

TD-обучение

Сергей Николенко

Академия MADE — Mail.Ru

14 октября 2020 г.

Random facts:

- 14 октября в Грузии — Мцхетоба, христианский государственный праздник, проводящийся в кафедральном храме Светицховели города Мцхета; по легенде, храм основан на том месте, где была захоронена риза Иисуса Христа, привезённая в Грузию грузинскими евреями Элиозом и Лонгинозом, которые присутствовали при распятии
- 14 октября 1843 г. на премьере спектакля «Сон в летнюю ночь» впервые прозвучал «Свадебный марш» Мендельсона, а 14 октября 1860 г. был открыт Мариинский театр
- 14 октября 1892 г. Артур Конан Дойль опубликовал книгу «Приключения Шерлока Холмса», а 14 октября 1926 г. в Лондоне вышла книга Алана Милна «Винни-Пух»
- 14 октября 1943 г. произошло восстание в концлагере Собибор, единственное удачное в истории Третьего рейха
- 14 октября 2012 г. Феликс Баумгартнер прыгнул с парашютом с высоты 39 км и успешно приземлился в окрестностях, что характерно, города Розуэлл, Нью-Мексико

Основные задачи

- Теоретически всё готово, но у нас много проблем:
 - уравнения знаем, но пока не знаем, как их решать, то есть как найти V^π для данного π ?
 - разных стратегий очень, очень много — как найти оптимальную стратегию поведения агента в данной модели и соответствующие V^* ?
 - но уравнений тоже не знаем — в реальности обычно P и R не даны, их тоже нужно обучить; как?
 - более того, их обычно даже записать не получится, слишком уж много состояний в любой реальной задаче... что делать?



- Давайте есть слона по частям...

Методы Монте-Карло

- В прошлый раз мы ввели основные понятия динамики марковских процессов принятия решений:
 - собственно динамику процесса:

$$p(s', r | s, a) = p(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a);$$

- награды за каждый эпизод, начиная со времени t :

$$G_t = R_{t+1} + \gamma G_{t+1} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1};$$

- функцию значения для состояний и пар состояние-действие:

$$V_{\pi}(s) = \mathbb{E}_{\pi} [G_t \mid S_t = s] = \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right],$$

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi} [G_t \mid S_t = s, A_t = a] = \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

- В прошлый раз мы выписали уравнения Беллмана на V и Q и научились их решать.
- Теперь будем обучать одновременно и модель, и оптимальную стратегию; вознаграждения и переходы не даны.
- Начнём со стохастических алгоритмов (метода Монте-Карло); но начнём опять с простой задачи.
- Как обучить вознаграждения $V^\pi(s)$, ожидаемые от состояния s в эпизодической задаче?

- Да очень просто: будем накапливать данные и усреднять.

Алгоритм Monte Carlo estimation:

- инициализировать случайно π и $V(s)$, пустые списки $\text{Ret}(s)$;
- повторять до сходимости:
 - сгенерировать эпизод $S_0, A_0, R_1, S_1, A_1, \dots, S_T$ по стратегии π ;
 - $G := 0$
 - для каждого $t = T - 1, T - 2, \dots, 0$:
 - $G := \gamma G + R_{t+1}$
 - если надо, то добавить G в $\text{Ret}(S_t)$ и обновить $V(S_t) := \text{Avg}(\text{Ret}(S_t))$.
- «Если надо» скрывает тонкую разницу между first-visit и every-visit Monte Carlo.
- На выходе этот алгоритм выдаст V_π для данной π , которой порождаются эпизоды.

- Но вообще без модели гораздо удобнее оценивать Q_π . Сразу можно и стратегию обновлять, тот же policy iteration.

Алгоритм Monte Carlo control with exploring starts:

- инициализировать случайно π и $Q(s)$, пустые списки $\text{Ret}(s)$;
- повторять до сходимости:
 - выбрать S_0, A_0 случайно так, чтобы $\forall (s, a) \quad p(s, a) > 0$;
 - сгенерировать эпизод $S_0, A_0, R_1, S_1, A_1, \dots, S_T$ по стратегии π ;
 - $G := 0$
 - для каждого $t = T - 1, T - 2, \dots, 0$:
 - $G := \gamma G + R_{t+1}$
 - если надо, то добавить G в $\text{Ret}(S_t, A_t)$ и обновить:

$$Q(S_t, A_t) := \text{Avg}(\text{Ret}(S_t, A_t)),$$

$$\pi(S_t) := \arg \max_a Q(S_t, a).$$

- Этот алгоритм выдаст π_* и соответствующую ей функцию Q_* .
- Здесь важно предположение exploring starts, без него мы не исследуем все действия.

Метод Монте-Карло

- А что делать, если оно не выполняется? Придётся исследовать самостоятельно.

Алгоритм on-policy Monte Carlo control с мягкими стратегиями:

- инициализировать случайно ϵ -мягкую π и $Q(s)$, пустые $\text{Ret}(s)$;
- повторять до сходимости:
 - выбрать S_0, A_0 случайно так, чтобы $\forall (s, a) \quad p(s, a) > 0$;
 - сгенерировать эпизод $S_0, A_0, R_1, S_1, A_1, \dots, S_T$ по стратегии π ;
 - $G := 0$
 - для каждого $t = T - 1, T - 2, \dots, 0$:
 - $G := \gamma G + R_{t+1}$
 - если надо, то добавить G в $\text{Ret}(S_t, A_t)$ и обновить:

$$Q(S_t, A_t) := \text{Avg}(\text{Ret}(S_t, A_t)),$$

$$a_* := \arg \max_a Q(S_t, a),$$

$$\pi(a \mid S_t) := \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(S_t)|}, & \text{если } a = a_*, \\ \frac{\epsilon}{|\mathcal{A}(S_t)|}, & \text{если } a \neq a_*. \end{cases}$$

- Этот алгоритм умеет искать оптимальную мягкую π .

- На самом деле для ϵ -мягких стратегий тоже верен аналог policy improvement теоремы, и для ϵ -мягких стратегий метод policy iteration тоже вполне работает.
- Но это on-policy алгоритм, он найдёт мягкую стратегию, а в реальности шахматист, который играет как Магнус Карлсен 90% ходов, а 10% ходов делает случайно, вряд ли продвинется сильно дальше третьего разряда.
- Но ведь и исследовать тоже нужно! Хорошо было бы научиться исследовать по одной стратегии, а оценивать другую...

Метод Монте-Карло

- ...и такой трюк действительно можно сделать!
- Вспомним сэмплирование со значимостями (importance sampling): если мы умеем брать сэмплы по распределению $q(\mathbf{x})$, а оценивать хотим ожидание по распределению $p(\mathbf{x})$, то можно сделать так:

$$\mathbb{E}_{p(\mathbf{x})} [f(\mathbf{x})] = \int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{q(\mathbf{x})} \left[f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} \right].$$

- А у нас в качестве p и q выступают распределения на траекториях:

$$\begin{aligned} p(\text{Traj}|\pi) &= p(A_t, S_{t+1}, A_{t+1}, \dots, S_{T-1}, A_{T-1}, S_T|\pi) = \\ &= \pi(A_t|S_t) p(S_{t+1}|S_t, A_t) \dots \pi(A_{T-1}|S_{T-1}) p(S_T|S_{T-1}, A_{T-1}) = \\ &= \prod_{k=t}^{T-1} \pi(A_k|S_k) p(S_{k+1}|S_k, A_k). \end{aligned}$$

- И получается, что для двух стратегий π , b (от слова behaviour) определить веса

$$\begin{aligned}\rho_{t:T-1}^{\pi,b} &= \frac{p(\text{Traj}|\pi)}{p(\text{Traj}|b)} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k) p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k | S_k) p(S_{k+1}|S_k, A_k)} = \\ &= \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k | S_k)},\end{aligned}$$

то неизвестные вероятности сократятся и останется, что когда мы порождаем эпизоды по b , нужно просто усреднять не G_t , а $\rho_{t:T-1}^{\pi,b} G_t$, и будут получаться оценки π !

- Единственное условие – покрытие (coverage): должно быть верно, что если $\pi(a|s) > 0$, то и $b(a | s) > 0$.

- Ещё есть тонкая разница между обычным importance sampling и взвешенным (weighted importance sampling):
 - в обычном мы берём оценку среднего через сэмплы

$$V(s) = \frac{1}{N} \sum_{t=1}^N \rho_{t:T-1}^{\pi,b} G_t,$$

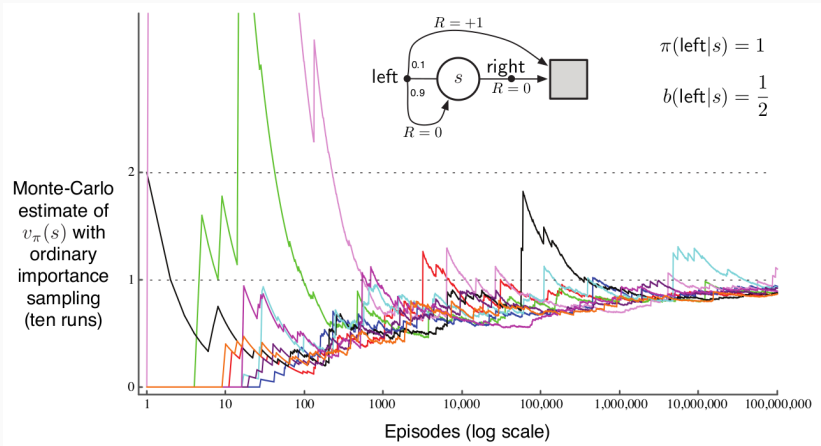
- а во взвешенном ещё нормируем суммой весов

$$V(s) = \frac{\sum_{t=1}^N \rho_{t:T-1}^{\pi,b} G_t}{\sum_{t=1}^N \rho_{t:T-1}^{\pi,b}}.$$

- Вторая оценка смещённая, но сходится куда надо и у неё нормальная человеческая дисперсия.

Метод Монте-Карло

- А у первого варианта дисперсия очень большая, и может быть даже бесконечная! Пример:



- Итого вот какой алгоритм получается

Алгоритм on-policy Monte Carlo control с мягкими стратегиями:

- инициализировать ϵ -мягкую b , π , $Q(s)$, пустые $\text{Ret}(s)$ и $c(s)$;
- повторять до сходимости:
 - сгенерировать эпизод $S_0, A_0, R_1, S_1, A_1, \dots, S_T$ по мягкой стратегии b ;
 - $G := 0$, $W := 1$
 - для каждого $t = T - 1, T - 2, \dots, 0$:
 - $G := \gamma G + R_{t+1}$
 - $c(S_t, A_t) := c(S_t, A_t) + W$
 - $Q(S_t, A_t) := Q(S_t, A_t) + \frac{W}{c(S_t, A_t)} (G - Q(S_t, A_t))$
 - $\pi(a | S_t) := \arg \max_a Q(S_t, a)$
 - если $A_t \neq \pi(S_t)$, то перейти к следующему эпизоду
 - $W := \frac{W}{b(A_t | S_t)}$.
- А если убрать $\arg \max$, то получится просто алгоритм оценки данной стратегии π .

TD-обучение

- Общий принцип TD-обучения: давайте обучать оценки состояний на основе обученных нами ранее оценок для последующих состояний.
- $TD(0)$ -обучение: инициализировать $V(s)$ и π произвольно, затем на каждом эпизоде обучения:
 - инициализировать s ;
 - для каждого шага t в эпизоде:
 - выбрать A_t в состоянии S_t по стратегии π ;
 - сделать A_t , пронаблюдать результат R_{t+1} и следующее состояние S_{t+1} ;
 - $V(S_t) := V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$.

- Здесь по сути методы Монте-Карло и TD-обучение расходятся в том, как строить оценку для $V(s)$:
 - MC-методы оценивают $V(s) = \mathbb{E}_{\pi} [G_t \mid S_t = s]$, собирая статистику из G_t ;
 - а TD-методы оценивают на один шаг вперёд как $V(s) = \mathbb{E}_{\pi} [R_{t+1} + \gamma V_{\pi}(S_{t+1}) \mid S_t = s]$.
- Смысл TD-обучения в том, чтобы использовать уже обученные закономерности для поиска более глубоких закономерностей.
- В результате обучение получится целенаправленным, обучается гораздо быстрее, чем другие стратегии.

- Здесь тоже есть on-policy и off-policy варианты

Алгоритм Sarsa (on-policy TD control):

- инициализировать случайно $Q(s, a)$;
- повторять до сходимости:
 - инициализировать S_0 , выбрать A_0 по стратегии, полученной из Q (например, по ϵ -жадной стратегии);
 - для каждого шага в эпизоде $t = 0, \dots, T$:
 - сделать действие A_t , получить награду R_{t+1} , перейти в состояние S_{t+1} ;
 - выбрать A_{t+1} по стратегии, полученной из Q (например, по ϵ -жадной стратегии);
 - обновить Q :

$$Q(S_t, A_t) := Q(S_t, A_t) + \alpha (R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$$

- Этот алгоритм умеет искать оптимальную *мягкую* π (т.е. опять нужно самому исследовать)

- А off-policy вариант называется Q-обучение; он ещё проще, и это была очень мощная идея, которая до сих пор определяет многое в RL (Watkins, 1989)

Алгоритм Q-learning (off-policy TD control):

- инициализировать случайно $Q(s, a)$;
- повторять до сходимости:
 - инициализировать S_0
 - для каждого шага в эпизоде $t = 0, \dots, T$:
 - выбрать A_t по стратегии, полученной из Q (например, по ϵ -жадной стратегии);
 - сделать действие A_t , получить награду R_{t+1} , перейти в состояние S_{t+1} ;
 - обновить Q :

$$Q(S_t, A_t) := Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right)$$

- Этот алгоритм умеет искать оптимальную жёсткую π_* , делая ходы по мягкой стратегии

Спасибо!

Спасибо за внимание!