Лекция 7

Обучение с подкреплением

Никита Юдин, iudin.ne@phystech.edu

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

20 марта 2024

ФФПМИ

Что делали?

Поставили проблему

В сложны средах рандомное исследование плохо работает! Нужно придумать что-то получше.

Что делали?

Поставили проблему

В сложны средах рандомное исследование плохо работает! Нужно придумать что-то получше.

Придумали решение

Нужно поощрять умное исследование.

Что делали?

Markov decision process

- S — множество состояний
- A — множество действий
- $S \times A \times S \to [0, 1]$ — функция вероятности перехода
- $R^{\text{extr}} : S \times A \to R$ — функция наград для исходной задачи (внешняя)

Auxiliary task

- S — множество состояний
- A — множество действий
- $S \times A \times S \to [0, 1]$ — функция вероятности перехода
- $R^{\text{intr}} : S \times A \to R$ — функция наград для лучшего *exploration* (внутренняя)

Что делали?

Пришли к трём решениям

1) Наивно: выдавать награду за посещение любого состояния, в котором за этот эпизод мы ещё не были.
2) Random Network Distillation: по ошибке на вспомогательной задаче регрессии оцениваем новизну состояния и идём в самые новые состояния.
3) Любопытство: шаг в сторону model based алгоритмов — иногда спрашиваем агента: «что будет, если из состояния s совершить действие a?». Если агент угадывает, то тройка (s, a, s') нам неинтересна, иначе возникает любопытство.

Что делали?

Решили проблему неуправляемого шума

Встроили в агента модель, которая по двум соседним состояниям среды предсказывает действие между ними.

Не решили проблему управляемого шума

Агент очень любит залипать над объектами, на которые он может воздействовать, но не может предсказать результаты этих воздействий. Может это и к лучшему?

Напоминание

Определение

$V-$**функцией ценности** (или *value function*) называется функция:

$$V^{\pi}(s) = \mathbb{E}_{p(\tau_t|\pi)}[R_t|s_t = s].$$

Это есть средняя награда по политике π, если агент начинает действовать в момент времени t из состояния s.
$Q-$**функцией ценности** (или *quality function*) называется функция:

$$Q^{\pi}(s, a) = \mathbb{E}_{p(\tau_t|\pi)}[R_t|s_t = s, a_t = a].$$

То же, что V функция, только теперь из состояния $s_t = s$ обязательно сначала совершается действие $a_t = a$.

Напоминание

Определение

Обозначим оптимальную политику (политику, на которой масимизируется $\mathbb{E}_{p(\tau|\pi)}[R]$) как π^*. Тогда оптимальной V-**функцией ценности** называется функция:

$$V^*(s) = \max_\pi \left\{\mathbb{E}_{p(\tau_t|\pi)}[R_t|s_t = s]\right\} = \max_\pi\{V^\pi(s)\} = V^{\pi^*}(s).$$

А оптимальной $Q-$**функцией ценности** называется функция:

$$Q^*(s,a) = \max_\pi \{Q^\pi(s,a)\} = Q^{\pi^*}(s,a).$$

Опять про политику

Напоминание

- *On-policy алгоритмы* — алгоритмы, которые оценивают и улучшают ту же самую политику, которую используют для выбора действий (Target Policy $=$ Behavior Policy).
- *Off-policy алгоритмы* — алгоритмы, которые оценивают и улучшают одну политику, а для выбора действий используют другую политику (Target Policy $\neq$ Behavior Policy).

Что получим в этой лекции

DQN

- off-policy
- одношаговое оценивание политики (смещённая)
- ε-жадная политика
- учим оптимальную Q-функцию Q^*

Policy Gradient

- on-policy
- оценка до конца эпизода (большая дисперсия)
- обучение явной политики как распределения $\pi(a|s)$
- учим V^π

Постановка задачи и вывод алгоритма

План

1) Ввести оптимизационную задачу для *Policy gradient*.
2) Вывести градиент оптимизируемой функции первым способом.
3) Вывести градиент оптимизируемой функции вторым способом.
4) Доказать эквивалентность двух подходов.
5) Продемонстрировать физический смысл.

Первый способ

Постановка задачи

Рассмотрим θ-параметризованное семейство политик $\pi(a|s,\theta)$ или $\pi_\theta(a|s)$. Тогда будем максимизировать следующую величину:

$$V^\pi := J(\theta) := \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t\geq 0}\gamma^t r_t] \to \max_\theta.$$

Первый способ

Подсчет градиента

$$V^{\pi} := \mathbb{E}_{a\sim\pi_\theta} Q^{\pi_\theta}(s,a) = \int\limits_A \pi_\theta(a|s) Q^{\pi_\theta}(s,a) da,$$

тогда

$$\nabla_\theta J(\theta) = \nabla_\theta \int\limits_A \pi_\theta(a|s) Q^{\pi_\theta}(s,a) da = \int\limits_A \nabla_\theta [\pi_\theta(a|s) Q^{\pi_\theta}(s,a)] da =$$

$$= \int\limits_A \nabla_\theta \pi_\theta(a|s) Q^{\pi}(s,a) da + \int\limits_A \pi(a|s) \nabla_\theta Q^{\pi_\theta}(s,a) da.$$

Первый способ

Трюк производной логарифма

$$\nabla_\theta p_\theta(x) = p_\theta(x)\frac{\nabla_\theta p_\theta(x)}{p_\theta(x)} = p_\theta(x)\nabla_\theta \log p_\theta(x)$$

Первое слагаемое

$$\int_A \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a) da = \int_A \pi_\theta(a|s)\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a) da =$$
$$= \mathbb{E}_a \nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)$$

Первый способ

Второе слагаемое

$$\int_A \pi(a|s)\nabla_\theta Q^{\pi_\theta}(s,a)da = \mathbb{E}_a \nabla_\theta Q^{\pi_\theta}(s,a)$$

Итого

$$\nabla_\theta J(\theta) = \mathbb{E}_a[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi}(s,a) + \nabla_\theta Q^{\pi_\theta}(s,a)]$$

Замечание

Мы смогли выразить градиент V-функции через градиент Q-функции, попробуем сделать наоборот.

Первый способ

Q через V

$$\nabla_\theta Q^{\pi_\theta}(s,a) = \nabla_\theta r(s,a) + \nabla_\theta \gamma \, \mathbb{E}_{s'} \, V^{\pi_\theta}(s') =$$
$$= \nabla_\theta \gamma \int_S V^{\pi_\theta}(s') p(s'|s,a) ds' = \gamma \, \mathbb{E}_{s'} \, \nabla_\theta V^{\pi_\theta}(s')$$

Подставляя одно в другое

$$\nabla_\theta J(\theta) = \mathbb{E}_a \, \mathbb{E}_{s'} [\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a) + \gamma \nabla_\theta V^{\pi_\theta}(s')]$$

Первый способ

Замечание

Получили что-то в духе уравнения Беллмана. В правой части стоит матожидание по действию a, совершаемому из состояния s, и по следующему состоянию s'. Раскрутим рекурсию до бесконечности и получим:

Окончательный результат *Policy Gradient Theorem*

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^\pi(s_t, a_t),$$

где τ — траектория согласно политике.

Второй способ

Постановка задачи

Рассмотрим θ-параметризованное семейство политик $\pi(a|s,\theta)$ или $\pi_\theta(a|s)$, которое порождает траектории τ с вероятностями $p_\theta(\tau)$.
Тогда будем максимизировать следующую величину:

$$V^\pi := J(\theta) := \mathbb{E}_{\tau\sim p_\theta(\tau)}[\sum_t \gamma^t r_t] \to \max_\theta.$$

Обозначим награду на траектории τ за $r(\tau) := \sum\limits_{t\geq 0} \gamma^t r_t$.

Второй способ

Вычисляем градиент

По определению матожидания:

$$J(\theta) := \mathbb{E}_{\tau \sim p_\theta(\tau)}[r(\tau)] = \int p_\theta(\tau) r(\tau) d\tau,$$

тогда

$$\nabla_\theta J(\theta) = \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau.$$

Трюк производной логарифма

$$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \frac{\nabla_\theta p_\theta(\tau)}{p_\theta(\tau)} = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$$

Второй способ

Вычисляем градиент

Используем трюк

$$\nabla_\theta J(\theta) = \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau = \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) \cdot r(\tau) d\tau =$$
$$= \mathbb{E}_{\tau \sim p_\theta(\tau)} [\nabla_\theta \log p_\theta(\tau) r(\tau)].$$

Вероятность траектории

$$p_\theta(\tau) = p_\theta(s_0, a_0, \ldots, s_T, a_T, s_{T+1}) = p(s_0) \cdot \prod_{t=0}^{T} (\pi_\theta(a_t|s_t) p(s_{t+1}|a_t, s_t))$$

Второй способ

Логарифм вероятности траектории

$$\log p_\theta(\tau) = \log p(s_0) + \sum_{t=0}^{T} [\log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|a_t, s_t)]$$

Градиент логарифма вероятности траектории

$$\nabla_\theta \log p_\theta(\tau) = \nabla_\theta \sum_{t=0}^{T} [\log \pi_\theta(a_t|s_t)] + \underbrace{\nabla_\theta \sum_{t=0}^{T} [\log p(s_{t+1}|a_t, s_t)]}$$

в среднем, нулевой вектор

по трюку производной логарифма

Второй способ

Конечная формула второго способа

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left(\sum_{t \geqslant 0} \nabla_\theta [\log \pi_\theta(a_t | s_t) r(\tau)] \right)$$

Вывод

Видим, что мы более простым способом получили очень похожую формулу, но с суммарной наградой за игры вместо Q-функции из первого способа. Математически эти формы будут эквивалентны, то есть равны, как интегралы, но их Монте-Карло оценки могут начать вести себя совершенно по-разному.

Эквивалентность

Потихоньку идем к эквивалентности

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim p_\theta(\tau)}[\left(\sum_{t\geqslant 0}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{\hat{t}\geqslant 0}\gamma^{\hat{t}} r_{\hat{t}}\right)] =$$

$$= \mathbb{E}_{\tau\sim p_\theta(\tau)}\sum_{t\geqslant 0}\sum_{\hat{t}\geqslant 0}\nabla_\theta \log \pi_\theta(a_t|s_t)\gamma^{\hat{t}} r_{\hat{t}},$$

выпишем одно из слагаемых этой двойной суммы:

$$j_t := \sum_{\hat{t}\geqslant 0}\nabla_\theta \log \pi_\theta(a_t|s_t)\gamma^{\hat{t}} r_{\hat{t}}.$$

Эквивалентность

Замечание

Видим, что на слагаемое $j_t = \sum_{\hat{t} \geqslant 0} \nabla_\theta \log \pi_\theta(a_t|s_t)\gamma^{\hat{t}} r_{\hat{t}}$, отвечающее за градиент решения выбрать a_t в момент времени t, влияют не только награды после принятия этого решения ($\hat{t} \geqslant t$), но и награды из прошлого ($\hat{t} < t$), то есть некая величина, на которую наше только что принятое решение никак не могло повлиять.

Пример почему это плохо

До момента времени $t_{\text{near end}} \approx T$ агент мог выполнять максимально хоршие действия и $\sum_{\hat{t} < t_{\text{near end}}} \gamma^{\hat{t}} r_{\hat{t}}$ очень большая величина, а вот решение на $t_{\text{near end}}$ может быть просто ужасным и после него награда всегда минимально возможная. Но на градиенте это плохое решение мы не увидим, потому что в сумме награда всё еще большая.

Эквивалентность

Теорема

Для произвольного распределения $\pi_\theta(a)$ верно:

$$\mathbb{E}_{a\sim\pi_\theta(a)} \nabla_\theta \log \pi_\theta(a) = 0.$$

Доказательство

$$\mathbb{E}_{a\sim\pi_\theta(a)} \nabla_\theta \log \pi_\theta(a) = \mathbb{E}_{a\sim\pi_\theta(a)} \frac{\nabla_\theta \pi_\theta(a)}{\pi_\theta(a)} =$$

$$= \int_A \nabla_\theta \pi_\theta(a) da = \nabla_\theta \int_A \pi_\theta(a) da = \nabla_\theta 1 = 0$$

Эквивалентность

Теорема — Принцип причинности

При $\hat{t} < t$:

$$\mathbb{E}_{\tau \sim \pi_\theta} \nabla_\theta \log \pi_\theta(a_t|s_t)\gamma^{\hat{t}} r_{\hat{t}} = 0.$$

Доказательство

По теореме выше:

$$\mathbb{E}_{\tau \sim \pi_\theta} \nabla_\theta [\log \pi_\theta(a_t|s_t)\gamma^{\hat{t}} r_{\hat{t}}] =$$

$$\mathbb{E}_{a_1,s_1,\ldots,s_{\hat{t}},a_{\hat{t}}} \mathbb{E}_{s_{\hat{t}+1},a_{\hat{t}+1},\ldots,s_t,a_t,\ldots} [\nabla_\theta \log \pi_\theta(a_t|s_t)\gamma^{\hat{t}} r_{\hat{t}}] =$$

$$= \mathbb{E}_{a_1,s_1,\ldots,s_{\hat{t}},a_{\hat{t}}} [\gamma^{\hat{t}} r_{\hat{t}} \cdot \mathbb{E}_{s_{\hat{t}+1},a_{\hat{t}+1},\ldots,s_t,a_t,\ldots} \nabla_\theta \log \pi_\theta(a_t|s_t)] = 0.$$

Эквивалентность

Вывод

В формуле полученной вторым способом из суммы можно убрать все слагаемые с $\hat{t} < t$, поскольку они после взятия матожидания обратятся в нуль. Плюсом ко всему, вычеркивание этих слагаемых уменьшит дисперсию при оценке градиента, полученного вторым способом, по методу Монте-Карло.

Замечание

Дисконтирование в среде идет с самого начала, поэтому дисконтирующий фактор при слагаемых $\hat{t} \geqslant t$ своей степени не поменяет, а значит его можно переписать как:

$$\sum_{\hat{t} \geqslant t} \gamma^{\hat{t}} r_{\hat{t}} = \gamma^t \sum_{\hat{t} \geqslant t} \gamma^{\hat{t}-t} r_{\hat{t}} =: \gamma^t r_t(\tau).$$

Эквивалентность

Приблизись к эквивалентной форме

На данный момент имеем:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \sum_{t \geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) r_t(\tau).$$

Неформальное доказательство эквивалентности

$r_t(\tau)$ — очень похож на Q-функцию, поскольку является её несмещенной Монте-Карло оценкой, а в формуле выше всё равно берется матожидание, поэтому формулы из первого способа и из второго — одно и то же.

Эквивалентность

Теорема об эквивалентности

Следующие формулы эквивалентны:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^\pi(s_t, a_t),$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim p_\theta(\tau)} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) r_t(\tau).$$

Эквивалентность

Доказательство

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim p_\theta(\tau)} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) r_t(\tau) =$$

$$= \sum_{t\geqslant 0} \mathbb{E}_{\tau\sim p_\theta(\tau)}\, \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) r_t(\tau) =$$

$$= \sum_{t\geqslant 0} \mathbb{E}_{a_0,s_1,\ldots,s_t,a_t}\, \mathbb{E}_{s_{t+1},a_{t+1},\ldots}\, \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) r_t(\tau) =$$

$$= \sum_{t\geqslant 0} \mathbb{E}_{a_0,s_1,\ldots,s_t,a_t}\, \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t)\, \mathbb{E}_{s_{t+1},a_{t+1},\ldots}\, r_t(\tau) =$$

$$= \sum_{t\geqslant 0} \mathbb{E}_{a_0,s_1,\ldots,s_t,a_t}\, \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^\pi(s_t,a_t) =$$

$$= \mathbb{E}_{\tau\sim\pi} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^\pi(s_t,a_t)$$

Физический смысл

Заметим

Оказывается, градиент нашего функционала имеет вид градиента взвешенных логарифмов правдоподобий. Чтобы ещё лучше увидеть это, рассмотрим **суррогатную функцию** — другой функционал, который будет иметь в точке текущих значений параметров стратегии π такой же градиент, как и $J(\theta)$:

Определение суррогатной функции

$$\mathcal{L}_{\tilde{\pi}}(\theta) := \mathbb{E}_{\tau \sim \tilde{\pi}} \sum_{t \geqslant 0} \gamma^t \log \pi_\theta(a|s) Q^{\tilde{\pi}}(s, a)$$

Физический смысл

Что дальше?

Получили суррогатную функция от двух стратегий: стратегии π_θ, которую мы оптимизируем, и ещё одной стратегии $\tilde{\pi}$. Давайте рассмотрим эту суррогатную функцию в точке θ такой, что эти две стратегии совпадают: $\pi_\theta = \tilde{\pi}$, и посмотрим на градиент при изменении θ, только одной из них. Буквально мы «заморозим» оценочную Q-функцию, и «заморозим» распределение, из которого приходят пары (s, a).

Утверждение

$$\nabla_\theta \mathcal{L}_{\tilde{\pi}}(\theta)\big|_{\tilde{\pi}=\pi_\theta} = \nabla_\theta J(\theta)$$

Физический смысл

Доказательство

Поскольку мат.ожидание по траекториям не зависит в этой суррогатной функции от θ, то градиент просто можно пронести внутрь:

$$\nabla_\theta \mathcal{L}_{\tilde{\pi}}(\theta)|_{\tilde{\pi}=\pi_\theta} = \mathbb{E}_{\tau\sim\tilde{\pi}} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a|s)|_{\tilde{\pi}=\pi_\theta} Q^{\tilde{\pi}}(s,a).$$

В точке θ такой, что $\pi_\theta = \tilde{\pi}$ верно, что $p(\tau|\tilde{\pi}) \equiv p(\tau|\pi_\theta)$ и $Q^{\tilde{\pi}}(s,a) = Q^{\pi}(s,a)$; следовательно, значение градиента в этой точке совпадает со значением формулы для $\nabla_\theta J(\theta)$.

Физический смысл

Вывод

Значит, направление максимизации $J(\theta)$ в текущей точке θ просто совпадает с направлением максимизации этой суррогатной функции! Таким образом, можно считать, что в текущей точке мы на самом деле «как бы» максимизируем, а это уже в чистом виде логарифм правдоподобия каких-то пар (s, a), для каждой из которых дополнительно выдан «вес» в виде значения $Q^{\pi}(s, a)$.

Физический смысл

Например

Если в машинном обучении в задачах регрессии и классификации мы для данной выборки (x, y) максимизировали правдоподобие:

$$\sum_{(x,y)} \log p(y|x,\theta) \to \max_{\theta},$$

то теперь в RL, когда выборки нет, мы действуем по-другому: мы сэмплируем сами себе входные данные s и примеры выходных данных a, выдаём каждой паре какой-то «кредит доверия», некую скалярную оценку хорошести, выраженную в виде $Q^{\pi}(s,a)$, и идём в направлении максимизации:

$$\sum_{(x,y)} \log p(y|x,\theta) Q^{\pi}(s,a) \to \max_{\theta}.$$

Способ получения

Monte-Carlo

Воспользуемся первой формулой для подсчета *Policy Gradient*:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi} \sum_{t\geqslant 0} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^\pi(s_t, a_t),$$

в которой заменим всё неизвестное на оценку по Монте-Карло.

Способ получения

Что будем заменять?

- $\mathbb{E}_{\tau \sim \pi} \rightarrow$ сыграем несколько полных игр при помощи текущей стратегии π — алгоритм будет *on-policy*.
- $Q^{\pi}(s_t, a_t) \rightarrow r(\tau)$ — можно сказать, что мы воспользовались вторым способом подсчета *Policy Gradient* с Монте-Карло оценкой $\mathbb{E}_{\tau \sim \pi}$, что не удивляет, ведь подходы, как мы уже показали, эквиваленты.

Итоговый алгоритм

Reinforce

Гиперпараметры: N — количество игр, $\pi(a|s,\theta)$ — стратегия с параметрами θ, SGD-оптимизатор.

0. Произвольно инициализируем θ.

На очередном шаге t

1. Играем N игр $\tau_1, \tau_2, \ldots, \tau_N \sim \pi$.
2. Для каждого t в каждом τ_i считаем $r_t(\tau_i) := \sum_{\hat{t} \geq t} \gamma^{\hat{t}-t} r^i_{\hat{t}}$.
3. Считаем оценку градиента:

$$\nabla_\theta J(\pi) := \frac{1}{N} \sum_{i=1}^{N} \sum_{t \geqslant 0} \gamma^t \nabla_\theta \log \pi(a^i_t|s^i_t,\theta) r_t(\tau_i).$$

4. Делаем шаг градиентного подъёма по θ, используя $\nabla_\theta J(\pi)$.

Итоговый алгоритм

Недостатки

1. Для одного шага градиентного подъёма нам необходимо играть несколько игр **до конца** при помощи текущей стратегии.
2. Колоссальная дисперсия нашей оценки градиента — на практике дождаться каких-то результатов от такого алгоритма в сколько-то сложных задачах не получится.

Два типа стохастики

Мотивация

До этого мы часто работали с функционалами вида $\mathbb{E}_{\tau\sim\pi}\sum_{t\geqslant 0}\gamma^t f(s_t, a_t)$, где f — какая-то функция от пар состояние-действие.

Стохастика

В MDP есть два вида стохастики:

1) **внешняя** связанная со случайностью в самой среде и неподконтрольная агенту; она заложена в функции переходов $p(s'|s, a)$;
2) **внутренняя** , связанная со случайностью в стратегии самого агента; она заложена в $\pi(a|s)$. Это стохастика нам подконтрольна при обучении.

Два типа стохастики

Мотивация

Матожидание $\mathbb{E}_{\tau \sim \pi}$ плохо тем, что мат.ожидания по внешней и внутренней стохастике чередуются. При этом во время обучения из внешней стохастики мы можем только получать сэмплы, поэтому было бы здорово переписать наш функционал как-то так, чтобы он имел вид мат.ожидания по всей внешней стохастике.

Два типа стохастики

Утверждение

Состояния, которые встречает агент со стратегией π, приходят из некоторой стационарной марковской цепи.

Доказательство

Выпишем вероятность оказаться на очередном шаге в состоянии s', если мы используем стратегию π:

$$p(s'|s) = \int_A \pi(a|s)p(s'|s,a)da.$$

Эта вероятность не зависит от времени и от истории, следовательно, цепочка состояний образует марковскую цепь.

Два типа стохастики

Допустим, начальное состояние s_0 фиксировано. Обозначим вероятность оказаться в состоянии s в момент времени t при использовании стратегии π как $p(s_t = s|\pi)$.

Определение

Для данного MDP и политики π **state visitation frequency** называется:

$$\mu_\pi(s) := \lim_{T\to\infty} \frac{1}{T} \sum_{t\geqslant 0}^{T} p(s_t = s|\pi).$$

Введём ещё один, «дисконтированный счётчик посещения состояний» для стратегии взаимодействия π. При дисконтировании отпадают проблемы с нормировкой.

Два типа стохастики

Определение

Для данного MDP и политики π **discounted state visitation distribution** называется

$$d_\pi(s) := (1-\gamma)\sum_{t\geqslant 0}\gamma^t p(s_t = s|\pi).$$

Утверждение

State visitation distribution есть распределение на множестве состояний, то есть:

$$\int_S d_\pi(s)ds = 1.$$

Два типа стохастики

Доказательство

$$= \int_S d_\pi(s)ds = \int_S (1-\gamma)\sum_{t\geqslant 0}\gamma^t p(s_t = s|\pi)ds =$$

$$= (1-\gamma)\sum_{t\geqslant 0}\gamma^t \int_S p(s_t = s|\pi)ds = 1$$

Два типа стохастики

Теорема

Для произвольной функции $f(s, a)$:

$$\mathbb{E}_{\tau \sim \pi} \sum_{t \geqslant 0} \gamma^t f(s_t, a_t) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\pi(s)} \mathbb{E}_{a \sim \pi(a|s)} f(s, a).$$

Начало доказательства

$$\mathbb{E}_{\tau \sim \pi} \sum_{t \geqslant 0} \gamma^t f(s_t, a_t) = \sum_{t \geqslant 0} \gamma^t \mathbb{E}_{\tau \sim \pi} f(s_t, a_t) =$$

$$= \sum_{t \geqslant 0} \gamma^t \int_S \int_A p(s_t = s, a_t = a|\pi) f(s, a) da ds =$$

Два типа стохастики

Продолжение доказательства

$$= \sum_{t \geqslant 0} \gamma^t \int_S \int_A p(s_t = s|\pi)\pi(a|s) f(s,a) da ds =$$

$$= \sum_{t \geqslant 0} \gamma^t \int_S p(s_t = s|\pi)\, \mathbb{E}_{\pi(a|s)} f(s,a) ds =$$

$$= \int_S \sum_{t \geqslant 0} \gamma^t p(s_t = s|\pi)\, \mathbb{E}_{\pi(a|s)} f(s,a) ds =$$

$$= \int_S \frac{d_\pi(s)}{1-\gamma} \mathbb{E}_{\pi(a|s} f(s,a) ds = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\pi(s)} \mathbb{E}_{\pi(a|s)} f(s,a)$$

Два типа стохастики

Пример

$$J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\pi(s)} \mathbb{E}_{\pi(a|s)} r(s,a)$$

Пример

$$\nabla_\theta J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\pi(s)} \mathbb{E}_{\pi(a|s)} \nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)$$

Уменьшаем дисперсию Reinforce. Baseline.

Мотивация

При стохастической оптимизации ключевым фактором является дисперсия оценки градиента. Когда мы заменяем мат.ожидания на Монте-Карло оценки, дисперсия увеличивается. Понятно, что замена Q-функции — выинтегрированных будущих наград — на её Монте-Карло оценку в REINFORCE повышало дисперсию. Однако, в текущем виде основной источник дисперсии заключается в другом.

Уменьшаем дисперсию Reinforce. Baseline.

Причина большой дисперсии

Градиент логарифма правдоподобия в среднем равен нулю. Это значит, что если для данного s мы выдаём некоторое распределение $\pi(a|s)$, для увеличения вероятностей в одной области A нужно данный вес θ_i параметризации увеличивать, а в другой области — уменьшать. В среднем «магнитуда изменения» равна нулю. Но у нас в Монте-Карло оценке только $a \sim \pi(a|s)$, и для него направление изменения домножится на кредит: на нашу оценку $Q^\pi(s,a)$). Если эта оценка в одной области 100, а в другой 1000 — дисперсия получаемых значений $\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)$становится колоссальной.

Вывод

Кредит надо центрировать нулем!

Уменьшаем дисперсию Reinforce. Baseline.

Утверждение

Для произвольной функции $b(s) : S \to \mathbb{R}$, называемой бэйзлайном, верно:

$$\nabla_\theta J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{\pi(a|s)} \nabla_\theta \log \pi_\theta(a|s)(Q^\pi(s,a) - b(s)).$$

Доказательство

Добавленное слагаемое есть ноль в силу формулы теоремы о среднем градиента логарифма.

Уменьшаем дисперсию Reinforce. Baseline.

Замечание

Это верно для произвольной функции от состояний и становится неверно, если вдруг бэйзлайн $b(s)$ начинает зависеть от a. Мы вольны выбрать бэйзлайн произвольно; он не меняет среднего значения оценок градиента, но изменяет дисперсию.

Уменьшаем дисперсию Reinforce. Baseline.

Теорема

Бэйзлайном, максимально снижающим дисперсию Монте-Карло оценок формулы градиентов, является

$$b^*(s) := \frac{\mathbb{E}_a \|\nabla_\theta \log \pi(a,s)\|_2^2 Q^\pi(s,a)}{\mathbb{E}_a \|\nabla_\theta \log \pi(a,s)\|_2^2}$$

Проблема

Практическая ценность результата невысока. Знать норму градиента для всех действий a вычислительно будет труднозатратно даже в дискретных пространствах действий.

Уменьшаем дисперсию Reinforce. Baseline.

На практике

$$b^*(s) := \frac{\mathbb{E}_a \|\nabla_\theta \log \pi(a,s)\|_2^2 Q^\pi(s,a)}{\mathbb{E}_a \|\nabla_\theta \log \pi(a,s)\|_2^2} =$$

$$= [\|\nabla_\theta \log \pi(a,s)\|_2^2 \approx \text{const}(a)] \approx \mathbb{E}_a\, Q^\pi(s,a) = V^\pi(s)$$

Итого

$$\nabla_\theta J(\pi) = \frac{1}{1-\gamma}\, \mathbb{E}_{s\sim d_\pi(s)}\, \mathbb{E}_{\pi(a|s)}\, \nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)$$

Уменьшаем дисперсию Reinforce. Baseline.

Итого

$$\nabla_\theta J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{\pi(a|s)} \nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)$$

Определение

Для данного MDP **Advantage-функцией** политики π называется

$$A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s).$$

Схемы «актор-критик»

- Хотим оптимизировать параметры стратегии при помощи формулы градиента, не доигрывая эпизоды до конца.
- Введём вторую сетку, которая будет «оценивать» наши собственные решения — критика (critic). Нейросеть, моделирующую стратегию, соответственно будем называть актёром или актором (actor), и такие алгоритмы, в которых обучается как модель критика, так и модель актора, называются Actor-Critic.

Схемы «актор-критик»

- В качестве критика обычно учат именно V-функцию.
- Возможность не обучать сложную Q^* является одним из преимуществ подхода прямой оптимизации $J(\theta)$.
- Из соображений эффективности:

$$Q^\pi(s,a) = r(s,a) + \gamma\, \mathbb{E}_{s'}\, V^\pi(s') \approx r(s,a) + \gamma V_\phi(s'), \quad s' \sim p(s' \mid s,a).$$

Схемы «актор-критик». Bias-variance trade-off

Собираемся вместо честного advantage подставить некоторую его оценку (advantage estimator) и провести таким образом credit assingment:

$$\nabla_\theta J(\pi) \approx \frac{1}{1-\gamma} \mathbb{E}_{d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)} \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\approx A^\pi(s,a)}.$$

В контексте policy gradient, речь напрямую идёт о дисперсии и смещении оценок градиента.

$\Psi(s,a)$	**Дисперсия**	**Смещение**
$R_t - V_\phi(s)$	высокая	нет
$r(s,a) + \gamma V_\phi(s') - V_\phi(s)$	низкая	большое

Схемы «актор-критик». Построение оценки Q-функции

$$Q^{\pi}(s,a) \approx \sum_{t=0}^{N-1} \gamma^t r^{(t)} + \gamma^N V_{\phi}(s^{(N)}).$$

Для credit assingment-а N-шаговой оценкой Advantage, или N-шаговой временной разностью:

$$\Psi_{(N)}(s,a) := \sum_{t=0}^{N-1} \gamma^t r^{(t)} + \gamma^N V_{\phi}(s^{(N)}) - V_{\phi}(s).$$

С ростом N дисперсия такой оценки увеличивается: всё больший фрагмент траектории мы оцениваем по Монте-Карло, нам становятся нужны сэмплы $a_{t+1} \sim \pi(a_{t+1} \mid s_{t=1})$, $s_{t+2} \sim \pi(s_{t+2} \mid s_{t+1}, a_{t+1})$, $\ldots$, $s_{t+N} \sim \pi(s_{t+N} \mid s_{t+N-1}, a_{t+N-1})$.

Схемы «актор-критик». Построение оценки Q-функции

Определение

Для пар s_t, a_t из роллаута $s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_N$ длины N будем называть оценкой максимальной длины (max trace estimation) оценку с максимальным заглядыванием в будущее: для Q-функции

$$y^{\text{MaxTrace}}(s_t, a_t) := \sum_{\hat{t}=t}^{N-1} \gamma^{\hat{t}-t} r_{\hat{t}} + \gamma^{N-t} V^{\pi}(s_N), \tag{1}$$

для Advantage функции, соответственно:

$$\Psi^{\text{MaxTrace}}(s_t, a_t) := y^{\text{MaxTrace}}(s_t, a_t) - V^{\pi}(s_t). \tag{2}$$

Generalized Advantage Estimation (GAE)

Решение дилеммы bias-variance trade-off подсказывает теория TD(λ) оценки. Нужно применить формулу TD(λ) и просто заансамблировать N-шаговые оценки разной длины:

Определение

GAE-оценкой Advantage-функции называется ансамбль многошаговых оценок, где оценка длины N берётся с весом λ^{N-1}, где $\lambda \in (0, 1)$ — гиперпараметр:

$$\Psi_{\mathrm{GAE}}(s, a) := (1 - \lambda) \sum_{N>0} \lambda^{N-1} \Psi_{(N)}(s, a).$$

Как мы помним, при $\lambda \to 0$ такая GAE-оценка соответствует одношаговой оценке; при $\lambda = 1$ GAE-оценка соответствует Монте-Карло оценке Q-функции.

Generalized Advantage Estimation (GAE)

В текущем виде в формуле суммируются все N-шаговые оценки вплоть до конца эпизода. В реальности собранные роллауты могут прерваться в середине эпизода: допустим, для данной пары s, a через M шагов роллаут «обрывается». Тогда на практике используется чуть-чуть другим определением GAE-оценки: если мы знаем $s^{(M)}$, но после этого эпизод ещё не доигран до конца, мы пользуемся формулой TD(λ) и оставляем от суммы только «доступные» слагаемые:

$$\Psi_{\mathrm{GAE}}(s, a) := \sum_{t \geq 0}^{M-1} \gamma^t \lambda^t \Psi_{(1)}(s^{(t)}, a^{(t)}). \tag{3}$$

Generalized Advantage Estimation (GAE)

Утверждение

Формула (3) эквивалентна следующему ансамблю N-шаговых оценок:

$$\Psi_{\mathrm{GAE}}(s,a) = (1-\lambda)\sum_{N>0}^{M-1}\lambda^{N-1}\Psi_{(N)}(s,a) + \lambda^{M-1}\Psi_{(M)}(s,a).$$

В такой «обрезанной» оценке $\lambda = 1$ соответствует оценке максимальной длины (2), а $\lambda = 0$ всё ещё даст одношаговую оценку.

Generalized Advantage Estimation (GAE)

В коде формула (3) очень удобна для рекурсивного подсчёта оценки; также для практического алгоритма осталось учесть флаги done_t. Формулы подсчёта GAE-оценки для всех пар (s, a) из роллаута $s_0, a_0, r_0, s_1, a_1, r_1, \cdots s_N$ приобретают такой вид:

$$
\begin{aligned}
\Psi_{\mathrm{GAE}}(s_{N-1}, a_{N-1}) &:= \Psi_{(1)}(s_{N-1}, a_{N-1}) \\
\Psi_{\mathrm{GAE}}(s_{N-2}, a_{N-2}) &:= \Psi_{(1)}(s_{N-2}, a_{N-2}) + \\
&\quad + \gamma\lambda(1 - \mathrm{done}_{N-2})\Psi_{\mathrm{GAE}}(s_{N-1}, a_{N-1}) \\
&\vdots \\
\Psi_{\mathrm{GAE}}(s_0, a_0) &:= \Psi_{(1)}(s_0, a_0) + \gamma\lambda(1 - \mathrm{done}_0)\Psi_{\mathrm{GAE}}(s_1, a_1)
\end{aligned}
$$

Заметим, что эти формулы очень похожи на расчёт кумулятивной награды за эпизод, где «наградой за шаг» выступает $\Psi_{(1)}(s, a)$.

Обучение критика

Воспользуемся идеей перехода к регрессии, которую мы обсуждали раньше в контексте DQN. Нам нужно просто решать методом простой итерации уравнение Беллмана:

$$V_{\phi_{k+1}}(s) \leftarrow \mathbb{E}_a \left[r + \gamma \, \mathbb{E}_{s'} \, V_{\phi_k}(s') \right].$$

Воспользуемся преимуществами on-policy режима и поймём, что мы можем поступить точно также, как с оценкой Q-функции в формуле градиента: решать многошаговое уравнение Беллмана вместо одношагового. Например, можно выбрать любое N-шаговое уравнение и строить целевую переменную как

$$y := r + \gamma r' + \gamma^2 r'' + \cdots + \gamma^N V_{\phi_k}(s^{(N)}). \tag{4}$$

Обучение критика

Тогда если мы оцениваем Advantage как

$$\Psi(s, a) = y - V_\phi(s),$$

где y — некоторая оценка Q-функции, то y же является и таргетом для V-функции, и наоборот. Используя функцию потерь MSE с таким таргетом, мы как раз и учим среднее значение наших оценок Q-функции, то есть бэйзлайн.
Конечно же, мы можем использовать и GAE-оценку (3) Advantage, достаточно «убрать бэйзлайн»:

$$Q^\pi(s, a) = A^\pi(s, a) + V^\pi(s) \approx \Psi_{\text{GAE}}(s, a) + V_\phi(s).$$

Утверждение

Таргет $\Psi_{\mathrm{GAE}}(s,a) + V_\phi(s)$ является несмещённой оценкой правой части «ансамбля» уравнений Беллмана:

$$V_\phi(s) = (1-\lambda)\sum_{N>0}\lambda^{N-1}\left[\mathfrak{B}^N V_\phi\right](s),$$

где $\mathfrak{B}$ — оператор Беллмана для V-функции.

Доказательство.

По определению, поскольку $\Psi_{(N)}(s,a) + V(s)$ является несмещённой оценкой правой части N-шагового уравнения Беллмана (т. е. несмещённой оценкой $\left[\mathfrak{B}^N V^\pi\right](s)$), а

$$(1-\lambda)\sum_{N>0}\lambda^{N-1}(\Psi_{(N)}(s,a) + V(s)) = \Psi_{\mathrm{GAE}}(s,a) + V(s).$$

□

Обучение критика

Делаем несколько шагов взаимодействия со средой, собирая таким образом роллаут некоторой длины N; считаем для каждой пары s, a некоторую оценку Q-функции $y(s, a)$, например, оценку максимальной длины (1); оцениваем Advantage каждой пары как $\Psi(s, a) := y(s, a) - V_\phi(s)$; далее по Монте-Карло оцениваем градиент по параметрам стратегии

$$\nabla_\theta J(\pi) \approx \frac{1}{N} \sum_{s,a} \nabla_\theta \log \pi_\theta(a \mid s) \Psi(s, a)$$

и градиент для оптимизации критика (допустим, критик — Q-функция):

$$\text{Loss}^{\text{critic}}(\phi) = \frac{1}{N} \sum_{s,a} \big(y(s, a) - V_\phi(s)\big)^2$$

Advantage Actor-Critic (A2C)

Гиперпараметры: M — количество параллельных сред, N — длина роллаутов, $V_\phi(s)$ — нейросеть с параметрами ϕ, $\pi_\theta(a \mid s)$ — нейросеть для стратегии с параметрами θ, SGD оптимизатор.

Инициализировать θ, ϕ

На каждом шаге:

1. в каждой параллельной среде собрать роллаут длины N, используя стратегию π_θ:

$$s_0, a_0, r_0, s_1, \ldots, s_N$$

Advantage Actor-Critic (A2C)

2 для каждой пары s_t, a_t из каждого роллаута посчитать оценку Q-функции максимальной длины, игнорируя зависимость оценки от ϕ:

$$Q(s_t, a_t) := \sum_{\hat{t}=t}^{N-1} \gamma^{\hat{t}-t} r_{\hat{t}} + \gamma^{N-t} V_\phi(s_N)$$

3 вычислить лосс критика:

$$\text{Loss}^{\text{critic}}(\phi) := \frac{1}{MN} \sum_{s_t, a_t} \big(Q(s_t, a_t) - V_\phi(s_t)\big)^2$$

4 делаем шаг градиентного спуска по ϕ, используя $\nabla_\phi \text{Loss}^{\text{critic}}(\phi)$

Advantage Actor-Critic (A2C)

5 вычислить градиент для актора:

$$\nabla_\theta^{\text{actor}} := \frac{1}{MN} \sum_{s_t, a_t} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \left(Q(s_t, a_t) - V_\phi(s_t) \right)$$

6 сделать шаг градиентного подъёма по θ, используя $\nabla_\theta^{\text{actor}}$

Advantage Actor-Critic (A2C). Примеры

Пример

Представьте, что вы стоите перед кофеваркой и у вас есть два действия: получить кофе и не получить. Вы прекрасно знаете, какое действие лучше другого, но при этом не оцениваете, сколько кофе вы сможете выпить в будущем при условии, например, что сейчас вы выберете первый или второй вариант: то есть не строите в явном виде прогноз значения оценочной функции.

Advantage Actor-Critic (A2C). Примеры

Пример

Допустим, вы играете в видео-игру, и в начале обучения мало что умеете, всё время действуя примерно случайно и падая в первую же яму. Среда выдаёт вам всё время ноль, и вы продолжаете вести себя случайно. Вдруг в силу стохастичности стратегии вы перепрыгиваете первую яму и получаете монетку +1. В DQN этот ценнейший опыт будет сохранён в буфере, и постепенно критик выучит, какие действия привели к награде. В A2C же агент сделает один небольшой шаг изменения весов моделей и тут же выкинет все собранные данные в мусорку, потому что на следующих итерациях он никак не может переиспользовать их. Агенту придётся ждать ещё много-много сессий в самой игре, пока он не перепрыгнет яму снова, чтобы сделать следующий шаг обучения перепрыгиванию ям.