Обучение с подкреплением (Reinforcement Learning)

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ • 28 апреля 2023

Содержание

1. Задача о многоруком бандите
   - Простая постановка задачи
   - Жадные и полужадные стратегии
   - Адаптивные стратегии
2. Среда с состояниями
   - Постановка задачи
   - Метод Q-обучения
   - Параметризация стратегий и функций ценности
3. Моделирование среды
   - Контекстный бандит и томпсоновское сэмплирование
   - Линейная регрессионная модель премии
   - Оценивание модели по историческим данным

Задача о многоруком бандите (multi-armed bandit)

Имеется множество допустимых *действий* (ручек, arm),
с различными распределениями размера *премии* (reward, payoff).

Как быстрее найти самое выгодное действие?
Какие возможны стратегии?

Задача о многоруком бандите (multi-armed bandit)

A — множество возможных *действий*
$p(r|a)$ — неизвестное распределение *премии* $r \in \mathbb{R}$ для $a \in A$
$\pi_t(a)$ — *стратегия* (policy) агента в раунде t, распределение на A

Игра агента со средой:

инициализация стратегии $\pi_1(a)$;
для всех раундов $t = 1, \ldots, T, \ldots$
  агент выбирает действие $a_t \sim \pi_t(a)$;
  среда генерирует премию $r_t \sim p(r|a_t)$;
  агент корректирует стратегию $\pi_{t+1}(a)$;

$$Q_t(a) = \frac{\sum_{i=1}^{t} r_i[a_i = a]}{\sum_{i=1}^{t}[a_i = a]} \quad \text{— средняя премия в } t \text{ раундах}$$

$$Q^*(a) = \lim_{t\to\infty} Q_t(a) \to \max_{a\in A} \quad \text{— \textit{ценность действия} } a$$

Примеры прикладных задач

- Управление роботами, технологическими процессами
- Генерация движений персонажей в мультипликации
- Рекомендация новостных статей пользователям
- Показ рекламы в Интернете
- Управление портфелем ценных бумаг, игра на бирже
- Управление ценами и ассортиментом в сетях продаж
- Маршрутизация в телекоммуникационных сетях
- Стратегические игры: шахматы, го, Dota2, StarCraft2, ...

Обобщения постановки задачи:

- Есть информация о состоянии среды или о контексте
- Есть параметрическая модель стратегии/ценности/среды

H. Robbins. Some aspects of the sequential design of experiments. 1952.

Жадная стратегия

Множество действий с максимальной текущей оценкой ценности:

$$A_t = \operatorname{Arg}\max_{a \in A} Q_t(a)$$

Жадная стратегия — выбирать любое действие из A_t:

$$\pi_{t+1}(a) = \frac{1}{|A_t|}[a \in A_t]$$

Недостаток жадной стратегии — по некоторым действиям a можем так и не набрать статистику для оценки $Q_t(a)$.

Компромисс «изучение–применение» (exploration–exploitation)
ε*-жадная стратегия*:

$$\pi_{t+1}(a) = \frac{1-\varepsilon}{|A_t|}[a \in A_t] + \frac{\varepsilon}{|A|}$$

Эвристика: параметр ε уменьшать со временем.

Метод UCB (upper confidence bound)

Выбор действия с максимальной верхней оценкой ценности:

$$A_t = \operatorname*{Arg\,max}_{a \in A} \left(Q_t(a) + \varepsilon \sqrt{\frac{2 \ln t}{k_t(a)}} \right),$$

где $k_t(a) = \sum_{i=1}^{t} [a_i = a]$, ε — параметр exr/ext-компромисса.

Интерпретация:
чем меньше $k_t(a)$, тем менее исследована стратегия,
тем выше должна быть вероятность выбрать a;

чем больше ε, тем стратегия более исследовательская.

Эвристика: параметр ε уменьшать со временем.

P. Auer, N. Cesa-Bianchi, P. Fischer. Finite-time analysis of the multiarmed bandit problem, Machine Learning, 2002.

Экспоненциальное скользящее среднее

Рекуррентная формула Moving Average для усреднения Q_t:

$$Q_t(a) = \alpha r_t + (1-\alpha) Q_{t-1}(a) = \mathrm{MA}_\alpha(r_t)$$

При $\alpha = \mathrm{const}$ это экспоненциальное скользящее среднее (EMA)
При $\alpha = \frac{1}{k_t(a)}$ это среднее арифметическое

Условие сходимости к среднему: $\sum\limits_{t=1}^{\infty} \alpha_t = \infty, \quad \sum\limits_{t=1}^{\infty} \alpha_t^2 < \infty$

Среднее арифметическое подходит для стационарных задач, экспоненциальное скользящее среднее — для нестационарных (в этом случае сходимости нет, но она и не нужна)

Задачи обучения с подкреплением, как правило, не стационарные

Напоминание. Экспоненциальное скользящее среднее

Задача прогнозирования временного ряда $y_0, \dots, y_t, \dots$:
— простейшая регрессионная модель — константа $y_t = c$,
— наблюдения учитываются с весами, убывающими в прошлое,
— прогноз $\hat{y}_{t+1}$ методом наименьших квадратов:

$$\sum_{i=0}^{t} \beta^{t-i}(y_i - c)^2 \to \min_c, \quad \beta \in (0,1)$$

Аналитическое решение — формула Надарая-Ватсона:

$$c \equiv \hat{y}_{t+1} = \frac{\sum_{i=0}^{t} \beta^i y_{t-i}}{\sum_{i=0}^{t} \beta^i}$$

Запишем аналогично $\hat{y}_t$, оценим $\sum_{i=0}^{t} \beta^i \approx \sum_{i=0}^{\infty} \beta^i = \frac{1}{1-\beta}$,

получим $\hat{y}_{t+1} = \hat{y}_t \beta + (1-\beta) y_t$, заменим $\alpha = 1 - \beta$:

$$\hat{y}_{t+1} = (1-\alpha)\hat{y}_t + \alpha y_t$$

Использование EMA для конструирования стратегий

Метод преследования (pursuit) жадной стратегии:

$$\pi_{t+1}(a) = \mathrm{EMA}_\alpha\left(\frac{[a \in A_t]}{|A_t|}\right), \quad a \in A$$

Сравнение с подкреплением (reinforcement comparison):
$\bar{r}_t = \mathrm{EMA}_\alpha(r_t)$ — *средняя премия* по всем действиям,
$p_t(a_t) = \mathrm{EMA}_\beta(r_t - \bar{r}_t)$ — *преимущество* (advantage) действия,

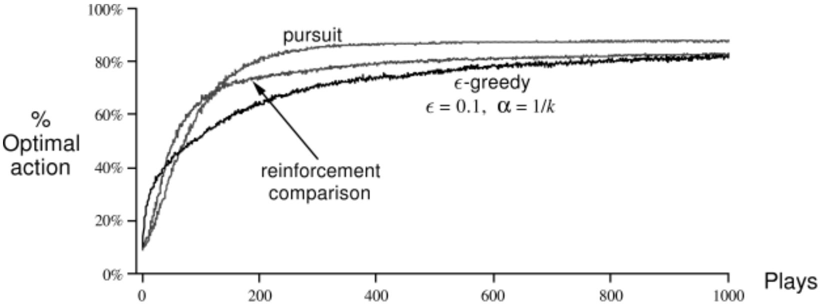
$$\pi_{t+1}(a) = \frac{\exp\left(\frac{1}{\tau} p_t(a)\right)}{\sum_{a'} \exp\left(\frac{1}{\tau} p_t(a')\right)},$$

при $\tau \to 0$ стратегия стремится к жадной,
при $\tau \to \infty$ — к равномерной, т.е. чисто исследовательской.

Экспериментальный факт:
не существует метода, универсально лучшего для всех задач

Сравнение стратегий в имитационных экспериментах

Зависимость доли оптимальных действий (% optimal action) от числа шагов t, усреднённая по 2000 синтетическим задачам



Richard Sutton, Andrew Barto. Reinforcement Learning: An Introduction. The MIT Press. 1998, 2004, 2018

Р. Саттон, Э. Барто. Обучение с подкреплением. 2011, 2020

Постановка задачи в случае, когда агент влияет на среду

A — конечное множество возможных *действий* (action)
S — конечное множество состояний среды (state)

Игра агента со средой:

инициализация стратегии $\pi_1(a \,|\, s)$ и состояния среды s_1;
для всех $t = 1, \dots, T, \dots$
- агент выбирает действие $a_t \sim \pi_t(a \,|\, s_t)$;
- среда генерирует премию $r_t \sim p(r \,|\, a_t, s_t)$
- и новое состояние $s_{t+1} \sim p(s \,|\, a_t, s_t)$;
- агент корректирует стратегию $\pi_{t+1}(a \,|\, s)$;

Марковский процесс принятия решений (МППР, MDP):

$$\mathsf{P}\bigl(s_{t+1}, r_t \,|\, s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, r_{t-2}, \dots, s_1, a_1\bigr) = \\ = \mathsf{P}\bigl(s_{t+1}, r_t \,|\, s_t, a_t\bigr)$$

Понятия выгоды и ценности действия

Суммарная выгода (return) на конечном горизонте T:

$$R_t = r_t + r_{t+1} + \dots + r_{t+T}$$

Дисконтированная выгода (discounted return):

$$R_t = r_t + \gamma r_{t+1} + \dots + \gamma^k r_{t+k} + \dots$$

где $\gamma \in [0, 1]$ — коэффициент дисконтирования,
$1 + \gamma + \gamma^2 + \dots = \frac{1}{1-\gamma}$ — горизонт дальновидности агента.

Функции ценности состояния $V^\pi(s)$ *и ценности действия в состоянии* $Q^\pi(s, a)$ при условии, что агент следует стратегии π:

$$V^\pi(s) = \mathsf{E}_\pi(R_t \,|\, s_t = s) \qquad = \mathsf{E}_\pi\biggl(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \biggm| s_t = s\biggr)$$

$$Q^\pi(s, a) = \mathsf{E}_\pi(R_t \,|\, s_t = s,\ a_t = a) = \mathsf{E}_\pi\biggl(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \biggm| s_t = s,\ a_t = a\biggr)$$

Жадные стратегии максимизации ценности

Рекуррентная формула для функции ценности $Q^\pi(s,a)$:

$$\begin{aligned} Q^\pi(s,a) &= \mathsf{E}_\pi\bigl(\textstyle\sum_{k=0}^\infty \gamma^k r_{t+k} \bigm| s_t = s,\ a_t = a\bigr) \\ &= \mathsf{E}_\pi\bigl(r_t + \gamma \textstyle\sum_{k=0}^\infty \gamma^k r_{t+k+1} \bigm| s_t = s,\ a_t = a\bigr) \\ &= \mathsf{E}_\pi\bigl(r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}) \bigm| s_t = s,\ a_t = a\bigr) \end{aligned}$$

Уравнение Беллмана для оптимальной функции ценности Q^*:

$$Q^*(s,a) = \mathsf{E}_\pi\bigl(r_t + \gamma \max_{a' \in A} Q^*(s_{t+1}, a') \bigm| s_t = s,\ a_t = a\bigr)$$

Утв. *Жадная стратегия* π относительно $Q^*(s,a)$ «выбирать то действие, на котором достигается максимум в уравнениях Беллмана», является оптимальной:

$$A_t = \operatorname*{Arg\,max}_{a \in A} Q^*(s_t, a)$$

Метод Q-обучения

Аппроксимируем $Q^*(s,a)$ экспоненциальным скользящим средним:

$$Q(s_t, a_t) = \text{EMA}_\alpha\bigl(r_t + \gamma \max_a Q(s_{t+1}, a)\bigr)$$

инициализация стратегии $\pi_1(a \,|\, s)$ и состояния среды s_1;
для всех $t = 1, \dots, T, \dots$
- агент выбирает действие $a_t \sim \pi_t(a \,|\, s_t)$;
- среда генерирует $r_t \sim p(r \,|\, a_t, s_t)$ и $s_{t+1} \sim p(s \,|\, a_t, s_t)$;
- $Q(s_t, a_t) := Q(s_t, a_t) + \alpha\bigl(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\bigr)$;

Утв. Если α_t уменьшается ($\sum_t \alpha_t = \infty$, $\sum_t \alpha_t^2 < \infty$), и все s посещаются бесконечное число раз, то $Q \overset{\text{пн}}{\to} Q^*$, $t \to \infty$

Возможны два способа выбора действий:
- **on-policy**: $a_t \sim \pi_t(a \,|\, s_t) \quad \Leftrightarrow \quad a_t \in \operatorname{Arg\,max}_a Q(s_t, a)$
- **off-policy**: $a_t \sim \pi_t(a \,|\, s_t)$ — другая стратегия на основе Q

Отличия от обычных задач машинного обучения

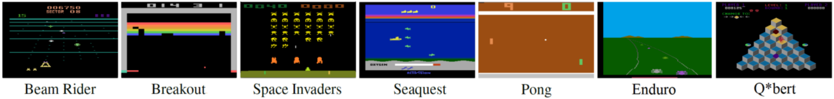
- выборка (s_t, a_t, r_t) не является независимой
- распределение $p(s_t, a_t, r_t)$ может меняться во времени и зависеть от стратегии агента π
- премии могут быть
  — отложенными (оценивать действия с задержкой)
  — разреженными (почти всё время $r_t = 0$)
  — зашумлёнными (не ясно, за что именно премия)

Какие параметрические модели можно обучать:

- функцию ценности действия в состоянии $Q(s, a; \theta)$
- функцию ценности состояния $V(s; \theta)$
- стратегию $\pi_{t+1}(a|s; \theta)$
- модель среды $(r_t, s_{t+1}) = \mu(s_t, a_t; \theta)$

Пример. Обучение играм Атари

Среда — эмулятор 7 игр Atari, каждый кадр 210×160pix 128col

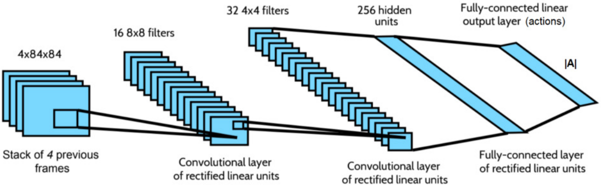


Состояния s — 4 последовательных кадра, сжатых до 84×84

Действия a — от 4 до 18, в зависимости от игры

Премии r — текущий SCORE согласно правилам игры

Функция ценности $Q(s, a; w)$ — CNN со входом s и $|A|$ выходами



V. Mnih et al. (DeepMind). Playing Atari with deep reinforcement learning. 2013

Метод DQN (Deep Q-Learning Network)

Сохранение траекторий $(s_t, a_t, r_t)_{t=1}^{T}$ в памяти (reply memory) для многократного *воспроизведения опыта* (experience replay)

Аппроксимация оптимальной функции ценности $Q(s_t, a_t)$ при фиксированных текущих параметрах сети w_t:

$$y_t = \begin{cases} r_t, & \text{если состояние } s_{t+1} \text{ терминальное} \\ r_t + \gamma \max_a Q(s_{t+1}, a; w_t), & \text{иначе} \end{cases}$$

Функция потерь для обучения нейросевой модели $Q(s, a; w)$:

$$\mathscr{L}_t(w) = \bigl(Q(s_t, a_t; w) - y_t\bigr)^2$$

Стохастический градиент SGD (по мини-батчам длины 32):

$$w_{t+1} = w_t - \eta\bigl(Q(s_t, a_t; w_t) - y_t\bigr)\nabla_w Q(s_t, a_t; w_t)$$

V.Mnih et al. (DeepMind). Playing Atari with deep reinforcement learning. 2013

Метод DQN: собираем всё воедино

инициализация reply-памяти и параметров сети w;
для всех эпизодов $m = 1, \dots, M$
- инициализация состояния среды s_1;
- **для всех** $t = 1, \dots, T_m$ (длина m-го эпизода)
  - $a_t = \begin{cases} \text{случайное действие}, & \text{с вероятностью } \varepsilon; \\ \arg\max_a Q(s_t, a, w), & \text{с вероятностью } 1 - \varepsilon; \end{cases}$
  - среда генерирует $r_t \sim p(r \,|\, a_t, s_t)$ и $s_{t+1} \sim p(s \,|\, a_t, s_t)$;
  - запомнить (s_t, a_t, r_t) в reply-памяти;
  - выбрать случайный фрагмент траектории из памяти;
  - **для всех** $j = 1, \dots, J$ (длина мини-батчей)
    - оценить y_j;
    - сделать градиентный шаг, обновить w;

V.Mnih et al. (DeepMind). Playing Atari with deep reinforcement learning. 2013

Градиентная оптимизация стратегии (policy gradient)

$\pi(a \,|\, s, \theta)$ — параметризованная стратегия агента
$f(s_t, a_t)$ — функция ценности или её оценка (например, R_t)

Задача максимизации $\mathsf{E}_\pi f$ по вектору параметров стратегии θ:

$$\mathsf{E}_\pi f(s,a) \equiv \mathsf{E}_{a\sim\pi(a|s,\theta)} f(s,a) \to \max_\theta$$

Градиентный метод: $\theta^{(t+1)} := \theta^{(t)} + \eta \nabla_\theta \mathsf{E}_{a\sim\pi} f(s,a)$

$$\nabla_\theta \mathsf{E}_{a\sim\pi} f(s,a) = \nabla_\theta \sum_{a\in A} f(s,a)\pi(a|s,\theta) = \sum_{a\in A} f(s,a)\nabla_\theta \pi(a|s,\theta) =$$

$$= \sum_{a\in A} f(s,a)\pi(a|s,\theta)\frac{\nabla_\theta \pi(a|s,\theta)}{\pi(a|s,\theta)} =$$

$$= \mathsf{E}_{a\sim\pi}\bigl[f(s,a)\nabla_\theta \ln \pi(a|s,\theta)\bigr]$$

Градиентная оптимизация стратегии (policy gradient)

Замена E_π эмпирической оценкой EMA градиента g_t:

$$g_{t+1} := g_t + \alpha\bigl(f(s_t, a_t)\nabla_\theta \ln \pi(a_t \,|\, s_t, \theta) - g_t\bigr)$$

Фактически, это стохастический градиент SGD с методом инерции Б.Т.Поляка для максимизации log-правдоподобия:

$$\sum_t f(s_t, a_t) \ln \pi(a_t \,|\, s_t, \theta) \;\to\; \max_\theta$$

Основные отличия от максимизации log-правдоподобия:

- вместо предсказания меток классов y_t — действия a_t
- вместо обучения по бинарным y_t — вещественные $f(s_t, a_t)$

Что можно использовать в качестве $f(s_t, a_t)$:

- функцию *выгоды* R_t,
- функцию *ценности* $Q(s_t, a_t)$,
- функцию *преимущества* (advantage) $Q(s_t, a_t) - V(s_t)$

Алгоритм REINFORCE

Отложенная выгода становится известна *в конце эпизода*

инициализация стратегии $\pi_1(a \,|\, s)$ и состояния среды s_1;
для всех эпизодов $m = 1, \dots, M$
- **для всех** $t = 1, \dots, T$
  - агент выбирает действие $a_t \sim \pi_t(a \,|\, s_t, \theta)$;
  - среда генерирует $r_t \sim p(r \,|\, a_t, s_t)$ и $s_{t+1} \sim p(s \,|\, a_t, s_t)$;
- $\theta := \theta + \eta \sum\limits_{t=1}^{T} R_t \nabla_\theta \ln \pi(a_t \,|\, s_t, \theta)$;

Преимущество policy gradient и алгоритма REINFORCE:

- легко обобщается на задачи с непрерывным множеством A

Недостаток:

- медленная сходимость, надо дожидаться конца эпизода

Алгоритм «актёр–критик» (Actor–Critic)

- *Актёр* оптимизирует стратегию, *критик* оценивает
- Вместо R_t — *преимущество* $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ (уменьшается дисперсия, улучшается сходимость)
- Параметр θ обновляется на каждом шаге

инициализация стратегии $\pi_1(a \,|\, s)$ и состояния среды s_1;
для всех $t = 1, \ldots, T$
- агент выбирает действие $a_t \sim \pi_t(a \,|\, s_t, \theta)$;
- среда генерирует $r_t \sim p(r \,|\, a_t, s_t)$ и $s_{t+1} \sim p(s \,|\, a_t, s_t)$;
- $V(s_t) := V(s_t) + \beta\bigl(r_t + V(s_{t+1}) - V(s_t)\bigr)$;
- $A(s_t, a_t) := r_t + V(s_{t+1}) - V(s_t)$;
- $\theta := \theta + \eta A(s_t, a_t) \nabla_\theta \ln \pi(a_t \,|\, s_t, \theta)$;

Преимущество Actor–Critic:

- легко параметризуется как стратегия, так и оценка $A(s_t, a_t)$

Алгоритм Actor–Critic с параметризацией

- Стратегия параметризуется как в policy gradient
- Оценка преимущества параметризуется моделью регрессии

$$\sum_t \bigl(A(s_t, a_t; w) - A_t\bigr)^2 \to \min_w$$

инициализация стратегии $\pi_1(a \,|\, s)$ и состояния среды s_1;
для всех эпизодов $m = 1, \dots, M$
  для всех $t = 1, \dots, T$
    агент выбирает действие $a_t \sim \pi_t(a \,|\, s_t, \theta)$;
    среда генерирует $r_t \sim p(r \,|\, a_t, s_t)$ и $s_{t+1} \sim p(s \,|\, a_t, s_t)$;
  $A_t = Q(s_t, a_t) - V(s_t)$ для всех $t = 1, \dots, T$;
  $\theta := \theta + \eta_1 \sum_{t=1}^T R_t \nabla_\theta \ln \pi(a_t \,|\, s_t, \theta)$;
  $w := w - \eta_2 \sum_{t=1}^T \bigl(A(s_t, a_t; w) - A_t\bigr) \nabla_w A(s_t, a_t; w_t)$;

Моделирование среды в обучении с подкреплением

Отличие Model-Based подходов от Model-Free:

- моделируется поведение среды $(r_t, s_{t+1}) = \mu(s_t, a_t; w)$
- возможно долгосрочное планирование действий
- возможна непрерывная параметризация как A, так и S
- в несложных технических системах управления адекватная параметрическая модель среды может быть известна

Трудность задачи:

- сложные среды требуют больших выборок для обучения моделей большой размерности
- RL может хорошо функционировать в смоделированной среде, и гораздо хуже — в настоящей

T.Moerland, J.Broekens, C.Jonker. Model-based reinforcement learning: a survey. 2020

Томпсоновское сэмплирование (Thompson sampling)

$x_{ta} \in \mathbb{R}^n$ — контекст, вектор признаков действия $a \in A$ на шаге t
$p(r_t|x, w)$ — вероятностная модель премии, $w \in \mathbb{R}^n$

Игра агента и среды:

инициализация априорного распределения $p_1(w)$;
для всех $t = 1, \dots, T$
- среда сообщает агенту контексты x_{ta} для всех $a \in A$;
- агент сэмплирует вектор модели премии $w_t \sim p_t(w)$;
- агент выбирает действие $a_t = \arg\max\limits_{a \in A} \langle x_{ta}, w_t \rangle$;
- среда генерирует премию r_t;
- агент корректирует распределение по формуле Байеса: $p_{t+1}(w) \propto p(r_t|x_{ta_t}, w)\, p_t(w)$;

S.Agrawal, N.Goyal. Thompson sampling for contextual bandits with linear payoffs. 2013.

Томпсоновское сэмплирование с гауссовскими распределениями

$$p(r|x,w) = \mathcal{N}(r; \langle x, w\rangle, \sigma^2), \quad p(w_t) = \mathcal{N}(w_t; w, \sigma^2 B^{-1})$$

Игра агента и среды (contextual bandit with linear payoff)

инициализация: $B = I_{n\times n}$; $w = 0_n$; $f = 0_n$;

для всех $t = 1, \dots, T$

- среда сообщает агенту контексты x_{ta} для всех $a \in A$;
- агент сэмплирует вектор линейной модели премии $w_t \sim \mathcal{N}(w, \sigma^2 B^{-1})$;
- агент выбирает действие $a_t = \arg\max\limits_{a\in A} \langle x_{ta}, w_t\rangle$;
- среда генерирует премию r_t;
- агент корректирует параметры распределения: $B := B + x_{ta_t} x_{ta_t}^{\mathsf{T}}$; $f := f + x_{ta_t} r_t$; $w := B^{-1} f$;

S.Agrawal, N.Goyal. Thompson sampling for contextual bandits with linear payoffs. 2013.

Регрессия с инкрементным обучением и доверительной оценкой

$r(s, a)$ — функция премии за действие a в состоянии s
$\hat{r}(s, a; w)$ — регрессионная модель премии с параметром w
$\mathrm{UCB}(s, a)$ — верхняя оценка отклонения $\hat{r} - r$
δ — параметр (чем больше, тем больше exploration)

Игра агента со средой:

инициализация стратегии $\pi_1(a|s)$;
для всех $t = 1, \dots, T, \dots$
- агент выбирает действие
  $a_t = \arg\max\limits_{a \in A} \Big(\hat{r}(s, a; w) + \delta \mathrm{UCB}(s, a)\Big)$;
- среда генерирует премию $r_t = r(s_t, a_t)$;
- регрессия $\hat{r}(s, a; w)$ дообучается на точке (s_t, a_t, r_t);

Lihong Li, Wei Chu, John Langford, Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. WWW-2010.

Пример. Рекомендация новостных статей пользователям

Агент — рекомендательная система для персонализации показов новостных статей (Yahoo! Today).

F1..F4 — позиции для показа заголовков новостей.

A — новостные статьи, действия системы
s_t — состояние = пользователь, которому даём рекомендацию
$x_{ta} \in \mathbb{R}^n$ — признаковое описание пары (s_t, a)
$r_{ta} \in \{0, 1\}$ — пользователь s_t кликнул на статью a
$Q_t(a)$ — средняя премия, CTR (click-through rate) статьи

Цель — повышение среднего CTR и «счастья пользователя»

Lihong Li, Wei Chu, John Langford, Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. WWW-2010.

Линейная модель премий и гребневая регрессия

Пусть $x_{ta} \in \mathbb{R}^n$, $w_a \in \mathbb{R}^n$.
Линейная модель премий для действия $a \in A$ в состоянии s_t:

$$\mathsf{E}\bigl[r_{ta} \,\big|\, x_{ta}\bigr] = \langle x_{ta}, w_a \rangle.$$

Гребневая регрессия: обучение w_a для действия a в момент t:

$$\sum_{i=1}^{t} \bigl[a_i = a\bigr] \bigl(\langle x_{ia}, w_a\rangle - r_{ia}\bigr)^2 + \frac{\tau}{2}\|w_a\|^2 \;\to\; \min_{w_a}.$$

$w_a = \bigl(F_a^{\mathsf{T}} F_a + \tau I_n\bigr)^{-1} F_a^{\mathsf{T}} y_a$ — решение задачи МНК, где

$F_a = \bigl(x_{ia}\bigr)_{i=1:\, a_i=a}^{t}$ — $\ell \times n$-матрица объекты–признаки,

$y_a = \bigl(r_{ia}\bigr)_{i=1:\, a_i=a}^{t}$ — $\ell \times 1$-вектор ответов,

$\ell = k_t(a) = \sum\limits_{i=1}^{t} [a_i = a]$ — объём обучающей выборки.

LinUCB: линейная модель с верхней доверительной оценкой

Доверительный интервал с коэффициентом доверия $1 - \alpha$ для линейной модели регрессии w: $\|Fw - y\| \to \min_w$:

$$y = \langle x, w \rangle \pm \hat{\sigma} Z_\alpha \sqrt{x^\intercal (F^\intercal F)^{-1} x},$$

$Z_\alpha \equiv t_{\ell-n, 1-\frac{\alpha}{2}}$ — квантиль распределения Стьюдента,
$\hat{\sigma}^2 = \frac{1}{\ell-n} RSS$ — оценка дисперсии отклика y.

Стратегия $\pi_t(a, x) = \frac{1}{|A_t|}[a \in A_t]$ — действие с максимальной верхней оценкой ценности UCB (upper confidence bound):

$$A_t = \operatorname{Arg}\max_{a \in A} \Bigl(\langle x_{ta}, w_a \rangle + \delta \hat{\sigma} Z_\alpha \sqrt{x_{ta}^\intercal \bigl(F_a^\intercal F_a + \tau I_n\bigr)^{-1} x_{ta}} \Bigr).$$

Чем больше параметр δ, тем больше исследования.

LinUCB: особенности реализации и обобщения

- *Инкрементный алгоритм* пересчёта w_a и матрицы $\bigl(F_a^{\mathsf{T}} F_a + \tau I_n\bigr)^{-1}$ при добавлении каждой строки в F_a.
- Гибридная линейная модель $Q^*(a) = \langle \tilde{x}_t, v\rangle + \langle x_{ta}, w_a\rangle$, где $\tilde{x}_t$ — часть контекста, не зависящая от действия a.
- «Сырые признаки»:
  пользователи: 12 соцдем, 200 география, $\sim$1000 категорий,
  статьи: $\sim$100 категорий.
- Используется кластеризация и *понижение размерности*: $\dim w_a = 6$, $\dim v = 36$.
- Можно было бы использовать любую другую модель с инкрементным обучением и доверительными оценками.

Lihong Li, Wei Chu, John Langford, Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. WWW-2010.

Оценивание модели по историческим данным

Проблема off-line оценивания стратегии π:
исторические данные накоплены при использовании
другой стратегии (logging policy) $\pi_0(a)$, отличной от π

Идея:
для оценивания $Q_t(a)$ отбираются только те события (x_{ta}, a, r_{ta}),
для которых стратегии π и π_0 выбирали одинаковое действие:

$$a = \arg\max_a \pi(a, x_{ta}) = \arg\max_a \pi_0(a)$$

(нужны очень большие данные или сходство стратегий)

Утв. Если $\pi_0(a)$ — равномерное распределение, то
оценка $Q_t(a)$ по отобранной выборке является несмещённой.

Lihong Li, Wei Chu, John Langford, Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. WWW-2010.

Резюме в конце лекции

- В обучении с подкреплением нет ответов учителя, есть только ответная реакция среды
- Что можно обучать в Model-Free подходах:
  — функцию ценности $Q(s, a; w)$, например, методом SGD
  — стратегию $\pi(a|s; w)$, методом Policy Gradient
  — модели актора $a(s; w_1)$ и критика $Q(s, a; w_2)$
- Что можно обучать в Model-Based подходах:
  — только модель премии $r(s, a; w)$
  — модель среды $(r_t, s_{t+1}) = \mu(s_t, a_t; w)$
- Компромисс «изучение–применение» при любом обучении с подкреплением подбирается экспериментальным путём

R.S.Sutton, A.G.Barto. Reinforcement Learning: An Introduction. 2018
`https://spinningup.openai.com/en/latest`
Yuxi Li. Resources for Deep Reinforcement Learning. 2018