8. ЛЕКЦИЯ. Policy gradient подход (Подход градиентной политики)

Теперь рассмотрим третий, Policy Gradient подход (Подход градиентной политики) к решению задачи, в котором целевой функционал будет оптимизироваться градиентными методами. Для этого мы выведем формулу градиента средней награды по параметрам стратегии и обсудим различные способы получения его стохастических оценок. В итоге сможем получить общие алгоритмы, основным ограничением которых будет жёсткий on-policy (с политикой) режим.

Policy Gradient Theorem (Теорема градиентной политики)

Вывод первым способом

Часто говорят, что функционал в задаче обучения с подкреплением не дифференцируем. Имеется в виду, что функция награды $r(s,a)$ не дифференцируема по действиям a; например, просто потому что пространство действий дискретно, например, в состоянии s агент выбрал действие $a=0$ и значение полученной награды можно лишь сравнивать со значениями для других действий. Однако, это не мешает дифференцируемости по параметрам стратегии в ситуации, когда стратегия ищется в семействе стохастичных стратегий. Фактически, оптимизация в пространстве стохастичных стратегий является этакой «релаксацией» нашей задачи.

Пусть политика $\pi_\theta(a|s)$ параметризована θ и дифференцируема по параметрам. Тогда наш оптимизируемый функционал $J(\pi_\theta)=V^{\pi_\theta}(s_0)$ тоже дифференцируем по θ, и далее мы рассмотрим формулу этого градиента. Для этого нам понадобится стандартная техника вычисления градиента мат.ожидания по распределению. Сейчас в оптимизируемом функционале у нас стоит целая цепочка вложенных мат.ожиданий, и наш вывод будет заключаться просто в последовательном применении той же техники к каждому стоящему там мат.ожиданию $\mathbb{E}_{a\sim\pi(a|s)}(\cdot)$.

Заранее оговоримся, что при минимальных технических условиях регулярности мы имеем право менять местами знаки градиента, мат.ожиданий, сумм и интегралов.

Теорема:

$$\nabla_\theta V^{\pi_\theta}(s) = \mathbb{E}_a[\nabla_\theta log\, \pi_\theta(a \mid s) Q^\pi(s,a) + \nabla_\theta Q^{\pi_\theta}(s,a)] \qquad (8.1)$$

Эта техника вычисления градиента через «стохастичный узел нашего вычислительного графа», когда мы сэмплируем a∼π(a|s), носит название REINFORCE (УКРЕПИТЬ). Как видно, эта техника универсальна: применима всегда для любых пространств действий, а также в ситуации, когда функция

$Q^{\pi}(s,a)$ не дифференцируема по действиям. Мы смогли выразить градиент V-функции через градиент Q-функции, попробуем сделать наоборот. Для этого нам нужно посчитать градиент от мат.ожидания по функции переходов, не зависящей от параметров нашей стратегии, поэтому здесь всё тривиально.

Утверждение:

$$\nabla_{\theta} V^{\pi_{\theta}}(s,a) = \gamma \mathbb{E}_{s'} \nabla_{\theta} V^{\pi_{\theta}}(s') \tag{8.2}$$

Подставляя (8.2) в (8.1), получаем:

Утверждение:

$$\nabla_{\theta} V^{\pi_{\theta}}(s) = \mathbb{E}_{a} \mathbb{E}_{s'} [\nabla_{\theta} log\, \pi_{\theta}(a \mid s) Q^{\pi}(s,a) + \nabla_{\theta} Q^{\pi_{\theta}}(s,a)] \tag{8.3}$$

Следовательно,мы получили рекурсивное выражение градиента $V^{\pi_{\theta}}(s)$ через него же само.Очень похоже на уравнение Беллмана, кстати: в правой части стоит мат.ожидание по выбранному в s действию a и следующему состоянию.

Осталось раскрутить эту рекурсивную цепочку, продолжая раскрывать $\nabla_{\theta} V^{\pi_{\theta}}(s')$ в будущее до бесконечности. Аккуратно собирая слагаемые, а также собирая из мат.ожиданий мат.ожидание по траектории, получаем следующее

Утверждение — Policy Gradient Theorem: Выражение для градиента оптимизируемого функционала можно записать следующим образом

$$\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\mathcal{T} \sim \pi} \sum_{t \geq 0} \gamma^{t}\ log\, \pi_{\theta}(a_t \mid s_t) Q^{\pi}(s_t, a_t) \tag{8.4}$$

Вывод вторым способом

Применим REINFORCE (УКРЕПИТЬ) не к мат.ожиданиям по отдельным действиям, а напрямую к распределению всей траектории следующим образом:

Теорема:

$$\nabla_{\theta} V^{\pi_{\theta}}(s) = \mathbb{E}_{\mathcal{T} \sim \pi | s_0 = s} \sum_{t \geq 0} \nabla_{\theta}\, log\, \pi_{\theta}(a_t \mid s_t) R(\mathcal{T})$$

Видим, что мы более простым способом получили очень похожую формулу, но с суммарной наградой за игры вместо Q-функции из первого доказательства (8.4). В силу корректности всех вычислений, уже можно утверждать равенство между этими формулами, что наводит на мысль, что градиент можно записывать в нескольких математически эквивалентных формах. Математически эти формы будут эквивалентны, то есть равны, как интегралы, но их Монте-Карло оценки могут начать вести себя совершенно по-разному.

Теорема: Для произвольного распределения $\pi_{\theta}(a)$ с параметрами θ, верно:

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

Следующее утверждение формализует этот тезис о том, что «будущее невлияет на прошлое» :выбор действий в некоторый момент времени никак не влияет на те слагаемые из награды, которые были получены в прошлом.

Теорема — Принцип причинности (causality): При $t > \hat{t}$:

$$\mathbb{E}_{\mathcal{T}\sim\pi}\nabla_\theta log\ \pi_\theta(a_t \mid s_t)\gamma^{\hat{t}}\, r_{\hat{t}} = 0$$

Утверждение:

$$\nabla_\theta V^{\pi_\theta}(s) = \mathbb{E}_{\mathcal{T}\sim\pi|s_0=s} \sum_{t\geq 0} \gamma^t\, \nabla_\theta log\ \pi_\theta(a_t \mid s_t) R_t \qquad (8.5)$$

Reward-to-go (награда на вынос) очень похож на Q-функцию, так как является Монте-Карло оценкой Q-функции, а мат.ожидание по распределениям всё равно берётся. Формальное обоснование эквивалентности выглядит так:

Утверждение: Формула (8.5) эквивалентна

$$\nabla_\theta V^{\pi_\theta}(s) = \mathbb{E}_{\mathcal{T}\sim\pi|s_0=s} \sum_{t\geq 0} \gamma^t\, \nabla_\theta log\ \pi_\theta(a_t \mid s_t) Q^{\pi}(s_t, a_t) \qquad (8.6)$$

Итак, мы получили формулу (8.4) вторым способом.

Физический смысл

Обсудим, а в каком направлении, собственно, указывает полученная формула градиента (8.4). Оказывается, градиент нашего функционала имеет вид градиента взвешенных логарифмов правдоподобий. Чтобы ещё лучше увидеть это, рассмотрим суррогатную функцию (surrogate objective) — другой функционал, который будет иметь в точке текущих значений параметров стратегии π такой же градиент, как и $J(\theta)$:

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

Это суррогатная 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: $\tilde{\pi} \equiv \pi_\theta$, и посмотрим на градиент при изменении θ, только одной из них. То есть что мы сказали: давайте «заморозим» оценочную Q-функцию, и «заморозим» распределение, из которого приходят пары s, a. Тогда:

Утверждение:

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

Значит, направление максимизации $J(\theta)$ в текущей точке θ просто совпадает с направлением максимизации этой суррогатной функции. Это принципиально единственное свойство введённой суррогатной функции. Таким образом, можно считать, что в текущей точке мы на самом деле «как бы» максимизируем (8.6), а это уже в чистом виде логарифм правдоподобия каких-то пар (s, a), для каждой из которых дополнительно выдан «вес» в виде значения $Q^{\pi}(s, a)$.

Проведём такую аналогию с задачей обучения с учителем: если в машинном обучении в задачах регрессии и классификации мы для данной

выборки (x, y) максимизировали правдоподобие

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

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» (credit), некую скалярную оценку хорошести, выраженную в виде $Q^{\pi}(s, a)$, и идём в направлении максимизации

$$\sum_{(s,a)} \log \pi(a \mid s, \theta) Q^{\pi}(s, a) \to \max_{\theta}.$$

REINFORCE (УКРЕПИТЬ)

Попробуем сразу построить какой-нибудь практический RL алгоритм при помощи формулы (8.4). Нам достаточно лишь несмещённой оценки на градиент, чтобы воспользоваться методами стохастической градиентной оптимизации, и поэтому мы просто попробуем всё неизвестное в формуле заменить на Монте-Карло оценки. Во-первых, для оценки мат.ожидания по траекториям просто сыграем несколько полных игр при помощи текущей стратегии π. Сразу заметим, что мы тогда требуем эпизодичности среды и сразу ограничиваем себя on-policy (с политикой) режимом: для каждого следующего шага нам требуется сыграть эпизоды при помощи именно текущей стратегии. Во-вторых, воспользуемся Монте-Карло оценкой для приближения $Q^{\pi}(s_t, a_t)$, заменив его просто на reward-to-go (награда на вынос):

$$Q^{\pi}(s, a) \approx R(\mathcal{T}), \mathcal{T} \sim \pi | s_t = s, a_t = a$$

Можно сказать, что мы воспользовались формулой градиента в форме (8.5).

Алгоритм: REINFORCE

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

Инициализировать θ произвольно
На очередном шаге t:

1. играем N игр $\mathcal{T}_1, \mathcal{T}_2 \ldots \mathcal{T}_N \sim \pi$
2. для каждого t в каждой игре $\mathcal{T}$ считаем reward-to-go: $R_t(\mathcal{T}) := \sum_{\hat{t}=t} \gamma^{\hat{t}-t} r_{\hat{t}}$
3. считаем оценку градиента:

$$\nabla_{\theta} J(\pi) := \frac{1}{N} \sum_{\mathcal{T}} \sum_{t \geq 0} \gamma^t \nabla_{\theta} \log \pi(a_t \mid s_t, \theta) R_t(\mathcal{T})$$

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

Первая беда такого алгоритма очевидна: для одного шага градиентного подъёма нам необходимо играть несколько игр целиком при помощи текущей стратегии. Такой алгоритм просто неэффективен в плане сэмплов, и это

негативная сторона on-policy режима. Чтобы как-то снизить этот эффект, хотелось бы научиться как-то делать шаги обучения, не доигрывая эпизоды до конца.

Вторая проблема алгоритма —колоссальная дисперсия нашей оценки градиента. На одном шаге направление оптимизации указывает в одну сторону, на следующем — совсем в другую. В силу корректности нашей оценки все гарантии стохастичной оптимизации лежат у нас в кармане, но на практике дождаться каких-то результатов от такого алгоритма в сколько-то сложных задачах не получится.

Чтобы разобраться с этими двумя проблемами, нам понадобится чуть подробнее познакомиться с конструкцией (8.4).

State visitation frequency (Состояние посещения частоты)

Мы получили, что градиент оптимизируемого функционала по параметрам стратегии (8.4) имеет вид мат.ожиданий по траекториям стратегии π. Казалось бы, для его оценки нам придётся играть полные эпизоды. Однако видно, что внутри интеграла по траекториям и суммы по времени стоит нечто, зависящее только от пар состояние-действие:

$$f(s,a) := log\pi_\theta(a|s)Q^\pi(s,a)$$

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

Допустим, начальное состояние s_0 фиксировано. Обозначим вероятность оказаться в состоянии s в момент времени t при использовании стратегии π как $p(s_t = s|\pi)$. Мы могли бы попробовать посчитать, сколько раз мы в среднем оказываемся в некотором состоянии s, просто просуммировав по времени:

$$\sum_{t\geq 0} p(s_t = s|\pi)$$

однако, как легко видеть, такой ряд может оказаться равен бесконечности (например, если в MDP всего одно состояние). Поскольку мы хотели получить распределение, мы можем попробовать отнормировать этот «счётчик»:

Определение: Для данного MDP и политики π state visitation frequency (состояние посещения частоты) называется

$$\mu_\pi(s) := \lim_{T\to\infty} \sum_{t\geq 0}^{T} p(s_t = s|\pi) \qquad (8.7)$$

Вообще говоря, мы мало что можем сказать про это распределение. При некоторых технических условиях у марковской цепи встречаемых состояний может существовать некоторое стационарное распределение $\lim_{t\to\infty} p(s_t = s|\pi)$, из которого будут приходить встречающиеся состояния, условно, через

бесконечное количество шагов взаимодействия; если так, то (8.7) совпадает с ним, поскольку, интуитивно, начиная с некоторого достаточно большого момента времени t все слагаемые в ряде будут очень похожи на стационарное распределение.

Можно примерно считать, что во время обучения при взаимодействии со средой состояния приходят из $\mu_\pi(s)$, где π — стратегия взаимодействия. Конечно, это верно лишь в предположении, что марковская цепь уже «разогрелась» и распределение действительно похоже на стационарное; то есть, в предположении, что обучение продолжается достаточно долго (обычно это так), и предыдущие стратегии, использовавшиеся для взаимодействия, менялись достаточно плавно.

Естественно, надо помнить, что сэмплы из марковской цепи скоррелированы, и соседние состояния будут очень похожи — то есть независимости в цепочке встречаемых состояний, конечно, нет. При необходимости набрать мини-батч независимых сэмплов из $\mu_\pi(s)$ для декорреляции необходимо воспользоваться параллельными средами: запустить много сред параллельно и для очередного мини-батча собирать состояния из разных симуляций взаимодействия. Вообще, если в батч попадает целая цепочка состояний из одной и той же среды, то это нарушает условие независимости сэмплов и мешает обучению.

Расцепление внешней и внутренней стохастики

Итак, давайте попробуем формально понять, из какого распределения приходят состояния в формуле градиента (8.4), и отличается ли оно от $\mu_\pi(s)$. Для этого мы сейчас придумаем, как можно записывать функционалы вида

$$\mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t\geq 0}\gamma^t f(s_t, a_t),$$

где f — какая-то функция от пар состояние-действие, немного по-другому.

В MDP есть два вида стохастики:

• внешняя (extrinsic), связанная со случайностью в самой среде и неподконтрольная агенту; она заложена в функции переходов $p(s'|s,a)$.

• внутренняя (intrinsic), связанная со случайностью в стратегии самого агента; она заложена в π(a|s). Это стохастика нам подконтрольна при обучении.

Мат. ожидание по траектории 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.

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

Определение: Для данного MDP и политики π discounted state visitation distribution (распределение посещений со скидкой) называется

$$d_\pi(s) := (1-\gamma)\sum_{t\geq 0}\gamma^t \mathrm{p}(\mathrm{s_t} = \mathrm{s}\,|\pi) \tag{8.8}$$

Утверждение: State visitation distribution (состояние распределение посещений) есть распределение на множестве состояний, то есть:

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

State visitation distribution (состояние распределение посещений) (8.8) является важным понятием ввиду следующей теоремы, благодаря которой мы можем чисто теоретически расцепить (decouple) внешнюю и внутреннюю стохастику:

Теорема: Для произвольной функции $f(s,a)$:

$$\mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t\geq 0}\gamma^t f(s_t,a_t) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\pi(s)}\mathbb{E}_{a\sim\pi(a|s)}f(s,a) \tag{8.9}$$

Итак, мы научились переписывать мат.ожидание по траекториям в другом виде. В будущем мы будем постоянно пользоваться формулой (8.9) для разных f(s,a), поэтому полезно запомнить эту альтернативную форму записи. Например, мы можем получить применить этот результат к нашему функционалу:

Утверждение: Оптимизируемый функционал можно записать в таком виде:

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

Интерпретация у полученного результата может быть такая: нам не столько существенна последовательность принимаемых решений, сколько частоты посещений «хороших» состояний с высокой наградой.

Рассмотренная теорема позволяет переписать и формулу градиента:

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

Итак, множитель γ^t в формуле (8.4) имеет смысл «дисконтирования частот посещения состояний». Для нас это представляет собой, мягко говоря, очень странную проблему. Если мы рассмотрим формулу градиента $J(\theta)$, то есть зафиксируем начальное состояние в наших траекториях, то все слагаемые, соответствующие состояниям, встречающиеся в эпизодах только после, условно, 100-го шага, будут домноженына γ^{100}.То есть одни слагаемые в нашем

оптимизируемом функционале имеют один масштаб, а другие — домножаются на близкий к нулю γ^{100}, совершенно иной. Градиентная оптимизация с такими функционалами просто не справится: в градиентах будет доминировать информация об оптимизации наших решений в состояниях около начального, имеющих большой вес.

На практике во всех Policy Gradient методах от дисконтирования частот посещения отказываются. Это означает, что мы заменяем $d_\pi(s)$ на $\mu_\pi(s)$:

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

В формуле (8.4) это эквивалентно удалению множителя γ^t:

$$\nabla_\theta J(\pi) \approx \mathbb{E}_{\mathcal{T}\sim\pi}\sum_{t\geq 0}\nabla_\theta\, log\pi_\theta(a_t|s_t)Q^\pi(s_t,a_t)$$

Мы вовсе не отказались от дисконтирования: оно всё ещё сидит внутри оценочной функции и даёт приоритет ближайшей награде перед получением той же награды в будущем, как мы и задумывали изначально. Итак, при таком соглашении мы можем получить как бы оценку Монте-Карло на градиент:

$$\nabla_\theta J(\pi) \approx \frac{1}{1-\gamma}\mathbb{E}_{\mathrm{a}\sim\pi(\mathrm{a|s})}\nabla_\theta log\pi_\theta(a_t|s_t)Q^\pi(s_t,a_t) \quad \mathrm{s}\sim\mu_\pi(\mathrm{s})$$

Такую оценку можно считать условно несмещённой (с учётом нашего забивания на множитель γ^t и приближённого сэмплирования из $\mu_\pi(\mathrm{s})$),если у нас на руках есть точная (или хотя бы несмещённое) оценка «кредита» $Q^\pi(s,a)$.

Связь с policy improvement (улучшение политики)

Формула градиента даёт ещё одну интересную интерпретацию. Давайте введём ещё одну суррогатную функцию (surrogate objective). Как и суррогатная функция (5.8), это будет ещё один функционал, который имеет в точке текущих значений параметров стратегии π такой же градиент, как и $J(\theta)$:

$$\mathcal{L}_{\tilde{\pi}}(\theta) := \frac{1}{1-\gamma}\mathbb{E}_{d_{\tilde{\pi}}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)}\mathrm{Q}^{\tilde{\pi}}(\mathrm{s},\mathrm{a})$$

Эта суррогатная функция, опять же, от двух стратегий: стратегии π_θ, которую мы оптимизируем, и ещё одной стратегии $\tilde{\pi}$. Снова смотрим на эту суррогатную функцию в такой точке θ, что две стратегии совпадают: $\tilde{\pi}\equiv\pi_\theta$; будем «шевелить» θ и смотреть на градиент. То есть теперь мы «заморозили» распределение частот посещений состояний и, как и в прошлый раз, оценочную функцию. Тогда:

Утверждение:

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

Это значит, что в текущей точке градиенты указывают туда же, куда и при

максимизации $\mathcal{L}_{\widetilde{\pi}}(\theta)$ по θ. А куда указывает направление градиентов для неё? Выражение $\mathbb{E}_{a\sim\pi_\theta(a|s)}\mathrm{Q}^{\widetilde{\pi}}(\mathrm{s},\mathrm{a})$ говорит максимизировать Q-функцию по выбираемым нами действиям.

Но если в табличном сеттинге policy improvement (улучшение политики) мы делали, так сказать, «жёстко», заменяя целиком стратегию на жадную по действиям, то формула градиента теперь говорит нам, что это лишь направление максимального увеличения $J(\theta)$; после любого шага в этом направлении наша Q-функция тут же, формально, меняется, и в новой точке направление уже должно быть скорректировано.

Второе уточнение, которое дарит нам эта формула, это распределение, из которого должны приходить состояния. В табличном случае мы не знали, в каких состояниях проводить improvement (улучшение) «важнее», теперь же мы видим, что s должно приходить из $d_\pi(s)$. Если какое-то состояние посещается текущей стратегией часто, то улучшать стратегию в нём важнее, чем в состояниях, которые мы посещаем редко.

Но ещё это наблюдение, что градиент будущей награды и policy improvement (улучшение политики) связаны, в частности, даёт одно из оправданий тому, что мы отказались от дисконтирования частот посещения состояний. Мы используем формулу градиента для максимизации $V^\pi(s_0)$. Но, в общем-то, мы хотим максимизировать $V^\pi(s)$ сразу для всех s, и мы можем делать это с некоторыми весами. Тогда, выходит, частоты появления s в оптимизируемом функционале определяются в том числе этими весами. Другими словами, $d_\pi(s)$ указывает на «самое правильное» распределение, из которого должны приходить состояния, которые дадут направление именно максимального увеличения функционала, но теория policy improvement-а (улучшение политики) подсказывает нам, что теоретически корректно выбрать и любое другое. Даже если мы совсем другое какое-то распределение выберем для сэмплирования состояний s, то функционал

$$\mathbb{E}_s\mathbb{E}_{a\sim\pi_\theta(a|s)}Q^{\tilde{\pi}}(s,a)\to\max_\theta$$

будет давать какое-то направление подъёма, какое-то годное направление оптимизации.

Если у нас есть модель Q-функции, мы даже с оговорками можем обучать стратегию с буфера, если будем брать из него только состояния, а мат.ожидание по a (или его Монте-Карло оценку) считать, используя текущую стратегию π. Однако, коли уж мы хотим off-policy (вне политики) алгоритм, то и эту Q-функцию тогда нужно учить с буфера. В итоге, мы получим алгоритм, очень

похожий на DQN, со схожими недостатками и преимуществами.

Бэйзлайн

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.

Для этого вспомним утверждение: 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 $\mathcal{A}$ нужно данный вес θ_i параметризации увеличивать, а в другой области — уменьшать. В среднем «магнитуда изменения» равна нулю.

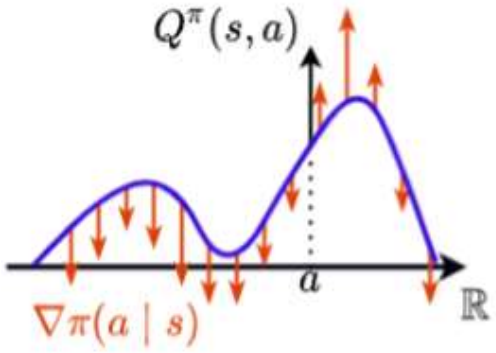

Рис. 8.1.

Но у нас в Монте-Карло оценке только один сэмпл $a \sim \pi(a|s)$, и для него направление изменения домножится на кредит, на нашу оценку Qπ(s,a). Если эта оценка в одной области 100, а в другой 1000 — дисперсия получаемых значений $\nabla_\theta log\pi_\theta(a \mid s)Q^\pi(s,a)$ становится колоссальной. Было бы сильно лучше, если бы «кредит» — вес примеров — был в среднем центрирован, и тоже колебался возле нуля. Тогда для «плохих действий» мы правдоподобие этих действий уменьшаем, а для «хороших действий» — увеличиваем, что даже чисто интуитивно логичнее. И для центрирования весов правдоподобия в Policy Gradient методах всегда вводится бэйзлайн (baseline), без которого алгоритмы обычно не заведутся.

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

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

Это верно для произвольной функции от состояний и становится неверно, если вдруг бэйзлайн начинает зависеть от a. Мы вольны выбрать бэйзлайн

произвольно; он не меняет среднего значения оценок градиента, но изменяет дисперсию.

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

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

Практическая ценность результата невысока. Знать норму градиента для всех действий a вычислительно будет труднозатратно даже в дискретных пространствах действий. Поэтому мы воспользуемся небольшим предположением: мы предположим, что норма градиента примерно равна для всех действий. Тогда:

$$b^*(s) = \frac{\mathbb{E}_a \|\nabla_\theta \log \pi_\theta(a \mid s)\|_2^2 Q^\pi(s,a)}{\mathbb{E}_a \|\nabla_\theta \log \pi_\theta(a \mid s)\|_2^2} = \{\|\nabla_\theta \log \pi_\theta(a \mid s)\|_2^2 \approx \text{const}(a)\} \approx \mathbb{E}_a Q^\pi(s,a) = V^\pi(s)$$

Эта аппроксимация довольно интуитивная: по логике, для дисперсии хорошо, если значения функции $Q^\pi(s,a) - \mathrm{b(s)}$ вертятся вокруг нуля, то есть в среднем дают ноль, и поэтому хорошим (но неоптимальным) бэйзлайном будет

$$\mathbb{E}_{a\sim\pi(a|s)}\left(Q^\pi(s,a) - b(s)\right) := 0 \quad \Rightarrow \quad b(s) := \mathbb{E}_{a\sim\pi(a|s)} Q^\pi(s,a) = V^\pi(s)$$

Итак, всюду далее будем в качестве бэйзлайна использовать $b(s) := V^\pi(s)$. Подставляя и вспоминая определение Advantage (Преимущество)-функции, получаем:

Утверждение

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

Таким образом, «кредит», который мы выдаём каждой паре s,a, будет являться оценкой Advantage (Преимущество), и состоять из двух слагаемых: оценки Q-функции и бэйзлайна. Именно поэтому этот вес и называется «кредитом»: задача оценки Advantage и есть в точности тот самый credit assingment (предоставление кредита) который мы обсуждали.