СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Майданюк Дмитрий,
Maxymizely, Noosphere
1.07.2016

Что такое обучение с подкреплением ?

- Обучение с подкреплением - это методология обнаружения «оптимальной» стратегии, в соответствии с которой должен действовать интеллектуальный агент в некотором окружении для максимизации долговременного выигрыша;
- Окружение, понимается в соответствии с марковским процессом принятия решений со множеством конечных состояний, вероятности выигрышей являются случайными величинами, стационарными в рамках задачи;
- Принятие субоптимальных решений явно не ограничивается. Обучение с подкреплением пытается найти компромисс между *исследованием неизученных областей и применением имеющихся знаний*;

Простейшая модель обучения с подкреплением

Пусть S — множество состояний окружения,
A — множество действий

В некоторый момент времени t агент характеризуется состоянием s_t ,$s_t \in S$ и множеством возможных действий $\mathrm{A}(s_t)$, выбирая действие $\alpha \in \mathrm{A}(s_t)$ он переходит в состояние s_{t+1} и получает вещественный выигрыш r_t.

Требуется построить стратегию $a: S \to A$, для максимизации величины выигрыша:

$$R = r_0 + r_1 + \cdots + r_n$$

или в общем случае $\mathrm{R} = \sum_t \gamma^t r_t$, γ – дисконтирующий множитель для предстоящего выигрыша.

Задача об многоруком бандите

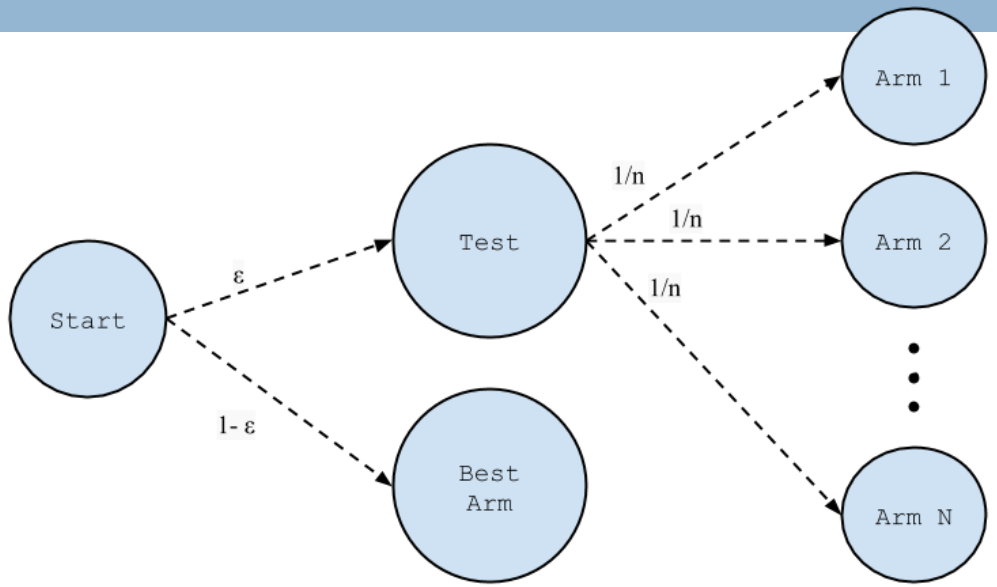
- Имеется только одно состояние и возможность выбора одного действия из множества
- Агент находится в комнате с множеством одноруких бандитов и ему необходимо максимизировать выигрыш за конечное число шагов
- Каждый бандит имеет свое ожидание выигрыша, неизвестное заранее агенту
- За каждое действие система дает вознаграждение и возвращает агента в исходное состояние
- К этой задача можно формализовать множество прикладных задач интернет-маркетинга (показ рекламы, тестирование лендинг страниц и т.п.)

p_1	p_2	p_3	...	p_n

Общие характеристики подходов к решению задач обучения с подкреплением

Группа методов/стратегий	*Преимущества*	*Ограничения*
Переборные стратегии (Динамическое программирование, индексы Гитинса, UCB1)	Доказуемо оптимальные стратегии	Конечный горизонт, небольшая последовательность шагов, вычислительная сложность при увеличении масштаба задачи
Линейное вознаграждение действия	Динамическое обновление веса, быстрый пересчет	Чувствительны к начальному приближению.
Эвристические методы (ε-greedy, Softmax, exp3)	Нечувствительны к увеличению масштаба задачи	Чаще всего приводят к субоптимальным решениям

Алгоритм $\varepsilon - greedy$



В основе алгоритма лежит относительная простая стратегия:

1. Существует 2 фазы: исследование (тест) и эксплуатация
2. На исследование отводится ε часть экспериментов на эксплуатацию 1-ε
3. В фазе эксплуатации используется «оптимальный» на текущий момент времени вариант

Алгоритм UCB1

В основе алгоритма лежит относительная простая формула выбора лучшей стратегии:

$$j = \arg\max\left(\bar{x}_j + \sqrt{\frac{2\ln n}{n_j}}\right)$$

$\bar{x}_j$ - Средний доход от ручки j

n_j - Количество проб ручки j

n -Общее количество проб

Алгоритм линейного вознаграждения действия

Алгоритм линейно увеличивает вероятность определенного действия, если оно привело к успеху. Если действие на i-й ручке привело успеху

$$p_i = p_i + \alpha(1 - p_i)$$

$$p_j = p_j - \alpha p_j, j \neq i$$

Алгоритм Softmax

В основе алгоритма лежит формула выбора лучшей стратегии в соотв. с распределением Гиббса:

$$j = \arg\max \frac{e^{E[R_j]/t}}{\sum_i e^{E[R_i]/t}}$$

t -константа, называемая температурой

$E[R_j]$ -наблюдаемый средний доход от ручки j

Алгоритм exp3

Модифицированный вариант алгоритма Softmax с экспоненциально взвешенными вероятностями:

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$$

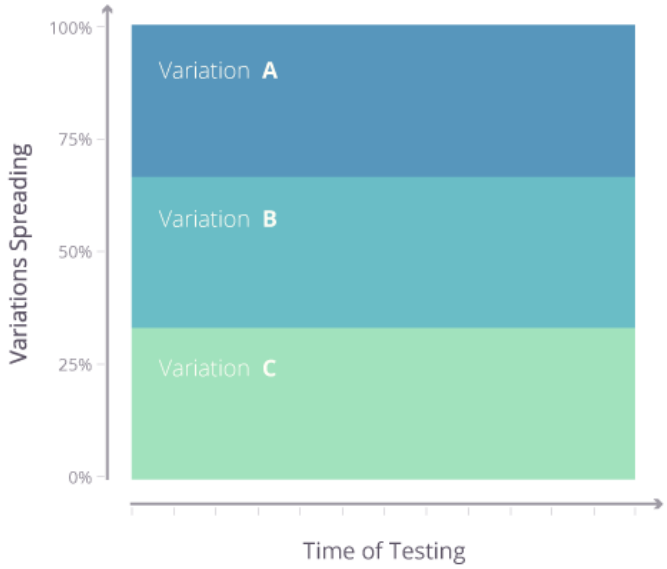
$$w_j(t+1) = w_j(t)\exp\left(\gamma \frac{E[R_j(t)]}{p_j(t)K}\right)$$

$\gamma \in (0;1)$ -некоторая константа

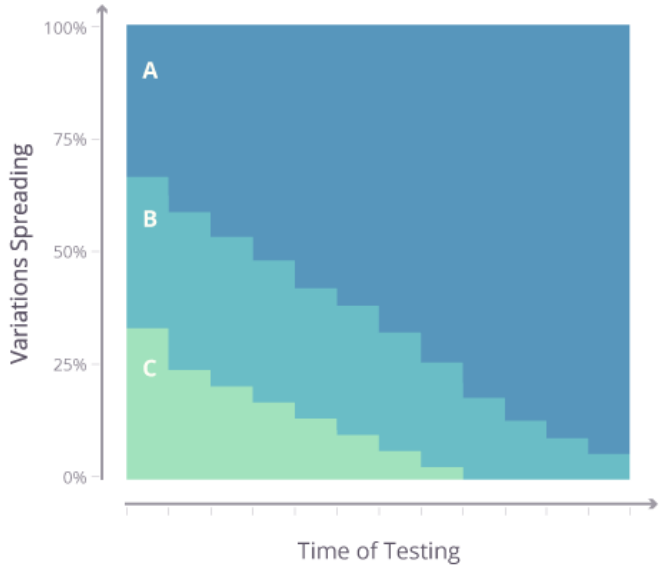
K -число ручек

Пример использования обучения с подкреплением в A/B тестировании

Показ с равномерным распределением вариаций контента



Показ с динамическим распределением вариаций контента



Сравнительный анализ результатов. Постановка задачи

Дано 3 варианта контента: A, B, C, таких что вероятности выигрышей соотв. равны

$$P\{A\} = 0.4 \qquad P\{B\} = 0.45 \qquad P\{C\} = 0.5$$

Интеллектуальный агент «не знает их» заранее. Необходимо максимизировать выигрыш R за N подходов, в каждом подходе можно осуществить M попыток

Распределение числа попыток по подходам

1

	Подход , N							
	1	2	3	4	5	6	7	8
A	33.33%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%
B	33.33%	10.00%	80.00%	80.00%	10.00%	80.00%	80.00%	80.00%
C	33.33%	80.00%	10.00%	10.00%	80.00%	10.00%	10.00%	10.00%
	Подход (накопительно)							
A	33.33%	21.67%	17.78%	15.83%	14.67%	13.89%	13.33%	12.92%
B	33.33%	21.67%	41.11%	50.83%	42.67%	48.89%	53.33%	56.67%
C	33.33%	56.67%	41.11%	33.33%	42.67%	37.22%	33.33%	30.42%

2

	Подход,N							
	1	2	3	4	5	6	7	8
A	33.33%	33.33%	33.33%	33.00%	33.00%	33.33%	33.33%	33.33%
B	33.33%	33.33%	33.33%	33.00%	33.00%	33.33%	33.33%	33.33%
C	33.33%	33.33%	33.33%	33.00%	33.00%	33.33%	33.33%	33.33%

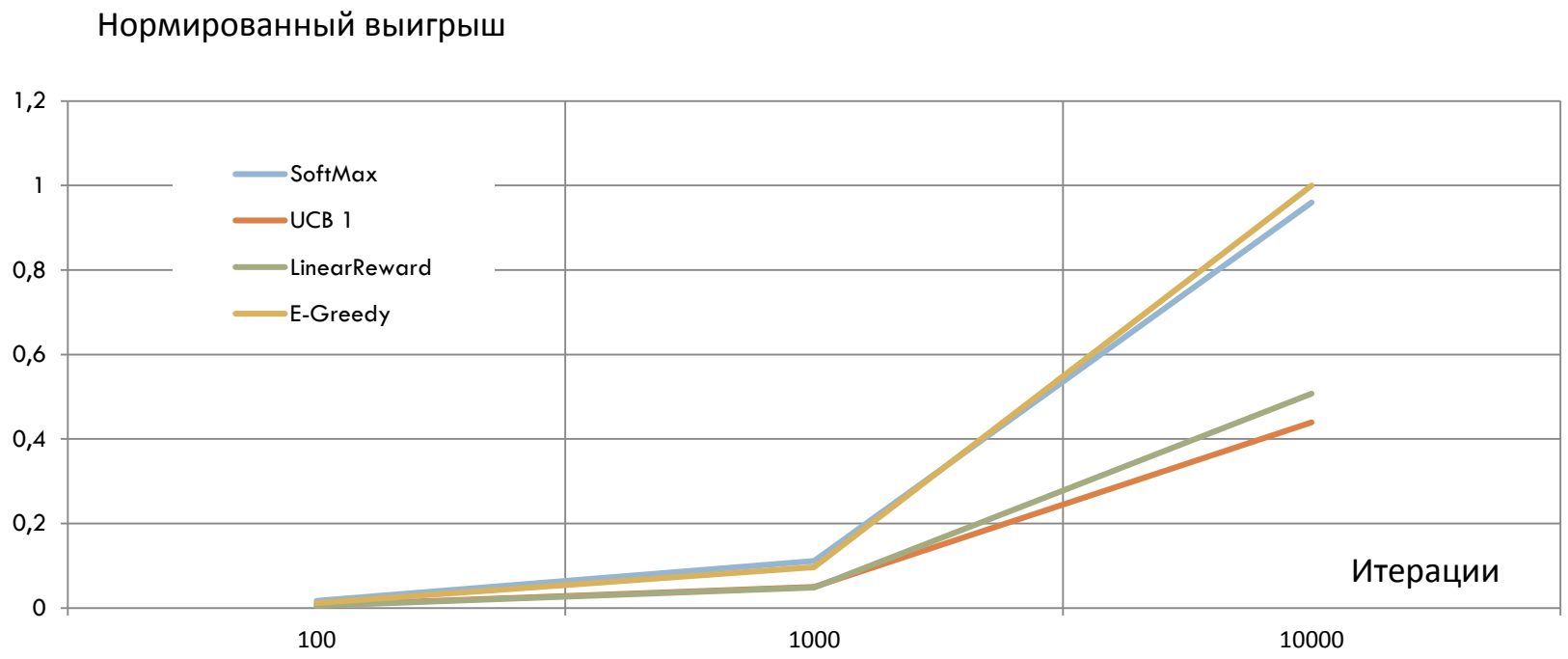
1. «Лучшая» стратегия
2. «Тривиальная» стратегия

Вероятности выигрышей

	Вероятности выиграша на подход							
A	33.36%	34.85%	44.70%	46.44%	38.08%	34.92%	43.29%	44.52%
B	44.94%	43.77%	43.49%	45.56%	45.64%	49.94%	47.05%	43.01%
C	48.34%	56.48%	48.72%	49.13%	51.95%	46.50%	52.49%	51.09%

	Вероятность выигрыша (накопительно)							
	1	2	3	4	5	6	7	8
A	33.36%	33.70%	35.76%	37.45%	37.54%	37.22%	37.87%	38.52%
B	44.94%	44.67%	44.45%	44.63%	44.76%	45.38%	45.56%	45.32%
C	48.34%	54.09%	51.87%	51.07%	51.27%	50.39%	50.71%	50.76%

Результаты анализа работы различных алгоритмов



Выводы:

- Многие задачи интернет-маркетинга (показ баннерной рекламы, тестирование посадочных страниц, ранжирование вариантов контента) могут быть формализованы как задачи о «многоруких бандитах», одной из простейших задач обучения с подкреплением
- Существует достаточно хорошо исследованная методология решения такого класса задач, алгоритмы UCB1, Softmax, E-greedy, Linear reward-inaction и др.
- Наилучшим образом в практическом плане оказались лучшими алгоритмы Softmax, E-greedy