

ГРАДИЕНТ ПО СТРАТЕГИЯМ

Сергей Николенко

СПбГУ — Санкт-Петербург

20 мая 2022 г.

Random facts:

- 20 мая в ООН — Всемирный день пчёл, отмечаемый в день рождения основоположника современного пчеловодства, словенца, пчеловода и художника Антона Янши
- 20 мая в Камбодже — День праведного гнева, ранее Национальный день ненависти к геноциду Пол Пота; именно 20 мая 1976 года красные кхмеры начали массовые убийства
- 20 мая 1862 г. в США был принят Homestead Act, по которому все граждане стали наделяться почти бесплатными наделами в 65 га на неосвоенном западе; по легенде, первый гомстед был зарегистрирован на имя Дэна Фримена через 10 минут после вступления закона в силу (то есть в 0:10 ночи 1 января 1863 г.)
- 20 мая 1875 г. в Зале Часов здания МИД Франции в Париже 17 государств, в том числе Россия, подписали «Метрическую конвенцию»
- 20 мая 1948 г. генералиссимус Чан Кай-Ши стал первым президентом Китайской республики (частично признанной, на Тайване), а 20 мая 1970 г. Мао Цзэдун призвал к началу мировой революции против американского империализма
- 20 мая 2003 г. Святой Престол выпустил словарь современного латинского языка с такими древнеримскими понятиями, как видеотелефон, ФБР, бестселлер и так далее

POLICY GRADIENT

- Мы до сих пор говорили про action-value методы: берём марковский процесс принятия решений и ищем V и/или Q .
- Но это не единственный подход!
- Что если мы будем оптимизировать стратегии напрямую?

$$\pi(a|s, \theta) = p(A_t = a \mid S_t = s, \theta_t = \theta)$$

- Может быть, есть ещё и $\hat{V}(s, \mathbf{w})$, это отдельно.

- Тогда можно попробовать сформулировать цель $J(\theta)$ и просто двигаться как

$$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta).$$

- Такие методы называются policy gradient методами.
- А если ещё и $\hat{V}(s, \mathbf{w})$, то actor-critic методами: критик – это \hat{V} , которая «критикует» стратегию π .

- Если действий не слишком много, можно искать $h(s, a, \theta)$ (хоть линейно, хоть нейросетью) и строить

$$\pi(a|s, \theta) = \text{softmax}(h).$$

- Преимущества:
 - можно сходиться к детерминированной стратегии без явной стратегии уменьшения ϵ ;
 - а можно строить стохастическую стратегию, которую через $Q(s, a)$ вообще непонятно как найти;
 - оптимизировать π может быть проще, чем Q .
- Но как же это сделать?

- Policy gradient theorem:

$$\begin{aligned}\nabla V_{\pi}(s) &= \nabla \left(\sum_a \pi(a|s) Q_{\pi}(s, a) \right) = \\ &= \dots = \\ &= \sum_{x \in S} \sum_{k=0}^{\infty} p(\text{перейти из } s \text{ в } x \text{ за } k \text{ шагов по } \pi) \sum_a \nabla \pi(a|x) Q_{\pi}(x, a),\end{aligned}$$

то есть

$$\nabla J(\theta) \propto \nabla V_{\pi}(s) = \sum_s \mu(s) \sum_a \nabla \pi(a|s) Q_{\pi}(s, a),$$

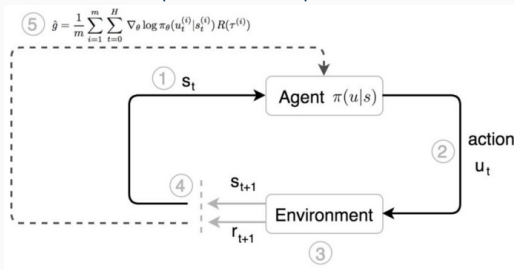
где $\mu(s)$ – ожидаемая доля времени, проведённого в состоянии s .

POLICY GRADIENT

- Иначе говоря,

$$\nabla J(\theta) \propto \mathbb{E}_{\pi} \left[\sum_a \nabla \pi(a|s) Q_{\pi}(s, a) \right],$$

и теперь можно и алгоритмы построить.



- Можно all-actions алгоритм:

$$\theta_{t+1} = \theta_t + \alpha \sum_a \hat{Q}(S_t, a, \mathbf{w}) \nabla_{\theta} \pi(a|S_t, \theta).$$

- Но лучше уж сразу REINFORCE (Williams, 1992):

$$\begin{aligned}\nabla J(\theta) &\propto \mathbb{E}_\pi \left[\sum_a \nabla \pi(A_t | S_t, \theta) Q_\pi(S_t, a) \right] = \\ &= \mathbb{E}_\pi \left[Q_\pi(S_t, A_t) \frac{\nabla \pi(A_t | S_t, \theta)}{\pi(A_t | S_t, \theta)} \right] = \mathbb{E}_\pi \left[G_t \frac{\nabla \pi(A_t | S_t, \theta)}{\pi(A_t | S_t, \theta)} \right],\end{aligned}$$

потому что $Q_\pi(S_t, A_t) = \mathbb{E}_\pi [G_t | S_t, A_t]$.

- И алгоритм такой:

$$\theta_{t+1} = \theta_t + \alpha G_t \nabla \log \pi(A_t | S_t, \theta).$$

- Т.е. двигаем туда, где максимальный return, всё логично.

- G_t можно оценить по методу Монте-Карло:

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \theta)$

Algorithm parameter: step size $\alpha > 0$

Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):

 Generate an episode $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \theta)$

 Loop for each step of the episode $t = 0, 1, \dots, T - 1$:

$$G \leftarrow \sum_{k=t+1}^T \gamma^{k-t-1} R_k$$

$$\theta \leftarrow \theta + \alpha \gamma^t G \nabla \ln \pi(A_t | S_t, \theta)$$

(G_t)

- Но это ещё не всё. Ещё можно добавить baseline:

$$\nabla J(\theta) \propto \nabla V_{\pi}(s) = \sum_s \mu(s) \sum_a (Q_{\pi}(s, a) - b(s)) \nabla \pi(a|s, \theta),$$

и это не повлияет ни на что, мы вычитаем

$$b(s) \nabla \sum_a \pi(a|s, \theta) = b(s) \nabla 1 = 0.$$

- Это не меняет ожидание, но может сильно изменить дисперсию! В том числе в хорошую сторону.
- Естественный baseline – это оценка состояния $\hat{V}(S_t, \mathbf{w})$.

- Его нужно будет оценить, как мы обычно оцениваем \hat{V} , то есть теперь два разных градиента получается:

REINFORCE with Baseline (episodic), for estimating $\pi_{\theta} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$

Input: a differentiable state-value function parameterization $\hat{v}(s, \mathbf{w})$

Algorithm parameters: step sizes $\alpha^{\theta} > 0$, $\alpha^{\mathbf{w}} > 0$

Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):

Generate an episode $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \theta)$

Loop for each step of the episode $t = 0, 1, \dots, T - 1$:

$$G \leftarrow \sum_{k=t+1}^T \gamma^{k-t-1} R_k \quad (G_t)$$

$$\delta \leftarrow G - \hat{v}(S_t, \mathbf{w})$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S_t, \mathbf{w})$$

$$\theta \leftarrow \theta + \alpha^{\theta} \gamma^t \delta \nabla \ln \pi(A_t|S_t, \theta)$$

- А чтобы получился полноценный actor-critic, нужно через \hat{V} оценить и результаты действия тоже:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha \left(G_{t:t+1} - \hat{V}(S_t, \mathbf{w}) \right) \nabla \log \pi(A_t | S_t, \theta) = \\ &= \theta_t + \alpha \left(R_{t+1} + \gamma \hat{V}(S_{t+1}, \mathbf{w}) - \hat{V}(S_t, \mathbf{w}) \right) \nabla \log \pi(A_t | S_t, \theta) = \\ &= \theta_t + \alpha \delta_t \frac{\nabla \pi(A_t | S_t, \theta)}{\pi(A_t | S_t, \theta)}.\end{aligned}$$

- Алгоритм получается такой:

One-step Actor–Critic (episodic), for estimating $\pi_{\theta} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$

Input: a differentiable state-value function parameterization $\hat{v}(s, \mathbf{w})$

Parameters: step sizes $\alpha^{\theta} > 0$, $\alpha^{\mathbf{w}} > 0$

Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):

 Initialize S (first state of episode)

$I \leftarrow 1$

 Loop while S is not terminal (for each time step):

$A \sim \pi(\cdot|S, \theta)$

 Take action A , observe S', R

$\delta \leftarrow R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$ (if S' is terminal, then $\hat{v}(S', \mathbf{w}) \doteq 0$)

$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S, \mathbf{w})$

$\theta \leftarrow \theta + \alpha^{\theta} I \delta \nabla \ln \pi(A|S, \theta)$

$I \leftarrow \gamma I$

$S \leftarrow S'$

- Можно с eligibility traces, как мы раньше обсуждали:

Actor-Critic with Eligibility Traces (episodic), for estimating $\pi_{\theta} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$

Input: a differentiable state-value function parameterization $\hat{v}(s, \mathbf{w})$

Parameters: trace-decay rates $\lambda^{\theta} \in [0, 1]$, $\lambda^{\mathbf{w}} \in [0, 1]$; step sizes $\alpha^{\theta} > 0$, $\alpha^{\mathbf{w}} > 0$

Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):

 Initialize S (first state of episode)

$\mathbf{z}^{\theta} \leftarrow \mathbf{0}$ (d' -component eligibility trace vector)

$\mathbf{z}^{\mathbf{w}} \leftarrow \mathbf{0}$ (d -component eligibility trace vector)

$I \leftarrow 1$

 Loop while S is not terminal (for each time step):

$A \sim \pi(\cdot|S, \theta)$

 Take action A , observe S', R

$\delta \leftarrow R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$ (if S' is terminal, then $\hat{v}(S', \mathbf{w}) \doteq 0$)

$\mathbf{z}^{\mathbf{w}} \leftarrow \gamma \lambda^{\mathbf{w}} \mathbf{z}^{\mathbf{w}} + \nabla \hat{v}(S, \mathbf{w})$

$\mathbf{z}^{\theta} \leftarrow \gamma \lambda^{\theta} \mathbf{z}^{\theta} + I \nabla \ln \pi(A|S, \theta)$

$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \mathbf{z}^{\mathbf{w}}$

$\theta \leftarrow \theta + \alpha^{\theta} \delta \mathbf{z}^{\theta}$

$I \leftarrow \gamma I$

$S \leftarrow S'$

- А можно попробовать сделать off-policy вариант; для другой стратегии β :

$$\begin{aligned}\nabla J(\theta) &\propto \nabla_{\theta} \sum_s \mu_{\pi}(s) \sum_a \pi(a|s) Q_{\pi}(s, a) = \\ &= \sum_s \mu_{\pi}(s) \sum_a [\nabla_{\theta} \pi(a|s) Q_{\pi}(s, a) + \pi(a|s) \nabla_{\theta} Q_{\pi}(s, a)],\end{aligned}$$

- Теперь первое слагаемое можно оценить как обычно:

$$\sum_s \mu_{\pi}(s) [\dots] = \mathbb{E}_{\pi} [\dots] = \mathbb{E}_{\beta} \left[\frac{\pi(a|s)}{\beta(a|s)} \dots \right], \text{ то есть}$$

$$\nabla J(\theta) \propto \mathbb{E}_{\beta} \left[\frac{\pi(a|s)}{\beta(a|s)} Q_{\pi}(s, a) \nabla_{\theta} \ln \pi(a|s) \right].$$

- На первый взгляд кажется, что всё равно всё пропало, потому что $\nabla_{\theta} Q_{\pi}(s, a)$ мы никак не оценим.
- Но оказывается (Degris et al., 2012), что его можно просто выбросить, и полученная аппроксимация всё равно неплохая и минимум у неё там же.
- Так что в алгоритме просто добавятся importance weights:

Algorithm 1 The Off-PAC algorithm

Initialize the vectors \mathbf{e}_v , \mathbf{e}_u , and \mathbf{w} to zero

Initialize the vectors \mathbf{v} and \mathbf{u} arbitrarily

Initialize the state s

For each step:

 Choose an action, a , according to $b(\cdot|s)$

 Observe resultant reward, r , and next state, s'

$$\delta \leftarrow r + \gamma(s')\mathbf{v}^T\mathbf{x}_{s'} - \mathbf{v}^T\mathbf{x}_s$$

$$\rho \leftarrow \pi_{\mathbf{u}}(a|s)/b(a|s)$$

 Update the critic (GTD(λ) algorithm):

$$\mathbf{e}_v \leftarrow \rho(\mathbf{x}_s + \gamma(s)\lambda\mathbf{e}_v)$$

$$\mathbf{v} \leftarrow \mathbf{v} + \alpha_v [\delta\mathbf{e}_v - \gamma(s')(1 - \lambda)(\mathbf{w}^T\mathbf{e}_v)\mathbf{x}_s]$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_w [\delta\mathbf{e}_v - (\mathbf{w}^T\mathbf{x}_s)\mathbf{x}_s]$$

 Update the actor:

$$\mathbf{e}_u \leftarrow \rho \left[\frac{\nabla_{\mathbf{u}}\pi_{\mathbf{u}}(a|s)}{\pi_{\mathbf{u}}(a|s)} + \gamma(s)\lambda\mathbf{e}_u \right]$$

$$\mathbf{u} \leftarrow \mathbf{u} + \alpha_u\delta\mathbf{e}_u$$

$$s \leftarrow s'$$

- В policy gradient можно рассмотреть даже, например, непрерывные действия.
- Например, параметризуем действие $a \in \mathbb{R}$ через гауссиан:

$$\pi(a|s, \theta) = \frac{1}{\sigma(s, \theta)\sqrt{2\pi}} e^{-\frac{1}{2\sigma(s, \theta)^2} (a - \mu(s, \theta))^2}.$$

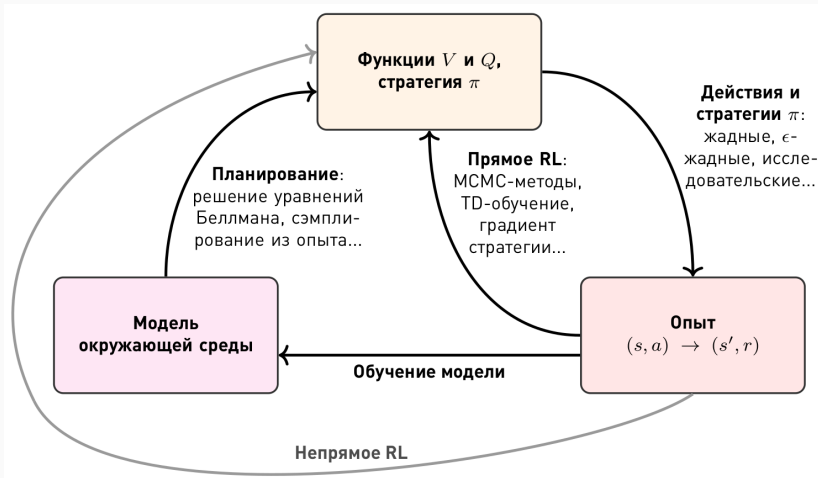
- Если, например,

$$\mu(s, \theta) = \theta_\mu^\top \mathbf{x}_\mu(s), \quad \sigma(s, \theta) = e^{\theta_\sigma^\top \mathbf{x}_\sigma(s)},$$



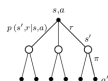



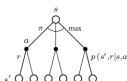

то можно подсчитать градиенты:

$$\begin{aligned} \nabla \ln \pi(a|s, \theta_\mu) &= \frac{1}{\sigma(s, \theta)^2} (a - \mu(s, \theta)) \mathbf{x}_\mu(s), \\ \nabla \ln \pi(a|s, \theta_\sigma) &= \left(\frac{1}{\sigma(s, \theta)^2} (a - \mu(s, \theta))^2 - 1 \right) \mathbf{x}_\sigma(s). \end{aligned}$$

ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ



ОБЗОР МЕТОДОВ

	Функция	Обновление в ожидании	Обновление по выборке
Оценка стратегии π	$V_{\pi}(S_t) :=$	Уравнение Беллмана $\sum_{a \in \mathcal{A}} \pi(a S_t) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r s, a) (r + \gamma V_{\pi}(s'))$ 	TD(0) $V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$ 
	$Q_{\pi}(S_t, A_t) :=$	Уравнение Беллмана $\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r S_t, A_t) (r + \gamma \sum_{a' \in \mathcal{A}} \pi(a' s') Q_{\pi}(s', a'))$ 	Sarsa, Sarsa с ожиданием $Q(S_t, A_t) + \alpha (R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$ $Q(S_t, A_t) + \alpha (R_{t+1} + \gamma \sum_a \pi(a S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t))$  
Управление, поиск π_*	$V_*(S_t) :=$	Уравнение Беллмана $\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r S_t, a) (r + \gamma V_*(s'))$ 	
	$Q_*(S_t, A_t) :=$	Уравнение Беллмана $\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r S_t, A_t) \left(r + \gamma \max_{a'} Q_*(s', a') \right)$ 	Q-обучение $Q(S_t, A_t) + \alpha (R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$ 

Спасибо за внимание!