Оптимизация в задачах обучения с подкреплением

Методы оптимизации в машинном обучении

Никита Юдин, iudin.ne@phystech.edu

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

3 мая 2024

Марковский процесс принятия решений (MDP)

В обучении с подкреплением взаимодействия между агентом и окружающей средой часто описываются марковским процессом принятия решений (MDP). Различают:

- дисконтированный марковский процесс принятия решений (γ-*Discounted Markov Decision Process*, DMDP);
- MDP с усредненным вознаграждением (*infinite-horison Average reward Markov Decision Process*, AMDP);
- эпизодический марковский процесс принятия решений (*H-episodic Markov Decision Process*, HMDP);
- другие, в том числе и *частично наблюдаемые*.

Марковский процесс принятия решений (MDP)

Марковский процесс принятия решений представляет собой систему, которая со временем $(t = 0, 1, 2, \dots)$ претерпевает случайные изменения и обозначается кортежем $M = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$ со следующими объектами:

(i) $\mathcal{S}$ – пространство состояний, $S := |\mathcal{S}|$ – количество уникальных состояний.

(ii) $\mathcal{A}$ – пространство действий, $A := |\mathcal{A}|$ – количество уникальных действий.

(iii) $p\left(s, a; s'\right)$ – вероятность перехода из состояния $s \in \mathcal{S}$ в момент времени t с определенным действием $a \in \mathcal{A}$ в состояние $s' \in \mathcal{S}$ в момент $(t+1)$ (при этом $\sum\limits_{s' \in \mathcal{S}} p\left(s, a; s'\right) = 1$, $p\left(s, a; s'\right) \equiv \mathrm{P}\left(s' \middle| s, a\right)$). Функция вероятности $p(\cdot)$ вместе с функцией вероятности $\mathrm{P}(a|s)$ задают вероятности перехода для Марковского ядра, оно же ядро MDP.

Марковский процесс принятия решений (MDP)

(iv) Функция награды
$r_\xi(s,a) : \Omega \times \mathcal{S} \times \mathcal{A} \to [0,1]$, ($\mathbb{E}_\xi \left[r_\xi(s,a) \right] = r(s,a)$, где $\mathbb{E}[\cdot]$ – математическое ожидание). В зависимости от постановки задачи функция награды может зависеть от следующего за состоянием s состояния s':

$$r_\xi \left(s, a; s' \right) : \Omega \times \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1], \ \mathbb{E}_\xi \left[r_\xi \left(s, a; s' \right) \right] = r \left(s, a; s' \right).$$

Стоит отметить, что мы предполагаем в общем случае стохастическую природу функции награды в зависимости от случайной величины $\xi \in \Omega$. При детерминированном вычислении награды относительно фиксированных (s,a) или (s,a,s') мы просто опускаем обозначение ξ в $r_\xi(\cdot)$ и математическое ожидание по нему.

Марковский процесс принятия решений (MDP)

(iv) *(продолжение)* Здесь и далее используется предположение об ограниченности награды за каждое действие, поэтому без ограничений общности использованы приведённые выше определения функции $r(\cdot)$. В работе используются детерминированные относительно своих аргументов награды, если не оговорено иное.

(v) $\gamma \in (0, 1]$ – коэффициент дисконтирования для DMDP, для AMDP $\gamma = 1$, но просто положив $\gamma = 1$ из DMDP не сделать AMDP, понадобится ещё усреднение суммарной награды агента за взаимодействие с MDP по времени.

Нередко рассматривается более общая форма $M = (\mathcal{S}, \mathcal{A}, p, r, \mu_0, \gamma)$, в которой μ_0 – вероятностное распределение начального состояния $s_0 \sim \mu_0$, при явном отсутствии μ_0 происходит обуславливание всех вычислений на $s_0 \in \mathcal{S}$.

MDP. Принятие решений

- Здесь и далее приводятся результаты для дискретных $\mathcal{S}$ и $\mathcal{A}$ с конечными мощностями, однако они могут быть обобщены на непрерывный случай заменой суммы по переменной в области её непрерывности на соответствующий интеграл по области.
- Стратегией принятия решений или политику агента, принимающего решение в MDP, обозначим через символ π и присвоим ему отображения, задающие вероятностную меру на пространстве действий:

$$\pi(a|s) \equiv \mathrm{P}(a|s) \text{ в общем случае, } \hat{a} \sim \pi(\cdot|s) \text{ или } \pi(s) \sim \pi(\cdot|s);$$

$$\hat{a} := \pi(s) \textit{ в случае вырожденного распределения: } \mathrm{P}(\hat{a}|s) = 1.$$

MDP. Ядро

- Введённое распределение позволяет явно записать Марковское ядро перехода между состояниями $s \mapsto s'$, оно же ядро MDP, его также корректно называть Марковским ядром, обусловленным политикой π:

$$\mathrm{P}^{\pi}(s' | s) := \sum_{a \in \mathcal{A}} \pi(a|s) p(s, a; s').$$

- В процессах с конечным количеством состояний Марковское ядро можно задать с помощью матрицы:

$$P^{\pi} = \left(\sum_{a \in \mathcal{A}} \pi(a|s) p(s, a; s') \right)_{s \in \mathcal{S},\ s' \in \mathcal{S}}, \quad s \text{ – строка, } s' \text{ – столбец.}$$

MDP. Ядро

В процессе взаимодействия с марковским процессом стратегия π собирает траекторию $\tau_t := (s_0, a_0, r_0, \dots, s_t, a_t, r_t)$. Её правдоподобие выражается следующим образом:

$$\mathrm{P}\left(\tau_{H-1} | \pi\right) = \mu_0(s_0) \prod_{t=0}^{H-2} \left(\pi(a_t | s_t) p(s_t, a_t; s_{t+1})\right) \pi(a_{H-1} | s_{H-1}).$$

DMDP. V-функция ценности

Для фиксированной политики и начального состояния $s_0 = s$ определяется V-функция значений (ценности) $V^\pi : \mathcal{S} \to \mathbb{R}$ как дисконтированная сумма будущих вознаграждений:

$$V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \middle| \pi,\ s_0 = s\right],$$

где s_t – состояние системы в момент времени t, $a(s_t)$ – выбор действия в соответствии с политикой $\pi(\cdot)$. Это есть средняя награда по политике π, если агент начинает действовать в момент времени t из состояния s. Иногда индекс политики опускают: $V(s) := V^\pi(s)$.

Замечание

В определении V-функции в левой части выражения отсутствует обозначение t в силу однородности MDP.

DMDP. Q-функция ценности

Схожим образом задается Q-функция ценности $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \middle| \pi,\; s_0 = s, a_0 = a\right].$$

То же, что V функция, только теперь из состояния $s_t = s$ обязательно сначала совершается действие $a_t = a$. Иногда индекс политики опускают: $Q(s,a) := Q^\pi(s,a)$.

Замечание

В определении V- и Q- функции в левой части выражения отсутствует обозначение t в силу однородности MDP.

DMDP. Дисконтированная награда

Дисконтированная кумулятивная награда за эпизод длины $H - t, \; t = \overline{0, H-1}$:

$$R_t^{H-1} := \sum_{j=t}^{H-1} \gamma^{j-t} r(s_j, a(s_j)) \text{ и } R_t := R_t^{\infty} := \sum_{j=t}^{\infty} \gamma^{j-t} r(s_j, a(s_j)).$$

При переходе к AMDP ($\gamma = 1$) наиболее естественным аналогом кумулятивной награды является среднее арифметическое наград по времени:

$$R_t^{H-1} := \frac{1}{H-t} \sum_{j=t}^{H-1} r(s_j, a(s_j)) \text{ и } R_t := R_t^{\infty} := \lim_{H \to \infty} \frac{1}{H-t} \sum_{j=t}^{H-1} r(s_j, a(s_j)).$$

DMDP. Дисконтированная награда

Мажоранта на $V^\pi(\cdot)$:

$$r(s,a) \in [0,1]: \quad 0 \leq V^\pi(s) \leq \frac{1}{(1-\gamma)} \quad \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

Q^π-функция обладает той же мажорантой, что и $V^\pi(\cdot)$:

$$r(s,a) \in [0,1]: \quad 0 \leq Q^\pi(s,a) \leq \frac{1}{(1-\gamma)} \quad \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

DMDP. Уравнения Беллмана

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \middle| \pi,\ s_0 = s\right] =$$
$$= \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} p(s, a; s') \left[r(s, a) + \gamma V^\pi(s')\right] = \mathbb{E}_\pi \left[Q^\pi(s, a)\right] =$$
$$= \mathbb{E}_\pi \left[r(s, a)\right] + \gamma \mathbb{E}_{p,\pi} \left[V^\pi(s')\right] = \mathbb{E}_\pi \left[r(s, a)\right] + \gamma \mathbb{E}_{p,\pi} \left[Q^\pi(s', a')\right];$$

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \middle| \pi,\ s_0 = s, a_0 = a\right] =$$
$$= \sum_{s' \in \mathcal{S}} p(s, a; s') \left[r(s, a) + \gamma V^\pi(s')\right] =$$
$$= r(s, a) + \gamma \mathbb{E}_p \left[V^\pi(s')\right] = r(s, a) + \gamma \mathbb{E}_{p,\pi} \left[Q^\pi(s', a')\right],$$
$$a' \sim \pi(\cdot|s').$$

DMDP. Задача RL

Цель задачи обучения с подкреплением (Reinforcement Learning, RL) – поиск политики, позволяющей получить максимальное кумулятивное вознаграждение в долгосрочной перспективе. В большинстве практических случаев задача RL формулируется как задача оптимизации следующего формата:

$$\pi^* \in \operatorname*{Arg\,max}_{\pi \in \hat{\Pi}} \left\{ \mathbb{E}_{\mathrm{P}(\tau_{H-1}|\pi)} \left[R_0^{H-1} \right] = \mathbb{E}_{s \sim \mu_0} \left[V^\pi(s) \right] \right\};$$

$$\hat{\Pi} := \left\{ \pi \,\middle|\, \pi(a|s) \geq 0,\ \sum_{\hat{a} \in \mathcal{A}} \pi(\hat{a}|s) = 1,\ a \in \mathcal{A}, s \in \mathcal{S} \right\}.$$

DMDP. Задача RL

Следующее утверждение [1] (детали: глава 3, утв. 21 и предшествующие) задаёт подкласс оптимальных политик, в рамках которого достаточно производить поиск интересующей π.

Пусть Π – набор всех нестационарных и рандомизированных политик. $V^\pi(s)$, $Q^\pi(s,a)$ зажаты между 0 и $\frac{1}{1-\gamma}$, следовательно, существуют конечные

$$V^*(s) := \sup_{\pi \in \Pi} \left\{ V^\pi(s) \right\}, \qquad Q^*(s,a) := \sup_{\pi \in \Pi} \left\{ Q^\pi(s,a) \right\};$$

$\exists\ \pi$ – стационарная, детерминированная, такая, что $\forall s \in \mathcal{S},\ a \in \mathcal{A}$:

$$V^\pi(s) = V^*(s), \qquad Q^\pi(s,a) = Q^*(s,a),$$

а, значит, π – оптимальная политика.

DMDP. Задача RL

В данном утверждении мы можем легко заменить операцию sup на операцию max, как минимум, в случае наград с достижимыми верхними гранями:

$$V^*(s) = \max_{\pi \in \Pi} \{V^\pi(s)\},$$

где V^* – оптимальная функция ценности. Введём обозначение класса всех отображений, описывающих детерминированные политики в данном процессе:

$$\mathbb{A} = \{a(\cdot) |\ a : \mathcal{S} \mapsto \mathcal{A}\}.$$

DMDP. Уравнения Беллмана

Если воспользоваться принципом динамического программирования, то удаётся вывести уравнение оптимальности Беллмана на V-функцию ценности:

$$
\begin{aligned}
V^*(s) &= \max_{a(\cdot)\in\mathbb{A}} \left\{ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \right] \right\} = \\
&= \max_{a(\cdot)\in\mathbb{A}} \left\{ \mathbb{E}\left[r(s, a(s)) + \gamma \sum_{t=0}^{\infty} \gamma^t r(s_{t+1}, a(s_{t+1})) \right] \right\} = \\
&= \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}\left[V^*(s') \right] \right\} = \\
&= \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s,a;s') V^*(s') \right\}.
\end{aligned}
$$

DMDP. Уравнения Беллмана

Соответственно, мы можем провести аналогичные рассуждения для Q-функции ценности:

$$
\begin{aligned}
Q^*(s,a) &= \max_{\pi \in \Pi} \left\{ Q^\pi(s,a) \right\};\\
Q^*(s,a) &= \mathbb{E}\left[r(s,a) + \gamma V^*(s') \middle| s_0 = s, a_0 = a \right] =\\
&= r(s,a) + \gamma \mathbb{E}\left[\max_{a' \in \mathcal{A}} \left\{ Q^*(s',a') \right\} \right] =\\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \max_{a' \in \mathcal{A}} \left\{ Q^*(s',a') \right\}.
\end{aligned}
$$

Критерий оптимальности относительно Q-функции ценности [1] (детали: глава 3.1.10.).

Функция Q представляет собой оптимальную функцию ценности Q^*, если и только если она удовлетворяет уравнениям оптимальности Беллмана:

$$Q(s,a) = \mathbb{E}\left[r(s, a(s)) + \gamma \max_{a' \in \mathcal{A}} \left\{ Q(s', a') \right\} \middle| s_0 = s, a_0 = a \right] =$$

$$= \sum_{s' \in \mathcal{S}} p(s, a; s') \left[r(s,a) + \gamma \max_{a' \in \mathcal{A}} \left\{ Q(s', a') \right\} \right], \quad \forall s \in \mathcal{S},\ a \in \mathcal{A}.$$

Кроме того, детерминированная политика, определенная как

$$\pi(s) \in \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ Q^*(s,a) \right\},$$

есть оптимальная политика.

DMDP. Уравнения оптимальности Беллмана

Таким образом, для оптимальной политики π^* выполнены следующие соотношения [2]:

1) $V^{\pi^*}(s) = \max\limits_{a \in \mathcal{A}} \left\{ Q^{\pi^*}(s, a) \right\}, \quad \forall s \in \mathcal{S};$
2) $Q^{\pi^*}(s, a) = r(s, a) + \gamma \sum\limits_{s' \in \mathcal{S}} p(s, a; s') V^{\pi^*}(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A};$
3) $V^{\pi^*}(s) = \max\limits_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \sum\limits_{s' \in \mathcal{S}} p(s, a; s') V^{\pi^*}(s') \right\}, \quad \forall s \in \mathcal{S};$
4) $Q^{\pi^*}(s, a) = r(s, a) + \gamma \sum\limits_{s' \in \mathcal{S}} p(s, a; s') \max\limits_{a' \in \mathcal{A}} \left\{ Q^{\pi^*}(s', a') \right\}, \quad s \in \mathcal{S}, a \in \mathcal{A}.$

Соответствующая этим соотношениям детерминированная политика:

$$\pi^*(s) \in \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ Q^{\pi^*}(s, a) \right\} = \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s, a; s') V^{\pi^*}(s') \right\}.$$

DMDP. Уравнения оптимальности Беллмана

Основное свойство оптимальных V/Q-функций

Если V^*, Q^* – оптимальные функции, то функции $V^\star(s) := V^*(s) + \alpha, Q^\star(s,a) := Q^*(s,a) + \alpha,\ \forall s \in \mathcal{S}, a \in \mathcal{A}, \alpha \in \mathbb{R}$ так же соответствуют оптимальной политике.

$$\pi^*(s) \in \operatorname*{Arg\,max}_{a \in \mathcal{A}} \{Q^*(s,a)\} = \operatorname*{Arg\,max}_{a \in \mathcal{A}} \{Q^*(s,a) + \alpha\} =$$

$$= \operatorname*{Arg\,max}_{a \in \mathcal{A}} \{Q^\star(s,a)\} = \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V^*(s') \right\} =$$

$$= \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \underbrace{\left(V^*(s') + \alpha\right)}_{=V^\star(s')} \right\},\ \forall s \in \mathcal{S}, \alpha \in \mathbb{R}$$

DMDP. Уравнения оптимальности Беллмана

Для уравнения Беллмана верна связь с константным сдвигом функции награды $\hat{r}(s,a) := r(s,a) + \alpha(1-\gamma)$, $\forall s \in \mathcal{S}, a \in \mathcal{A}, \alpha \in \mathbb{R}$:

$$\begin{aligned} Q^\star(s,a) = Q^*(s,a) + \alpha &= \left(r(s,a) + \alpha(1-\gamma)\right) + \\ &+ \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \max_{a' \in \mathcal{A}} \left\{ Q^*(s',a') + \alpha \right\} = \\ &= \hat{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \max_{a' \in \mathcal{A}} \left\{ Q^\star(s',a') \right\} = \\ &= \hat{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V^\star(s'); \end{aligned}$$

$$\begin{aligned} V^\star(s) &= \max_{a \in \mathcal{A}} \left\{ \left(r(s,a) + \alpha(1-\gamma)\right) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \left(V^*(s') + \alpha \right) \right\} = \\ &= \max_{a \in \mathcal{A}} \left\{ \hat{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V^\star(s') \right\} = \max_{a \in \mathcal{A}} \left\{ Q^\star(s,a) \right\}. \end{aligned}$$

DMDP. Уравнения оптимальности Беллмана

Аддитивное преобразование, не меняющее решение π^*

[a] $\hat{r}(s,a) := \beta \left(r(s,a) + f(s) - \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') f(s') \right)$, $\forall s \in \mathcal{S}, a \in \mathcal{A}, \beta > 0$, где $f : \mathcal{S} \mapsto \mathbb{R}$ – произвольное отображение.

[a] $\pi^*(s) = \arg\max_{a \in \mathcal{A}} \{Q^*(s,a)\} = \arg\max_{a \in \mathcal{A}} \{\beta\,(Q^*(s,a) + f(s))\}$, $s \in \mathcal{S}$.

$$\begin{aligned}
Q^\star(s,a) := \beta\,(Q^*(s,a) + f(s)) = \\
&= \beta\,(r(s,a) + f(s)) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \max_{a' \in \mathcal{A}} \{\beta Q^*(s',a')\} = \\
&= \hat{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \max_{a' \in \mathcal{A}} \{\beta\,(Q^*(s',a') + f(s'))\} = \\
&= \hat{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') \max_{a' \in \mathcal{A}} \{Q^\star(s',a')\}.
\end{aligned}$$

DMDP. Уравнения оптимальности Беллмана

Для функции награды, зависящей от $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, аддитивное преобразование, инвариантное относительно оптимальной политики, выглядит проще: $\hat{r}(s, a, s') := \beta\left(r(s, a, s') + f(s) - \gamma f(s')\right)$.

$$
\begin{aligned}
Q^{\star}(s, a) &:= \beta\left(Q^{*}(s, a) + f(s)\right) = \\
&= \sum_{s' \in \mathcal{S}} p(s, a; s')\left(\beta\left(r(s, a, s') + f(s) - \gamma f(s')\right) + \right. \\
&\left. + \gamma \max_{a' \in \mathcal{A}} \left\{\beta\left(Q^{*}(s', a') + f(s')\right)\right\}\right) = \\
&= \sum_{s' \in \mathcal{S}} p(s, a; s')\left(\hat{r}(s, a, s') + \gamma \max_{a' \in \mathcal{A}} \left\{Q^{\star}(s', a')\right\}\right).
\end{aligned}
$$

Предположения

Будем считать, что о нашей среде нам известны следующие функции:

- $p(s'|s,a)$ — вероятность попасть в состояние s' из состояния s с помощью действия a
- $r(s,a)$ — награда за выполнения действия a в состоянии s

Замечание

Эти допущения существенно сужают круг задач, которые мы можем решить, однако алгоритмы, предложенные в этой лекции будут оптимальными в данной постановке.

Построение алгоритма обучения π

Сам процесс можно разделить на два этапа: оценка качества текущей политики и поиск следующего приближения оптимальной политики. *Дополнительно предположим дискретность и конечность пространств* $\mathcal{S}$ *и* $\mathcal{A}$. Начнём с оценивания — вычислим $V^\pi(s)$ и $Q^\pi(s,a)$:

$$V^\pi(s) = \underbrace{\underbrace{\mathbb{E}_{\pi(a|s)}\left[r(s,a)\right]}_{:=u(s)} + \gamma \mathbb{E}_{\pi(a|s)}\mathbb{E}_{p(s'|s,a)}\left[V^\pi(s')\right]}_{:=F(V^\pi(s))}, \quad \forall s \in S.$$

Представим выражение выше в виде матрично векторных операций. V^π — вектор ценностей состояний. P — матрица вероятностей: $P_{ss'} = \sum\limits_{a \in \mathcal{A}} \pi(a|s)p(s'|s,a) = p(s'|s)$. u – вектор средних наград за шаг.

Построение алгоритма обучения π

$$V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) r(s,a) + \gamma \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s'|s,a) V^\pi(s') =$$
$$= u(s) + \gamma \sum_{s'\in\mathcal{S}} V^\pi(s') p(s'|s) \Longrightarrow V^\pi = F(V^\pi) = u + \gamma P V^\pi.$$

Полученное отображение F является сжимающим, для произвольных векторов V и W:

$$\|F(V) - F(W)\|_\infty = \|u + \gamma PV - u - \gamma PW\|_\infty = \gamma \|P(V - W)\|_\infty \leq$$
$$\leq \gamma \|P\|_\infty \|V - W\|_\infty = \gamma \|V - W\|_\infty,$$

$$\|P\|_\infty = \max_{x\neq 0} \left\{ \frac{\|Px\|_\infty}{\|x\|_\infty} \right\} = \max_{\|x\|_\infty \leq 1} \max_{s\in\mathcal{S}} \left\{ \sum_{s'\in\mathcal{S}} p(s'|s) \underbrace{x_{s'}}_{=1} \right\} = 1.$$

Построение алгоритма обучения π

Получили алгортим Iterative Policy Evaluation для оценки политики π с заданной точностью $\varepsilon > 0$:

1) Инициализировать $V(s), \forall s \in \mathcal{S}$;
2) Повторять в цикле:

2.1) $\Delta := 0$

2.2) для всех $s \in \mathcal{S}$:

$$\delta := V(s);$$
$$V(s) := \mathbb{E}_{\pi(a|s)}\left[r(s,a)\right] + \gamma \mathbb{E}_{\pi(a|s)} \mathbb{E}_{p(s'|s,a)}\left[V(s')\right];$$
$$\Delta := \max\{\Delta, |\delta - V(s)|\}.$$

2.3) Если $\Delta \leq \varepsilon$, то выход, иначе — переход на шаг 2.

Построение алгоритма обучения π

Теперь построим процедуру улучшения оценённой политики π:

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)}\left[V^{\pi}(s')\right].$$

Определение

Политика $\hat{\pi} \succeq \pi$ (монотонно лучше политики π), если $V^{\hat{\pi}}(s) \geq V^{\pi}(s), \quad \forall s \in \mathcal{S}$.

Таким образом, изменяя действие для одного состояния $\hat{s} \in \mathcal{S}$ мы производим улучшение π:

$$\hat{\pi}(a|\hat{s}) = \delta_{\left\{\arg\max_{\hat{a}\in\mathcal{A}}\{Q^{\pi}(\hat{s},\hat{a})\}\right\}}(a),\ \hat{\pi}(\hat{s}) := \arg\max_{\hat{a}\in\mathcal{A}}\left\{Q^{\pi}(\hat{s},\hat{a})\right\};$$

$$\hat{\pi}(a|s) = \pi(a|s),\ \forall s \in \mathcal{S},\ \ s \neq \hat{s}.$$

То есть $Q^{\pi}(s,\hat{\pi}(s)) \geq V^{\pi}(s)$.

Построение алгоритма обучения π

Теорема об улучшении политики [1] (детали: глава 3.2.3, теорема 17)

Пусть π и π' – любая пара детерминированных политик, таких, что

$$\forall s \in \mathcal{S} \quad Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s). \tag{1}$$

Тогда π' должна быть не хуже, чем π, то есть ценность не хуже $\forall s \in \mathcal{S}$:

$$V^{\pi'}(s) \geq V^{\pi}(s). \tag{2}$$

Более того, если в каком-либо состоянии существует строгое (1), то и (2) должно быть строгим.

Построение алгоритма обучения π

Действительно,

$$V^{\pi}(s) \leq Q^{\pi}(s, \underbrace{\hat{\pi}(s)}_{\text{детерминирован}}) = r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V^{\pi}(s') \right] \leq$$

$$\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{p(s'|s,a)} \left[Q^{\pi}(s', \hat{\pi}(s')) \right] \leq \ldots \leq$$

$$\leq r(s, \hat{\pi}(s)) + \gamma \mathbb{E}_{p(s'|s,\hat{\pi}(s))} \left[r(s', \hat{\pi}(s')) + \gamma^2 \ldots \right] = V^{\hat{\pi}}(s), \quad \forall s \in \mathcal{S}.$$

Шаг для обновления всей политики

$$\pi_{\text{new}}(s) := \arg\max_{a \in \mathcal{A}} \left\{ Q^{\pi_{\text{old}}}(s, a) \right\}, \quad \forall s \in \mathcal{S}.$$

Построение алгоритма обучения π

Если после очередного шага обновления политики получилось, что $Q^{\pi}(s, \hat{\pi}(s)) = V^{\pi}(s),\ \forall s \in \mathcal{S}$, то это означает удовлетворение уравнению Беллмана:

$$\hat{\pi} = \pi;$$
$$Q^{\pi}(s, a) = r(s, a) + \gamma \mathbb{E}_{p(s'|s,a)}\left[V^{\pi}(s')\right], \quad \forall s \in \mathcal{S},\ a \in \mathcal{A}.$$

Построение алгоритма обучения π

Мы получили алгоритм Policy Iteration:

1) Инициализируем $V(s)$ и $\pi(a|s)$ для всех $s \in \mathcal{S}$.
2) Оценить $V^\pi(s)$ для текущей π, используя Iterative Policy Evaluation.
3) Улучшаем политику:

3.1) Flag := True;

3.2) Для всех $s \in \mathcal{S}$:

$$
\begin{aligned}
& a = \pi(s); \\
& \pi(s) := \arg\max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V^\pi(s') \right] \right\}; \\
& \text{если } a \neq \pi(s), \text{ то } \text{Flag} := \text{False}.
\end{aligned}
$$

4) Если Flag = True, то выход, иначе — шаг 2.

Построение алгоритма обучения π

Описанную ранее процедуру поиска оптимальной политики можно представить как последовательность монотонно улучшающихся политик и функций ценности:

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \ldots \xrightarrow{I} \pi^* \xrightarrow{E} V^*,$$

где $\xrightarrow{E}$ обозначает оценку политики, $\xrightarrow{I}$ – улучшение политики, то есть получен алгоритм итеративной оптимизации политики.

Построение алгоритма обучения π

Уравнение Беллмана относительно фиксированной политики по сути решается простым итеративным способом. Начальное приближение V_0 выбирается произвольно, а каждая последовательная итерация реализуется согласно уравнению Беллмана:

$$V_{t+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} p(s,a;s') \left(r(s,a) + \gamma V_t(s') \right),$$

где $V_t = V^\pi$ – фиксированная точка. Для получения каждого последующего приближения, V_{t+1} из V_t при итеративной оценке политики применяется та же операция к каждому состоянию s, и ее называют ожидаемым обновлением, а данный алгоритм – итерации функции ценности.

Построение алгоритма обучения π

Для решения уравнения оптимальности Беллмана можно проводить следующую процедуру:

$$V_{t+1}(s) = \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V_t(s') \right\}, \quad \forall s \in \mathcal{S};$$

$$\pi_{t+1}(s) \in \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V_{t+1}(s') \right\}, \quad \forall s \in \mathcal{S}.$$

Причём начальное значение V_0 может быть произвольным, а в качестве критерия останова может выступать $\| V_{t+1} - V_t \|_\infty \leq \varepsilon$.

Построение алгоритма обучения π

На предыдущем слайде предложен по сути частный случай алгоритма Policy Iteration — Value Iteration (вместо шагов 2 и 3 один шаг делаем):

1) Инициализируем $V(s)$ для всех $s \in \mathcal{S}$.
2) Повторять:
2.1) $\Delta := 0$;
2.2) Для всех $s \in \mathcal{S}$:

$$
\begin{aligned}
&v = V(s) \\
&V(s) := \max_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V(s') \right] \right\} \\
&\Delta := \max\{\Delta, |v - V(s)|\}
\end{aligned}
$$

2.3) Если $\Delta \leq \varepsilon$, то выход, иначе — переход на шаг 2.

Мы по сути схлопнули Policy Iteration и Policy Improvement, не вычисляя π.

Построение алгоритма обучения π

В результате работы алгоритма Value Iteration оптимизированная политика вычисляется следующим образом:

$$\pi(s) = \arg\max_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V(s') \right] \right\}.$$

Сходимость алгоритма Value Iteration

Для оптимальной V-функции выполнено соотношение
$V^*(s) = V^{\pi^*}(s) = \max\limits_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V^*(s') \right] \right\}, \ \forall s \in \mathcal{S}$.

Рассмотрим
$V^{\pi_{t+1}}(s), \ \pi_{t+1}(s) = \arg\max\limits_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V^{\pi_{t+1}}(s') \right] \right\}$.

Построим оценку на невязку по V-функции,
$V^*(s) \geq V^{\pi_t}(s), \ t \in \mathbb{Z}_+, \ s \in \mathcal{S}$:

$$\max_{s \in \mathcal{S}} \left\{ \left| V^*(s) - V^{\pi_{t+1}}(s) \right| \right\} = \max_{s \in \mathcal{S}} \left\{ \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[V^*(s') \right] \right\} - \right.$$

$$\left. - \max_{a' \in \mathcal{A}} \left\{ r(s,a') + \gamma \mathbb{E}_{p(s''|s,a')} \left[V^{\pi_t}(s'') \right] \right\} \right\} \leq \gamma \max_{\substack{s \in \mathcal{S} \\ a \in \mathcal{A}}} \left\{ \mathbb{E}_{p(s'|s,a)} \left[V^*(s') \right] - \right.$$

$$\left. - \mathbb{E}_{p(s''|s,a)} \left[V^{\pi_t}(s'') \right] \right\} = \gamma \max_{s \in \mathcal{S}, a \in \mathcal{A}} \left\{ \mathbb{E}_{p(s'|s,a)} \left[V^*(s') - V^{\pi_t}(s') \right] \right\} \leq$$

Сходимость алгоритма Value Iteration

продолжим с последнего неравенства, раскрывая рекуррентную зависимость:

$$\leq \gamma \max_{s' \in \mathcal{S}} \left\{ V^*(s') - V^{\pi_t}(s') \right\} = \gamma \max_{s' \in \mathcal{S}} \left\{ \left| V^*(s') - V^{\pi_t}(s') \right| \right\} \Rightarrow$$

$$\Rightarrow \max_{s \in \mathcal{S}} \left\{ \left| V^{\pi_t}(s) - V^*(s) \right| \right\} \leq \gamma^t \max_{s' \in \mathcal{S}} \left\{ \left| V^{\pi_0}(s') - V^*(s') \right| \right\}.$$

В качестве $V^{\pi_0}(s)$, $s \in \mathcal{S}$ можно взять произвольное начальное приближение, например, тождественную по всем состояниям константу. Имеем оценку относительно внешних итераций:

$$\max_{s \in \mathcal{S}} \left\{ \left| V^{\pi_t}(s) - V^*(s) \right| \right\} = \mathcal{O}\left(\gamma^t\right), \quad t \in \mathbb{Z}_+, \ \gamma \in (0, 1).$$

Value Iteration. Сложность решения

Введём следующий оператор $T : \mathbb{R}^S \mapsto \mathbb{R}^S$:
$T(V)_s := \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s')V(s') \right\}$. Теперь обновление V-функции в Value Iteration выражается следующим образом: $V_{t+1} := T(V_t)$, а V-функция кодируется вещественным вектором.

Теорема [3]

Существует DMDP, для которого последовательность оценок V-функции удовлетворяет:
$V_0 = 0_S, V_{n+1} \in \operatorname{span}\left\{V_0, \ldots, V_n, T(V_0), \ldots, T(V_n)\right\}$, $n \in \mathbb{Z}_+$, со следующим свойством $\forall n = \overline{0, N-1}$:

$$\|V_n - V^*\|_\infty \geq \frac{\gamma^n}{1+\gamma}, \; V^* = T(V^*).$$

Value Iteration. Сложность решения

Доказательство. Для произвольного $\gamma \in (0, 1)$ предложим DMDP с N состояниями и с одним действием. Награда для первого состояния $r_1 := 1$, для остальных состояний $r_i := 0, i = \overline{2, N}$. Действие из первого состояния оставляет в нём же, действие из $(i+1)$-го состояния переводит в i-ое. Оптимальное значение V-функции следующее: $V^*(i) = \frac{\gamma^{i-1}}{1-\gamma}$. Рассмотрим последовательность векторов $(V_n)_{n \geq 0}, V_0 = 0_S$ и

$$V_{n+1} \in \operatorname{span}\left\{V_0, \dots, V_n, T(V_0), \dots, T(V_n)\right\}, n \geq 0.$$

Докажем через раскрытие рекурсии, что $\forall n \geq 0, i \in \mathcal{S}$ имеем $V_n(i) = 0$, если $i \geq n+1$. Это верно для $n = 0$, так как $V_0 = 0_S$. Предположим, что верно и для $V_0, \dots V_{n-1}$. По определению оператора T и в силу того, что $r_i = 0, i \geq 2$, имеем $T(V_t)_i = 0$, если $i \geq t+2, \forall t \leq n-1$.

Value Iteration. Сложность решения

Следовательно, в силу $V_{n+1} \in \operatorname{span}\{V_0, \ldots, V_n, T(V_0), \ldots, T(V_n)\}$ заметим, что $V_n(i) = 0$, если $i \geq n+1$, и мы доказали нашу рекурсию. $r_1 > 0$ – единственное, по сути мы доказали, что для любого метода первого порядка требуется $n-1$ шаг для распространения награды первого состояния до состояния $1 \leq n \leq N$.

Value Iteration. Сложность решения

Теперь мы имеем для $1 \le n \le N-1$:

$$\begin{aligned}
\|\mathrm{V}_n - T(\mathrm{V}_n)\|_\infty &= \|\mathrm{V}_n - T(\mathrm{V}_n) - (\mathrm{V}^* - T(\mathrm{V}^*))\|_\infty && (3)\\
&\ge (1-\gamma)\|\mathrm{V}_n - \mathrm{V}^*\|_\infty && (4)\\
&= (1-\gamma)\max_{1\le i\le N}\{|V_n(i) - V^*(i)|\}\\
&\ge (1-\gamma)\max_{n+1\le i\le N}\{|V_n(i) - V^*(i)|\}\\
&\ge (1-\gamma)\max_{n+1\le i\le N}\{|V^*(i)|\} && (5)\\
&\ge (1-\gamma)\max_{n+1\le i\le N}\left\{\frac{\gamma^{i-1}}{1-\gamma}\right\}\\
&\ge \gamma^n,
\end{aligned}$$

где (3) следует из $\mathrm{V}^* = T(\mathrm{V}^*)$,

Value Iteration. Сложность решения

(4) следует из (6), и (5) следует из $V_n(i) = 0$ для $i \geq n + 1$. Можем заключить в силу (6):

$$\|\mathrm{V}_n - \mathrm{V}^*\|_\infty \geq \frac{1}{1+\gamma} \cdot \|\mathrm{V}_n - T(\mathrm{V}_n) - (\mathrm{V}^* - T(\mathrm{V}^*))\|_\infty = \\ = \frac{1}{1+\gamma} \cdot \|\mathrm{V}_n - T(\mathrm{V}_n)\|_\infty \geq \frac{\gamma^n}{1+\gamma}.$$

$$\forall V, W \in \mathbb{R}^S \text{ (проверяется непосредственно)} : \\ (1-\gamma) \cdot \|\mathrm{V} - \mathrm{W}\|_\infty \leq \|(\mathrm{I} - T)(\mathrm{V}) - (\mathrm{I} - T)(\mathrm{W})\|_\infty; \\ \|(\mathrm{I} - T)(\mathrm{V}) - (\mathrm{I} - T)(\mathrm{W})\|_\infty \leq (1+\gamma) \cdot \|\mathrm{V} - \mathrm{W}\|_\infty. \tag{6}$$

Утверждение доказано.

Value Iteration. Сложность решения

Из доказанного утверждения вместе с утверждением о сходимости Value Iteration следует оптимальность алгоритма Value Iteration:

$$\frac{\gamma^t}{1-\gamma} = \gamma^t \left\| V_0 - V^* \right\|_\infty \geq \left\| V_t - V^* \right\|_\infty \geq \frac{\gamma^t}{1+\gamma}, \; t \in \mathbb{Z}_+, \; \gamma \in (0, 1);$$

$$\left\| V_t - V^* \right\|_\infty = \Theta\left(\gamma^t\right), \quad V_0 = 0_N.$$

LP-релаксация MDP. AMDP [4]

Для AMDP LP-задача вводится следующим образом:

$$V^*(s) = \max_{a(\cdot)\in\mathbb{A}} \left\{ \lim_{H\to\infty} \frac{1}{H} \mathbb{E}\left[\sum_{t=0}^{H-1} r(s_t, a_t(s_t)) \middle| s_0 = s \right] \right\},$$

где H – эпизодическое ограничение, то есть максимальная длина эпизода. В случае эпизодов конечной длины предел опускается и используется максимальное значение H. Для политики $\pi(a|s)$ можно определить стационарное распределение:

$$\nu_\pi(s') = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} p(s,a;s')\pi(a|s)\nu_\pi(s), \quad s' \in \mathcal{S},$$

которое соответствует своему вектору из вероятностей $\nu_\pi = (\nu_\pi(s))_{s\in\mathcal{S}}$.

LP-релаксация MDP. AMDP [4]

И если MDP равномерно эргодично, то:

$$V^{\pi} := V(\pi) := \lim_{H\to\infty} \frac{1}{H}\mathbb{E}\left[\sum_{t=0}^{H-1} r(s_t, a_t(s_t))\right] = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} r(s,a)\pi(a|s)\nu_{\pi}(s).$$

Напоминаем, что в данном случае равномерная эргодичность соответствует:

$$\max_{i=\overline{1,S}} \left\{\left\|\left(P^{\pi}\right)^{n} e_i - \nu_{\pi}\right\|_{\infty}\right\} \to 0,\ \ n\to\infty, \quad e_i^{\top} = (0,\ldots,0,\underbrace{1}_{i},0,\ldots,0).$$

LP-релаксация MDP. AMDP [4]

Вводится распределение действий по состояниям –
$\mu(s,a) = \nu_\pi(s)\pi(a|s)$, следовательно, можно переписать задачу поиска оптимальной политики в AMDP как задачу LP со смыслом оценки ценности политики по распределению μ:

$$\max_{\mu \in \Delta^{\mathcal{S}\times\mathcal{A}}} \left[V(\mu) = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} r(s,a)\mu(s,a) = \langle r, \mu \rangle : \sum_{b\in\mathcal{A}} \mu(s',b) = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} p(s,a;s')\mu(s,a),\ s' \in \mathcal{S} \right];$$

$$\Delta^{\mathcal{S}\times\mathcal{A}} = \left\{ \mu : \mu(s,a) \geq 0, \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mu(s,a) = 1 \right\},\ \pi_\mu(a|s) = \frac{\mu(s,a)}{\sum\limits_{b\in\mathcal{A}} \mu(s,b)}.$$

LP-релаксация MDP. AMDP [4]

Данную задачу можно напрямую переписать в матричной форме:

$$\max_{\mu \in \Delta^{\mathcal{S}\times\mathcal{A}}} \langle r, \mu \rangle \, ;$$
$$s.t. \; (\widehat{I} - P)\mu = 0.$$

Единичная матрица $\widehat{I}$ имеет нестандартный формат: это прямоугольная матрица размера $S \times (SA)$, на каждой строке $s \in \mathcal{S}$ только элементы, соответствующие паре (s, a), $a \in \mathcal{A}$, равняются единице, остальные элементы данной строки равняются нулю, то есть на каждой строке $\widehat{I}$ ровно A единиц. У матрицы P размера $S \times (SA)$ в каждом столбце $(s, a) \in \mathcal{S} \times \mathcal{A}$ записано распределение $\mathrm{P}(\cdot | s, a)$.

LP-релаксация MDP. AMDP [4]

Для этой задачи LP напрямую строится двойственная задача, с условием, что $\mu \geq 0$, которая имеет смысл оценки ценности оптимальной политики через V-функцию:

$$\min_{\overline{V} \in \mathbb{R}, V \in \mathbb{R}^{|\mathcal{S}|}} \overline{V};$$
$$s.t.\ r - \overline{V} \cdot 1_{SA} - (\widehat{I} - P)^\top V \leq 0.$$

Таким образом, имеет место уравнение оптимальности Беллмана со средним вознаграждением:

$$V(s) = \max_{a \in \mathcal{A}} \left\{ r(s,a) - V^* + \sum_{s' \in \mathcal{S}} p(s,a;s') V(s') \right\}, \quad V^* = \langle r, \mu^* \rangle,$$

полученное из ограничений вида неравенства:

$$\widehat{I}^\top V \geq r - \overline{V} \cdot 1_{SA} + P^\top V.$$

LP-релаксация MDP. AMDP [4]

Таким образом, для AMDP мы получили аналог уравнения Беллмана — уравнение Пуассона:

$$V^\pi + V^\pi(s) = \mathbb{E}_\pi\left[r(s,a)\right] + \mathbb{E}_{p,\pi}\left[V^\pi(s')\right], \ \forall s \in \mathcal{S}.$$

Множество его решений:

$$\left\{(V^\pi(\cdot) + ce(\cdot), V^\pi) : \ c \in \mathbb{R}\right\}, \quad e(s) = 1, \ \forall s \in \mathcal{S}.$$

Критерий оптимальной политики:

$$V^* + V^*(s) = \max_{a \in \mathcal{A}} \left\{ r(s,a) + \mathbb{E}_p\left[V^*(s')\right] \right\}, \ \forall s \in \mathcal{S};$$

$$\pi^* \in \operatorname*{Arg\,max}_{\pi \in \Pi} \left\{ V^\pi \right\}, \quad V^* \equiv V^{\pi^*}.$$

LP-релаксация MDP. AMDP [4]

Уравнение Пуассона также существует для Q-функции ценности:

$$V^\pi + Q^\pi(s,a) = r(s,a) + \mathbb{E}_{p,\pi}\left[Q^\pi(s',a')\right], \ \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

Множество его решений:

$$\left\{(Q^\pi(\cdot) + c\bar{e}(\cdot), V^\pi) : \ c \in \mathbb{R}\right\}, \quad \bar{e}(s,a) = 1, \ \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

Критерий оптимальной политики:

$$V^* + Q^*(s,a) = r(s,a) + \mathbb{E}_p\left[\max_{a' \in \mathcal{A}}\left\{Q^*(s',a')\right\}\right], \ \forall s \in \mathcal{S}, a \in \mathcal{A};$$

$$\pi^* \in \operatorname*{Arg\,max}_{\pi \in \Pi}\left\{V^\pi\right\}.$$

LP-релаксация MDP. DMDP [4]

Для DMDP задача LP записывается в следующем виде (q – распределение начального состояния μ_0 в виде вектора):

$$\min_{V \in \mathbb{R}^{|\mathcal{S}|}} \langle q, V \rangle ;$$
$$s.t. \; r - (\widehat{I} - \gamma P)^\top V \leq 0.$$

И ей соответствует такая двойственная задача:

$$\max_{\mu \in \Delta^{\mathcal{S} \times \mathcal{A}}} \langle r, \mu \rangle ;$$
$$s.t. \; (\widehat{I} - \gamma P)\mu = q.$$

LP-релаксация MDP. DMDP [5]

Существует также постановка задачи LP для ограниченного DMDP – Constrained Markov Decision Process, CMDP:

$$\max_{\mu \in \Delta^{\mathcal{S}\times\mathcal{A}}} \langle r, \mu \rangle ;$$
$$s.t.\ (\widehat{I} - \gamma P)\mu = q,\ D\mu \geq c.$$

По сравнению с предыдущими задачами линейного программирования вводится дополнительно аффинное ограничение вида неравенства: $D\mu \geq c$.

LP-релаксация MDP. DMDP [5]

Заметим, что вместо обозначенного ранее распределения $\mu(s, a) = \nu_\pi(s)\pi(a|s)$ может быть полезно рассмотреть:

$$\mu(s, a) := \mu^\pi(s, a) = \mathbb{E}_{s_0 \sim \mu_0} \left[\sum_{t=0}^{\infty} \gamma^t \, \mathrm{P}(s_t = s, a_t = a | s_0) \right].$$

Или даже масштабированную сумму сверху в виде корректно определённой вероятностной меры:

$$\begin{aligned} \mu(s, a) &:= \tilde{\mu}^\pi(s, a) = (1-\gamma)\mu^\pi(s, a) = \\ &= \mathbb{E}_{s_0 \sim \mu_0} \left[(1-\gamma) \sum_{t=0}^{\infty} \gamma^t \, \mathrm{P}(s_t = s, a_t = a | s_0) \right]. \end{aligned}$$

LP-релаксация MDP. DMDP [5]

В обоих случаях получается одна и та же политика:

$$\pi(a|s) = \frac{\mu^\pi(s,a)}{\sum\limits_{b\in\mathcal{A}} \mu^\pi(s,b)} = \frac{\tilde{\mu}^\pi(s,a)}{\sum\limits_{b\in\mathcal{A}} \tilde{\mu}^\pi(s,b)}.$$

Связь решения DMDP с выпуклой оптимизацией

Обозначим за $v \in \mathbb{R}^S$ вектор, кодирующий V-функцию ценности. Тогда решение уравнения $v = T(v)$ соответствует поиску стационарной точки некторой функции $f : \mathbb{R}^S \mapsto \mathbb{R}$ со следующим градиентом и (суб)гессианом:

$$\nabla f(v) := v - T(v) = (I - T)(v) =: F(v);$$

$$\nabla^2 f(v) := I - \gamma \cdot P^{\pi^v} =: \partial F(v) \succeq 0;$$

$$T(v) = r^{\pi^v} + \gamma \cdot P^{\pi^v} v, \quad \left(r^{\pi^v}\right)_s := r(s, a_s^v), \ s \in \mathcal{S},$$

$$\pi^v(s) := a_s^v \in \operatorname*{Arg\,max}_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V(s') \right\}.$$

Таким образом, можно ввести формально следующую функцию:

$$f(v) := \frac{1}{2} \left\langle \left(I - \gamma \cdot P^{\pi^v}\right) v, v \right\rangle - \langle r^{\pi^v}, v \rangle + \mathrm{const}\,.$$

Связь решения DMDP с выпуклой оптимизацией [6]

Свойства функции $f(\cdot)$, $\quad \forall v, w \in \mathbb{R}^S$:

$$(1-\gamma)\cdot\|v-w\|_\infty \le \|\nabla f(v) - \nabla f(w)\|_\infty \le (1+\gamma)\cdot\|v-w\|_\infty\,;$$

$$\frac{1-\gamma}{\sqrt{S}}\|v-w\|_2 \le \|\nabla f(v) - \nabla f(w)\|_\infty \le (1+\gamma)\cdot\|v-w\|_2\,;$$

$$\frac{1-\gamma}{\sqrt{S}}\|v-w\|_2 \le \|\nabla f(v) - \nabla f(w)\|_2 \le \sqrt{S}(1+\gamma)\cdot\|v-w\|_2\,.$$

Связь решения DMDP с выпуклой оптимизацией [6]

Мы с помощью введёных обозначений можем переписать алгоритм Policy Iteration как шаг метода Ньютона для решения нелинейного уравнения $F(v) = 0, \; v \in \mathbb{R}^S$:

$$v_{t+1} := v_t - \left(\partial F(v_t)\right)^{-1} F(v_t), \quad t \in \mathbb{Z}_+.$$

Данная процедура не обладает глобальной сходимостью.

Связь решения DMDP с выпуклой оптимизацией [6]

Сглаженный оператор оптимальности Беллмана, $\beta > 0$:

$$T_\beta(v)_s := \frac{1}{\beta} \ln \left(\sum_{a \in \mathcal{A}} \exp \left(\beta \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a;s') V(s') \right) \right) \right), \forall s \in \mathcal{S}.$$

Если $v^*_\beta = T_\beta(v^*_\beta)$ и $v^* = T(v^*)$, то:

$$\| v^*_\beta - v^* \|_\infty \leq \frac{\gamma \ln(A)}{\beta(1-\gamma)}.$$

Метод Ньютона для решения $F_\beta(v) := v - T_\beta(v) = 0$ сходится квадратично для любого начального приближения.

Q-обучение

Несложно заметить, что если

$$V^*(s) = \max_{a \in \mathcal{A}} Q(s, a),$$

то Q^*-функция должна удовлетворять *Q-уравнению*:

$$Q(s, a) = \sum_{s' \in \mathcal{S}} p(s, a; s') \left(r(s, a; s') + \gamma \max_{a' \in \mathcal{A}} Q(s', a') \right),$$

в текущем случае рассматривается уравнение Беллмана для более общего процесса MDP, в котором награда зависит уже от (s, a, s'), то есть добавилось ещё и следующее за s состояние s'.

Q-обучение

С такой зависимостью наград удобнее рассматривать траекторию τ_{H-1} политики π как набор четвёрок (s, a, r, s') со следующим правдоподобием:

$$\tau_{H-1} = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, \dots, a_{H-1}, r_{H-1}, s_H);$$

$$\mathrm{P}\left(\tau_{H-1} | \pi\right) = \mu_0(s_0) \prod_{t=0}^{H-1} \left(\pi(a_t | s_t) p(s_t, a_t; s_{t+1})\right).$$

Q-обучение

Данное уравнение Беллмана может быть решено методом простых итераций; если смотреть на $Q = \{Q(s,a)\}_{s\in\mathcal{S},a\in\mathcal{A}}$ как на вектор, то можно записать в операторном виде $Q = F(Q)$ (метод простых итераций будет иметь вид $Q_{t+1} = F(Q_t)$), где по определению оператор в правой части F является сжимающим с коэффициентом γ в норме Чебышева:

$$\max_{s\in\mathcal{S},a\in\mathcal{A}} \left|F(\tilde{Q}(s,a)) - F(Q(s,a))\right| \le \gamma \max_{s\in\mathcal{S},a\in\mathcal{A}} \left|\tilde{Q}(s,a) - Q(s,a)\right|,$$

$\tilde{Q}$ и Q – произвольные.

Q-обучение

Основная идея Q-обучения заключается в замене невычислимой правой части в уравнении $Q_{t+1} = F(Q_t)$ на ее вычислимую несмещенную оценку:

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)\left(r(s,a;s'(s,a)) + \gamma \max_{a' \in \mathcal{A}} Q_t(s'(s,a),a') - \right.$$
$$\left. - Q_t(s,a)\right), \qquad (7)$$

где $s'(s,a)$ – положение процесса на шаге $t+1$, если на шаге t процесс был в состоянии s и было выбрано действие a, то параметр $0 < \alpha_t(s,a) \leq 1$, иначе $\alpha_t(s,a) = 0$. Правая часть (7) следует из перехода в $s'(s,a)$, $\{Q_t(s,a)\}_{s\in\mathcal{S}, a\in\mathcal{A}}$ известно с прошлой итерации (можно посчитать $\max\limits_{a' \in \mathcal{A}} Q_t(s'(s,a),a')$).

Q-обучение

Вознаграждение $r(s, a; s'(s, a))$ получается при переходе из состояния s при действии a в $s'(s, a)$, в ненаблюдаемых случаях $\alpha_t(s, a) = 0$, то есть значение r не интересно. Подход Q-обучения в отличие от ранее расмотренных полагается на сэмплирование непосредственно траекторий из MDP, что освобождает от знания всего пространства $\mathcal{S} \times \mathcal{A}$ в каждый конечный момент времени, при этом описанный процесс вычисления Q_{t+1} всё также реализован через сжимающее отображение, гарантирующее асимптотическую сходимость к оптимальной политике.

Теорема [7]

Если при стратегии $a(s)$ с вероятностью 1 каждая пара (s, a) будет неограниченное число раз встречаться на бесконечном горизонте наблюдения, то при

$$\sum_{t=0}^{\infty} \alpha_t(s, a) = \infty, \qquad \sum_{t=0}^{\infty} \alpha_t^2(s, a) < \infty,$$

следует сжимаемость (7):

$$\lim_{t \to \infty} Q_t(s, a) = Q(s, a), \qquad V^*(s) = \max_{a \in \mathcal{A}} Q(s, a).$$

Таким образом, после достаточно большого числа шагов, даже в отсутствие какой-либо информации об управляемом марковском процессе, можно определить оптимальную стратегию $a(s) \in \operatorname{Arg}\max\limits_{a \in \mathcal{A}} Q(s, a)$.

Q-обучение

Расширение Q-обучения для θ-параметрической аппроксимации функций $(Q_\theta; \theta \in \mathbb{R}^d)$ выражается через следующее соотношение:

$$\theta_{t+1} = \theta_t + \alpha_t(s,a)\left\{r(s,a;s'(s,a)) + \gamma \max_{a' \in \mathcal{A}} Q_{\theta_t}(s'(s,a),a') - \right.$$
$$\left. - \; Q_{\theta_t}(s,a)\right\} \nabla_\theta Q_{\theta_t}(s_t,a_t).$$

Для линейной параметризации $Q_\theta = \theta^\top \varphi, \; \varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ обновление параметров выглядит следующим образом [8]:

$$1: \; \delta \leftarrow r(s_t,a_t;s_{t+1}) + \gamma \cdot \max_{a' \in \mathcal{A}} \theta^\top \varphi(s_{t+1},a') - \theta^\top \varphi(s_t,a_t);$$
$$2: \; \theta \leftarrow \theta + \alpha_t(s_t,a_t) \cdot \delta \cdot \varphi(s_t,a_t).$$

Основные подходы в Deep Reinforcement Learning

- *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).

Основные подходы в Deep Reinforcement Learning

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

Первый способ вывода policy gradient

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 вывода policy gradient

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 вывода policy gradient

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

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

Первый способ вывода policy gradient

Второе слагаемое

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

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 вывода policy gradient

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 вывода policy gradient

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 вывода policy gradient

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:

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

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

Второй способ вывода policy gradient

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 вывода policy gradient

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

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

Второй способ вывода policy gradient

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

$$\nabla_\theta \log p_\theta(\tau) = \nabla_\theta \sum_{t=0}^{T} [\log \pi_\theta(a_t | s_t)] + \underbrace{\nabla_\theta \sum_{t=0}^{T} [\log p(s_{t+1} | a_t, s_t)]}_{\substack{\text{в среднем, нулевой вектор} \\ \text{по трюку производной логарифма}}}$$

Второй способ вывода policy gradient

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:

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

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

Приближаясь 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?

- $\mathbb{E}_{\tau \sim \pi} \to$ сыграем несколько полных игр при помощи текущей стратегии π — алгоритм будет *on-policy*.
- $Q^{\pi}(s_t, a_t) \to r(\tau)$ — 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 $\tau_1, \tau_2, \dots, \tau_N \sim \pi$.
2. Для каждого t в каждом τ_i считаем $r_t(\tau) := \sum_{\hat{t} \geq t} \gamma^{\hat{t}-t} r_{\hat{t}}$.
3. Считаем оценку градиента:

$$\nabla_\theta J(\pi) := \frac{1}{N} \sum_{\tau} \sum_{t \geqslant 0} \gamma^t \nabla_\theta \log \pi(a_t|s_t,\theta) r_t(\tau).$$

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:

$$p(s'|s) = \int\limits_{A} \pi(a|s)p(s'|s,a)da.$$

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

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

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

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

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

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

Два типа стохастики

Продолжение доказательства

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

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

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

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

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 «умным» 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 — <u>критика</u> (critic). Нейросеть, моделирующую стратегию, соответственно будем называть <u>актёром или актором</u> (actor), и такие алгоритмы, в которых обучается как модель критика, так и модель актора, называются <u>Actor-Critic</u>.

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 $a_{t+1} \sim \pi(a_{t+1} \mid s_{t=1})$, $s_{t+2} \sim \pi(s_{t+2} \mid s_{t+1}, a_{t+1})$, $\dots$, $s_{t+N} \sim \pi(s_{t+N} \mid s_{t+N-1}, a_{t+N-1})$.

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 <u>оценкой максимальной длины</u> (max trace estimation) оценку с максимальным заглядыванием в будущее: для Q-функции

$$y^{\text{MaxTrace}}(s_t, a_t) := \sum_{\hat{t}=t}^{N-1} \gamma^{\hat{t}-t} r_{\hat{t}} + \gamma^{N-t} V^{\pi}(s_N), \tag{8}$$

для Advantage функции, соответственно:

$$\Psi^{\text{MaxTrace}}(s_t, a_t) := y^{\text{MaxTrace}}(s_t, a_t) - V^{\pi}(s_t). \tag{9}$$

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:

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

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:

$$\Psi_{\mathrm{GAE}}(s, a) := \sum_{t \geq 0}^{M-1} \gamma^t \lambda^t \Psi_{(1)}(s^{(t)}, a^{(t)}). \tag{10}$$

Компромисс между смещением и разбросом

Дана траектория $s, r, s', r', s'', r'' \dots s^{(M)}$ по политике π и приближение $V^\pi(s)$
выполним <u>оценку advantage (credit assignment)</u> для пары s, a (хорошее ли решение принято было?)

Для актора:

$$\nabla := \rho(\theta) \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s, a)}_{\text{оценка advantage}}$$

Для критика:

$$\underbrace{y_Q}_{\text{целевое значение для регрессии}} := \Psi(s, a) + V(s)$$

	$\Psi(s, a)$	**Смещение**	**Разброс**
Монте-Карло	$\Psi_{(\infty)}(s, a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$	0	высокий
N-шагов	$\Psi_{(N)}(s, a) := r + \gamma r' + \dots + \gamma^N V(s^{(N)}) - V(s)$	промежуточное	промежуточный
1-шаг	$\Psi_{(1)}(s, a) := r + \gamma V(s') - V(s)$	высокое	низкий

Проблема: выбор N.

Generalized Advantage Estimation (GAE)

Утверждение

Формула (10) 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 (9), а $\lambda = 0$ всё ещё даст одношаговую оценку.

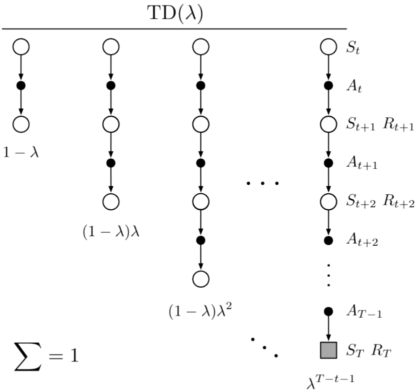
Generalized Advantage Estimation (GAE)

Шаг	Обновление	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	$\dots$	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0
1	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a')$	$1-\lambda$	λ	0		0
2	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a') +$ $+ (\gamma\lambda)^2\Psi_{(1)}(s'',a'')$	$1-\lambda$	$(1-\lambda)\lambda$	λ^2		0
$\vdots$						
N	$\sum_{t\geq 0}^{N}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$	$1-\lambda$	$(1-\lambda)\lambda$	$(1-\lambda)\lambda^2$		λ^N

Эквивалентные формы обновлений TD(λ)

$$\sum_{t=0}^{\infty}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)}) = (1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a)$$

Generalized Advantage Estimation (GAE)



Что если для некоторой пары s, a нам известно будущее только на T шагов вперёд?

$$\Psi^{\mathrm{GAE}}(s, a) := \sum_{t=0}^{T} (\gamma\lambda)^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$

Используемое на практике уравнение:

$$\Psi^{\mathrm{GAE}}(s_t, a_t) = \Psi_{(1)}(s_t, a_t) +$$
$$+ \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$$

Generalized Advantage Estimation (GAE)

В коде формула (10) 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:

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

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

$$y := r + \gamma r' + \gamma^2 r'' + \dots + \gamma^N V_{\phi_k}(s^{(N)}). \tag{11}$$

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 (10) Advantage, достаточно «убрать бэйзлайн»:

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

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

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

$\square$

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 (8); 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):

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

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 или другой оптимизатор первого порядка.

Инициализировать θ, ϕ

На каждом шаге:

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

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

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:

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

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 $\nabla_\theta^{\text{actor}}$

Off-policy оценка advantage

Данной траектории $s_0, r_0, s_1, r_1, s_2, r_2 \dots s_M$ по политике μ и приближению $V^\pi(s)$
требуется произвести <u>оценку advantage (credit assignment)</u> для пары состояние-действие s_0, a_0 в <u>off-policy</u> режиме: $\mu \neq \pi$.

Было бы замечательно воспользоваться GAE:

$$\sum_{t \geq 0} (\gamma\lambda)^t \Psi_{(1)}(s_t, a_t),$$

но $\Psi_{(1)}(s_t, a_t)$ зависит от случайных величин: $a_0, s_0, a_1, s_2, \dots s_{t+1}$.

Внимание!

Если $\pi(a_0 | s_0) = 0$, то ничего не получится.

Воспользуемся коррекцией с помощью выборки по значимости!

$$\Psi = \sum_{t \geq 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=1}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}}{\mu(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}} \right) \Psi_{(1)}(s_t, a_t) =$$

$$= \sum_{t \geq 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=1}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}})}{\mu(a_{\hat{t}} \mid s_{\hat{t}})} \right) \Psi_{(1)}(s_t, a_t), \quad \prod_{\hat{t}=1}^{0} \equiv 1.$$

Непрактично: очень высокая дисперсия!

- «Затухающий» след: $\mu(a|s) \gg \pi(a|s)$:
  - типичная ситуация μ делает примитивные случайные действия, которые π редко совершает. Не лечится.
- «Взрывающийся» след: $\mu(a|s) \ll \pi(a|s)$:
  - μ выбранное действие с малой $\mu(a|s)$, но *вероятное* для π. Причина большой дисперсии.

Присвоение ценности: общий вид

Давайте перепишем ценность следующим образом:

$$\Psi = \sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t), \quad \prod_{i=1}^{0} \equiv 1,$$

где c_i коэффициенты «отжига следа»:

Название оценки	**Коэффициенты c_i**	**Возникающая проблема**
GAE	λ	только on-policy
Одношаговая	0	большое смещение
Выборка по значимости	$\lambda \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)}$	легко «взрывается»

Retrace: основная теорема

$$\Psi = \sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t), \quad \prod_{i=1}^{0} \equiv 1.$$

Теорема о Retrace

В режиме on-policy возможен выбор произвольного коэффициента $c_i \in [0, 1]$, в off-policy режиме можно выбрать произвольный коэффициент в следующем интервале

$$c_i \in \left[0, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right].$$

- «затухающий» след: ничего не поделаешь;
- «взрывающийся» след: если вес из выборки по значимости больше 1, то прменяем клиппинг!

Retrace: финальный результат

$$\Psi = \sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t), \quad \prod_{i=1}^{0} \equiv 1,$$

где

$$c_i := \lambda \min \left(1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right).$$

Используется в:

- off-policy RL алгоритмах для теоретически корректных многошаговых целевых значений;
    - ($\lambda = 1$, потому что оно быстро затухает).
- дистрибутивных on-policy RL системах, где данные о градиенте от некоторых серверов могут задерживаться на несколько итераций обновления.

Источники I

[1] S. Ivanov, “Reinforcement learning textbook,” arXiv preprint arXiv:2201.09746, 2022.

[2] D. Bertsekas, Reinforcement learning and optimal control. Athena Scientific, 2019.

[3] V. Goyal and J. Grand-Clement, “A first-order approach to accelerated value iteration,” Operations Research, vol. 71, no. 2, pp. 517–535, 2023.

[4] M. L. Puterman, Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, 2014.

Источники II

[5] T. Liu, R. Zhou, D. Kalathil, P. Kumar, and C. Tian, "Policy optimization for constrained mdps with provable fast global convergence," arXiv preprint arXiv:2111.00552, 2021.

[6] J. Grand-Clément, "From convex optimization to mdps: A review of first-order, second-order and quasi-newton methods for mdps," arXiv preprint arXiv:2104.10677, 2021.

[7] J. N. Tsitsiklis, "Asynchronous stochastic approximation and q-learning," Machine learning, vol. 16, pp. 185–202, 1994.

[8] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton, "Toward off-policy learning control with function approximation.," in ICML, vol. 10, pp. 719–726, 2010.