Введение в обучение с подкреплением

Тема 2: Многорукие бандиты

Лектор: Кривошеин А.В.

Общие понятия RL

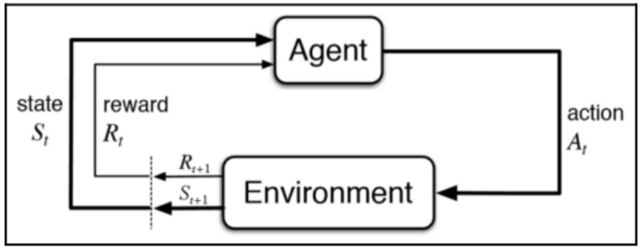
Суть RL:

научить обучаемого агента предпринимать "хорошие" последовательности действий в ходе взаимодействия со средой для эффективного достижения цели.

Цель агента: *поиск стратегии поведения, дающей максимальный доход.*

Схема взаимодействия:

обучаемый агент действует в окружающей его среде, среда реагирует на действия агента.



Обозначения:

$\mathcal{S}$ — множество состояний, S_t состояние в момент времени t,

$\mathcal{A}$ — множество действий, A_t действие в момент времени t,

$\mathcal{R}$ — множество вознаграждений, R_t вознаграждение в момент времени t.

Взаимодействие агента со средой — это **дискретный случайный процесс**: $S_0, A_0, R_1, S_1, A_1, \ldots$

Конкретная реализация этого процесса — это **траектория** взаимодействия агента со средой.

Многорукий бандит: базовая идея

Рассмотрим классическую задачу RL о **многоруких бандитах** (англ. Multi-Armed Bandits).

Многорукий бандит — это простая, но широко применяемая модель для принятия последовательности решений в условиях неопределённости.

Для начала рассмотрим пример. Пусть перед нами два игровых автомата.

Дёргая за рычаги этих автоматов можно с некоторой вероятностью получать либо 1 руб., либо не получать ничего.

Как действовать, чтобы в длительной перспективе получить наибольший выигрыш?

Многорукий бандит: общая постановка

Пусть перед агентом находится некоторое устройство с k рычагами. В каждый дискретный момент времени можно дёрнуть один из рычагов и получить вознаграждение.

С точки зрения общих понятий RL:

Агент всегда находится в одном состоянии среды.

Если выбор одного из рычагов условно обозначить числом от 1 до k, то $\mathcal{A} = \{1, \ldots, k\}$.

Вознаграждение за выбор того или иного рычага —

это случайная величина со стационарным (не меняющимся со временем) распределением вероятности.

Для каждого действия эта случайная величина своя.

Стратегия — это принцип выбора того или иного рычага в каждый дискретный момент времени.

Задача агента: поиск стратегии, максимизирующей доход. То есть надо найти самый выгодный рычаг.

Истинная ценность действия a, $a \in \{1, \ldots, k\}$, — это математическое ожидание случайной величины, соответствующей вознаграждению за это действие:

$$q_*(a) := \mathbb{E}[R_t \mid A_t = a].$$

Ясно, что стратегия, которая выбирает действие $a^* = \underset{a}{\operatorname{argmax}}\, q_*(a)$ решает задачу агента. Но эти ценности агенту неизвестны.

Многорукий бандит: приложения

Постановка задачи проста. Однако, методы её решения используются в работе рекомендательных систем и в системах поисковой выдачи.

Например, рассмотрим сайт интернет-магазина — это автомат.

Карточки с товарами — это рычаги.

Выбор рычага — это помещение товара в блоке рекомендаций для пользователя.

Вознаграждения зависят от действий пользователя, купил он товар или нет.

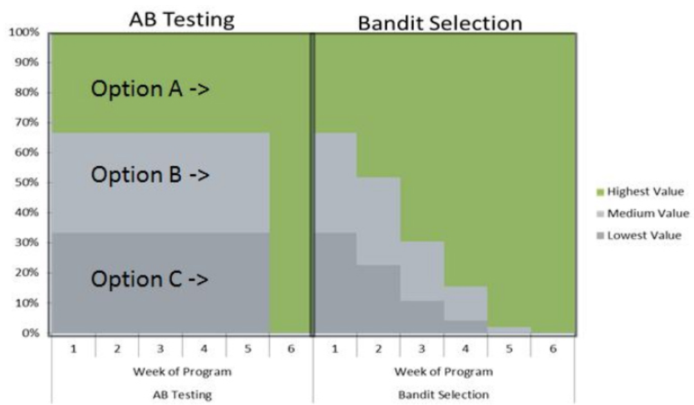
Вопрос: какие карточки лучше всего выдавать в блоке рекомендаций по запросу пользователя?

Классическим методом такого типа задач был метод АВ-тестирования. Суть метода в формировании и проверке гипотез об успешности тех или иных карточек с товарами. Гипотезы тестируются на наборе данных и выбираются наиболее успешные карточки товаров.

Схема не является гибкой, ведёт к “заморозке” выдачи:

успешные товары будут появляться всегда, менее успешные — никогда.

Out[•]//DisplayForm=



Многорукие бандиты начинают работать с нуля,
без тестирования, нет проблемы "заморозки" выдачи.
Более того, предпочтения пользователей могут
меняться со временем и многорукие бандиты
могут это учитывать в своей работе.

Многорукий бандит: базовые стратегии

Итак, истинные ценности действий $q_*(a) = \mathbb{E}[R_t \mid A_t = a]$ агенту неизвестны.
Но агент может получить **оценки** истинных ценностей, например, усредняя фактически получаемые вознаграждения за выбор рычага a. Обозначим оценку действия a к моменту времени t за $Q_t(a)$. Тогда

$Q_t(a)$ = ((сумма вознаграждений за выбор рычага a к моменту времени t) / (число выборов рычага a к моменту времени t)).

Закон больших чисел гарантирует, что оценка $Q_t(a)$ сойдётся к истинной ценности $q_*(a)$, если число выборов действия a стремится к бесконечности.

Пусть к моменту времени t есть некоторые оценки ценностей действий $Q_t(a),\ a \in \{1, \ldots, k\}$.

Сформулируем две базовые стратегии поведения:
1. **Жадный выбор** — это выбор действия с максимальной текущей оценкой $Q_t(a)$, то есть $A_t = \arg\max_a Q_t(a)$. Про такой выбор ещё говорят, что это **шаг использования** имеющегося знания (англ. exploiting step):
2. **Нежадный выбор** — это выбор случайного действия некоторым образом. Про такой выбор ещё говорят, что это **шаг исследования** (англ. exploring step).

Жадный выбор максимизирует вознаграждение на одном шаге. Однако, может оказаться, что лучшее в перспективе действие не будет выбираться.

Нежадный выбор может дать большую награду в перспективе (поскольку может улучшить оценки действий), хотя может дать меньшее вознаграждение в данный момент времени по сравнению с жадным выбором.

Оптимальный баланс между этими стратегиями зависит от точности оценок и числа шагов, оставшихся до конца.

Многорукий бандит: ε-жадная стратегия

Нежадный выбор можно осуществить по-разному.

Простейший способ заключается в выборе каждого действия с равной вероятностью. Ясно, что такой подход не способствует увеличению получаемых вознаграждений с течением времени.

Стратегия, которая совмещает жадный и равновероятный выбор — это **ε-жадная стратегия:**

— с вероятностью ε выбирается любое действие (например, равновероятно),

— с вероятностью $1 - \varepsilon$ выбирается жадное действие.

При такой стратегии на бесконечном промежутке времени каждое действие будет выбрано бесконечное число раз. Тогда все оценки действий $Q_t(a)$ сойдутся к истинным значениям $q_*(a)$. При жадной стратегии такой гарантии нет.

Нахождение баланса между использованием знаний о среде и исследованием среды при взаимодействии агента со средой является одной из основных задач в обучении с подкреплением (англ. **exploration vs exploitation trade-off**).

Многорукий бандит: обновление оценок

Зафиксируем действие a. Пусть r_n это вознаграждение, полученное после n-го выбора этого действия. Оценка ценности действия a после выбора этого действия n раз имеет вид:

$$Q_{n+1} = \frac{r_1 + r_2 + \ldots + r_{n-1} + r_n}{n}, \quad n \geq 1.$$

Q_1 — это начальная оценка (либо ноль, либо случайно, либо из каких-то иных соображений).

Чтобы найти Q_n, надо хранить все прошлые вознаграждения.
Поэтому имеет смысл обновлять оценку с помощью **инкрементной реализации:**

$$Q_{n+1} = \frac{n-1}{n} Q_n + \frac{1}{n} r_n = Q_n + \frac{1}{n}(r_n - Q_n).$$

Сформулируем полученное правило обновления более абстрактно:

НоваяОценка = СтараяОценка + Шаг (ЦельОбновления − СтараяОценка).

Это правило обновления будет далее встречаться довольно часто. Разницу (ЦельОбновления − СтараяОценка) называют **ошибкой оценки**. Если обновить оценку в направлении цели обновления, то новая ошибка оценки станет меньше.

В нашем случае цель обновления равна текущему вознаграждению r_n, которое можно трактовать как искажённое шумом истинное вознаграждение за действие: $r_n = q_*(a) + \text{noise}$.

Заметим, что в инкрементной реализации вычисления оценки **величина шага обновления меняется с каждой новой оценкой**.

Многорукий бандит: псевдокод

Приведём псевдокод алгоритма для работы агента, действующего по ε-жадной стратегии и формирующего оценки по методу выборочного среднего.

Инициализировать для i от 1 до k:

  $Q(i) := 0$ (оценка i-го действия)

  $N(i) := 0$ (число выборов i-го действия)

Повторять:

  $a := \begin{cases} \arg\max_i Q(i) & \text{с вероятностью } 1-\varepsilon \\ \text{случайный выбор из } \{1, \ldots, k\} & \text{с вероятностью } \varepsilon \end{cases}$

  $r := \text{bandit}(a)$

  $N(a) := N(a) + 1$

  $Q(a) := Q(a) + \frac{1}{N(a)}(r - Q(a))$

Неоднозначности выбора максимума разрешаются случайным образом.

Функция $\text{bandit}(a)$ в псевдокоде означает, что надо выполнить действие a и вернуть вознаграждение, соответствующее сделанному действию.

Многорукий бандит: величина шага

Рассмотрим модификацию формулы обновления оценок ценности действия в виде:

$$Q_{n+1} = Q_n + \alpha_n(r_n - Q_n) = (1 - \alpha_n)\, Q_n + \alpha_n\, r_n.$$

Когда параметр α_n близок к 1, то больший вклад в формирование новой оценки дают только что полученные вознаграждения.

Когда параметр α_n близок к 0, то больший вклад в формирование новой оценки даёт имеющаяся оценка.

Условия, при которых гарантируется сходимость оценок Q_n к истинным ценностям действий, называют **условиями Роббинса-Монро**:

$$\sum_n \alpha_n = \infty, \quad \sum_n \alpha_n^2 < \infty, \quad \text{например, можно выбрать } \alpha_n = \frac{1}{n}.$$

В стационарном случае, когда истинные ценности действий $q_*(a)$ не изменяются со временем, последовательность $\{\alpha_n\}$ имеет смысл выбирать именно так.

В нестационарном случае истинные ценности действий $q_*(a)$ могут меняться со временем. Это значит, что при обновлении оценок имеет смысл давать больший вес последним полученным вознаграждениям. В этом случае можно выбрать постоянный шаг обучения $\alpha_n = \alpha$, где α некоторая константа от 0 до 1. Причём,

$$Q_{n+1} = (1 - \alpha)\, Q_n + \alpha\, r_n = (1 - \alpha)[(1 - \alpha)\, Q_{n-1} + \alpha\, r_{n-1}] + \alpha\, r_n = \ldots = Q_{n+1} = (1 - \alpha)^n\, Q_1 + \sum_{i=1}^{n} \alpha (1 - \alpha)^{n-i}\, r_i,$$

то есть оценка Q_n является взвешенным средним прошлых вознаграждений $r_1, \ldots, r_n$ и начальной оценки Q_1.

Многорукий бандит: оптимистичные старты

Отметим, что оценки ценностей действий зависят от начальных оценок $Q_1(a)$.

Если оценки формируются по методу выборочного среднего, то эта зависимость сразу пропадает, так как $\alpha_1 = 1$ и

$Q_2 = (1-\alpha_1)\,Q_1 + \alpha_1\,r_1 = r_1$.

Если оценки формируются с постоянным шагом обучения, то эта зависимость убывает к нулю, так как $\alpha \in (0,\ 1)$ и

$$Q_{n+1} = (1-\alpha)^n\,Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i}\,r_i.$$

Работа некоторых стратегий на основе обновления оценок ценностей действий также может зависеть от начальных оценок $Q_1(a)$. На практике это можно использовать, чтобы включить априорную информацию о вознаграждениях. Эта модификация называется **оптимистичным стартом**.

Например, выставив начальные оценки выше уровня истинных оценок, мы поощрим агента к исследованию в начале процесса, даже если он действует по жадной стратегии. Например, если $q_*(a) < 3$ для всех a, то в качестве начальных оценок можно положить $Q_1(a) = 5$. Каждое действие будет опробовано несколько раз, пока уровень оценок не приблизится к истинным значениям.

Для нестационарной задачи это дополнительное поощрение к исследованию будет носить временный характер.
Старт происходит только один раз, а истинные ценности всё время меняются.

Многорукий бандит: ВДГ-стратегия

Стратегия, основанная на ВДГ-действии (ВДГ сокращение от “Верхняя Доверительная Граница”, англ. Upper Confident Bound, UCB), позволяет выбирать нежадные действия по их потенциалу оказаться оптимальными.

Идея: выбрав один рычаг a несколько раз, получается выборка из вознаграждений, у которой можно найти доверительный интервал, в котором с некоторой степенью уверенности лежит истинное значение ценности $q_*(a)$.

Суть ВДГ-стратегии: выбор максимальной оценки ценности с учётом недостоверности оценок.

Опуская детали вывода, для выбора действия надо пользоваться формулой

$$A_t = \arg\max_a \left(Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right),$$

где c число, отражающее степень доверия (подбирается эмпирически, $c \approx 1$),
$N_t(a)$ число выборов действия a к моменту времени t.

Второе слагаемое называют **мерой недостоверности**.

При каждом выборе действия a недостоверность оценки $Q_t(a)$ снижается,
однако, с течением времени недостоверность всех действий увеличивается.

Таким образом, потенциально будут выбираться все действия, но те у которых малы оценки будут выбираться с уменьшающейся частотой.
Стратегия на основе ВДГ-действий хорошо показывает себя на многоруком бандите, но её сложно обобщить на иные задачи.

Многорукий бандит: выборка Томпсона

Ещё один способ решения задачи основан на байесовском подходе. Он подходит для более частной постановки задачи: вознаграждений только два типа: 0 или 1 (“неудача” или “успех”), у каждого рычага своя вероятность получения “успеха” p_i. Надо найти самый “успешный” рычаг.

Представим, что мы выбрали каждый из рычагов некоторое количество раз. Для i-го рычага есть данные:

a_i число “успехов”, b_i число “неудач”,

$a_i + b_i$ число выборов рычага i.

По сути для каждого рычага проведена серия испытаний Бернулли.

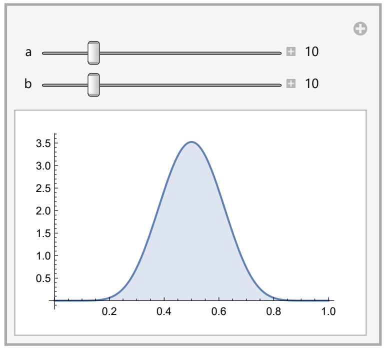
По принципу максимального правдоподобия, оценка истинной вероятности “успеха” p_i равна $\frac{a_i}{a_i+b_i}$.

Чтобы смоделировать степень уверенности в оценке, используем оценку апостериорного максимума, которая имеет вид **бета–распределения** $\mathrm{Beta}(a_i,b_i)$ с параметрами a_i, b_i.

Функция плотности имеет вид:

Out[3]=

$$f_{\mathrm{Beta}}(x,a_i,b_i) = \frac{1}{B(a_i,\,b_i)}\,x^{a_i}(1-x)^{b_i}, \quad \text{где} \quad B(a_i,b_i) = \frac{\Gamma(a_i)\,\Gamma(b_i)}{\Gamma(a_i+b_i)}$$



На каждом временном шаге будем оценивать истинные вероятности p_i с помощью чисел θ_i, полученных реализацией случайной величины с бета-распределением $\mathrm{Beta}(a_i,\,b_i)$. Этот факт будем обозначать в виде $\theta_i \sim \mathrm{Beta}(a_i,\,b_i)$.

Многорукий бандит: выборка Томпсона

Инициализация: $a_i = b_i = 1,\ i = 1,\ \dots,\ k$.

Повторять:

$\theta_i \sim \mathrm{Beta}(a_i,\ b_i),\ \ i = 1,\ \dots,\ k$.

$\mathrm{j} = \underset{i=1,\dots,k}{\mathrm{argmax}}\ \theta_i$.

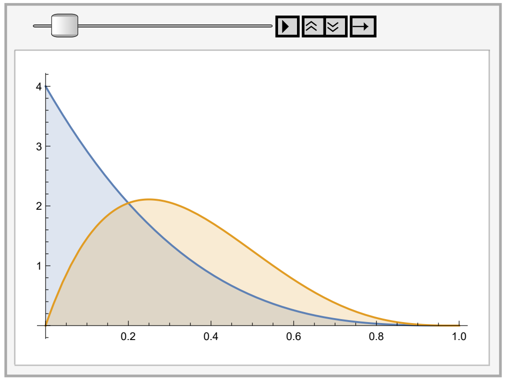
$\mathrm{r} = \mathrm{bandit(j)}$

$a_j := a_j + r$

$b_j := b_j + (1 - r)$

Неоднозначности выбора максимума разрешаются случайным образом.

Функция bandit(j) в псевдокоде означает, что надо выполнить действие j и вернуть вознаграждение, соответствующее сделанному действию. Ниже проиллюстрировано изменение плотностей вероятности бета-распределения $\mathrm{Beta}(a_i,\ b_i)$ при работе алгоритма в случае 2-х рычагов, у которых истинные вероятности “успеха” равны 0.45 и 0.3.



Сравнение различных алгоритмов проведено в iPython ноутбуке “2.0 RL_MAB.ipynb”