Лекции 3 и 4

Обучение с подкреплением: стратегии / критика

Дополнительные главы машинного обучения

Андрей Фильченков

05.03.2021 и 02.04.2021

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

- В презентации используются материалы курсов
  «Машинное обучение с подкреплением» А.И. Панова
  CS234: Reinforcement Learning, E. Brusnkill
- Слайды доступны: **shorturl.at/wGV59**
- Видео доступны: **shorturl.at/ovBTZ**

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Параметрическое Q (напоминание)

Будем считать, что $Q(s, a, \theta)$ параметрическое, с некоторым параметром $\theta \in \Theta$, который мы будем обучать.

Параметрическое π

Будем считать, что $Q(s, a, \theta)$ параметрическое, с некоторым параметром $\theta \in \Theta$, который мы будем обучать.

Но вместо того, чтобы думать про параметры функции ценности, можно сразу думать про параметры стратегии, $\pi(s, a, \theta)$.

Зачем это делать?

- Не будем в явном виде зависеть от размерности пространств состояний и действий
- Сможем работать с непрерывными действиями

- Более того, в непрерывных пространствах можно считать градиенты по аргументам

А что в случае с дискретным A?

Функция $r(s, a)$ не дифференцируема по действиям a, если $|A|$ дискретно.

Что делать?

А что в случае с дискретным A?

Что делать?

Использовать только стохастические стратегии. По сути, они работают не с самими действиями, а с вероятностными распределениями над действиями, что делает их непрерывными.

Если стратегия $\pi_\theta(s, a)$ дифференцируема по параметрам, то $V^\pi(s)$ тоже дифференцируема по θ.

Анализ

Плюсы
- Лучше сходится
- Чем больше состояний, тем лучше работает
- Работают со стохастическими стратегиями

Минусы
- Процесс вычисления стратегий обычно неэффективен и высокодисперсен
- Все проблемы градиентного спуска

Функция ценности стратегии

В эпизодических средах начальная ценность

$$J_1(\theta) = \mathrm{E}_{\pi_\theta} V^{\pi_\theta}(s_1)$$

В непрерывных средах придем к стационарному распределению марковской цепи $d^{\pi_\theta}(\mathrm{s})$ для π_θ

средняя ценность

$$J_{avgV}(\theta) = \mathrm{E}_{\pi_\theta} \sum_s d^{\pi_\theta}(s) V^{\pi_\theta}(s_1)$$

или среднее вознаграждение за шаг

$$J_{avgR}(\theta) = \mathrm{E}_{\pi_\theta} \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s,a)\, r(s,a)$$

Подходы к оптимизации

Матожидания в предыдущих формулах приближаются по Монте-Карло

Будем оптимизировать функционал для каждого запуска, избавляясь от усреднения

Можно применять произвольные методы оптимизации

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Переход к траекториям

Траектория $\tau = (s_1, a_1, r_1, \dots, s_T, a_T, r_T)$

Ценность траектории $r(\tau) = \sum_{t=1}^{T} r_t$

Распределение $p(\tau, \theta)$ над траекториями

$$J_1(\theta) = \mathrm{E}_{\pi_\theta} V^{\pi_\theta}(s_1) = \sum_{\tau} r(\tau) p(\tau, \theta)$$

Градиентный спуск

$$\nabla_\theta J_1(\theta) = \nabla_\theta \sum_\tau r(\tau) p(\tau,\theta) =$$

$$= \sum_\tau r(\tau) p(\tau,\theta) \frac{\nabla_\theta p(\tau,\theta)}{p(\tau,\theta)} =$$

$$= \sum_\tau r(\tau) p(\tau,\theta) \nabla_\theta \log p(\tau,\theta)$$

Оценим p по выборке размера m:

$$\nabla_\theta J_1(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} r(\tau_i) \nabla_\theta \log p(\tau_i,\theta)$$

Результирующая функция

Просто распишем, как будто мы знаем модель

Пусть $p_{prior}(s_1)$ — распределение начальных состояний

$$\nabla_\theta \log p(\tau_i, \theta) = \nabla_\theta \log \left[p_{prior}(s_1) \prod_{t=1}^{T} \pi_\theta(s_t, a_t)\, \mathcal{P}_{s_t s_{t+1}}^{a_t} \right] =$$

$$= \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t)$$

$\nabla_\theta \log \pi_\theta(s, a)$ — **результирующая функция**

Суммирование

Оптимизационная задача

$$\arg\max_{\theta} J_1(\theta)$$

По выборке из m траекторий

$$J_1(\theta) \approx \frac{1}{m}\sum_{i=1}^{m} r(\tau_i)\nabla_\theta \log p(\tau_i, \theta) =$$

$$= \frac{1}{m}\sum_{i=1}^{m} r(\tau_i)\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t)$$

Не нужна модель среды

Несмещенная, но очень дисперсная оценка

То же, но с ценностью действий

$$J_1(\theta) = \mathrm{E}_{\pi_\theta} V^{\pi_\theta}(s_1) = \mathrm{E}_{\pi_\theta} \mathrm{E}_a Q^{\pi_\theta}(s_1, a)$$

Тогда

$$\nabla_\theta J_1(\theta) = \mathrm{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) Q^{\pi_\theta}(s, a)]$$

Мы можем обобщить этот результат на произвольную функцию ценности действий

Интерпретация

Для функции $g(x_i) = f(x_i)\nabla_\theta \log p(x_i, \theta)$, где f измеряет полезность примера, движение в сторону роста g соответствует увеличению логарифма вероятности примера пропорционально его ценности.

Теорема о градиенте стратегии

Для любой дифференцируемой стратегии $\pi(s, a, \theta)$ и любой функции ценности стратегии $J = J_1, J_{avgR}$ или $\frac{1}{1-\gamma} J_{avgV}$

$$\nabla_\theta J(\theta) = \mathrm{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) Q^{\pi_\theta}(s, a)]$$

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

REINFORCE

В качестве оценки $Q^{\pi_\theta}(s,a)$ будем использовать R_t

1. Инициализируем $\theta_{(0)}$
2. Для всех траекторий $\tau_i = (s_1, \ldots, r_T)$
3. Для шагов в траектории t
4. $\theta_{(k+1)} = \theta_{(k)} + \alpha\, \nabla_\theta \log \pi_{\theta_{(k)}}(s_t, a_t)\, R_t$
5. Возвращаем последнее значение $\theta_{()}$

Анализ

Достоинства:

- Легко обобщается на задачи с большим множеством действий, в том числе на задачи с непрерывным множество действий
- Почти нет конфликта между исследованием и использованием в явном виде
- Имеет более сильные гарантии сходимости

Недостатки:

- Очень низкая скорость работы
- Высокая дисперсность
- Застревает в локальных оптимумах

Проблемы со скоростью работы

Будем пытаться ускорить сходимость

Можно применять наискорейший градиентный спуск, но он не эффективен

Идея: добиться максимального монотонного улучшения стратегий

Проблемы с дисперсностью

- Использовать смещенные, но низкодисперные оценки
- Использовать временную структуру МППС

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Опорное значение

Опорное значение (baseline) — некая функция от состояния $B(s)$, константная относительно a.

Перепишем:

$$\nabla_\theta J(\theta) = \mathrm{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)(Q^{\pi_\theta}(s,a) - B(s)]$$

Опорное значение не влияет на смещение, но позволяет уменьшить дисперсию.

Преимущество

Разницу между ценностью действия (или любой другой функцией ценности) и опорным значением будем называть **преимуществом (advantage):**

$$\hat{A}(s) = Q^{\pi_\theta}(s, a) - B(s)$$

Основная идея использования преимущества — это центрирование преимущества в районе нуля для минимизации дисперсии оценок.

Выбор B

Как выбирать B?

Выбор B

Как выбирать B?

Нужно задать параметрическое семейство функций $B_\beta(S)$ и адаптировать значения параметров в процессе обучения.

V как базовый уровень

Выбрать в качестве базового уровня функцию ценности действия $V^{\pi}(s)$:

$$\hat{A}(s) = Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$$

Оптимальное опорное значение

Теорема

Опорным значением, при котором дисперсия оценок Монте-Карло для

$$\nabla_\theta J(\theta) \; = \mathrm{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)(Q^{\pi_\theta}(s,a) - B(s)]$$

минимальна, является

$$B^*(s) = \frac{\mathrm{E}_a \|\nabla_\theta \log \pi_\theta(s,a)\|_2^2 \, Q^{\pi_\theta}(s,a)}{\mathrm{E}_a \|\nabla_\theta \log \pi_\theta(s,a)\|_2^2}$$

Теоретические предпосылки для V

В предположении, что норма градиента примерно равна для всех действий,

$$B^*(s) = \frac{\mathrm{E}_a \|\nabla_\theta \log \pi_\theta(s,a)\|_2^2 \, Q^{\pi_\theta}(s,a)}{\mathrm{E}_a \|\nabla_\theta \log \pi_\theta(s,a)\|_2^2} \approx$$

$$\approx \mathrm{E}_a Q^{\pi_\theta}(s,a) = V^{\pi_\theta}(s)$$

Побатчевая отпимизация

Обычно вычисление градиента неэффективно, для борьбы с этим возьмем батчи траекторий и будем считать суррогатную функцию потерь на батче:

$$\log \pi_\theta(s_t, a_t)\hat{A}_t$$

Policy gradient

1. Инициализируем $\theta_{(0)}$ и $\beta_{(0)}$
2. Повторяем итерации $i = 0,1, \ldots$
3. Собираем траектории $\{\tau^j\}$ согласно $\pi_{\theta_{(i)}}$
4. Для каждого шага t каждой траектории τ^j
5. $R_t^j = \sum_{t'=t}^{T-1} r_{t'}$
6. $A_t^j = R_t^i - B_{\beta_{(i)}}(s_t)$
7. $\beta_{(i+1)} = \arg\min_{\beta} \sum_j \sum_t \left(B_\beta(s_t) - R_t^j\right)^2$
8. $g = \nabla_\theta \log \pi_\theta(s_t, a_t)\hat{A}_t$
9. $\theta_{(i+1)} = \theta_{(i)} + \alpha g$
10. Возвращаем последнее значение $\theta_{()}$

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Общая идея

Оценки Монте-Карло дают большую дисперсию.

Идея: вместо использования Монте-Карло для оценки ценности действий, будем использовать некоторую параметрическую функцию.

Критик

Критик (critic) обновляет параметры функции, входящей в состав оценки A (либо саму A, либо Q).

Актор (actor) обновляет параметры стратегии с учетом критики критика.

QAC

1. Инициализируем $s, \theta_{(0)}, \psi_{(0)}$ совершаем действие a
2. Для всех шагов $i = 0,1, \ldots$
3. Получаем r, s', совершаем $a' \sim \pi_{\theta_{(i)}}$
4. $\theta_{(i+1)} = \theta_{(i)} + \alpha \nabla_\theta \log \pi_{\theta_{(i)}}(s, a) Q_\psi(s, a)$
5. $\delta = r + \gamma Q_\psi(s', a') - Q_\psi(s, a)$
6. $\psi_{(i+1)} = \psi_{(i)} + \beta \delta \nabla_\psi Q_\psi(s, a)$
7. Вернуть последнее значение $\theta_{()}$

Больше неопределенности

Алгоритмы актор-критик следуют по градиенту **приближенной** стратегии.

Аппроксимации градиента стратегии приводит к смещению оценки

Это повышает чувствительность алгоритма к изменению оценок

Совместимые функции аппроксимации

Теорема

Если

1. $\nabla_\psi Q_\psi(s,a) = \nabla_\theta \log \pi_{\theta_{(i)}}(s,a)$

2. Параметры функции ценности:

$$\psi^* = \arg\min_\psi \mathrm{E}_{\pi_\theta}\left[\left(Q^{\pi_\theta}(s,a) - Q_\psi(s,a)\right)^2\right]$$

то

$$\nabla_\theta J(\theta) = \mathrm{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(s,a)\, Q_\psi(s,a)$$

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Оценка преимущества

Можно так же параметризовать и оценивать $V^\pi(s) = V_\phi(s)$:

$$\hat{A}(s) = Q_\psi(s, a) - V_\phi(s),$$

но это избыточно.

TD-ошибка

Для $V^{\pi_\theta}(s)$ TD-ошибка равна

$$\delta^{\pi_\theta} = r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$$

Она является несмещенной оценкой функции преимущества:

$$\mathrm{E}_{\pi_\theta}[\delta^{\pi_\theta} | s, a] = Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$$

Поэтому

$$\nabla_\theta J(\theta) = \mathrm{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(s, a)\, \delta^{\pi_\theta}$$

В реальности используется

$$\delta_\chi = r + \gamma V_\phi(s') - V_\phi(s)$$

Сравнение Монте-Карло и TD

Для MC

$$R_t - V^{\pi}(s)$$

Для TD(0)

$$r + \gamma V^{\pi}(s') - V^{\pi}(s)$$

	Смещение	**Дисперсия**
MC	Нет	Высокая
TD	Есть	Низкая

Bias-Variance Trade-off

Займемся поиском **компромисса между смещением и дисперсией (bias-variance trade-off)** за счет построения промежуточных вариантов:

$$Q^{\pi}(s_t, a_t) \approx \hat{Q}_N^{\pi}(s_t, a_t) = \sum_{i=0}^{N} \gamma^i r_{t+i} + \gamma^N V^{\pi}(s_{t+N})$$

N-шаговая оценка функции преимущества:

$$\hat{A}_N(s) = \hat{Q}_N^{\pi}(s, a) - V^{\pi}(s)$$

Обобщенная оценка преимущества

Можно варьировать длины N. Построим ансамбль из разных оценок.

Обобщенная оценка преимущества (generalized advantage estimation, GAE):

$$\hat{A}_{\text{GAE}}(s) = (1 - \lambda) \sum_{i} \lambda^{N_i} \left(\hat{Q}^{\pi}_{N_i}(s, a) - V^{\pi}(s) \right)$$

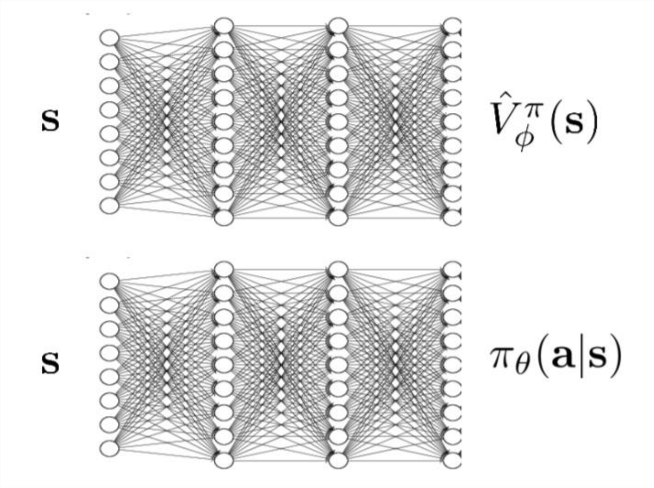
где $\{N_i\}$ — длины последовательностей, а $\lambda \in \{0; 1\}$ — гиперпараметр.

Как обучать?

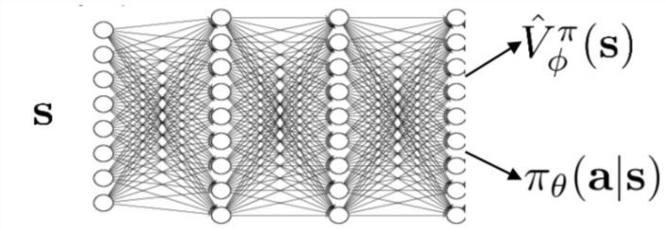
Обновление градиента стратегии по предсказанию критика

Обновление критика по наблюдаемым значениям ценности

Две отдельные сети



Совместное использование весов



A2C

Актор-критик с преимуществом (Advantage actor-critic, A2C)

1. Инициализируем $s, \theta_{(0)}, \phi_{(0)}$ совершаем действие a
2. Для всех шагов $i = 0,1, \dots$
3. Получаем r, s', совершаем $a' \sim \pi_{\theta_{(i)}}$
4. $V_{\phi_{(i+1)}} = (1 - \beta) V_{\phi_{(i)}}(s) + \beta (r + V_{\phi_{(i)}}(s'))$
5. $A(s,a) = r(s,a) + \gamma V_{\phi_{(i+1)}}(s') - V_{\phi_{(i+1)}}(s)$
6. $\theta_{(i+1)} = \theta_{(i)} + \alpha \nabla_\theta \log \pi_{\theta_{(i)}}(s,a) A(s,a)$
7. Вернуть последнее значение $\theta_{()}$

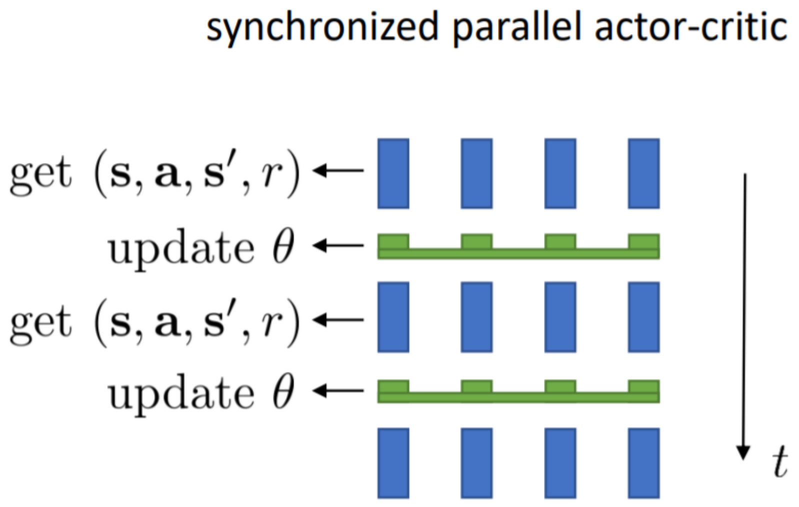
A3C

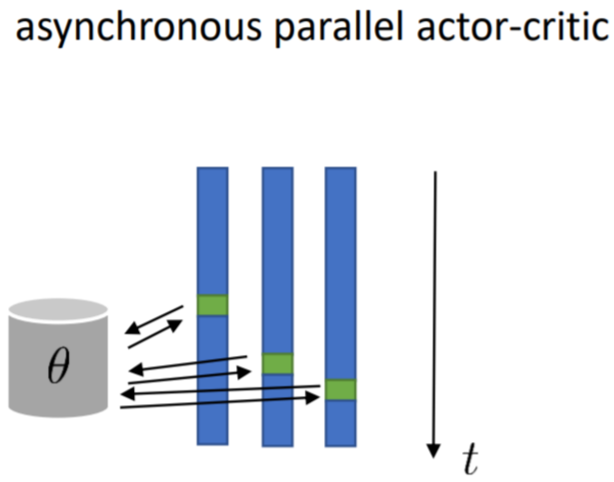
Асинхронный актор-критик с преимуществом (Asynchronous advantage actor-critic, A3C)

Основная идея: наблюдения одного агента скоррелированы, чтобы с этим бороться, построим ансамбль агентов, которые действуют относительно независимо.

Асинхронные агенты

Агенты исследуют среду независимо друг от друга, однако после каждого действия обновляют глобальные $V^{\pi}(s)$ и θ.





Анализ

Достоинства:

- за счет использования N-траекторий можно находить компромисс между смещением и дисперсией
- Совмещают сразу оба подхода
- Асинхронность работает быстрее повторов
- Работает стабильнее DQN

Недостатки:

- Работает только по собственному опыту
- Недостатки сразу обоих подходов

План лекции

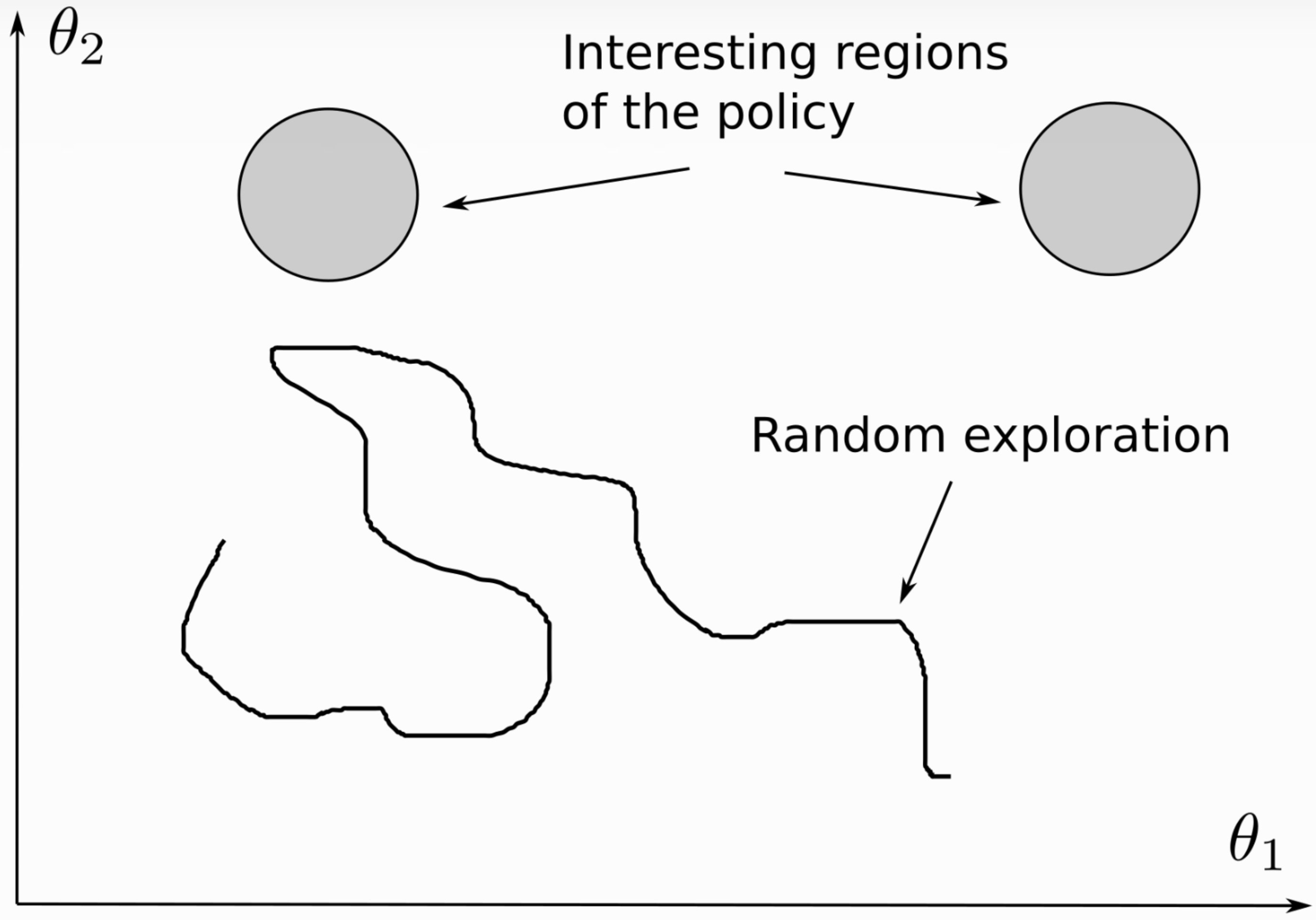
- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Проблема on-policy

Хорошие траектории попадаются для алгоритмов, использующих собственный опыт, только если нам повезло со стратегией.

Основная идея: попробуем добирать траектории для подсчета градиента похожими стратегиями

Интересные области



Стратегия для сбора траекторий

Поставим задачу оптимизировать стратегию π_θ, используя только траектории, собранные при помощи π_{old}

Теорема:

для любых двух стратегий π_1 и π_2

$$V^{\pi_1}(s) - V^{\pi_2}(s) = \mathrm{E}_{\pi_2} \sum_t \gamma^t A^{\pi_1}(s_t, a_t)$$

Градиент для новой стратегии

Благодаря теореме,

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{1-\gamma} \mathrm{E}_{s\sim d_{\pi_\theta}(s)} \mathrm{E}_{a\sim\pi_\theta(a|s)} A^{\pi_{old}}(s_t, a_t) =$$

$$= \nabla_\theta \frac{1}{1-\gamma} \mathrm{E}_{s\sim d_{\pi_\theta}(s)} \mathrm{E}_{a\sim\pi_{old}(a|s)} \frac{\pi_\theta(a|s)}{\pi_{old}(a|s)} A^{\pi_{old}}(s_t, a_t)$$

$A^{\pi_{old}}(s_t, a_t)$ оценивается критиком, обученным по траекториям, построенным π_{old}

Но состояния посещаются согласно новой стратегии

Суррогатная функция

$$L_{\pi_{old}}(\theta) = \frac{1}{1-\gamma} \mathrm{E}_{s\sim d_{\pi_{old}}(s)} \mathrm{E}_{a\sim\pi_{old}(a|s)} \frac{\pi_\theta(a|s)}{\pi_{old}(a|s)} A^{\pi_{old}}(s_t, a_t)$$

$$J(\pi) \approx J(\pi_{old}) + L_{\pi_{old}}(\theta)$$

Расстояние между стратегиями будем оценивать как $\mathrm{KL}^{\max}(\pi_{old}, \pi_\theta) = \max_a \mathrm{KL}\left(\pi_{old}(a|s) \middle\| \pi_\theta(a|s)\right)$.

Нижняя оценка

Теорема

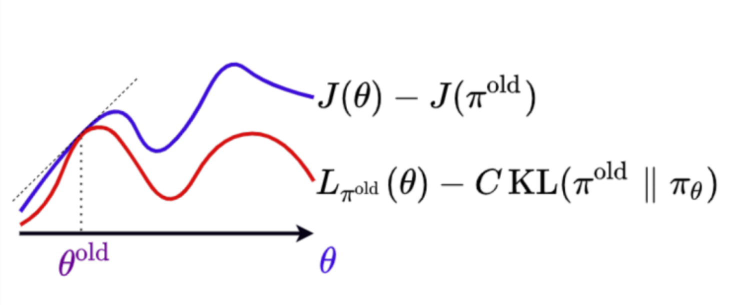
$$\left|J(\pi_\theta) - J(\pi_{old}) - L_{\pi_{old}}(\theta)\right| \leq$$
$$\leq C \cdot \mathrm{KL}^{\max}(\pi_{old}, \pi_\theta),$$

где

$$C = \frac{4\gamma}{(1-\gamma)^2} \max_{s,a} \left|A^{\pi_{old}}(s,a)\right|$$

Оптимизация нижней оценки

$$\theta^* = \arg\max_\theta \left(L_{\pi_{old}}(\theta) - C \cdot \mathrm{KL}^{\max}(\pi_{old}, \pi_\theta) \right)$$



Попеременно:

- Обновляем нижнюю оценку
- Оптимизируем по ней θ

План лекции

- Оптимизация стратегий
- Градиентный спуск по стратегиям
- REINFORCE
- Policy gradient
- Критик
- 2AC и 3AC
- Переиспользование сэмплов
- TRPO и PPO

Значение длины шага спуска

Если слишком далеко шагнуть при спуске, то:

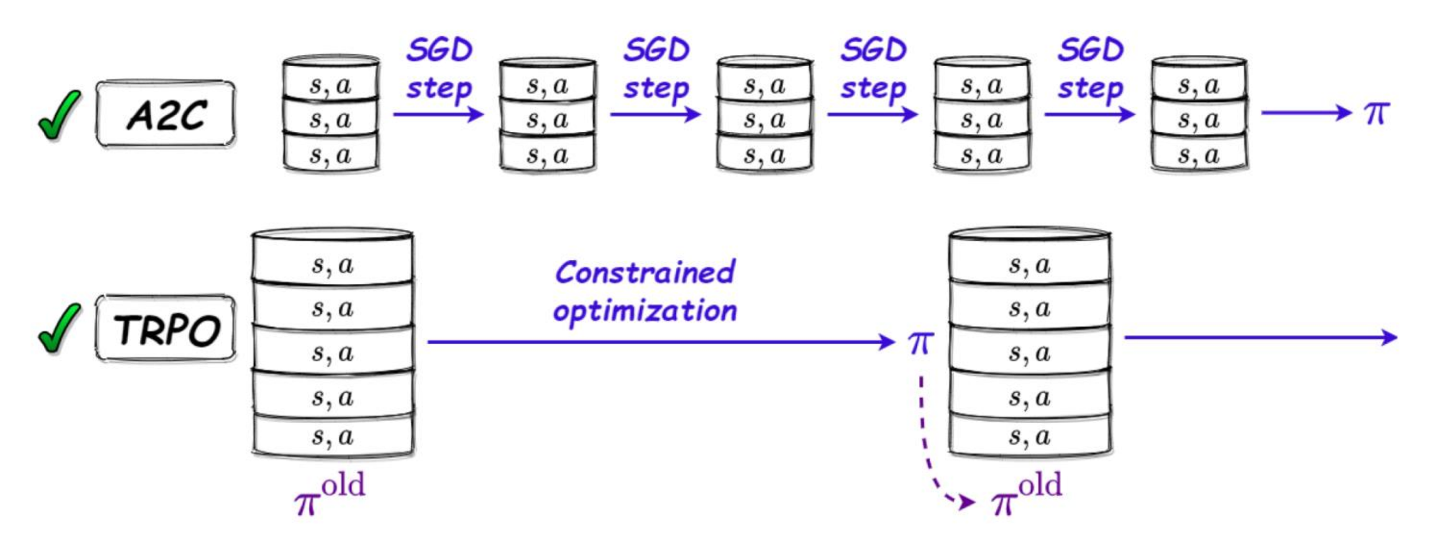
- Можно попасть в плохую стратегию
- Плохая стратегия даст плохие траектории, что может только усугубить ситуацию

Плохой выбор стратегии сложно исправлять.

TRPO

Trust region policy optimization (TRPO)

$$\begin{cases} L_{\pi_{old}}(\theta) \to \max\limits_{\theta} \\ C \cdot \mathrm{KL}(\pi_{old}, \pi_\theta) \leq \delta \end{cases}$$



Proximal Policy Loss

$$\mathrm{PPL} = \mathrm{E}_{s \sim d_{\pi_{old}}(s)} \mathrm{E}_{a \sim \pi_{old}(a|s)} \left[\frac{\pi_\theta(a|s)}{\pi_{old}(a|s)} A^{\pi_{old}}(s,a) - C\,\mathrm{KL}(\pi_{old}, \pi_\theta) \right] \to \max_\theta$$

Но для ее оптимизации требуется очень много данных.

Обрезка

$$\rho(\theta) = \frac{\pi_\theta(a|s)}{\pi_{old}(a|s)}$$

коэффициент значимости траектории.

Его значения могут быть около нуля или бесконечности.

Чтобы этого избежать, его стоит обрезать:

$$\rho^{clip}(\theta) = clip(\rho(\theta), 1 - \varepsilon, 1 + \varepsilon)$$

градиент с случае обрезки обнуляется

PPO

Proximal policy optimization

Используем градиентный спуск с PPL^{clip}:

1. Собрать траектории с π_{old}
2. Подсчитать преимущество по ним
3. Вычислить градиент для критика для новой стратегии и обновить
4. Вычислить градиент для актора для новой стратегии и обновить