обучение с учителем.

Behavior Cloning

Что, если попросить эксперта сказать, какие действия хорошие?
А затем применить обучение с учителем.

На примере self-driving cars (беспилотный автомобиль):

1. Берем хорошего водителя

Behavior Cloning

Что, если попросить эксперта сказать, какие действия хорошие?
А затем применить обучение с учителем.

На примере self-driving cars (беспилотный автомобиль):

1. Берем хорошего водителя
2. Отправляем ездить по маршруту

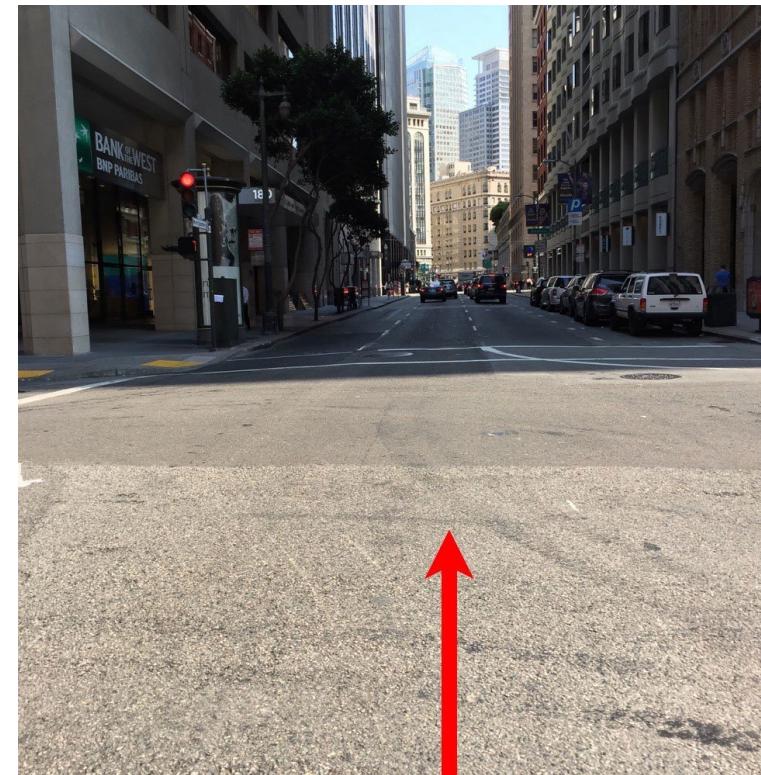
Behavior Cloning

Что, если попросить эксперта сказать, какие действия хорошие?

А затем применить обучение с учителем.

На примере self-driving cars (беспилотный автомобиль):

1. Берем хорошего водителя
2. Отправляем ездить по маршруту
3. Записываем видео его траекторий (s_i)



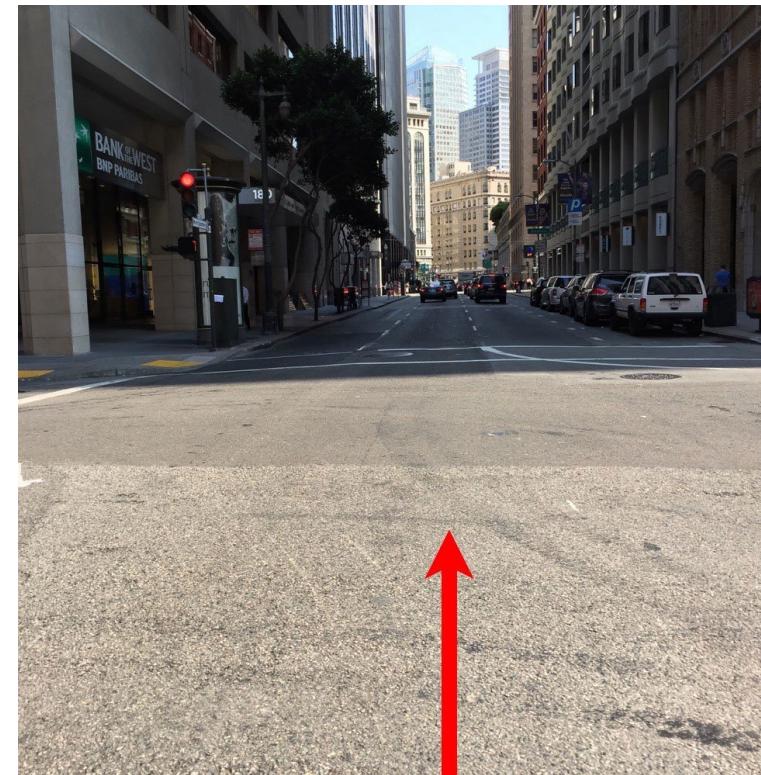
Behavior Cloning

Что, если попросить эксперта сказать, какие действия хорошие?

А затем применить обучение с учителем.

На примере self-driving cars (беспилотный автомобиль):

1. Берем хорошего водителя
2. Отправляем ездить по маршруту
3. Записываем видео его траекторий (s_i)
4. Сохраняем историю его действий (a_i)



Behavior Cloning

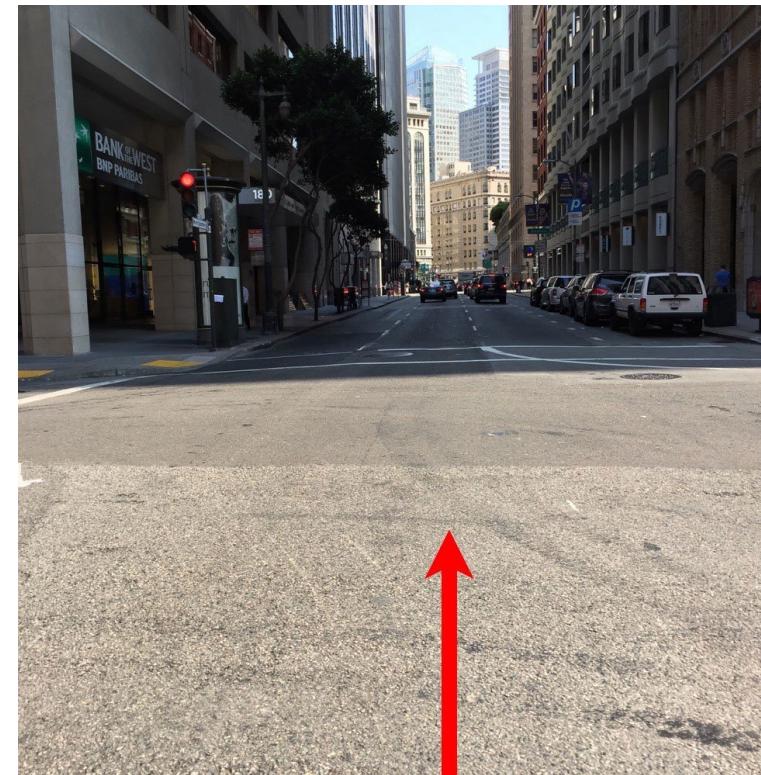
Что, если попросить эксперта сказать, какие действия хорошие?

А затем применить обучение с учителем.

На примере self-driving cars (беспилотный автомобиль):

1. Берем хорошего водителя
2. Отправляем ездить по маршруту
3. Записываем видео его траекторий (s_i)
4. Сохраняем историю его действий (a_i)
5. Обучаем ML модель π :

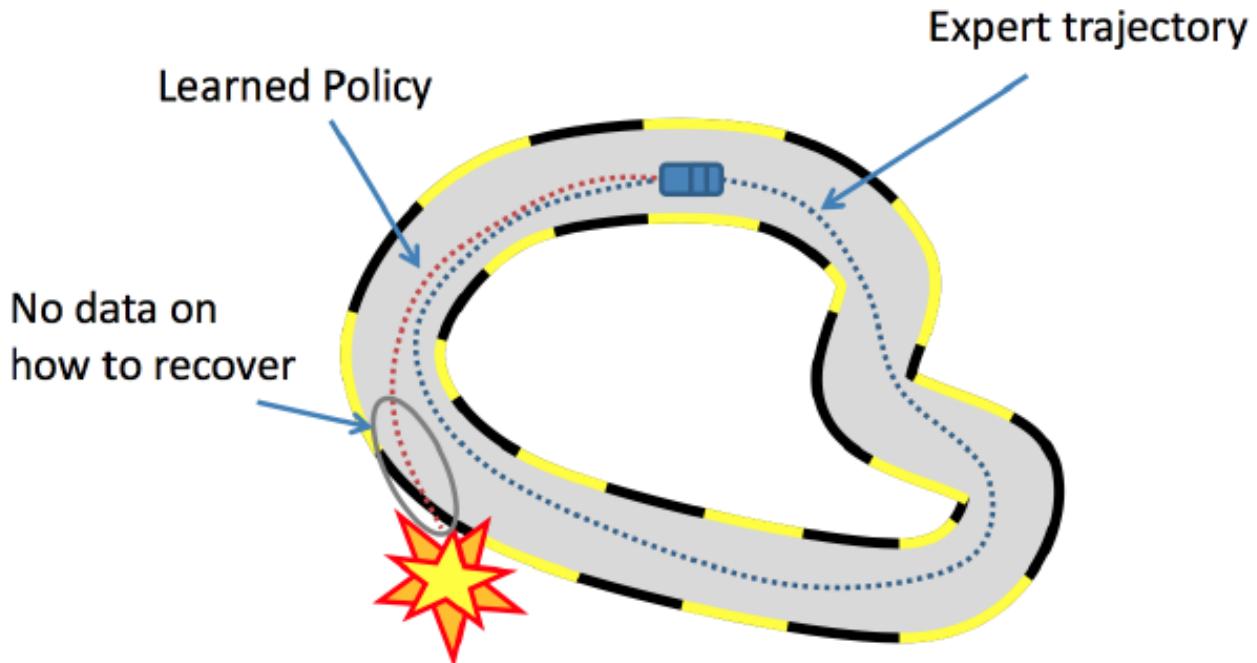
$$a_i \approx \pi(s_i)$$



Behavior Cloning

Неизбежно, наш агент заедет туда, где эксперт никогда не был.

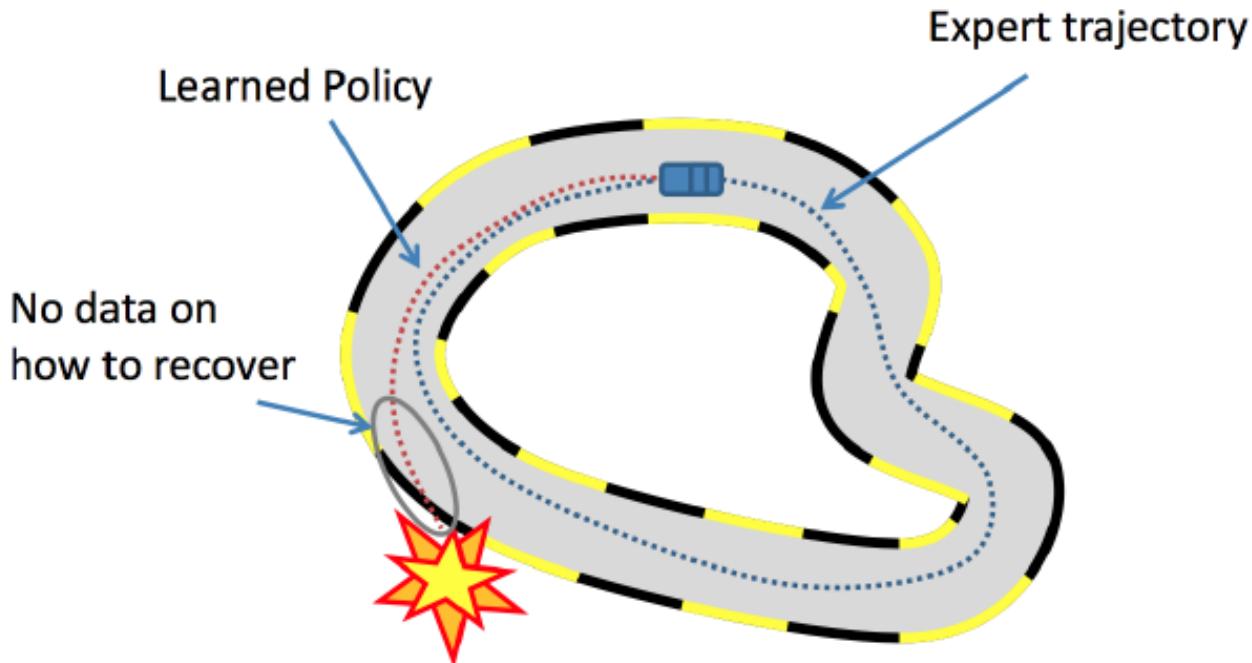
Агент не знает, как себя там вести.



Behavior Cloning

Неизбежно, наш агент заедет туда, где эксперт никогда не был.

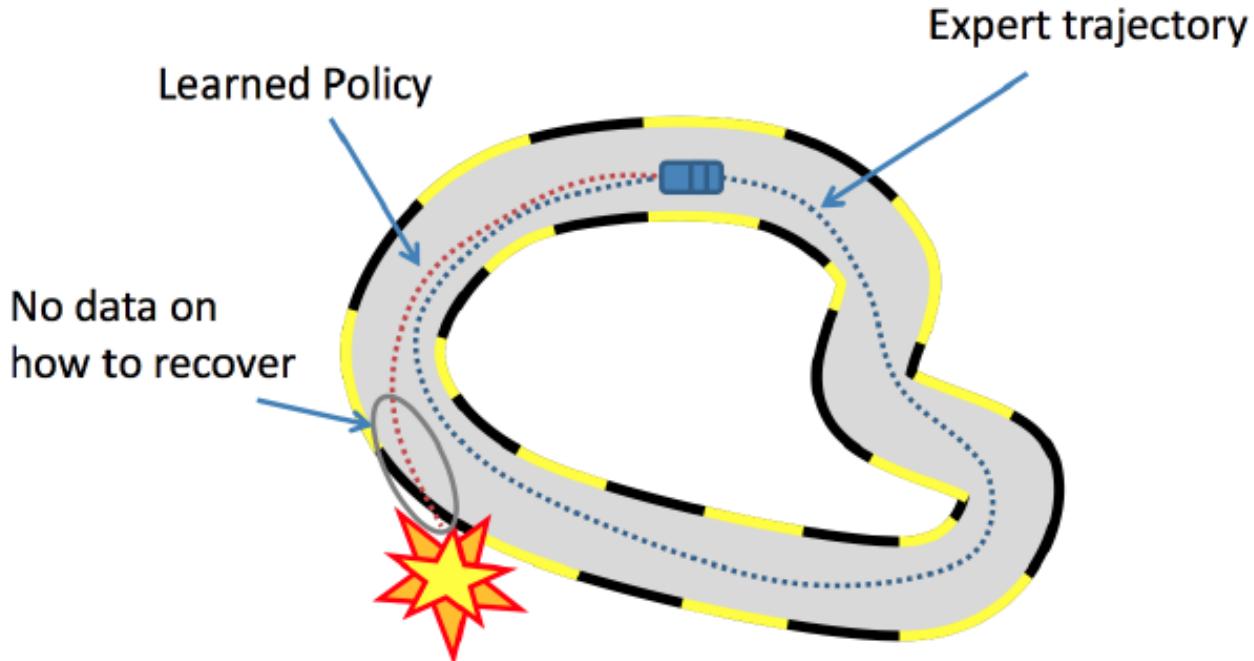
Агент не знает, как себя там вести.



Behavior Cloning

Неизбежно, наш агент заедет туда, где эксперт никогда не был.

Агент не знает, как себя там вести.



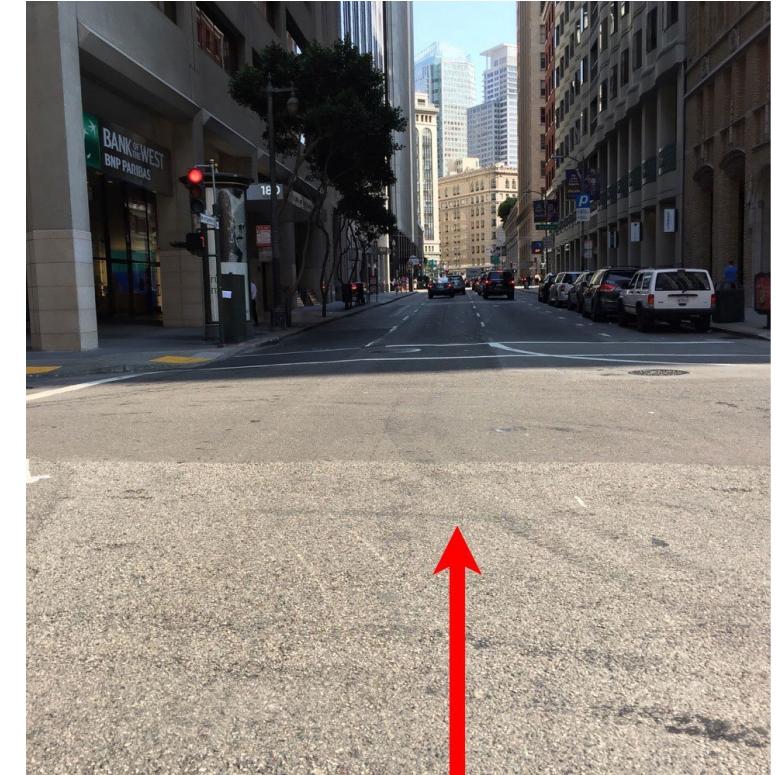
Проблема **Distributional Shift**:

Наши наблюдения меняются с изменением стратегии!

DAGGER

Можно просить эксперта возвращать агента на путь истинный.

На примере self-driving cars (беспилотный автомобиль):

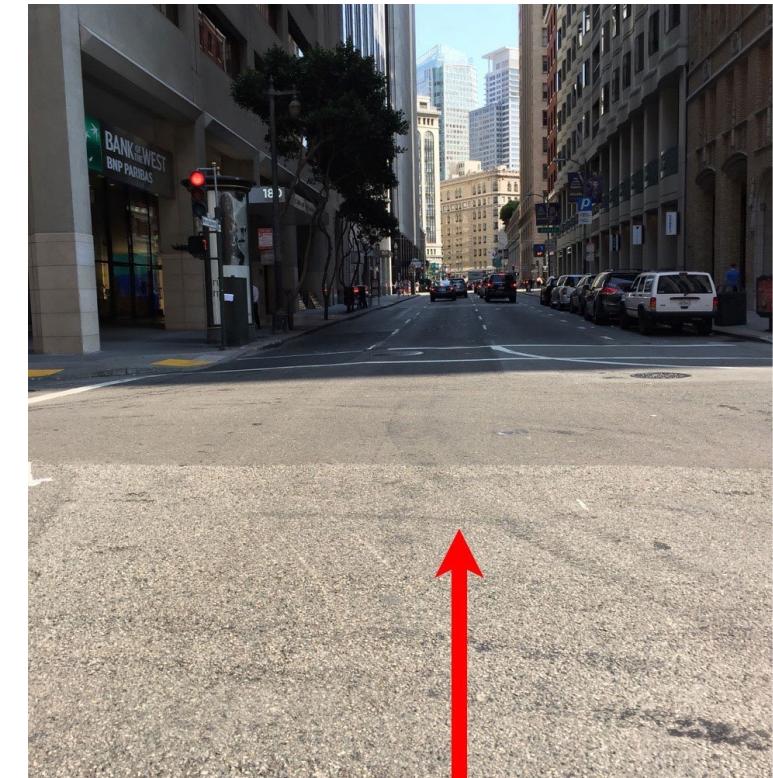


DAGGER

Можно просить эксперта возвращать агента на путь истинный.

На примере self-driving cars (беспилотный автомобиль):

- Отправляем водителя ездить по маршруту
- Записываем его траектории s_i и действия a_i
- Обучаем ML модель $\pi : a_i \approx \pi(s_i)$

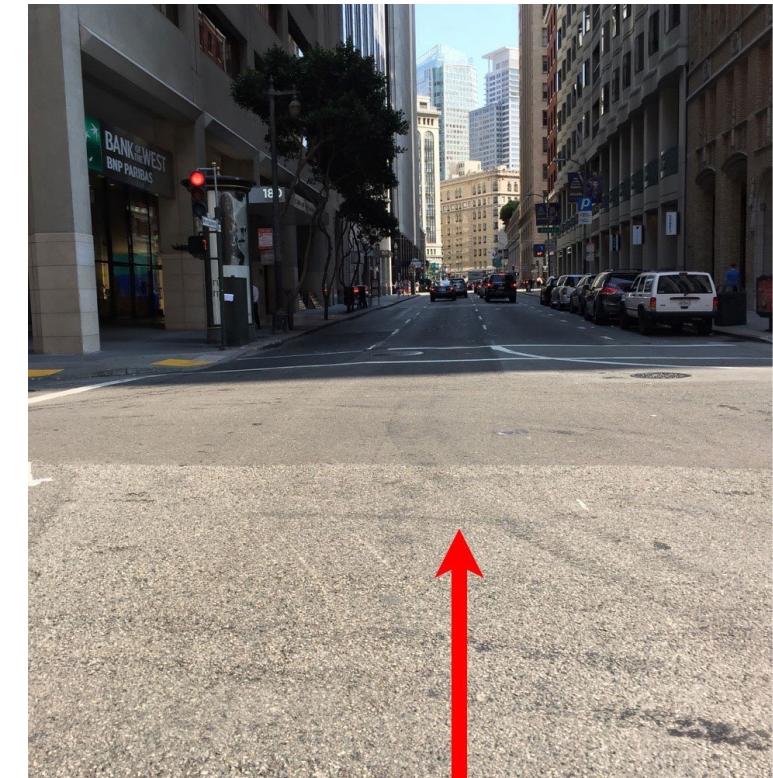


DAGGER

Можно просить эксперта возвращать агента на путь истинный.

На примере self-driving cars (беспилотный автомобиль):

- Отправляем водителя ездить по маршруту
- Записываем его траектории s_i и действия a_i
- Обучаем ML модель $\pi : a_i \approx \pi(s_i)$
- Делаем, пока не удовлетворены:

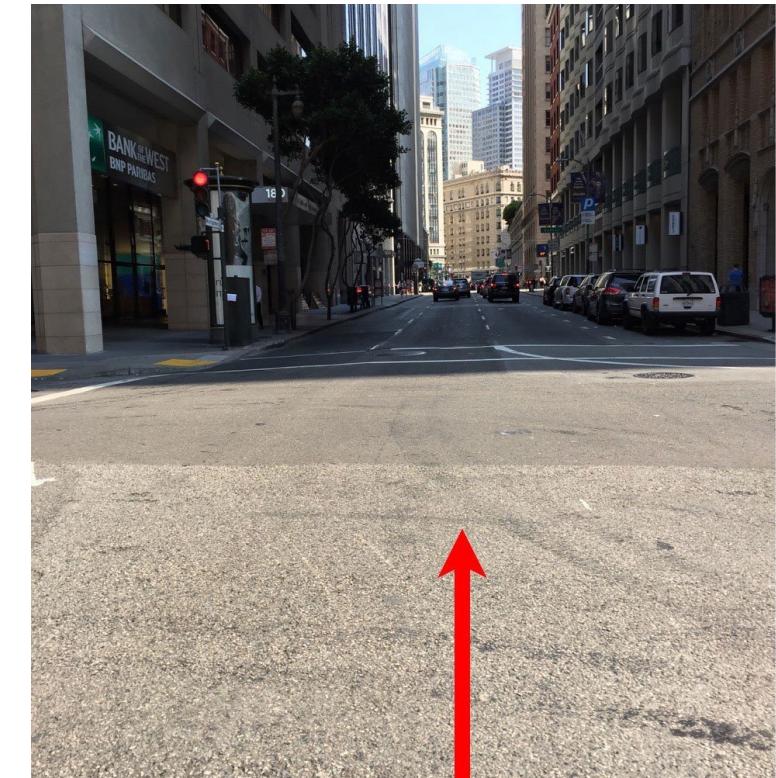


DAGGER

Можно просить эксперта возвращать агента на путь истинный.

На примере self-driving cars (беспилотный автомобиль):

- Отправляем водителя ездить по маршруту
- Записываем его траектории s_i и действия a_i
- Обучаем ML модель $\pi : a_i \approx \pi(s_i)$
- Делаем, пока не удовлетворены:
 - Пускаем агента в город, собираем траектории s_i



DAGGER

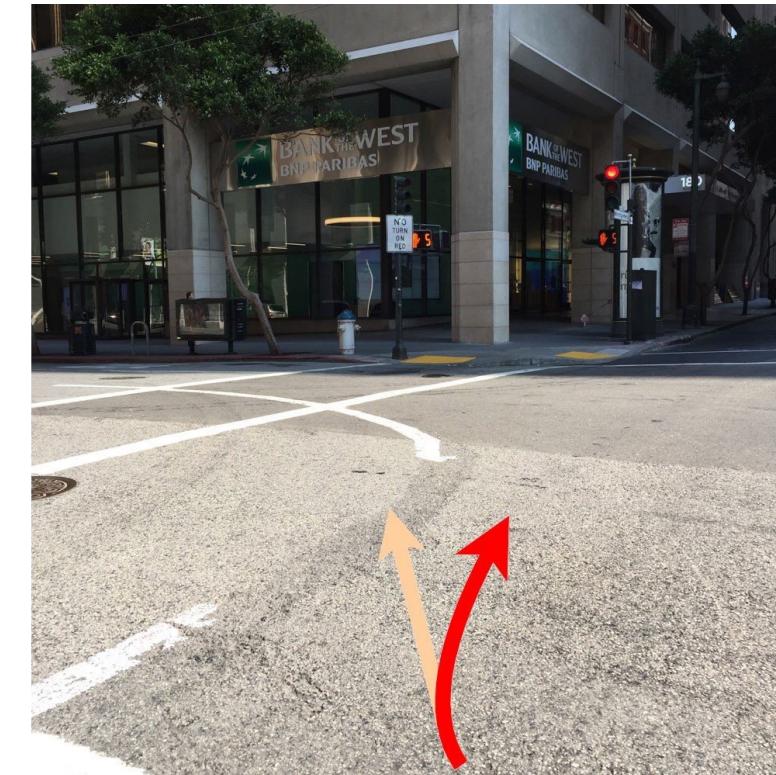
Можно просить эксперта возвращать агента на путь истинный.

На примере self-driving cars (беспилотный автомобиль):

- Отправляем водителя ездить по маршруту
- Записываем его траектории s_i и действия a_i
- Обучаем ML модель $\pi : a_i \approx \pi(s_i)$

Делаем, пока не удовлетворены:

- Пускаем агента в город, собираем траектории s_i
- Просим эксперта дать правильные действия a_i для собранных траекторий s_i



DAGGER

Плюсы:

- Очень прост
- Иногда хорошо работает

DAGGER

Плюсы:

- Очень прост
- Иногда хорошо работает

Минусы:

- Эксперт нужен в режиме онлайн
- Агент не будет лучше эксперта
- Не всегда эксперт вообще знает,
что делать!

DAGGER

Плюсы:

- Очень прост
- Иногда хорошо работает

Минусы:

- Эксперт нужен в режиме онлайн
- Агент не будет лучше эксперта
- Не всегда эксперт вообще знает, что делать!



Действия: напряжение, подаваемое на моторчики в суставах

DAGGER

<https://www.youtube.com/embed/YuyT2SDcYrU?t=137&enablejsapi=1>

Примеры Decision Process

Робототехника

Наблюдения:

- Изображения с видеокамеры
- Данные с датчиков

Действия:

- Усилие подаваемое на сочленения робота

Цели:

- Движение вперед
- Решение составных задач (перенос предметов)
- ...



Примеры Decision Process

Шахматы (или другие настольные игры)

Наблюдения:

- Расстановка фигур на доске

Действия:

- Выбор фигуры и хода ей

Цели:

- Победа
- Или, хотя бы, ничья



Примеры Decision Process

Автоматизированная торговля на бирже

Наблюдения:

- История изменения стоимости акций
- ...

Действия:

- Покупка и продажа акций

Цели:

- Максимизация прибыли



Награда за достижение целей

Вместо учителя



Награда за достижение целей

Вместо учителя



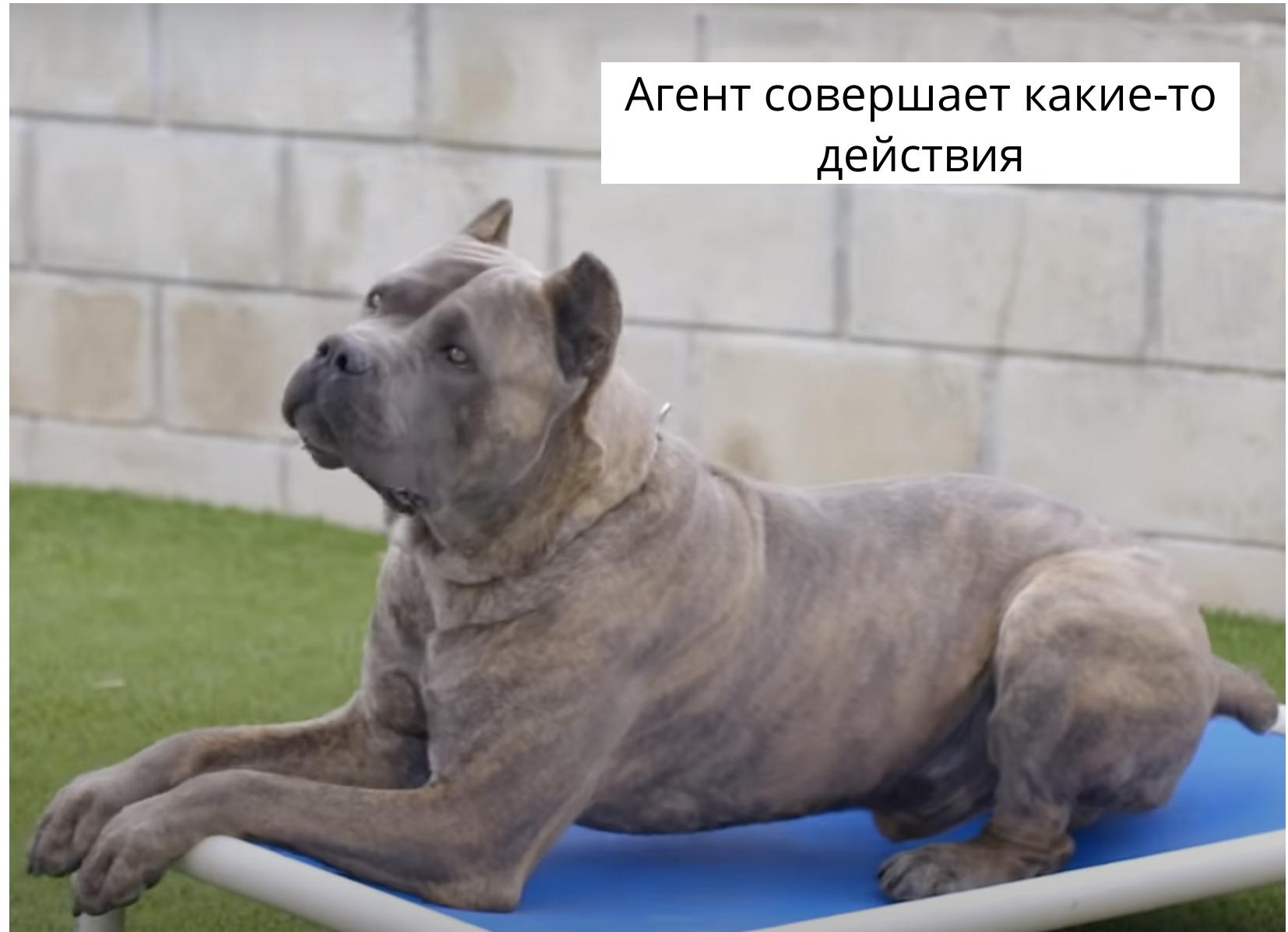
Награда за достижение целей

Вместо учителя



Награда за достижение целей

Вместо учителя



Агент совершает какие-то
действия

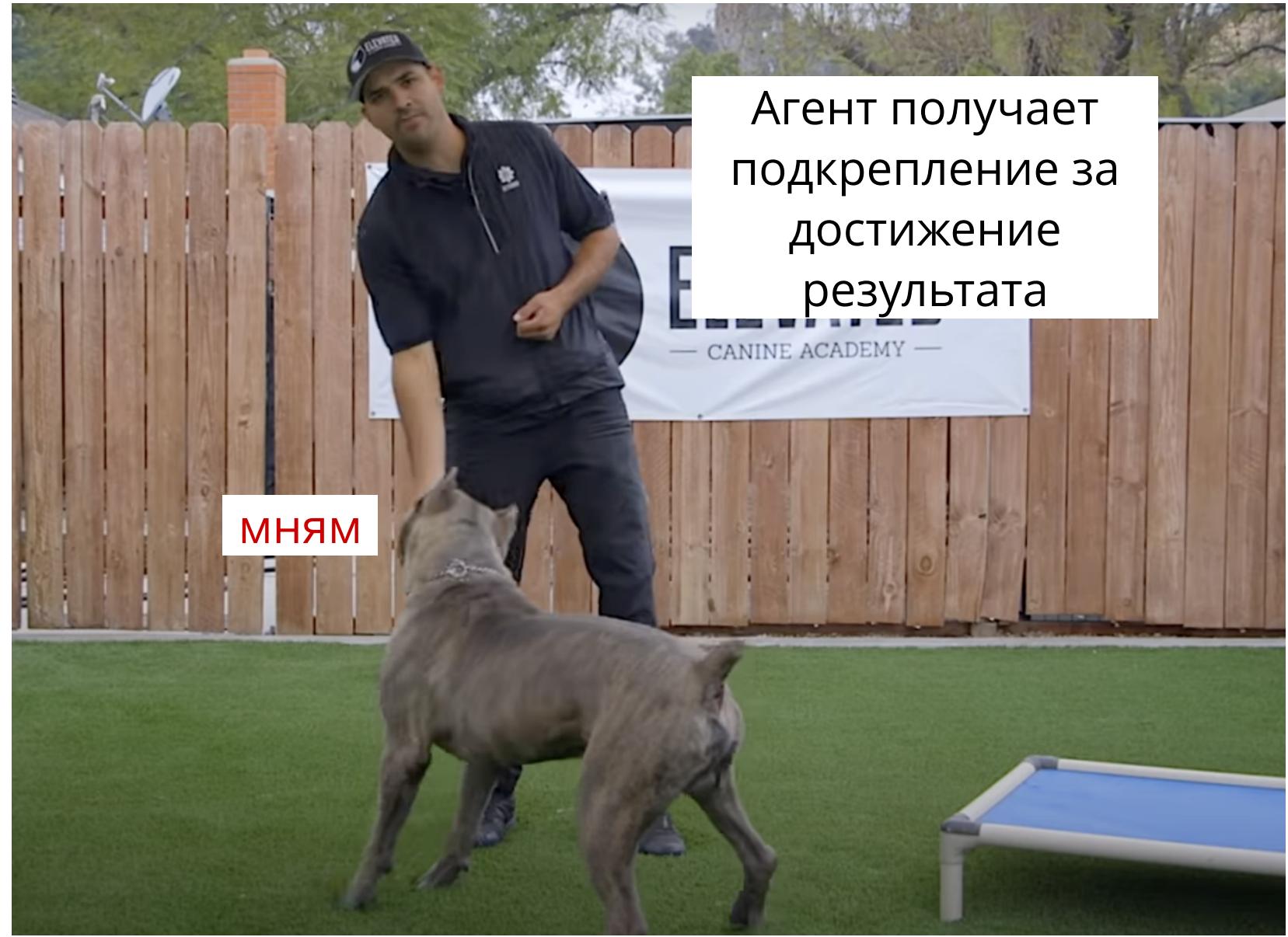
Награда за достижение целей

Вместо учителя



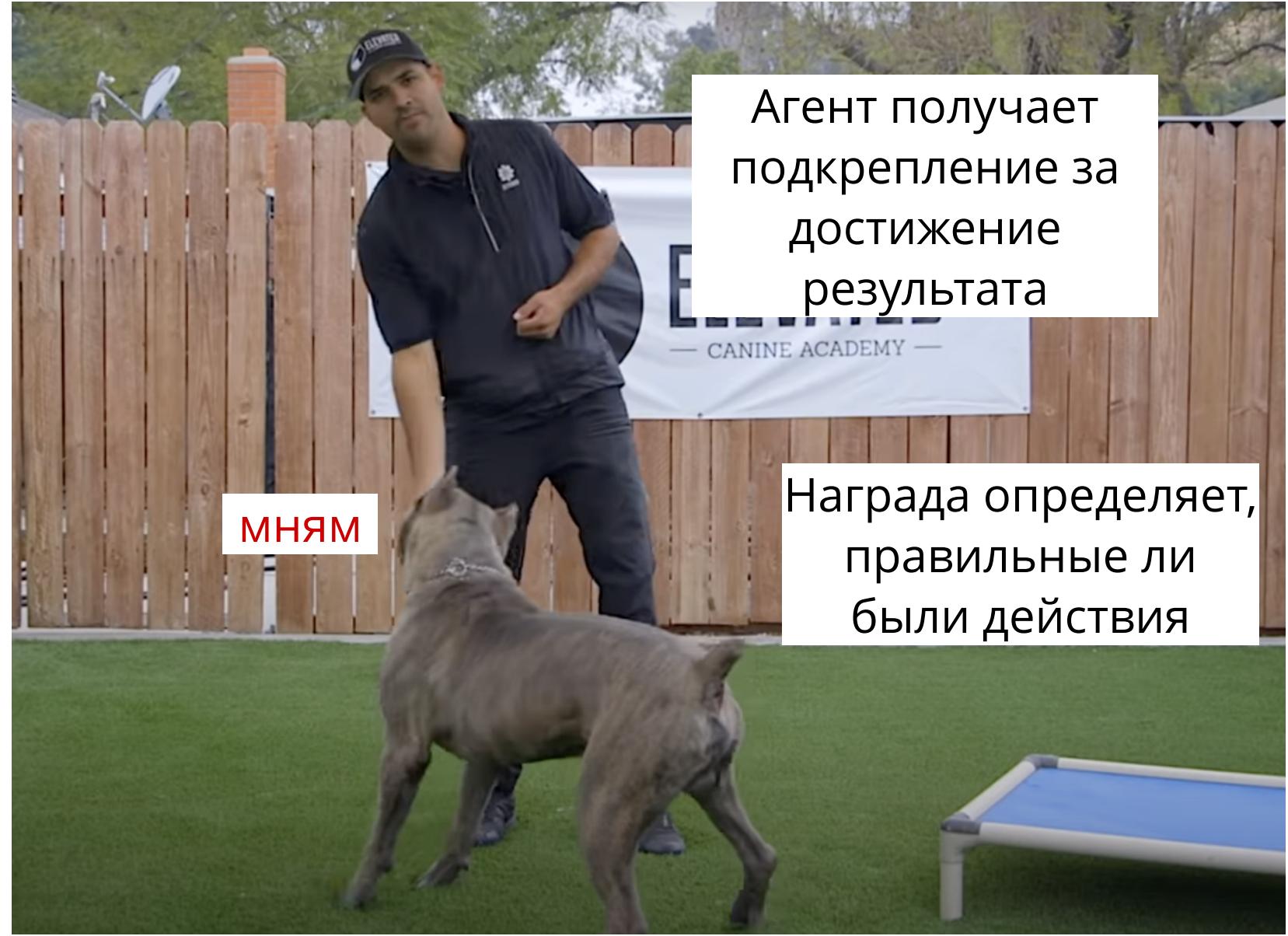
Награда за достижение целей

Вместо учителя



Награда за достижение целей

Вместо учителя



Агент получает
подкрепление за
достижение
результата

Награда определяет,
правильные ли
были действия

Награда за достижение целей

Вместо учителя

Это и есть

Обучение с
подкреплением

**Reinforcement
Learning**



Обучение с подкреплением

Если вы знаете, **чего** вы хотите, но не знаете, **как** этого достичь...

Используйте **НАГРАДЫ** (rewards)

Обучение с подкреплением

Если вы знаете, **чего** вы хотите, но не знаете, **как** этого достичь...

Используйте **НАГРАДЫ** (rewards)

- Победа: +1
- Проигрыш: -1
- Ничья: 0



Обучение с подкреплением

Если вы знаете, **чего** вы хотите, но не знаете, **как** этого достичь...

Используйте **НАГРАДЫ** (rewards)

- Победа: +1
- Проигрыш: -1
- Ничья: 0

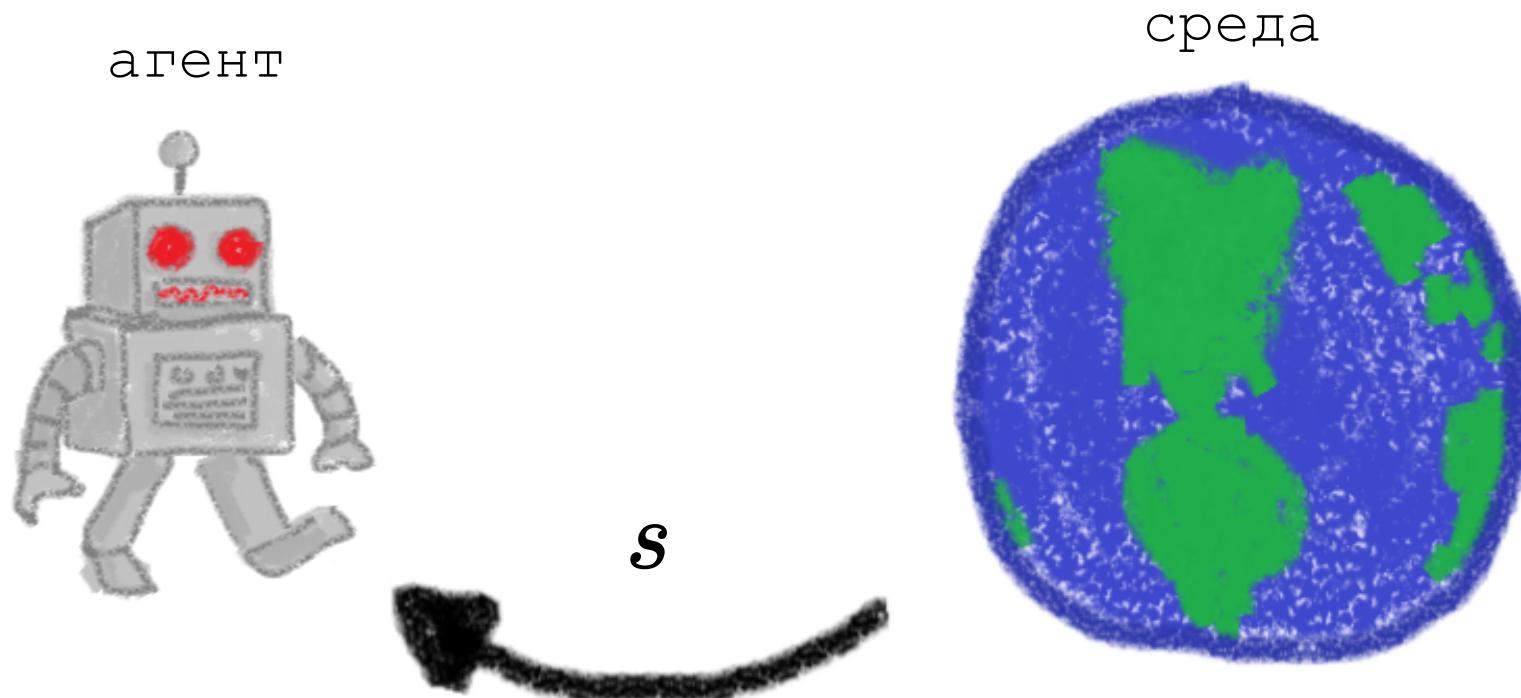
Стандартные предположения:

- Награду легко посчитать из:
 - текущего положения агента
 - принятого действия
- Из награды очевидна цель



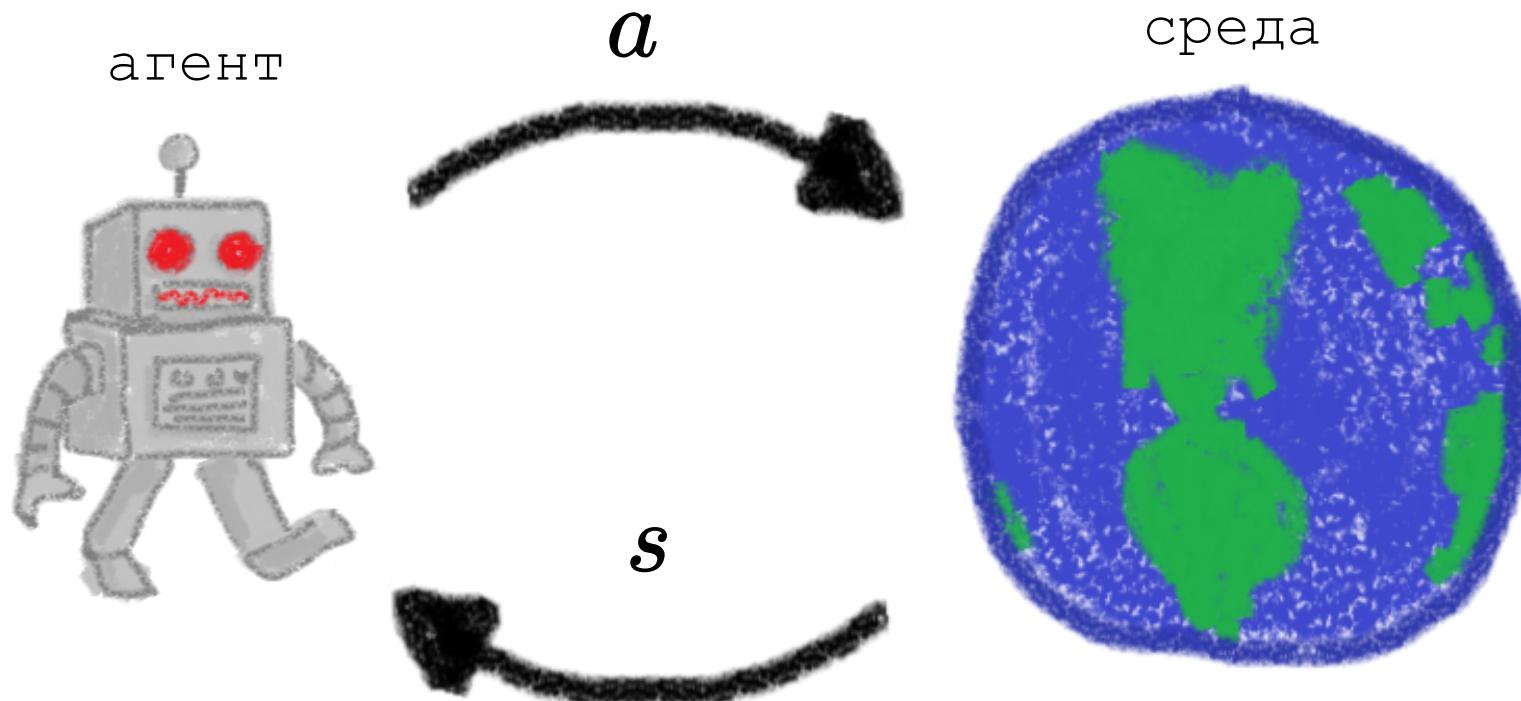
Задача Reinforcement Learning

- Агент взаимодействует со средой - наблюдает ее состояние s



Задача Reinforcement Learning

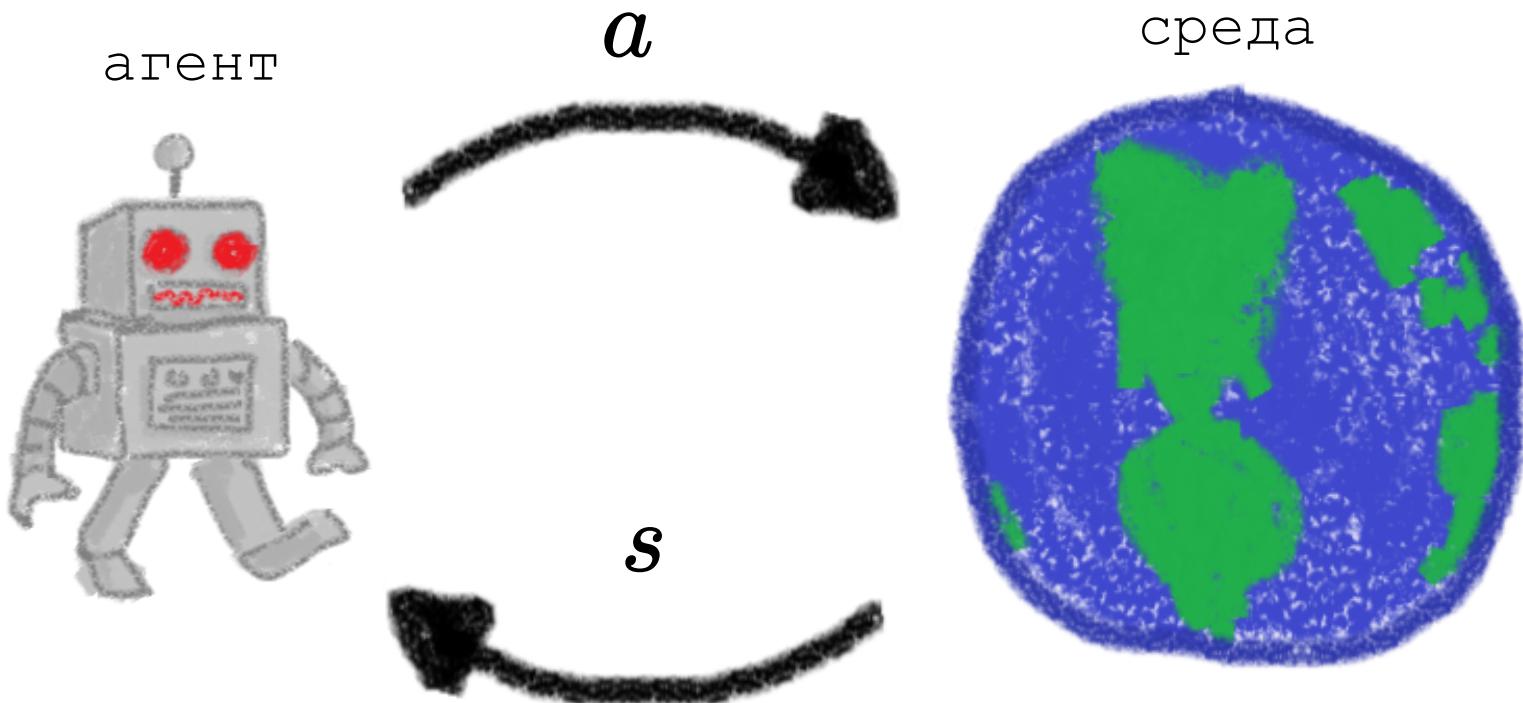
- Агент взаимодействует со средой - наблюдает ее состояние s
- Агент посыпает в среду действие a



Задача Reinforcement Learning

- Агент взаимодействует со средой - наблюдает ее состояние s
- Агент посыпает в среду действие a

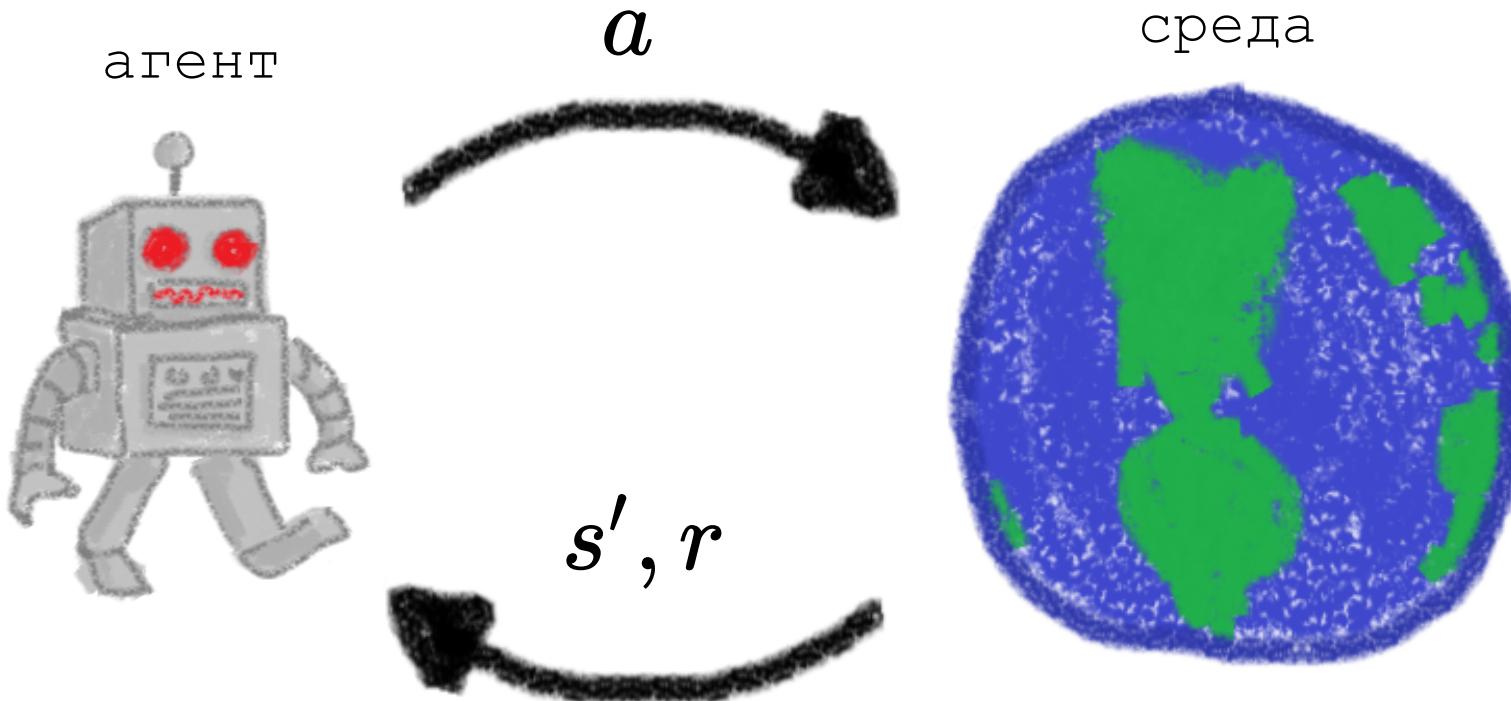
Политика
агента:
 $a = \pi(s)$



Задача Reinforcement Learning

- Агент взаимодействует со средой - наблюдает ее состояние s
- Агент посыпает в среду действие a
- Среда возвращает агенту следующее состояние s' и награду r

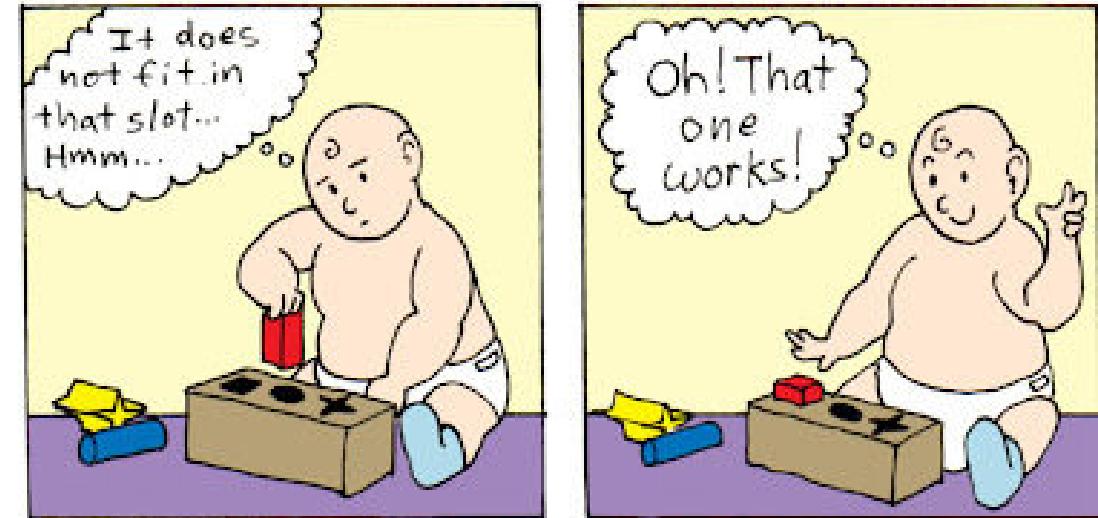
Политика
агента:
 $a = \pi(s)$



Задача Reinforcement Learning

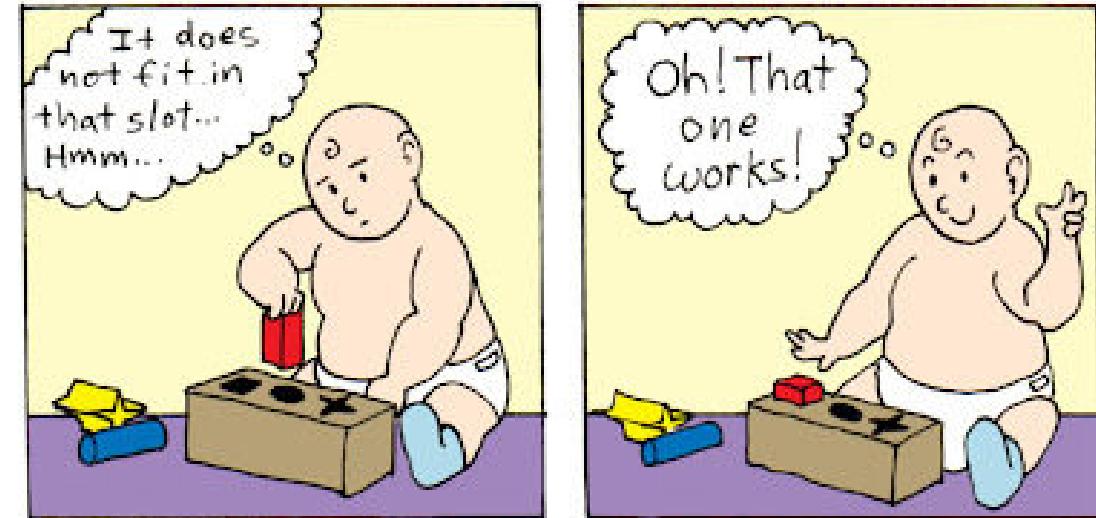
Задача Reinforcement Learning

- Мы не знаем, как устроена среда



Задача Reinforcement Learning

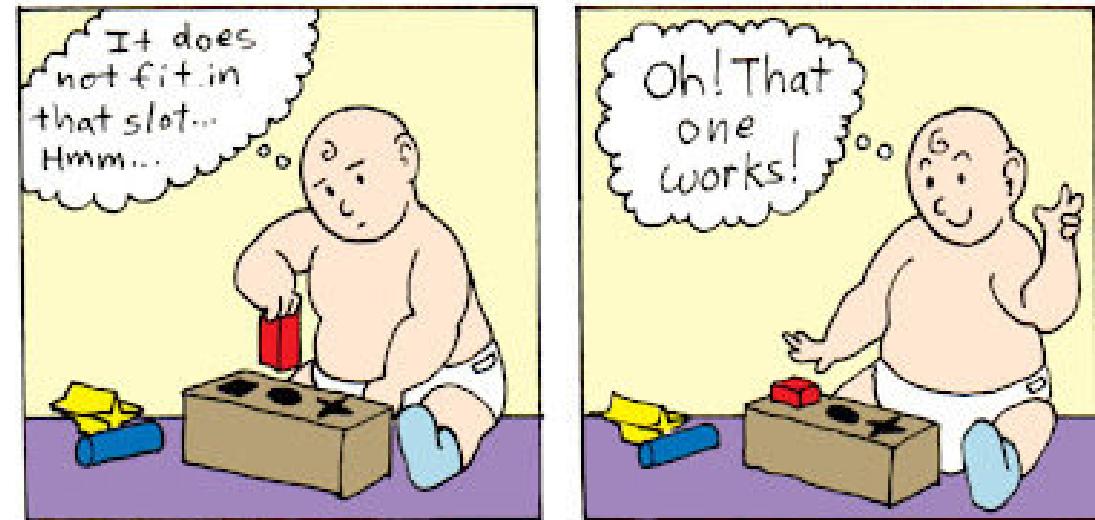
- Мы не знаем, как устроена среда
- Агент учится через взаимодействие



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

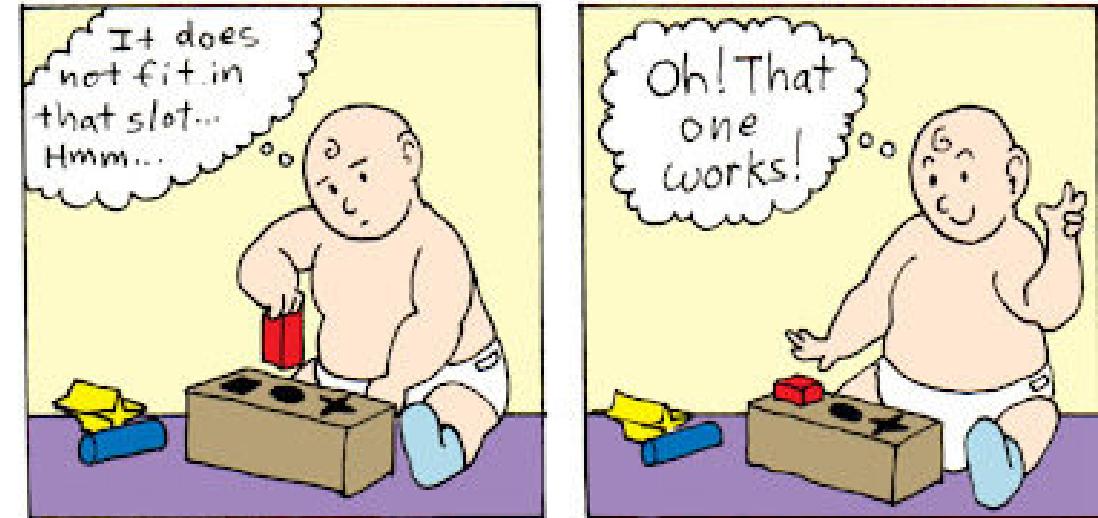
$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$

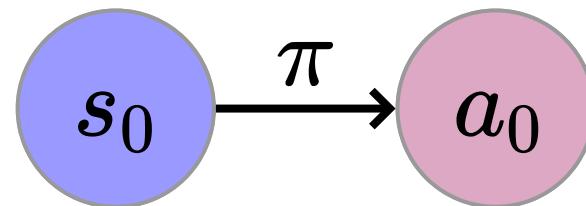
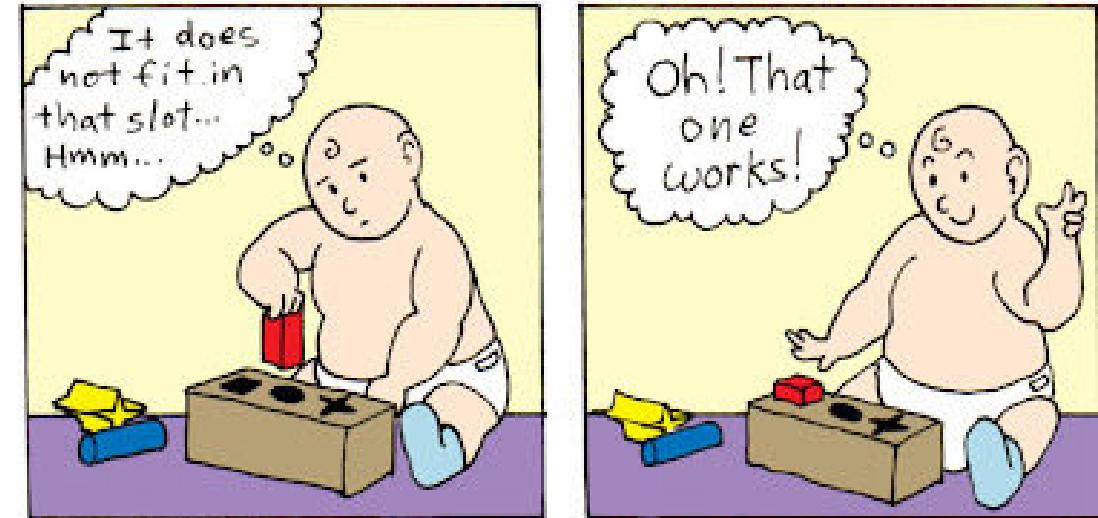


s_0

Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

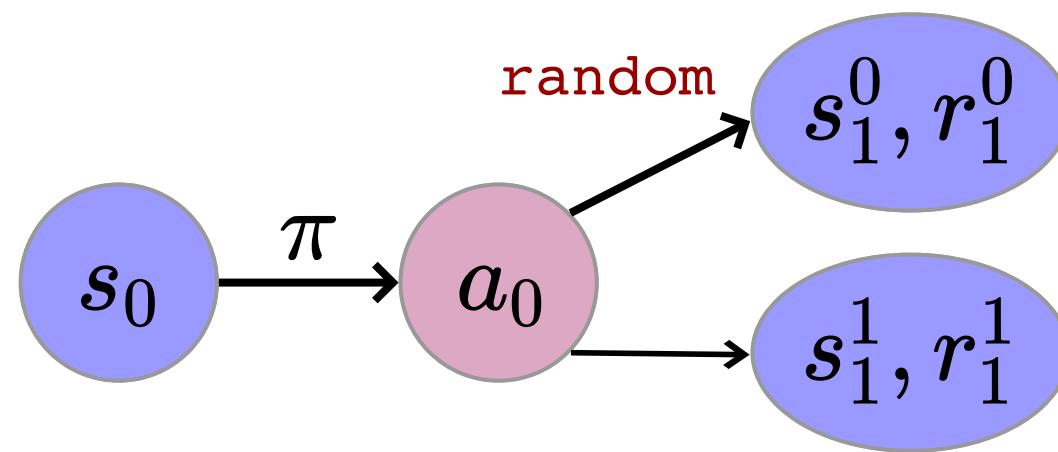
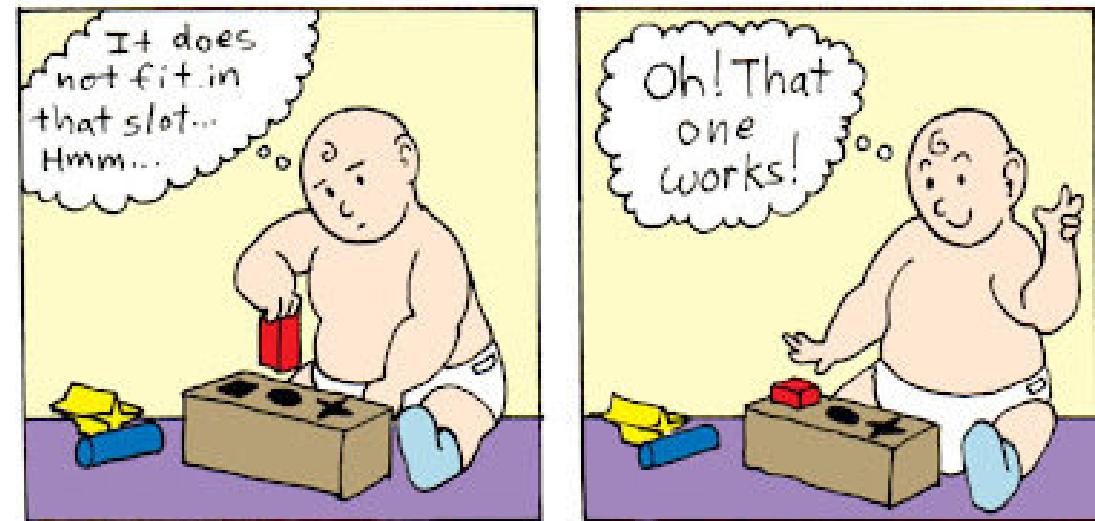
$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

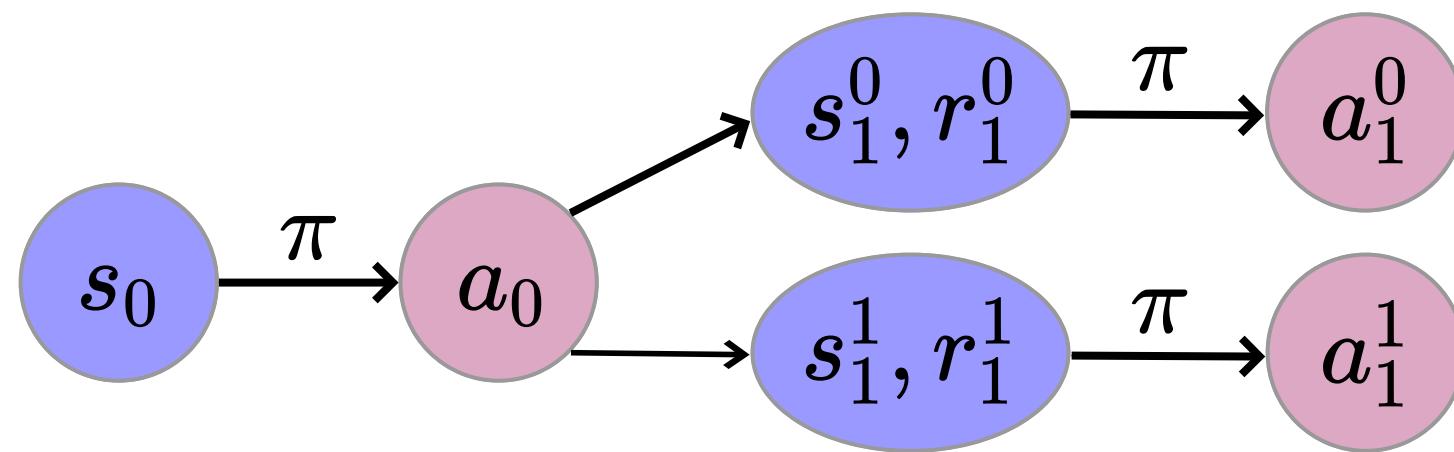
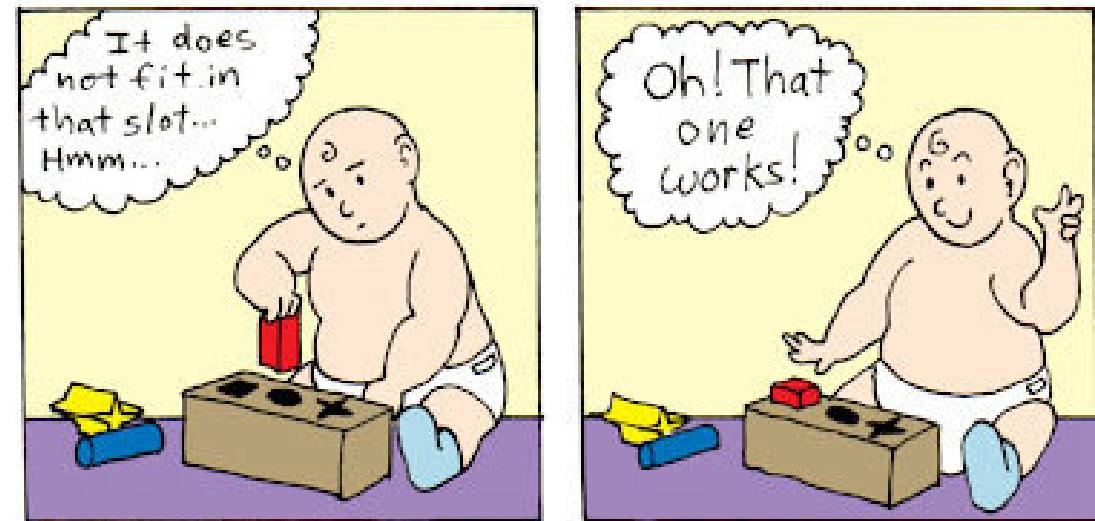
$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

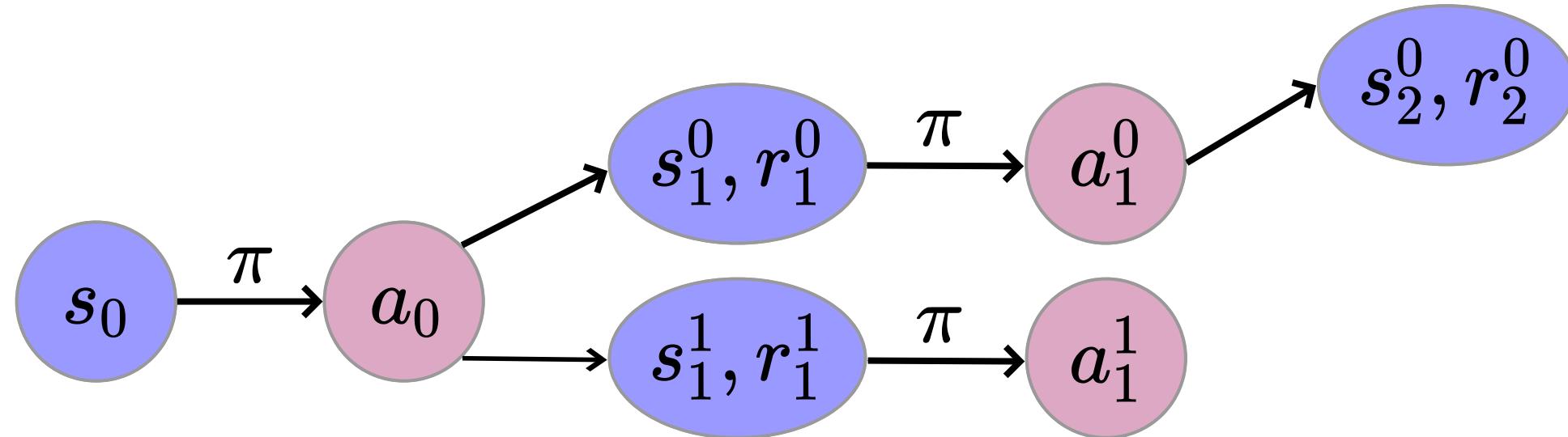
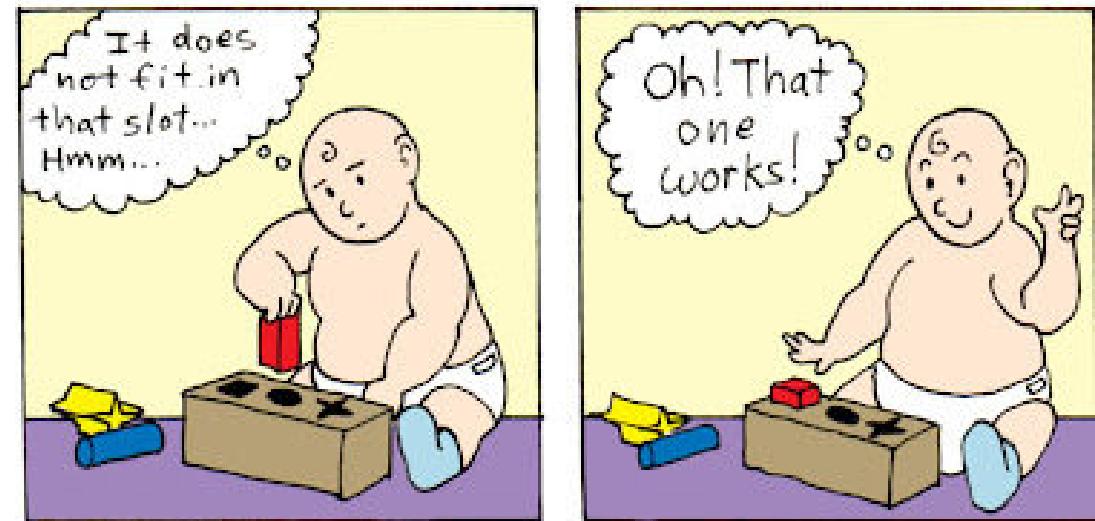
$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

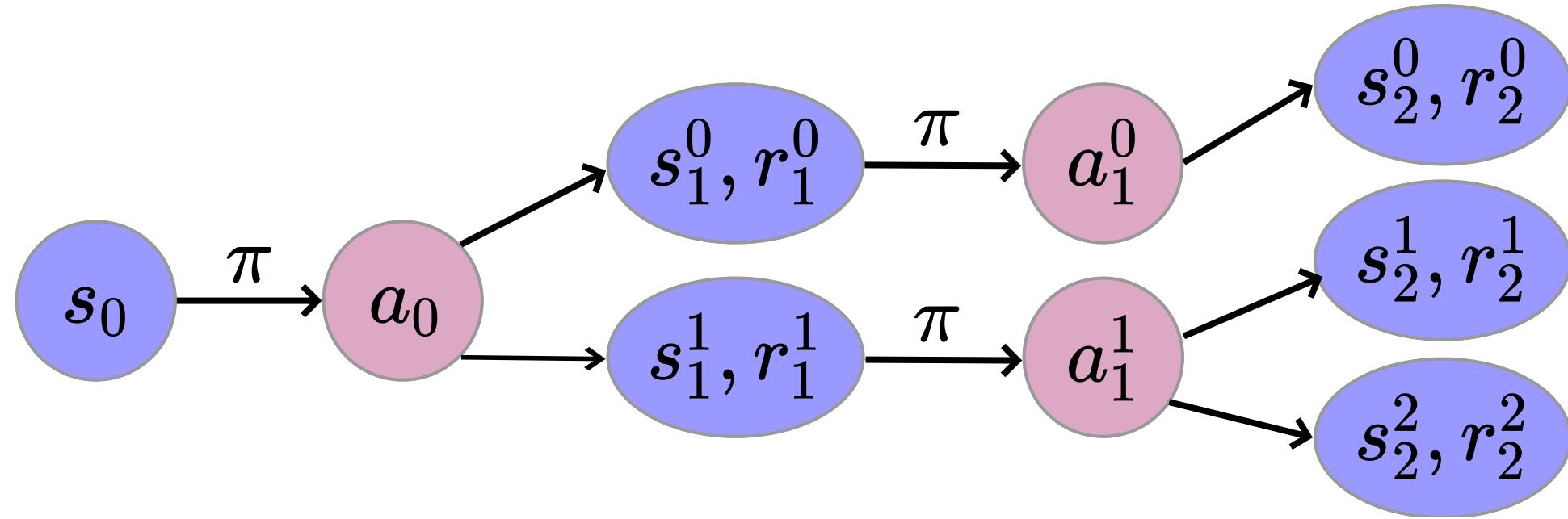
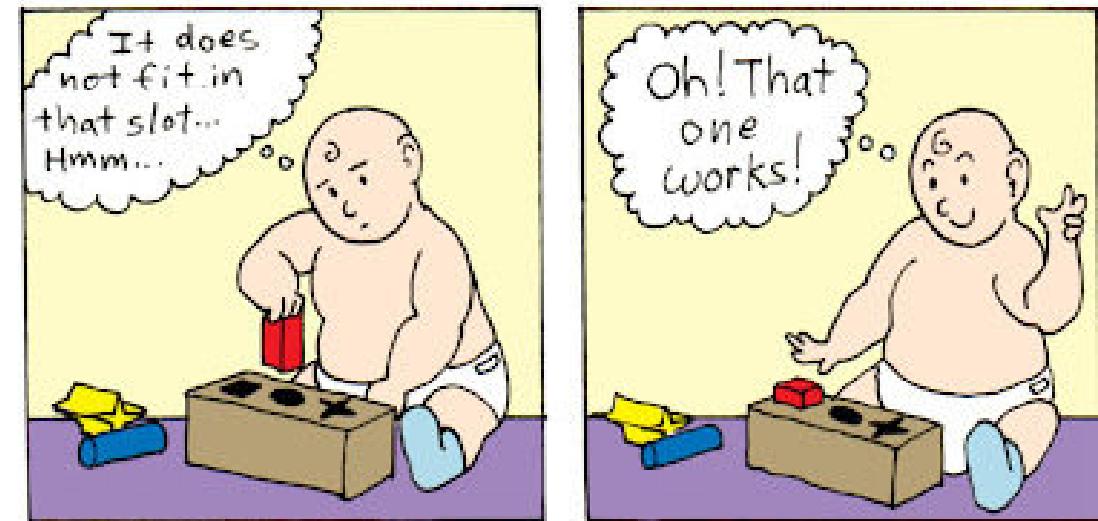
$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

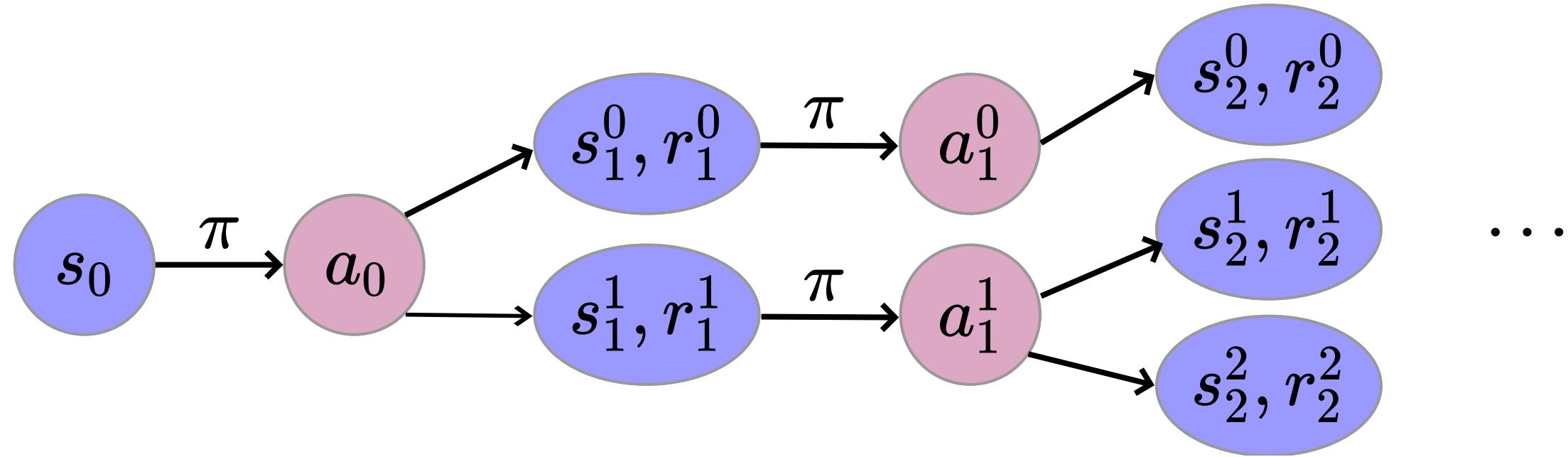
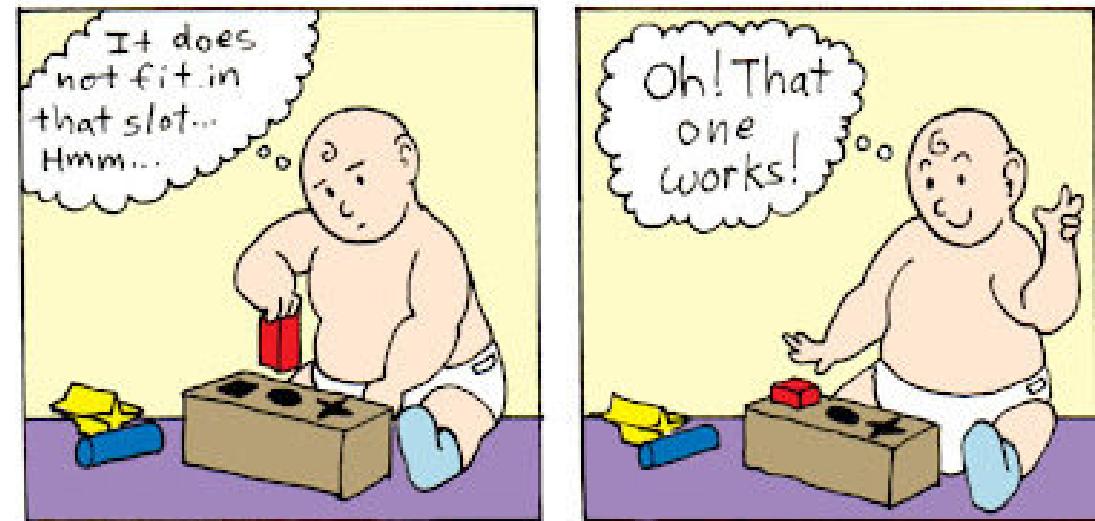
$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Задача Reinforcement Learning

- Мы не знаем, как устроена среда
- Агент учится через взаимодействие
- Пытается подобрать политику π , которая максимизирует **накопленную ожидаемую награду**:

$$R_\pi = \mathbb{E}_\pi [\sum_{t=0}^T r_t]$$



Как выбрать награду?

На примере шахмат

Какой выбор награды лучше?



Как выбрать награду?

На примере шахмат

Какой выбор награды лучше?

- победа: +1
- проигрыш: -1
- ничья: 0



Как выбрать награду?

На примере шахмат

Какой выбор награды лучше?

- победа: +1
 - проигрыш: -1
 - ничья: 0
-
- победа: +1
 - проигрыш: -1
 - ничья: 0
 - взятие фигуры: +1
 - потеря фигуры: -1



Как выбрать награду?

На примере шахмат

Какой выбор награды лучше?

- победа: +1
 - проигрыш: -1
 - ничья: 0
-
- победа: +1
 - проигрыш: -1
 - ничья: 0
 - взятие фигуры: +1
 - потеря фигуры: -1



Почему?

Как выбрать награду?

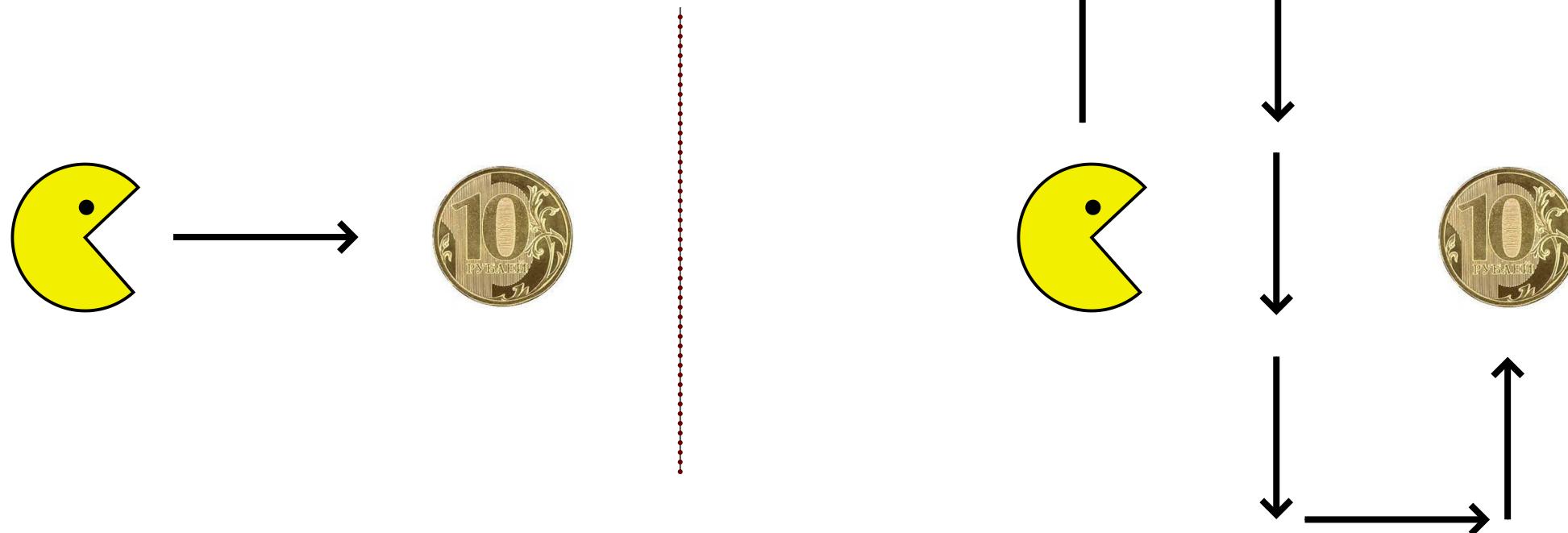
Агент, стремящийся получить максимум награды, должен лучше всего решать вашу задачу!

<https://www.youtube.com/embed/tI0IHko8ySg?enablejsapi=1>

Дисконтирование награды

Сравним эти две траектории агента

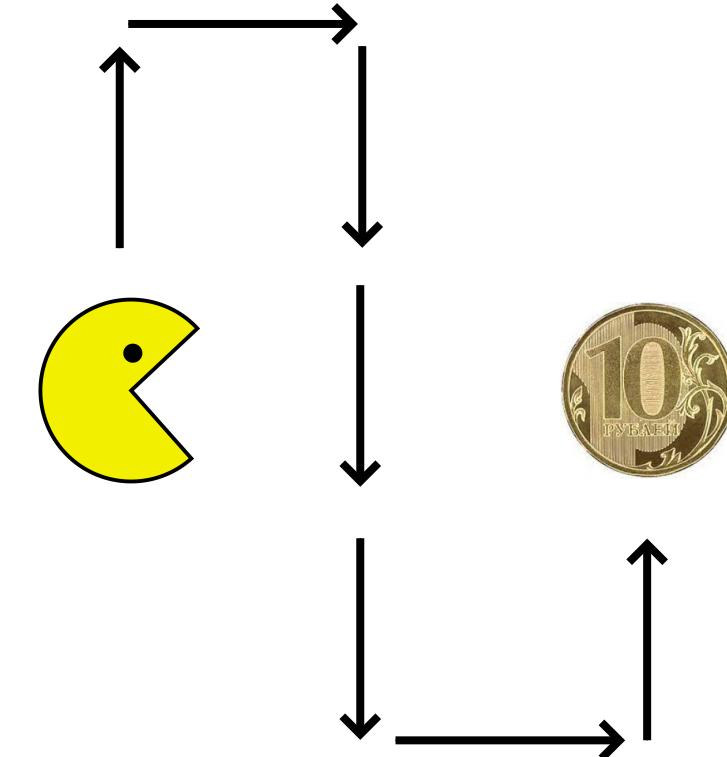
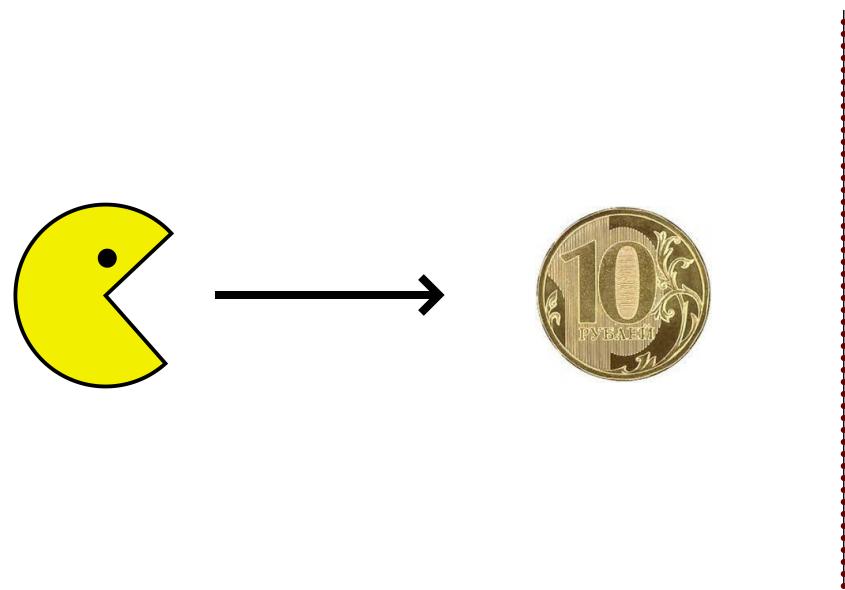
Какая лучше?



Дисконтирование награды

Сравним эти две траектории агента

Какая лучше?



Награда сегодня лучше, чем награда завтра!

Дисконтирование награды - уменьшение значимости наград, которые далеко

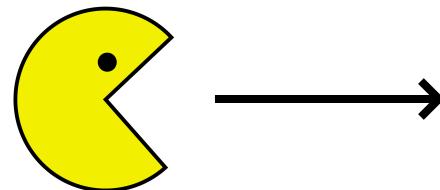
$$R_\pi = \sum_{t=0}^T \gamma^t r_t \quad \text{где } 0 < \gamma < 1$$

Дисконтирование награды

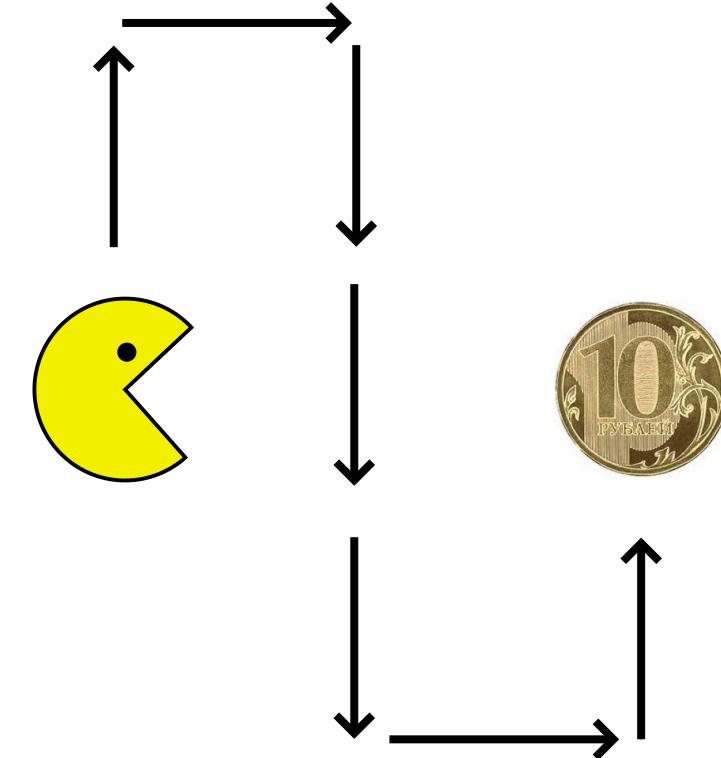
Сравним эти две траектории агента

Какая лучше?

пусть $\gamma = 0.9$



$r = 1$



Награда сегодня лучше, чем награда завтра!

Дисконтирование награды - уменьшение значимости наград, которые далеко

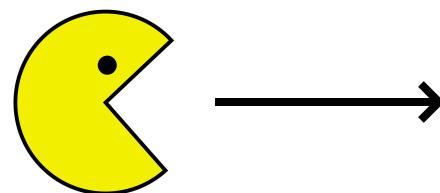
$$R_\pi = \sum_{t=0}^T \gamma^t r_t \quad \text{где } 0 < \gamma < 1$$

Дисконтирование награды

Сравним эти две траектории агента

Какая лучше?

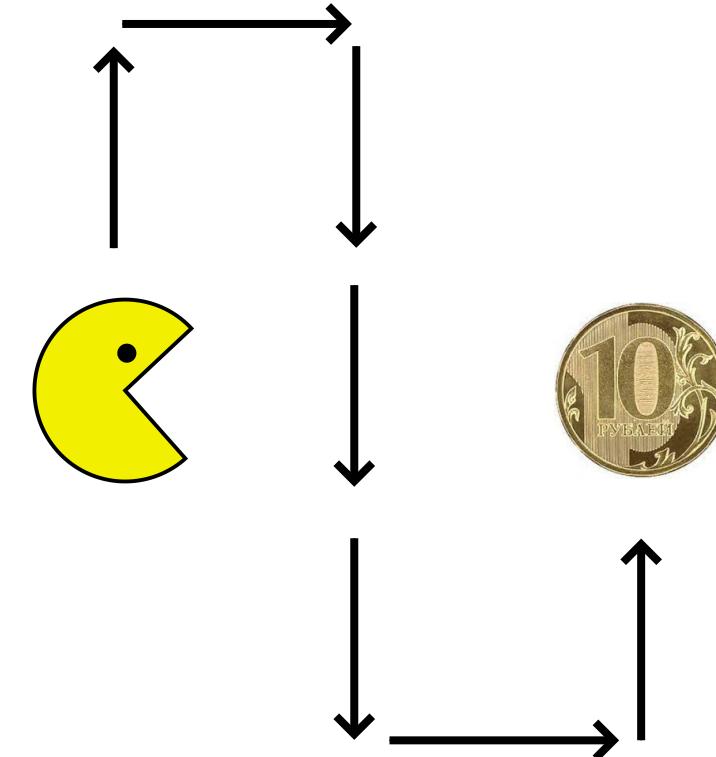
пусть $\gamma = 0.9$



$r = 1$



$$R = \gamma^1 r = 0.9$$



Награда сегодня лучше, чем награда завтра!

Дисконтирование награды - уменьшение значимости наград, которые далеко

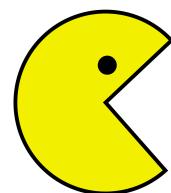
$$R_\pi = \sum_{t=0}^T \gamma^t r_t \quad \text{где } 0 < \gamma < 1$$

Дисконтирование награды

Сравним эти две траектории агента

Какая лучше?

пусть $\gamma = 0.9$



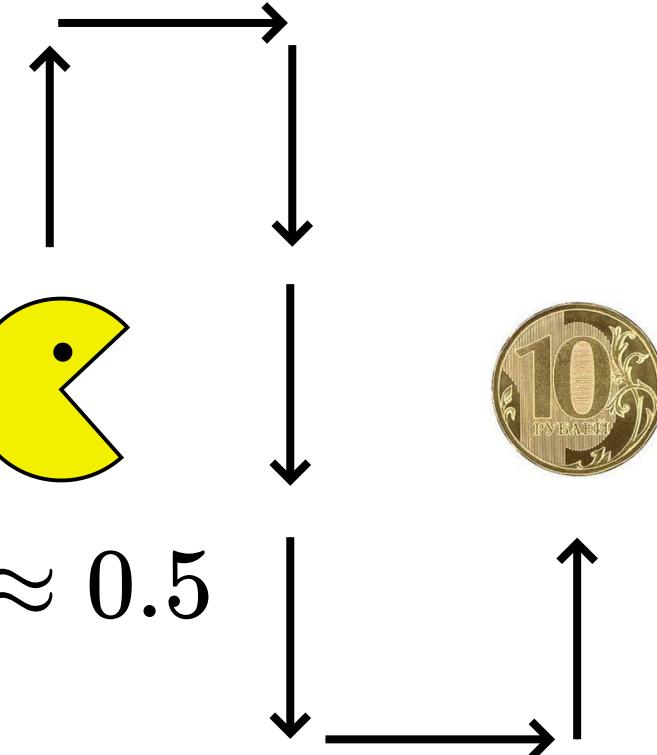
$r = 1$



$$R = \gamma^1 r = 0.9$$



$$R = \gamma^7 r \approx 0.5$$



Награда сегодня лучше, чем награда завтра!

Дисконтирование награды - уменьшение значимости наград, которые далеко

$$R_\pi = \sum_{t=0}^T \gamma^t r_t \quad \text{где } 0 < \gamma < 1$$

Среды и наблюдения

Пример: **робот-пылесос**



Среды и наблюдения

Пример: **робот-пылесос**

Цель: пропылесосить всю квартиру

Действия a : повороты, скорость вращения колес



Среды и наблюдения

Пример: **робот-пылесос**

Цель: пропылесосить всю квартиру

Действия a : повороты, скорость вращения колес

Какие могут быть **наблюдения** s у робота?



Среды и наблюдения

Пример: **робот-пылесос**

Цель: пропылесосить всю квартиру

Действия a : повороты, скорость вращения колес

Какие могут быть **наблюдения** s у робота?

- Данные лидаров и др. сенсоров
- Карта помещения
- Локация робота на карте



Среды и наблюдения

Пример: **робот-пылесос**

Цель: пропылесосить всю квартиру

Действия a : повороты, скорость вращения колес

Какие могут быть **наблюдения** s у робота?

- Данные лидаров и др. сенсоров
- Карта помещения
- Локация робота на карте

Нужно ли что-то еще?



Среды и наблюдения

Пример: **робот-пылесос**

Цель: пропылесосить всю квартиру

Действия a : повороты, скорость вращения колес

Какие могут быть **наблюдения** s у робота?

- Данные лидаров и др. сенсоров
- Карта помещения
- Локация робота на карте

Нужно ли что-то еще?

Хранение предыдущих наблюдений?



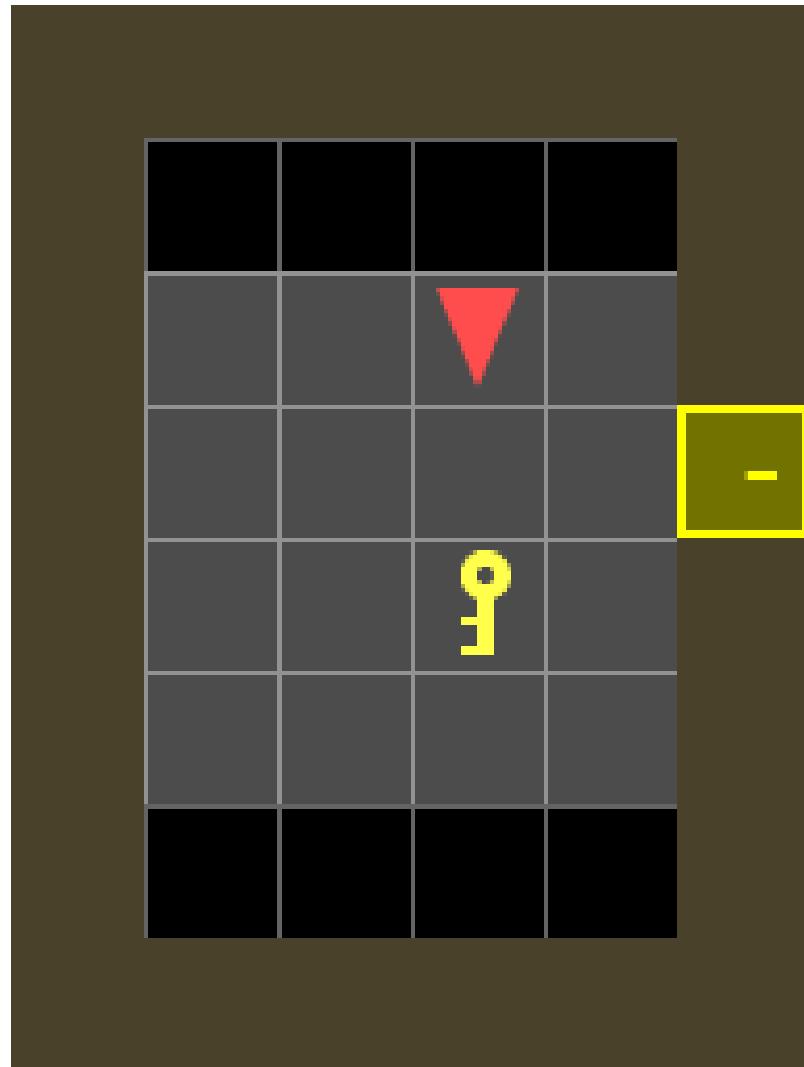
Марковское свойство

Markov Decision Process (MDP)

Задача: открыть дверь ключом

Действия:

- вверх \ вниз \ влево \ вправо
- взять объект
- использовать объект
(если находишься у двери)

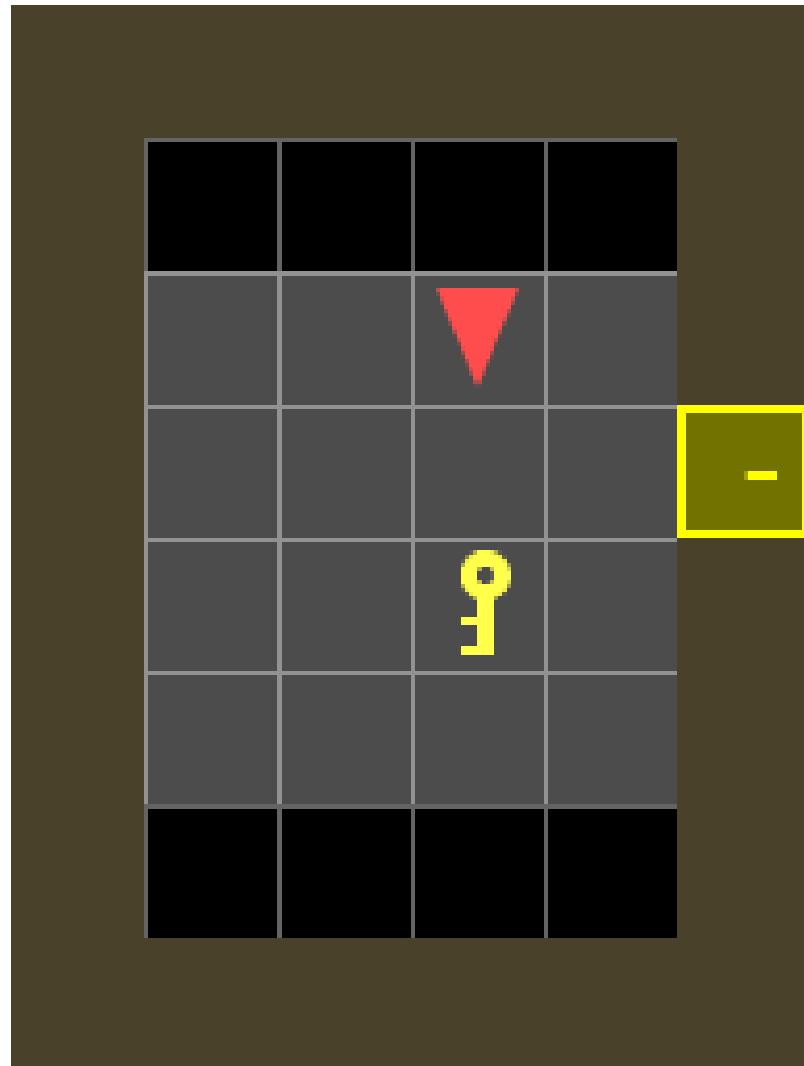


Марковское свойство

Markov Decision Process (MDP)

Нужно ли агенту помнить свою историю?

Возможные наблюдения:



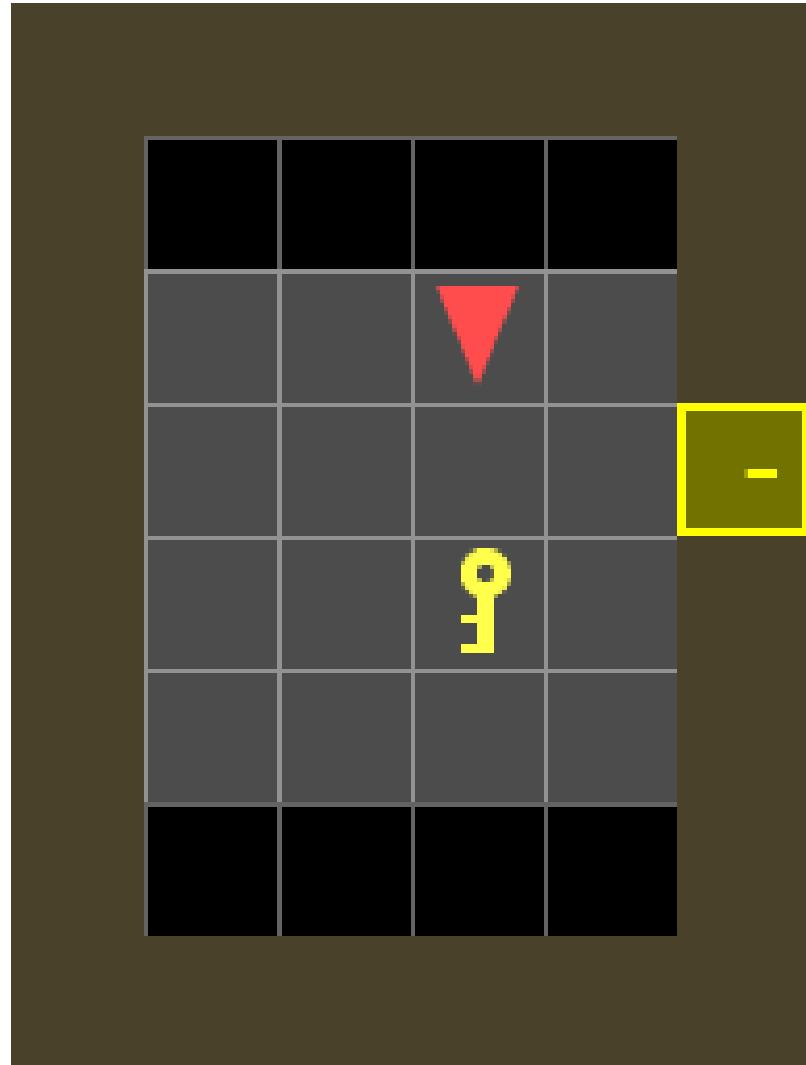
Марковское свойство

Markov Decision Process (MDP)

Нужно ли агенту помнить свою историю?

Возможные наблюдения:

1. координаты агента



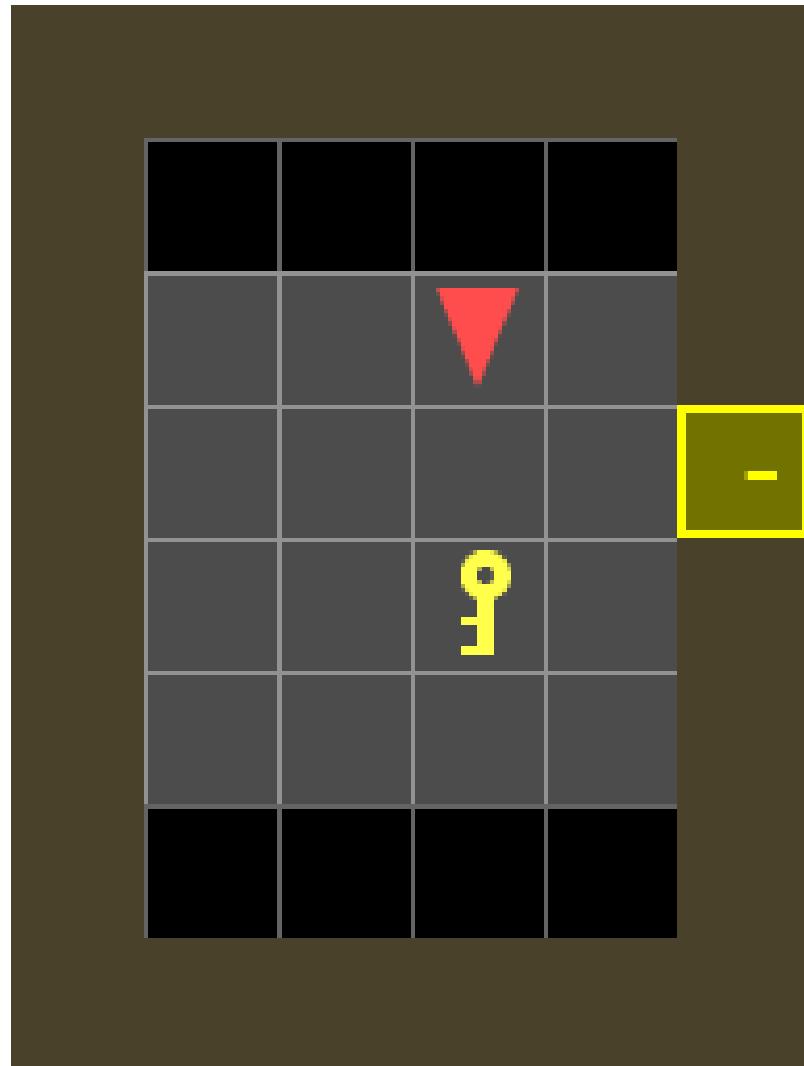
Марковское свойство

Markov Decision Process (MDP)

Нужно ли агенту помнить свою историю?

Возможные наблюдения:

1. координаты агента
2. полная картинка лабиринта



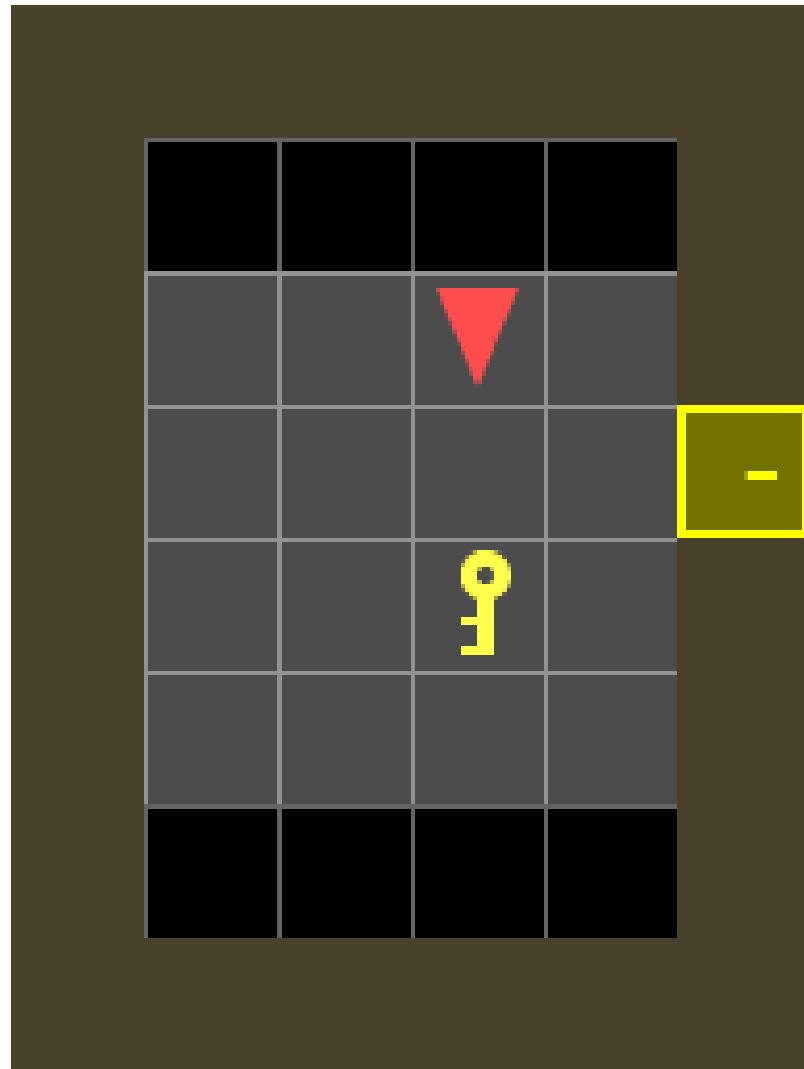
Марковское свойство

Markov Decision Process (MDP)

Нужно ли агенту помнить свою историю?

Возможные наблюдения:

1. координаты агента
2. полная картинка лабиринта
3. координаты агента + есть ли у него ключ



Марковское свойство

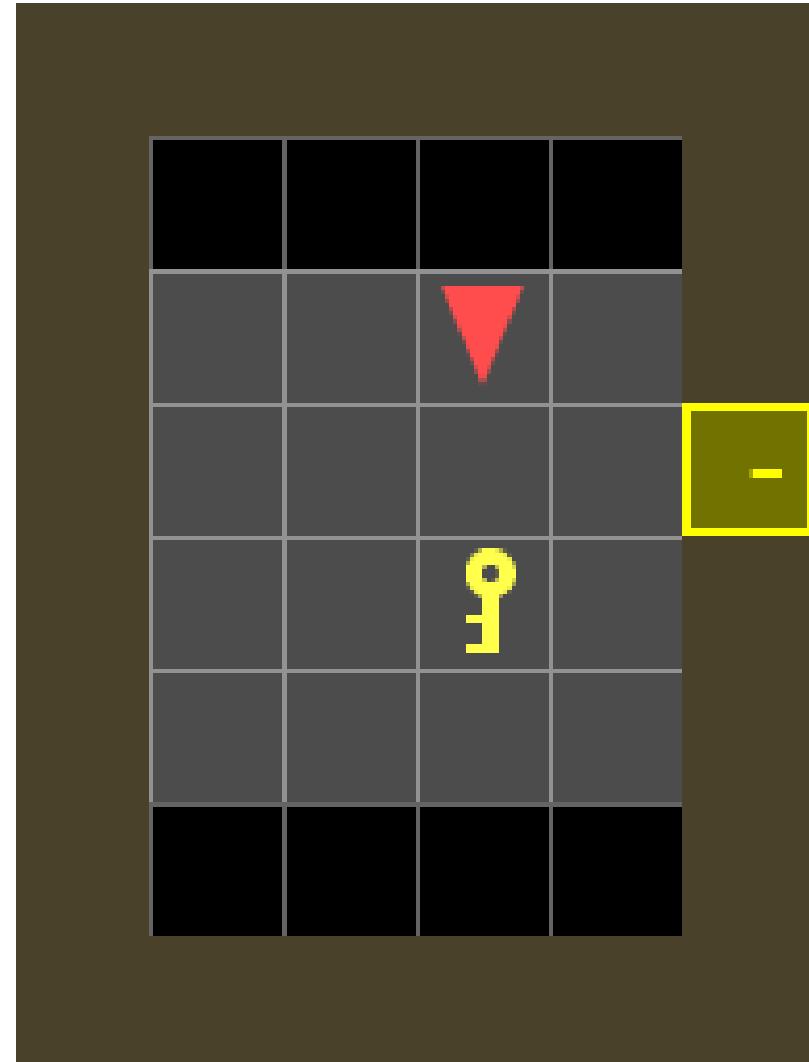
Markov Decision Process (MDP)

Нужно ли агенту помнить свою историю?

Возможные наблюдения:

1. координаты агента
2. полная картинка лабиринта
3. координаты агента + есть ли у него ключ

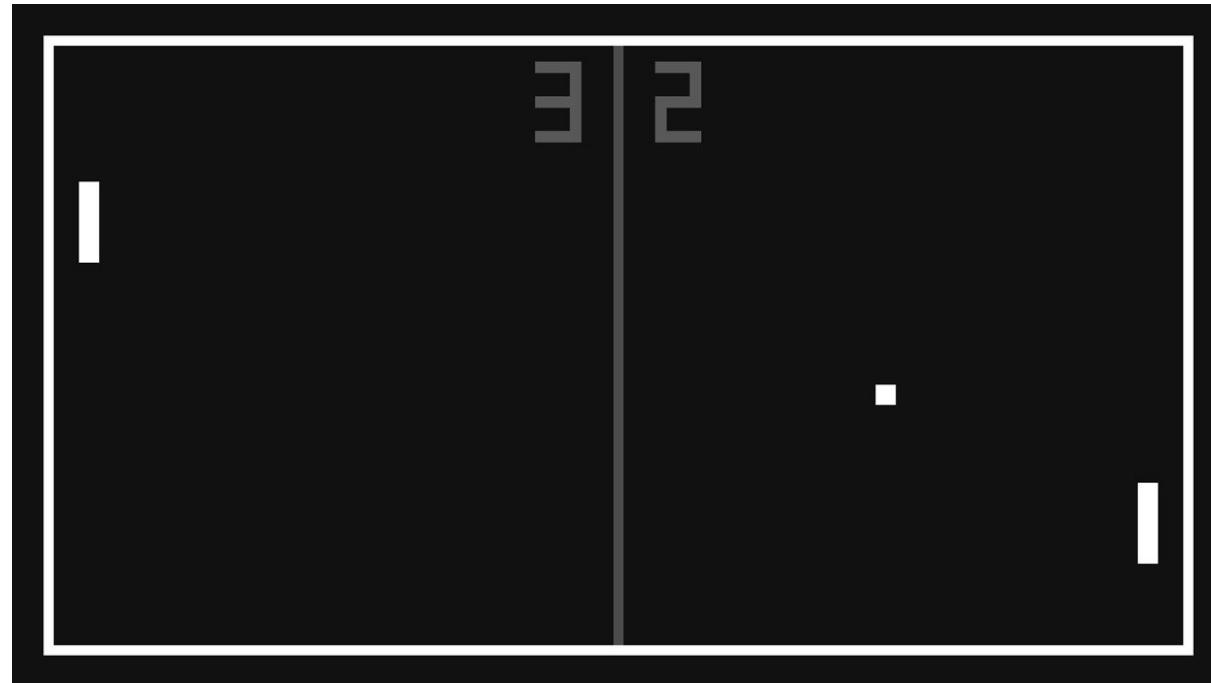
2 и 3 не требуют хранения истории



Марковость: "Будущее не зависит от прошлого, если известно настоящее"

Пример немарковости

Куда летит шарик?



Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

$p(s_{t+1}|s_t, a_t)$ - динамика переходов в среде (марковская)

Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

$p(s_{t+1}|s_t, a_t)$ - динамика переходов в среде (марковская)

$p(s_0)$ - распределение над начальными состояниями

Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

$p(s_{t+1}|s_t, a_t)$ - динамика переходов в среде (марковская)

$p(s_0)$ - распределение над начальными состояниями

$r(s, a)$ - награда за действие a в состоянии s

Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

$p(s_{t+1}|s_t, a_t)$ - динамика переходов в среде (марковская)

$p(s_0)$ - распределение над начальными состояниями

$r(s, a)$ - награда за действие a в состоянии s

$\pi(a|s)$ - политика агента

Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

$p(s_{t+1}|s_t, a_t)$ - динамика переходов в среде (марковская)

$p(s_0)$ - распределение над начальными состояниями

$r(s, a)$ - награда за действие a в состоянии s

$\pi(a|s)$ - политика агента

$p(\tau|\pi) = p(s_0) \prod_{t=0}^T \pi(a_t|s_t)p(s_{t+1}|a_t, s_t)$ - политика агента

где $\tau = (s_0, a_0, s_1, a_1, \dots, s_T, a_T)$ - траектория агента

Задача Reinforcement Learning

$s \sim \mathcal{S}$ - состояния (дискретные \ непрерывные)

$a \sim \mathcal{A}$ - действия (дискретные \ непрерывные)

$p(s_{t+1}|s_t, a_t)$ - динамика переходов в среде (марковская)

$p(s_0)$ - распределение над начальными состояниями

$r(s, a)$ - награда за действие a в состоянии s

$\pi(a|s)$ - политика агента

$p(\tau|\pi) = p(s_0) \prod_{t=0}^T \pi(a_t|s_t)p(s_{t+1}|a_t, s_t)$ - политика агента

где $\tau = (s_0, a_0, s_1, a_1, \dots, s_T, a_T)$ - траектория агента

Не знаем!
Узнаём,
взаимодействуя
со средой

Хотим найти!

Кросс-энтропийный метод для оптимизации

Вообще, мы бы хотели максимизировать по π
среднюю кумулятивную дисконтированную награду:

$$J(\pi) = \mathbb{E}_{p(\tau|\pi)} \sum_{t=0}^T \gamma^t r(s_t, a_t)$$

Пока абстрагируемся!

Пусть есть некоторая функция $f(x) : \mathcal{X} \rightarrow \mathbb{R}$

- Хотим ее максимизировать по x
- Но не умеем считать производную $\frac{\partial f}{\partial x}$

Кросс-энтропийный метод

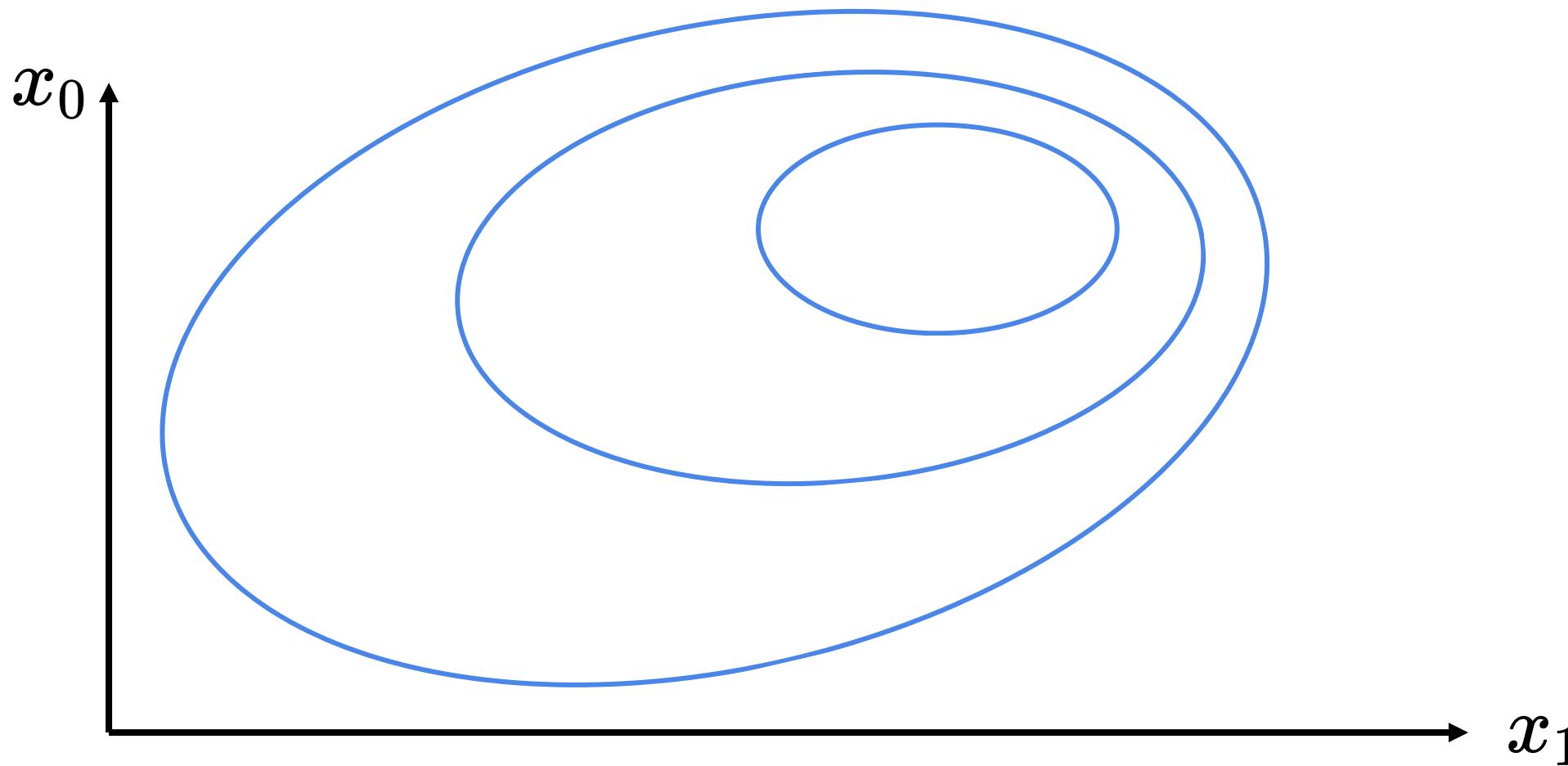
для оптимизации

Идея: вместо оптимального x^* найдем "оптимальное" распределение

Кросс-энтропийный метод

для оптимизации

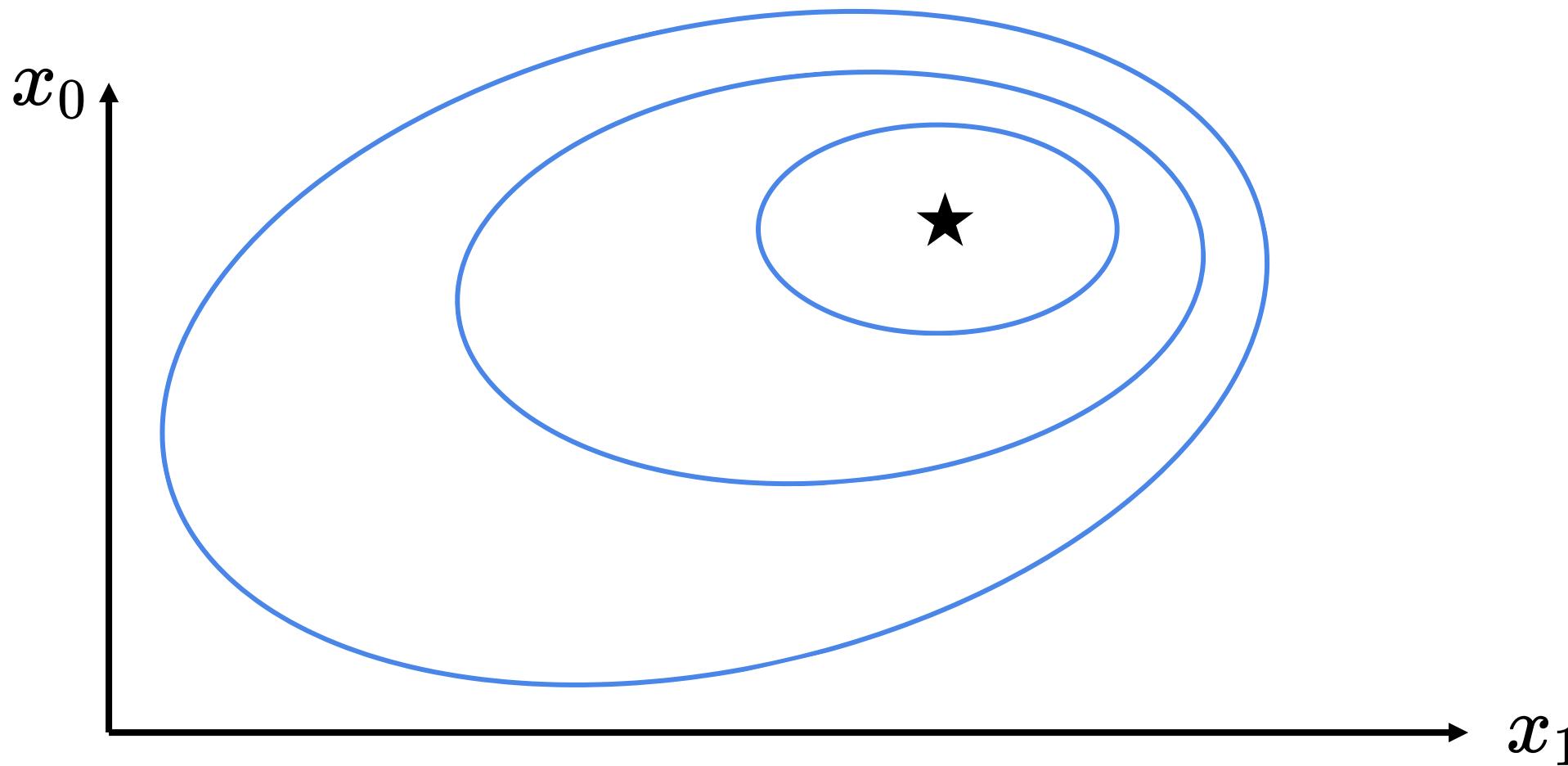
Идея: вместо оптимального x^* найдем "оптимальное" распределение



Кросс-энтропийный метод

для оптимизации

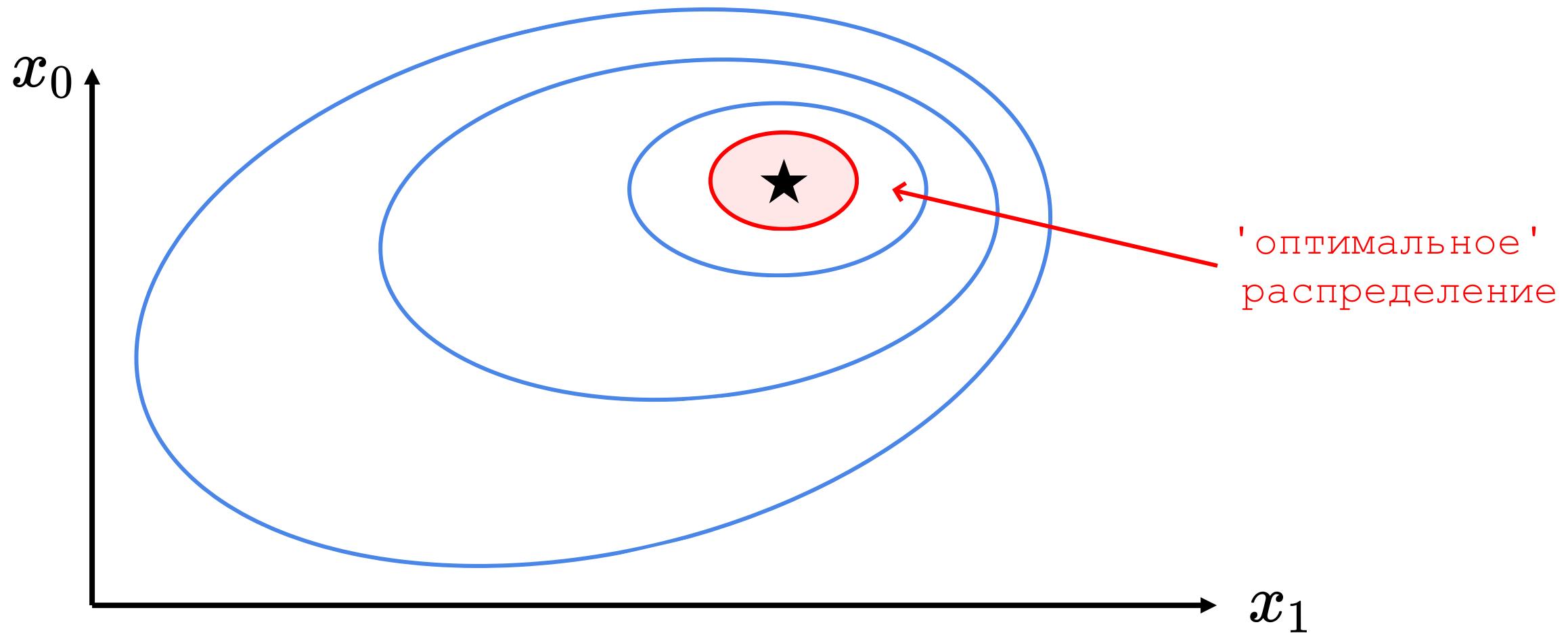
Идея: вместо оптимального x^* найдем "оптимальное" распределение



Кросс-энтропийный метод

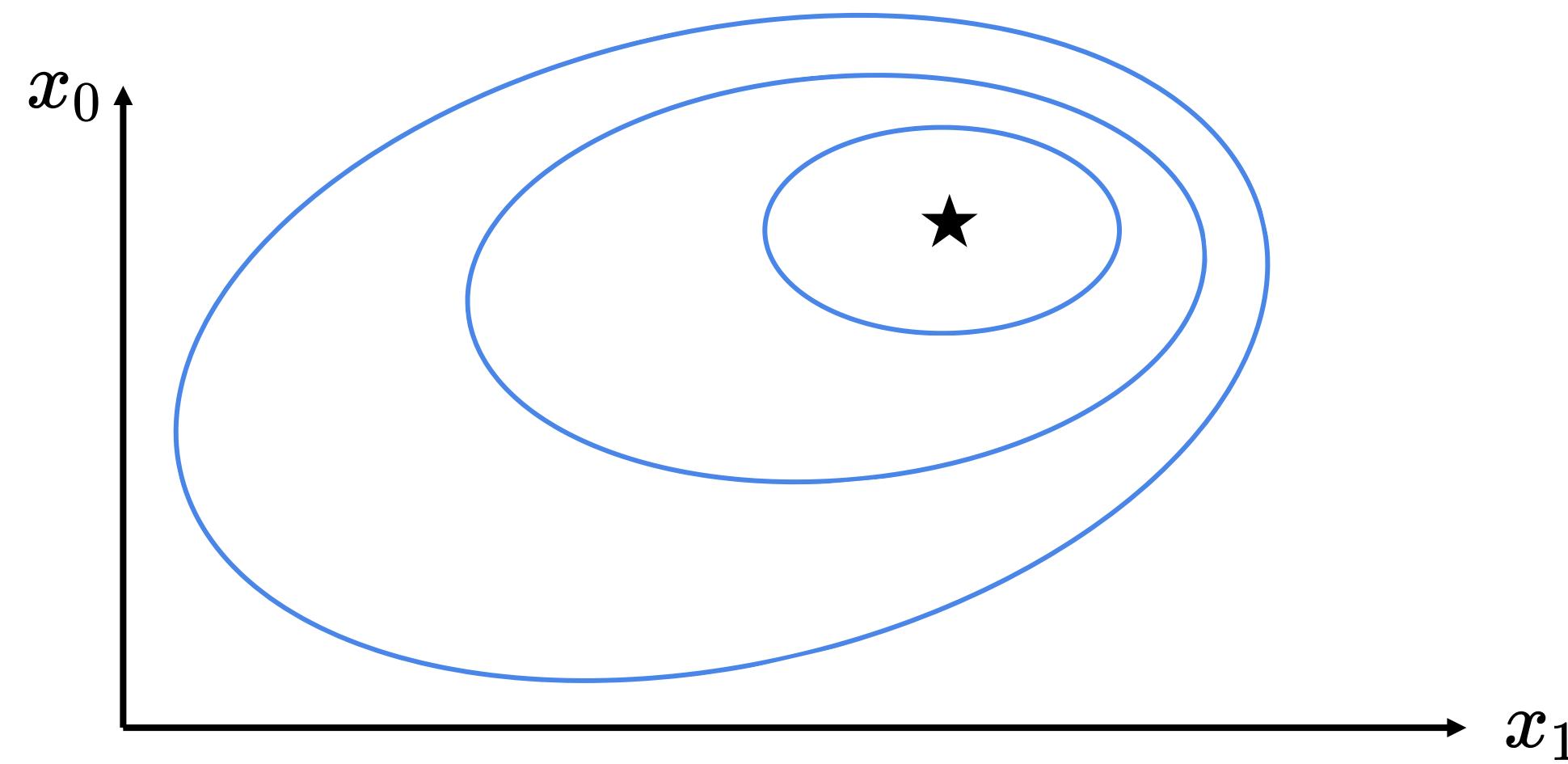
для оптимизации

Идея: вместо оптимального x^* найдем "оптимальное" распределение



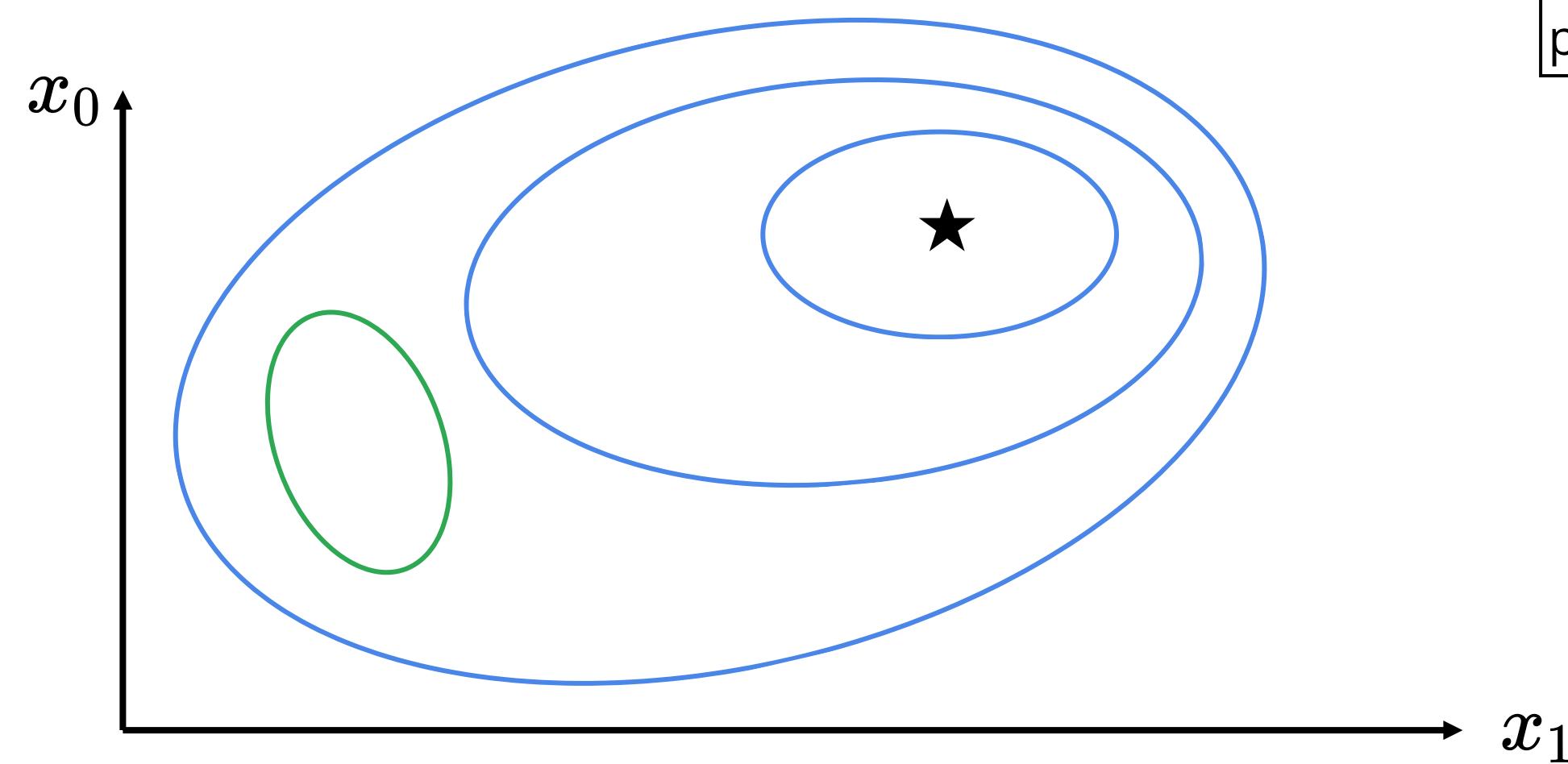
Кросс-энтропийный метод

для оптимизации



Кросс-энтропийный метод

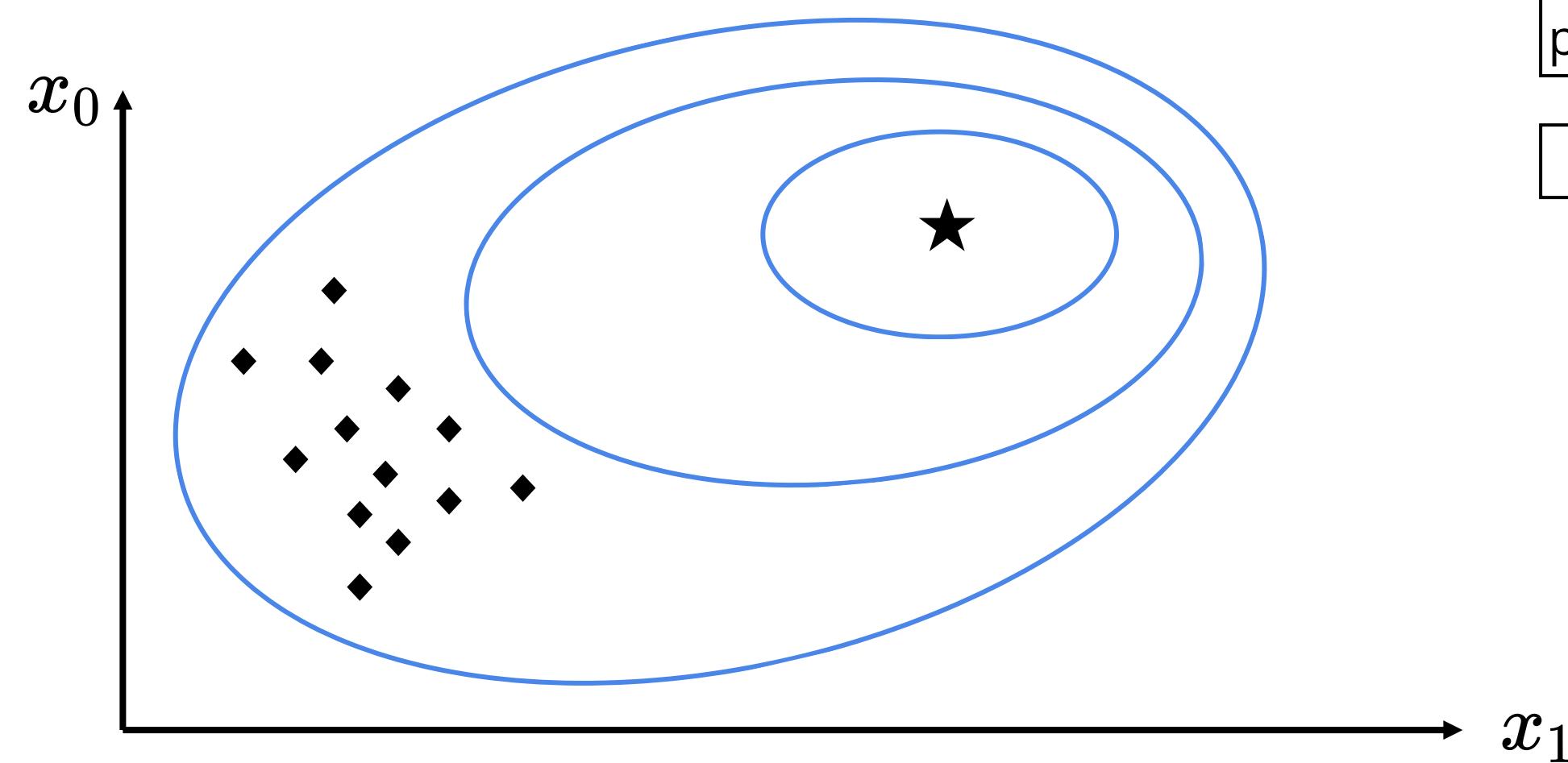
для оптимизации



Инициализируем
распределение q^0

Кросс-энтропийный метод

для оптимизации

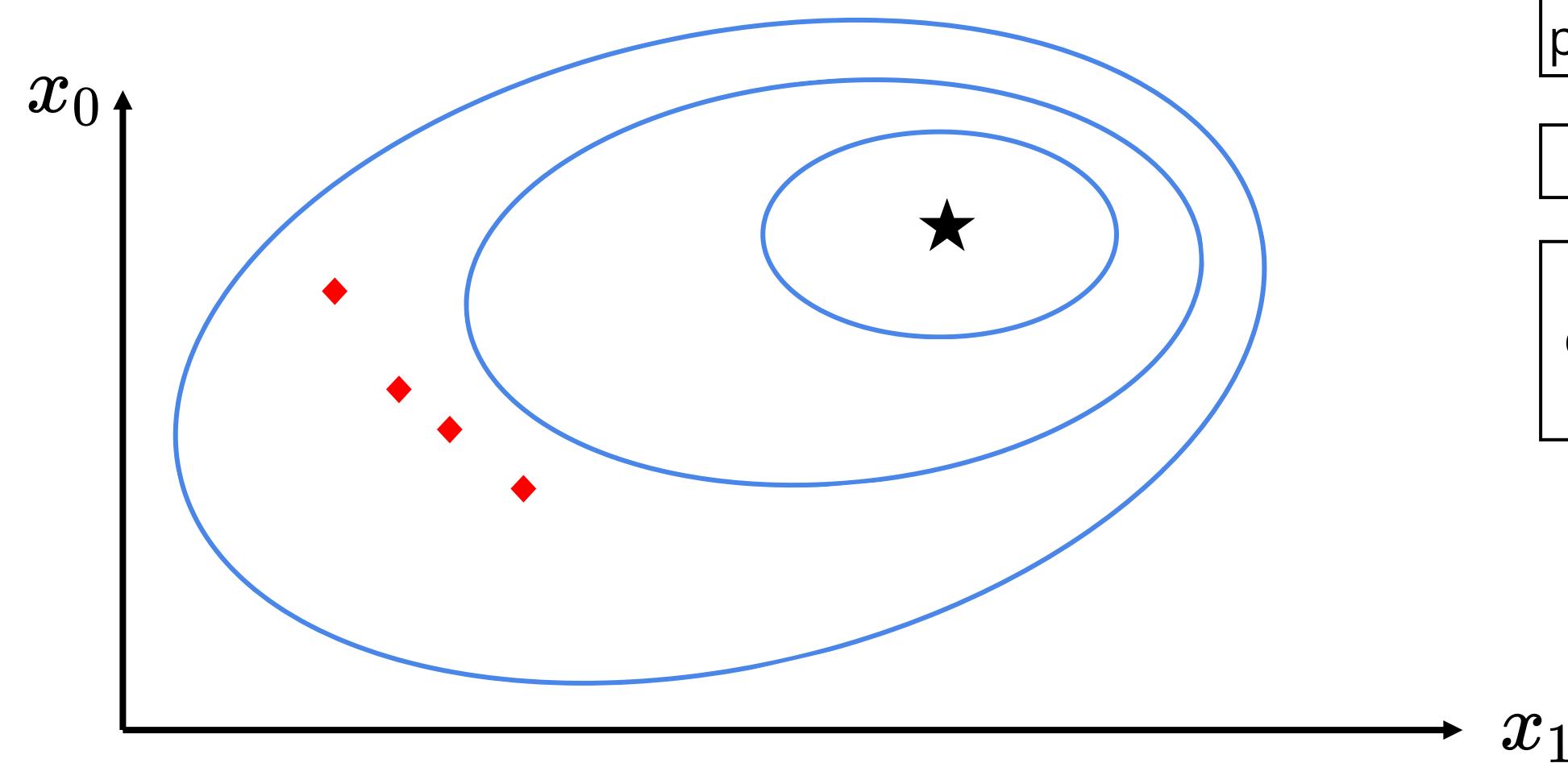


Инициализируем
распределение q^0

Сэмплим $x_i \sim q^0$

Кросс-энтропийный метод

для оптимизации



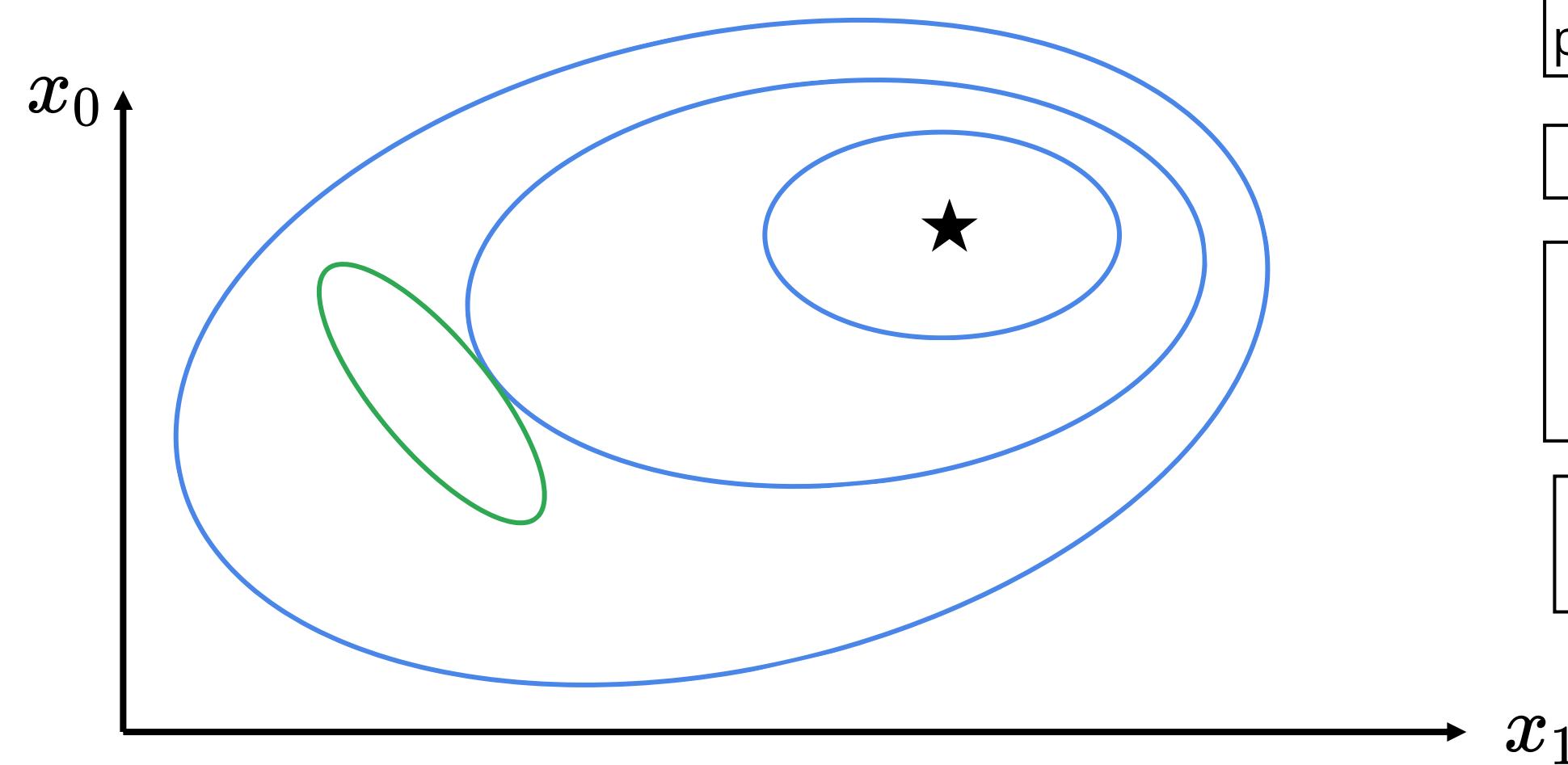
Инициализируем
распределение q^0

Сэмплим $x_i \sim q^0$

Выбираем M x_i
с наибольшим f
- Элиты

Кросс-энтропийный метод

для оптимизации



Инициализируем
распределение q^0

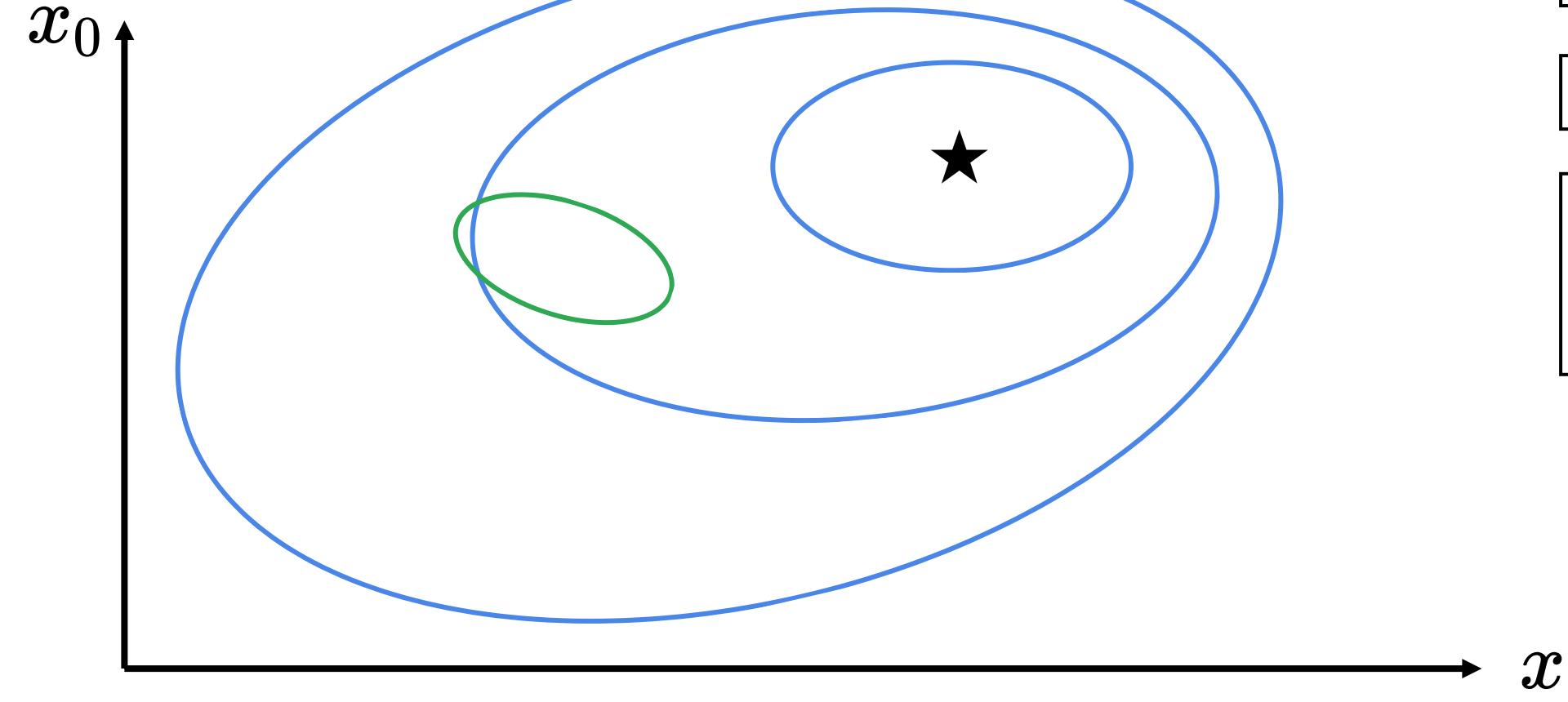
Сэмплим $x_i \sim q^0$

Выбираем M x_i
с наибольшим f
- элиты

Подстраиваем q^1
под элиты

Кросс-энтропийный метод

для оптимизации



Инициализируем
распределение q^0

Сэмплим $x_i \sim q^0$

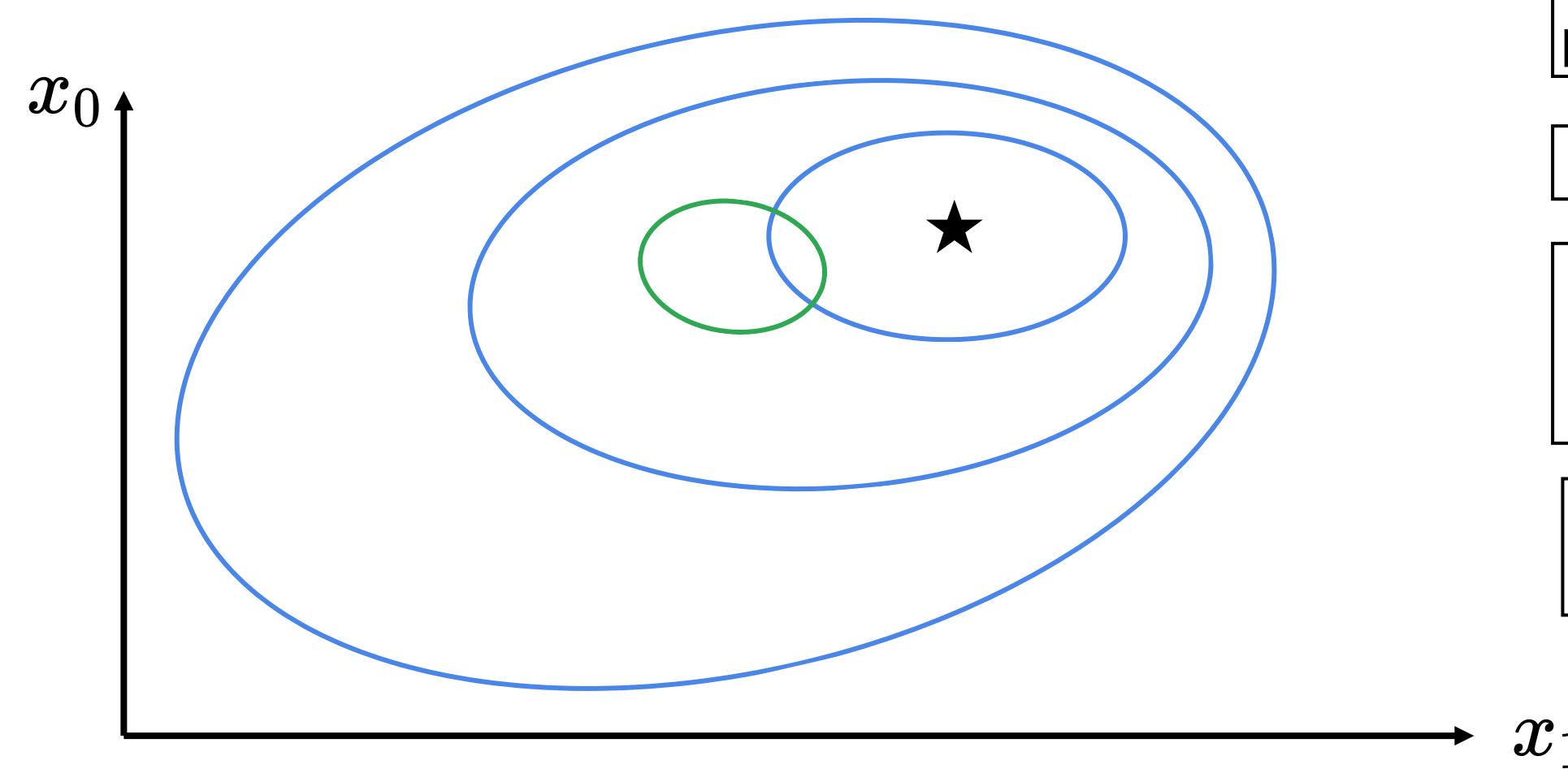
Выбираем M x_i
с наибольшим f
- элиты

Подстраиваем q^1
под элиты

Повторяем!

Кросс-энтропийный метод

для оптимизации



Инициализируем
распределение q^0

Сэмплим $x_i \sim q^0$

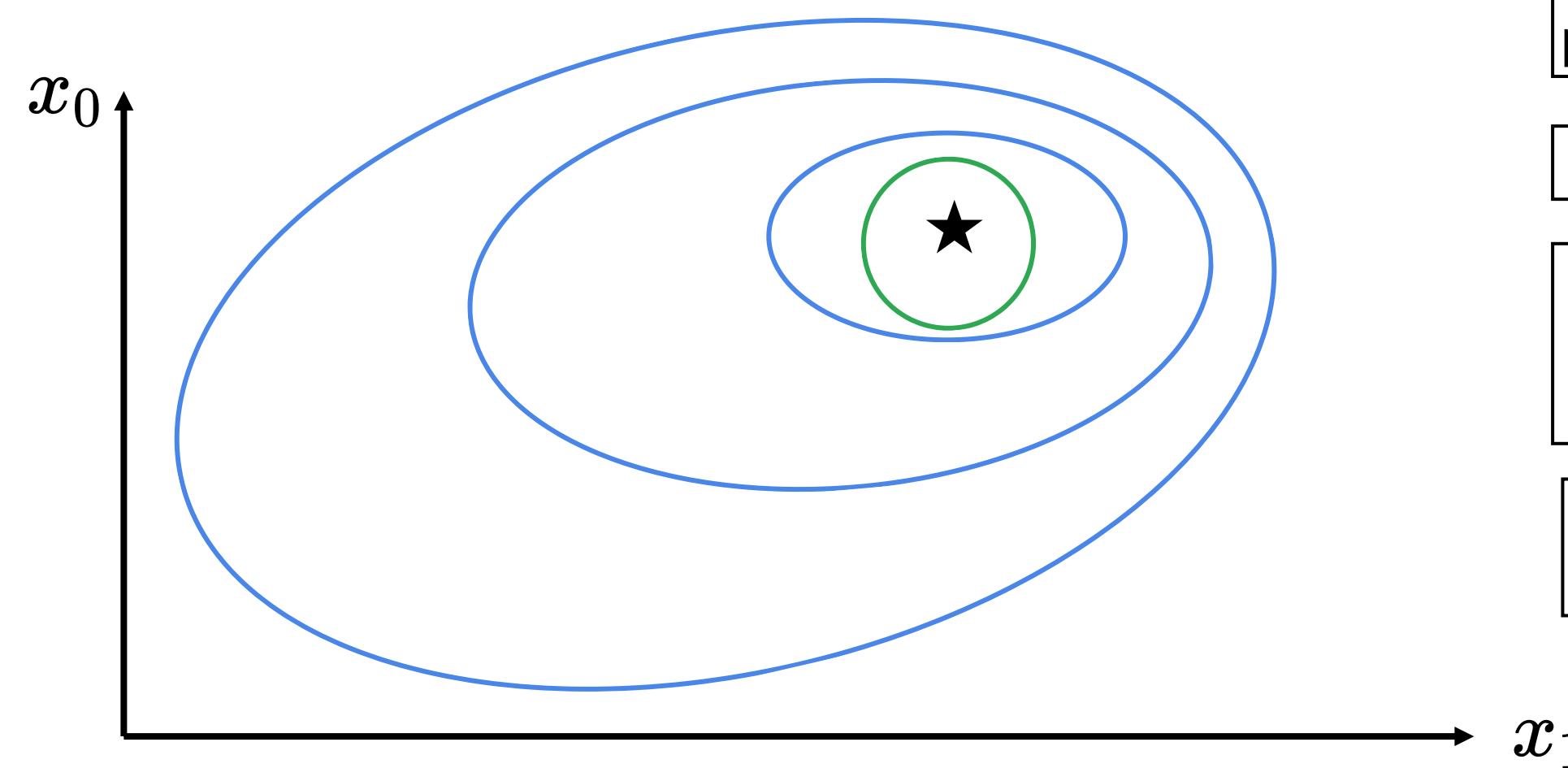
Выбираем M x_i
с наибольшим f
- элиты

Подстраиваем q^1
под элиты

Повторяем!

Кросс-энтропийный метод

для оптимизации



Инициализируем
распределение q^0

Сэмплим $x_i \sim q^0$

Выбираем М x_i
с наибольшим f
- элиты

Подстраиваем q^1
под элиты

Повторяем!

Кросс-энтропийный метод для оптимизации

Чтобы подстроить q под элиты, минимизируем KL-дивергенцию:

$$KL(p_{data} || q) = \int p_{data}(x) \log \frac{p_{data}(x)}{q(x)} dx$$

Кросс-энтропийный метод для оптимизации

Чтобы подстроить q под элиты, минимизируем KL-дивергенцию:

$$KL(p_{data} || q) = \int p_{data}(x) \log \frac{p_{data}(x)}{q(x)} dx$$

$$\min_q KL(p_{data} || q) = \min_q [-\mathbb{E}_{x \sim p_{data}} \log q(x)]$$



минимизация
кросс-энтропии

Кросс-энтропийный метод для оптимизации

Чтобы подстроить q под элиты, минимизируем KL-дивергенцию:

$$KL(p_{data} || q) = \int p_{data}(x) \log \frac{p_{data}(x)}{q(x)} dx$$

$$\min_q KL(p_{data} || q) = \min_q [-\mathbb{E}_{x \sim p_{data}} \log q(x)]$$

минимизация
кросс-энтропии

На k -й итерации:

$$q^{k+1} = \arg \min_q \left[- \sum_{x \in \mathcal{M}^k} \log q(x) \right]$$

где \mathcal{M}^k - элиты, собранные на прошлой итерации

Кросс-энтропийный метод

для Reinforcement Learning

Будем максимизировать $J(\pi_\theta)$ по параметрам θ .

Где π_θ - нейронная сеть, параметризующая распределение.

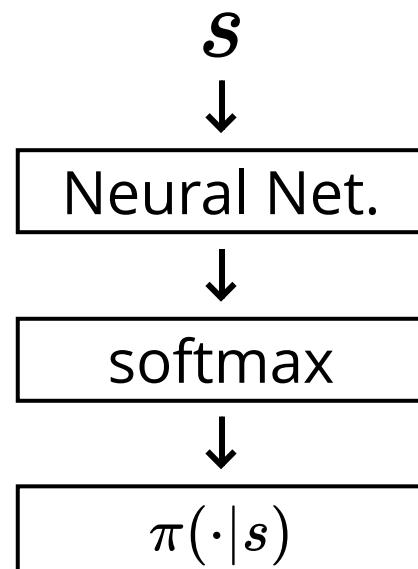
Кросс-энтропийный метод

для Reinforcement Learning

Будем максимизировать $J(\pi_\theta)$ по параметрам θ .

Где π_θ - нейронная сеть, параметризующая распределение.

Например:



(дискретные действия)

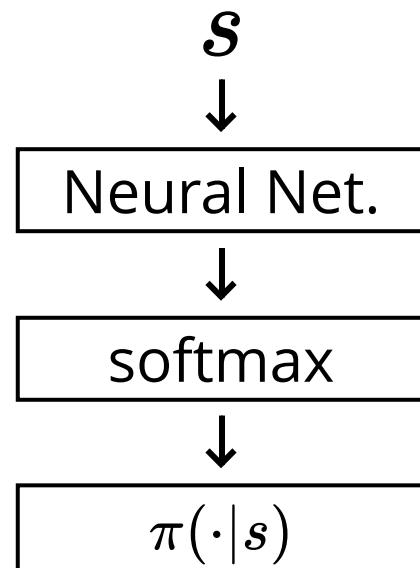
Кросс-энтропийный метод

для Reinforcement Learning

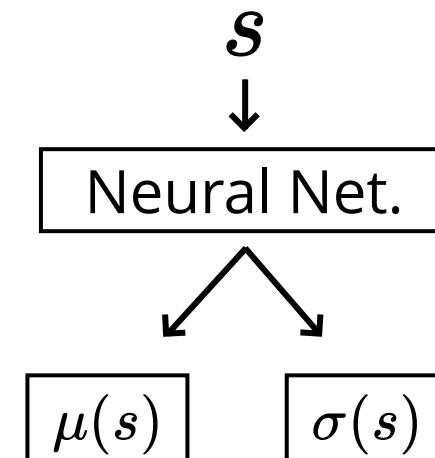
Будем максимизировать $J(\pi_\theta)$ по параметрам θ .

Где π_θ - нейронная сеть, параметризующая распределение.

Например:



(дискретные действия)



(непрерывные действия)

Кросс-энтропийный метод

для Reinforcement Learning

Алгоритм:

- **ввод:** α - процент сохраняемых элит
- инициализируем π_θ
- **повторять**
 - пускаем π_θ собирать траектории $\mathcal{T} = \{\tau_i\}$
 - вычисляем перцентиль $\delta = \text{percentile}(\mathcal{T}, p = \alpha)$
 - выбираем элиты $\mathcal{M} = \text{filter_elites}(\mathcal{T}, \delta)$
 - частично обучаем π_θ прогнозировать a_i по s_i , где $(a_i, s_i) \in \mathcal{M}$