Лекция 10

Обучение с подкреплением

Никита Юдин, iudin.ne@phystech.edu

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

24 апреля 2024

ФПМИ

Если уж мы рассматриваем задачу RL как попытку создания алгоритма «искусственного интеллекта», то мы должны дополнительно учесть следующие три факта:

- понятно, что в одной и той же среде агент может ставить себе совершенно разные задачи; интеллектуальное обучение должно позволять обобщать решения одних задач на другие, решать сложные задачи, состоящие из составных частей, и, наконец, уметь самостоятельно ставить самому себе «промежуточные» задачи.
- в общем случае, текущее наблюдение среды не описывает её состояние полностью, и агент, во-первых, должен обладать модулем памяти для запоминания предыдущих наблюдений, во-вторых, действовать в условиях неопределённости.
- наконец, в среде могут присутствовать другие агенты, которые могут иметь как схожие, так и противоположные цели, передавать вспомогательную информацию или, в частности, играть роль эксперта, демонстрирующих оптимальное (или полезное) поведение.

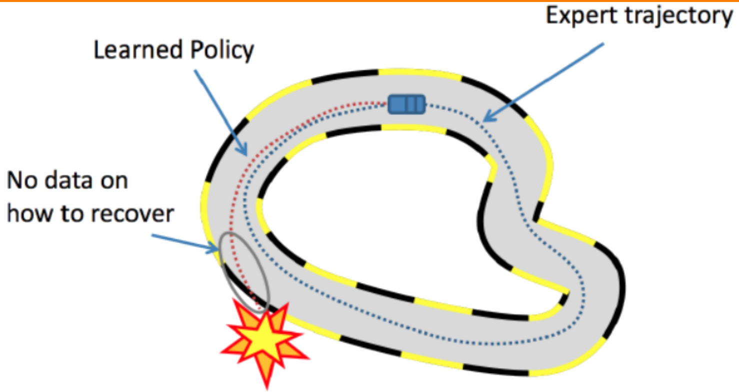
Клонирование поведения

Пример

Чтобы обучить self-driving car, проще посадить реального водителя за руль и попросить собрать примеры траекторий, чем описать функцию награды, описывающую правила движения.

Пример

Чтобы обучить робота переливать воду из стакана в стакан, проще не придумать функцию награды, описывающую такую задачу, а взять руку робота и несколько раз, «держа его за ручку», перелить воду из стакана в стакан.



Определение

Клонированием поведения *(behavioral cloning) называется обучение стратегии воспроизводить действия эксперта:*

$$\sum_{\mathcal{T}} \sum_{s,a \in \mathcal{T}} \log \pi_\theta(a \mid s) \to \max_\theta \,. \tag{1}$$

Пример (DAgger)

Одна из универсальных идей звучит так: после клонирования поведения запустить полученную стратегию в среду, собрать набор тех состояний, которые она посетила, и попросить эксперта «разметить» их: выбрать оптимальные действия. Но такой алгоритм предполагает, что у нас есть подобное «средство разметки», за счёт которого задача и сводится к обучению с учителем.

Пример 1 — Quadcopter Navigation in the Forest: Иногда возможно придумать какое-нибудь ухищрение, как всё-таки получить в RL «правильные ответы». Например, **в одном примере (ссылка)** квадрокоптер учится лететь вдоль лесной тропинки, выбирая на каждом шаге из трёх действий: влево, вперёд или вправо.
Чтобы собрать «обучающую выборку» для него, человек надел шлем с тремя камерами, смотрящими вправо, вперёд и влево, и пошёл по центру лесной тропинки. Собирается такой датасет: для камеры, смотрящей влево, правильный ответ — действие «вправо», для центральной — «вперёд», для правой — «влево». Всё свелось к обучению классификатора.

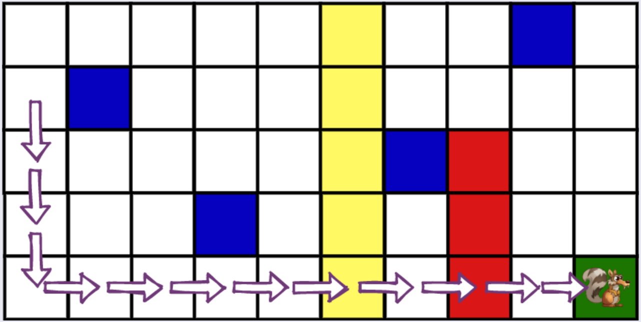
Обратное обучение с подкреплением (Inverse RL)

Определение

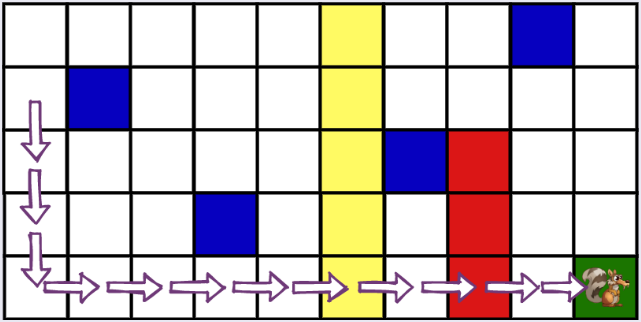
Задачей ***обратного обучения с подкреплением*** *(inverse reinforcement learning, IRL) называется задача по набору траекторий ($\mathcal{T}$) оптимального агента восстановить функцию награды, которую он максимизирует.*

Пример

Рассмотрим клеточный мир, в котором агент может ходить вправо-влево-вниз-вверх. Для простоты также допустим, что функция награды — детерминированная, зависит только от состояний, и на клетках одного цвета её значения совпадают. Что мы тогда можем о ней сказать, имея на руках одну экспертную траекторию, порождённую оптимальной стратегией?

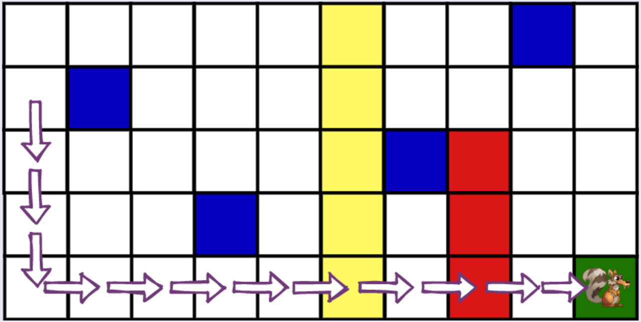


Пример



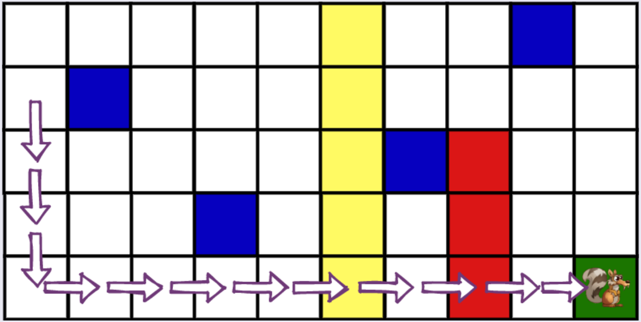
На самом деле, не так много. Агент, видимо, стремился в зелёную клетку; наверное, за неё полагается положительная награда. Через красную клетку агент прошёл, хотя мог бы обойти за счёт более позднего попадания в зелёную клетку; видимо, это того не стоило, и за красную клетку награда, если и отрицательная, то совсем маленькая.

Пример



Могла ли она быть положительной? Тогда бы эксперт, наверное, походил бы по красным клеткам; видимо, награда за зелёную сильно выгоднее. Синие клетки агент избегал; видимо, они или дают штраф, или ноль, поскольку если бы они давали бонус, то было бы выгодно добираться до зелёной клетки-цели через них.

Пример



Наконец, про жёлтые клетки сказать почти ничего нельзя: агенту пришлось бы в любом случае пройти через них, чтобы добраться до зелёной клетки, и поэтому в них может быть как штраф (но не очень большой), так и бонус (но тоже не очень большой).

Для упрощения формул будем везде далее полагать $\gamma = 1$. Введём следующее предположение: оптимальная стратегия π^* стохастична и генерирует такие траектории, что:

$$p(\mathcal{T} \mid \pi^*) \propto e^{R(\mathcal{T})} \prod_{t \geq 0} p(s_{t+1} \mid s_t, a_t). \tag{2}$$

Откуда это предположение свалилось? На самом деле, это уже знакомый нам Maximum Entropy RL. Действительно: рассмотрим задачу поиска стратегии, которая порождает траектории из распределения (2):

$$\mathrm{KL}(p(\mathcal{T} \mid \pi) \parallel p(\mathcal{T} \mid \pi^*)) \to \min_{\pi}. \tag{3}$$

Теорема 1: Задача (3) эквивалентна задаче Maximum Entropy RL.

Доказательство. Распишем (3):

$$\mathrm{KL}(p(\mathcal{T} \mid \pi) \parallel p(\mathcal{T} \mid \pi^*)) = \mathbb{E}_{\mathcal{T}\sim\pi} \overbrace{\sum_{t\geq 0} \log \pi(a_t \mid s_t) + \log p(s_{t+1} \mid s_t, a_t)}^{\log p(\mathcal{T} \mid \pi)} -$$
$$- \mathbb{E}_{\mathcal{T}\sim\pi} \underbrace{\sum_{t\geq 0} \log p(s_{t+1} \mid s_t, a_t) - r_t - \mathrm{const}(\pi)}_{\log p(\mathcal{T} \mid \pi^*) \text{ из } (2)},$$

где $\mathrm{const}(\pi)$ — нормировочная константа распределения (2). Убирая сокращающиеся логарифмы вероятностей переходов и домножая на минус единицу, получаем:

$$\mathbb{E}_{\mathcal{T}\sim\pi} \sum_{t\geq 0} [r_t - \log \pi(a_t \mid s_t)] \to \max_{\pi},$$

что есть в точности Maximum Entropy RL. ■

Утверждение

Ноль в задаче (3) не обязательно достижим.

Контрпример. Рассмотрим MDP с единственным состоянием, в котором от действий ничего не зависит, но с вероятностью 0.5 агент получает $+\log 2$, а с вероятностью 0.5 агент получает $+0$, после чего игра заканчивается. Распределение (2) говорит, что оптимальная стратегия так выбирает действия, что траектория с наградой $+\log 2$ встречает с вероятностью $\frac{2}{3}$, а с наградой $+0$ — с вероятностью $\frac{1}{3}$, однако это невозможно: любая стратегия будет получать их с вероятностями 0.5. ■

Guided Cost Learning

Аппроксимируем функцию награды нейросетью $r_\theta(s, a)$. Тогда правдоподобие одной траектории в предположении (2) равно:

$$p_\theta(\mathcal{T} \mid \pi^*) = \frac{e^{R_\theta(\mathcal{T})} \prod_{t \geq 0} p(s_{t+1} \mid s_t, a_t)}{Z(\theta)},$$

где $R_\theta(\mathcal{T}) := \sum_{s,a \in \mathcal{T}} r_\theta(s, a)$ — текущая аппроксимация кумулятивной награды, а нормировочная константа, внимание, зависит от параметров нейросети θ:

$$Z(\theta) := \int_{\mathcal{T}} e^{R_\theta(\mathcal{T})} \prod_{t \geq 0} p(s_{t+1} \mid s_t, a_t) \, \mathrm{d}\mathcal{T}. \tag{4}$$

Рассмотрим логарифм правдоподобия:

$$\log p_\theta(\mathcal{T} \mid \pi^*) = R_\theta(\mathcal{T}) + \underbrace{\sum_{t \geq 0} \log p(s_{t+1} \mid s_t, a_t)}_{\text{const}(\theta)} - \log Z(\theta). \tag{5}$$

Пусть у нас есть некоторая функция награды с параметрами θ. Пусть $\pi^*_{[\theta]}$ — стратегия, которая оптимально (в терминах Maximum Entropy фреймворка, в рамках предположения (2)) оптимизирует вот эту награду, которую мы предлагаем с текущими параметрами θ. Такая стратегия по определению будет в среде генерировать траектории из распределения:

$$p(\mathcal{T} \mid \pi^*_{[\theta]}) := \frac{e^{R_\theta(\mathcal{T})} \prod_{t \geq 0} p(s_{t+1} \mid s_t, a_t)}{Z(\theta)}. \tag{6}$$

Theorem (Guided Cost Learning)

Градиент для оптимизации правдоподобия (5) *траекторий эксперта по параметрам функции награды θ равен:*

$$\mathbb{E}_{\mathcal{T}\sim\pi^*}\nabla_\theta R_\theta(\mathcal{T}) - \mathbb{E}_{\mathcal{T}\sim\pi^*_{[\theta]}}\nabla_\theta R_\theta(\mathcal{T}). \tag{7}$$

Доказательство. Рассмотрим градиент логарифма правдоподобия одной траектории:

$$\nabla \log p_\theta(\mathcal{T} \mid \pi^*) = \nabla R_\theta(\mathcal{T}) - \nabla \log Z(\theta).$$

Мы хотим оптимизировать правдоподобие в среднем по траекториям эксперта, однако нам нужен градиент нормировочной константы (общий для всех траекторий эксперта, поэтому из мат.ожидания по ним эту константу можно вынести). Рассмотрим дифференцирование нормировочной константы отдельно:

$$\nabla \log Z(\theta) = \{\text{градиент логарифма}\} = \frac{1}{Z(\theta)} \nabla Z(\theta) =$$

$$= \{\text{определение } Z(\theta) \text{ (4)}\} =$$

$$= \frac{1}{Z(\theta)} \int\limits_{\mathcal{T}} \nabla_\theta e^{R_\theta(\mathcal{T})} \prod_{t \geq 0} p(s_{t+1} \mid s_t, a_t) \, \mathrm{d}\mathcal{T} =$$

$$= \{\text{дифференцируем экспоненту}\} =$$

$$= \frac{1}{Z(\theta)} \int\limits_{\mathcal{T}} e^{R_\theta(\mathcal{T})} \prod_{t \geq 0} p(s_{t+1} \mid s_t, a_t) \nabla_\theta R_\theta(\mathcal{T}) \, \mathrm{d}\mathcal{T} =$$

$$= \{\text{определение } \pi^*_{[\theta]} \text{ (6) }\} = \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T} \mid \pi^*_{[\theta]})} \nabla R_\theta(\mathcal{T}). \qquad \blacksquare$$

Generative Adversarial Imitation Learning (GAIL)

Утверждение

Оптимизация функции награды по формуле (7) соответствует оптимизации следующего функционала:

$$\mathbb{E}_{\mathcal{T}\sim\pi^*} R(\mathcal{T}) - \max_{\pi} \mathbb{E}_{\mathcal{T}\sim\pi} \sum_{t\geq 0} \left(r(s_t, a_t) + \mathcal{H}(\pi(\cdot \mid s_t)) \right) \to \max_{r}. \tag{8}$$

Пояснение. Градиент максимума от функции есть градиент этой функции в точке максимума, а энтропия $\mathcal{H}(\pi(\cdot \mid s_t))$ не зависит от r, поэтому градиент второго слагаемого совпадает со вторым слагаемым (7). ∎

Generative Adversarial Imitation Learning (GAIL)

Определение

Occupancy measure *для стратегии* π *будем называть*

$$\rho_\pi(s, a) := \pi(a \mid s) d_\pi(s), \tag{9}$$

где $d_\pi(s)$ *— частоты посещения состояний.*

По определению, мат.ожидания по траектории с таким обозначением можно записывать как

$$\mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t\geq 0} f(s_t, a_t) = \mathbb{E}_{\rho_\pi} f(s, a)$$

Давайте воспользуемся этим обозначением в (8). Раскрывая определение мат.ожидания и переписывая оптимизацию по π как минимизацию, получаем такую «игру» (читать — поиск седловой точки):

$$\min_r \max_\pi \mathbb{E}_{\rho_\pi} r(s, a) - \mathbb{E}_{\rho_{\pi^*}} r(s, a) + \mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t\geq 0}\mathcal{H}(\pi(\cdot \mid s_t)). \tag{10}$$

Утверждение

$$\pi(a \mid s) = \frac{\rho_\pi(s, a)}{\int\limits_{\mathcal{A}} \rho_\pi(s, a)\,\mathrm{d}a} \tag{11}$$

Значит, можно поиск стратегии интерпретировать как поиск occupancy measure: действительно, в (10) последнее слагаемое — суммарный энтропийный бонус стратегии π — тоже можно переписать в терминах $\rho_\pi(s, a)$. Обозначим его как

$$\tilde{\mathcal{H}}(\rho_\pi) := \mathbb{E}_{\mathcal{T} \sim \pi} \sum_{t \geq 0} \mathcal{H}(\pi(\cdot \mid s_t)) = \mathbb{E}_{\rho_\pi} - \log \pi(a \mid s) = \{(11)\} =$$

$$= \mathbb{E}_{\rho_\pi} - \log \frac{\rho_\pi(s, a)}{\int\limits_{\mathcal{A}} \rho_\pi(s, a) \, \mathrm{d}a}.$$

Получаем такой взгляд на процесс обучения:

$$\min_r \max_{\rho_\pi} \mathbb{E}_{\rho_\pi} r(s, a) - \mathbb{E}_{\rho_{\pi^*}} r(s, a) + \tilde{\mathcal{H}}(\rho_\pi). \tag{12}$$

Но как GAN-ы связаны с обратным обучением с подкреплением, с обучением награды? Оказывается, это одно и то же. Чтобы увидеть это в чистом виде, сделаем следующий важный шаг: мы добавим регуляризатор и для оптимизации по r. Это можно мотивировать тем, что, как мы обсуждали ранее, задача обратного обучения с подкреплением «некорректна» и может иметь много разных решений. Итак, добавим некоторое слагаемое $\psi(r)$, которое будет смотреть на нашу выдаваемую функцию награды и некоторым образом её дополнительно штрафовать:

$$\min_r \max_{\rho_\pi} \psi(r) + \mathbb{E}_{\rho_\pi} r(s, a) - \mathbb{E}_{\rho_{\pi^*}} r(s, a) + \tilde{\mathcal{H}}(\rho_\pi) \tag{13}$$

Оказывается, ванильный GAN, различающий сэмплы пар s, a из распределений $p(\mathcal{T} \mid \pi^*)$ и $p(\mathcal{T} \mid \pi)$, соответствует просто определённому выбору регуляризатора!

Теорема

Выберем в (13) следующий регуляризатор:

$$\psi(r) := \mathbb{E}_{\rho_{\pi^*}}\left[r(s,a) + \log(1 - e^{-r(s,a)})\right], \tag{14}$$

где штраф полагается бесконечно большим, если $r(s,a) \leq 0$. *Тогда задача* (13) *примет вид:*

$$\min_D \max_\pi -\mathbb{E}_{\rho_{\pi^*}} \log(1 - D(s,a)) - \mathbb{E}_{\rho_\pi} \log D(s,a) + \tilde{\mathcal{H}}(\rho_\pi), \tag{15}$$

где $D(s,a) \in (0,1)$.

Доказательство. Подставим в (13) выбранный регуляризатор:

$$\min_r \max_\pi \psi(r) + \mathbb{E}_{\rho_\pi} r(s,a) - \mathbb{E}_{\rho_{\pi^*}} r(s,a) + \tilde{\mathcal{H}}(\rho_\pi) =$$

$$= \min_r \max_\pi \mathbb{E}_{\rho_{\pi^*}} \log(1 - e^{-r(s,a)}) + \mathbb{E}_{\rho_\pi} r(s,a) + \tilde{\mathcal{H}}(\rho_\pi).$$

Сделаем замену переменных: вместо оптимизации по $r(s, a) > 0$ будем оптимизировать по $D(s, a)$, где $D(s, a) := e^{-r(s,a)}$ — произвольное число в диапазоне $(0, 1)$. Тогда $r(s, a) = -\log(D(s, a))$ и

$$\min_D \max_\pi -\mathbb{E}_{\rho_{\pi^*}} \log(1 - D(s, a)) - \mathbb{E}_{\rho_\pi} \log D(s, a) + \tilde{\mathcal{H}}(\rho_\pi). \qquad \blacksquare$$

Итак, что мы получили в формуле (15): вместо награды будем обучать дискриминатор, решающий задачу бинарной классификации, где пары s, a из ρ_{π^*} образуют класс 1, а пары s, a из ρ_π — класс 0. Действительно, оптимизация (15) по D при фиксированной π выглядит так:

$$\mathbb{E}_{\rho_{\pi^*}} \log(1 - D(s, a)) + \mathbb{E}_{\rho_\pi} \log D(s, a) \to \max_D,$$

то есть мы просто учимся отличать пары s, a, порождённые (встреченные) экспертом от тех, что встречает наша текущая стратегия. Оптимизация же по π (давайте вернёмся к оптимизации по стратегии) при фиксированном «дискриминаторе» D выглядит так:

$$\mathbb{E}_{\mathcal{T}\sim\pi} \sum_{t\geq 0} \left[-\log D(s_t, a_t) + \mathcal{H}(\pi(\cdot \mid s_t))\right] \to \max_\pi .$$

Generative Adversarial Imitation from Observation (GAIfO)

Часто эксперт предоставляет нам траектории, в которых отсутствует информация о совершённых агентом действиях: есть лишь цепочки состояний $s_0, s_1, s_2, \dots,$ которые посещал эксперт. Такая задача называется **имитационным обучением по наблюдениям** (imitation learning from observations).

Generative Adversarial Imitation from Observation (GAIfO)

Пример

Допустим, у вас есть покадровые анимации того, как персонаж делает сальто. Вы хотите научить робота с такими же конечностями делать тоже самое. В анимациях понятно, в каких координатах находились все конечности персонажа в каждый момент времени, и можно считать, что робот при выполнении задачи должен «посещать» те же цепочки состояний. Однако в анимации действий нету — вы не знаете, как нужно управлять роботом, чтобы получить ту же траекторию в реальной среде.

Мы просто хотим попадать в те же состояния, в которые попадал эксперт, поэтому дискриминатором $D(s) \in (0,1)$ теперь будем пытаться различать состояния — порождены ли они экспертом $s \sim d_{\pi^*}(s)$ или же нашей текущей стратегией $s \sim d_\pi(s)$:

$$\min_D \max_\pi -\mathbb{E}_{d_{\pi^*}(s)} \log(1 - D(s)) - \mathbb{E}_{d_\pi(s)} \log D(s) + \mathbb{E}_{d_\pi(s)} \mathcal{H}(\pi(\cdot \mid s)).$$

В методе Generative Adversarial Imitation from Observation (GAIfO) предлагается сделать хитрость, и различать не состояния, а пары «состояние-следующее состояние» s, s'. Другими словами, награда полагается зависящей от пары состояний $r(s, s')$, а дискриминатор $D(s, s')$ учится различать именно пары s, s' из экспертных траекторий и из порождаемых траекторий.

Формально говоря, вводится альтернативное определение occupancy measure как вероятность встретить ту или иную пару s, s' в траекториях из стратегии π:

$$\nu_\pi(s, s') \coloneqq d_\pi(s) \int_{\mathcal{A}} p(s' \mid s, a)\pi(a \mid s)\,\mathrm{d}a.$$

Тогда минимаксная задача оптимизации принимает следующий вид:

$$\min_D \max_\pi -\mathbb{E}_{\nu_{\pi^*}} \log(1 - D(s, s')) - \mathbb{E}_{\nu_\pi} \log D(s, s') + \mathbb{E}_{d_\pi(s)} \mathcal{H}(\pi(\cdot \mid s)).$$