Онлайн ценообразование с помощью структурированных многоруких бандитов

Гунаев Руслан

Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра интеллектуальных систем

Научный руководитель д.ф.-м.н. В. В. Стрижов
Научный консультант Ю. Дорн

Москва,
2021 г.

Онлайн ценообразование

Задача

Разработать алгоритм, решающий задачу динамического ценообразования в страховании.
Проблема динамического ценообразования определена следующим образом: учитывая количество товаров для продажи и заданный горизонт продаж, адаптивно корректировать цены с течением времени, чтобы максимизировать ожидаемую прибыль.

Требования к алгоритму

- на каждой итерации алгоритма цена должна подбираться так, чтобы избежать крупных трат во время проведения эксперимента;
- оптимальная цена должна удовлетворять ограничениям: $x_{\min} \leqslant x^* \leqslant x_{max}$;
- максимум прибыли должен достигаться в оптимальной цене x^*.

Предлагается

- алгоритм UCB+QBC,
- алгоритм WEIGHTED UCB+QBC.

Методы решения задачи динамического ценообразования

1. Ravi G., Matyas S. and Quoc T. Thompson Sampling for Dynamic Pricing. 2018.
2. Yichong X., Ruosong W., Lin F. Y., Aarti S. and Dubrawski A. Preference-based Reinforcement Learning with Finite-Time Guarantees. 2020.
3. Schlosser R. and Boissier M. Dynamic Pricing under Competition on Online Marketplaces: A Data-Driven Approach. 2019.

Постановка задачи динамического ценообразования

Обозначения

- $x \in \mathbb{R}$ – цена страховки,
- $r(x)$ – прибыль,
- $Q(x)$ – функция спроса от цены.

Требуется найти последовательность цен $\hat{\mathrm{X}}(T) = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^T$ такую, что

$$\hat{\mathrm{X}}(T) = \arg\max_{\mathrm{X}(T)} \sum_{t=1}^{T} \mathrm{E}[r(x_t)],$$

при условии, что для каждого момента выполнены ограничения

$$x_{\min} \leqslant x_t \leqslant x_{\max}.$$

Максимизация прибыли

Задача упрощается, если выразить прибыль через спрос:

$$r(x) = Q(x) \cdot x.$$

Зависимость спроса от цены неизвестна. Предлагается использовать модели спроса:

- линейная функция: $Q(x) = \max\{-ax + b, 0\}$;
- гиперболическая функция: $Q(x) = \max\{-\frac{a}{x} + b, 0\}$;
- экспоненциальная функция: $Q(x) = \max\{-\exp(ax + b)c + d, 0\}$;
- показательная функция: $Q(x) = \max\{ba^x + c, 0\}$.

Для нахождения максимума прибыли, в рамках данной работы, использована каждая из моделей спроса.

Используемые методы решения задачи

Upper Confidence Bound (UCB)

На каждой итерации алгоритма выбираем ручку согласно:

$$x_i = \arg\max_{x \in X} \left(\mathrm{E}[\hat{r}(x)] + \sqrt{\frac{2 \log n}{n_x}} \right),$$

n – число раз, которое мы дергали все ручки, n_x – сколько раз мы дергали ручку x, $\hat{r}(x)$ – значение функции прибыли в точке x.

Активное обучение. Несогласие в комитете (QBC)

Метод, в котором алгоритм оперирует не одной моделью, а сразу несколькими, которые формируют комитет.
У нас есть J моделей $M^J = \{m_1, m_2, \ldots, m_J\}$. Выбираем цену x так, чтобы модели в этой точке максимально расходились. В качестве критерия расхождения используем выборочную дисперсию.

Алгоритм: UCB+QBC

В предложенном алгоритме выбираем точку, максимизируя функционал:

$$\lambda \left(\frac{1}{J} \sum_{j=1}^{J} \mathrm{E}[\hat{r}_j(x)] + \sqrt{\frac{2 \ln n}{n_x}} \right) + (1-\lambda) \left(\frac{1}{J} \sqrt{\sum_{j=1}^{J} \mathrm{D}[\hat{r}_j(x)]} \right),$$

- $\lambda \in (0; 1)$ – некоторый параметр, с которым мы учитываем вес оценки в точке (UCB),
- $1 - \lambda$ учитывает вес расхождения в комитете,
- $\frac{1}{J} \sqrt{\sum_{j=1}^{J} \mathrm{D}[\hat{r}_j(x)]}$ – расхождение в комитете.

Алгоритм: WEIGHTED UCB+QBC

Изменим алгоритм, добавив веса, связанные с оценкой качества каждой модели:

$$\frac{\lambda}{JA}\sum_{j=1}^{J}\mathrm{E}[\hat{r}_j(x)]\alpha[r_j(x)] + \lambda\sqrt{\frac{2\ln n}{n_x}} + \\ + (1-\lambda)\left(\frac{1}{J}\sqrt{\frac{1}{B}\sum_{j=1}^{J}\mathrm{D}[\hat{r}_j(x)]\beta[r_j(x)]}\right),$$

- $\alpha[r_j(x)]$ – вес j-ой модели в точке x для UCB,
- $\beta[r_j(x)]$ – вес j-ой модели в точке x для QBC,
- $A = \sum_{j=1}^{J}\alpha[r_j(x)]$ – нормировочная константа,
- $B = \sum_{j=1}^{J}\beta[r_j(x)]$ – нормировочная константа.

Контроль качества моделей

Требуется подобрать такую параметризацию функции спроса, чтобы она верно указывала на оптимальное значение цены, поэтому вес j-ой модели в точке x для UCB будем находить согласно:

$$\alpha[r_j(x)] = \exp(-(x_j^* - a)^2), a = \frac{\sum_{i=1}^{T} r(x_i)x_i}{\sum_{i=1}^{T} r(x_i)},$$

x_j^* – оптимум j-ой модели.

Предлагается учитывать риск итераций в точках, удаленных от известных:

$$\beta[r_i(x)] = \begin{cases} 1, & \text{если } x_j^* \in [x_{\min}; x_{\max}], \\ 0, & \text{иначе.} \end{cases}$$

Вычислительный эксперимент

Цели эксперимента

- сравнение существующих алгоритмов с предложенными,
- получение максимальной прибыли с продажи страховок при помощи предложенного алгоритма.

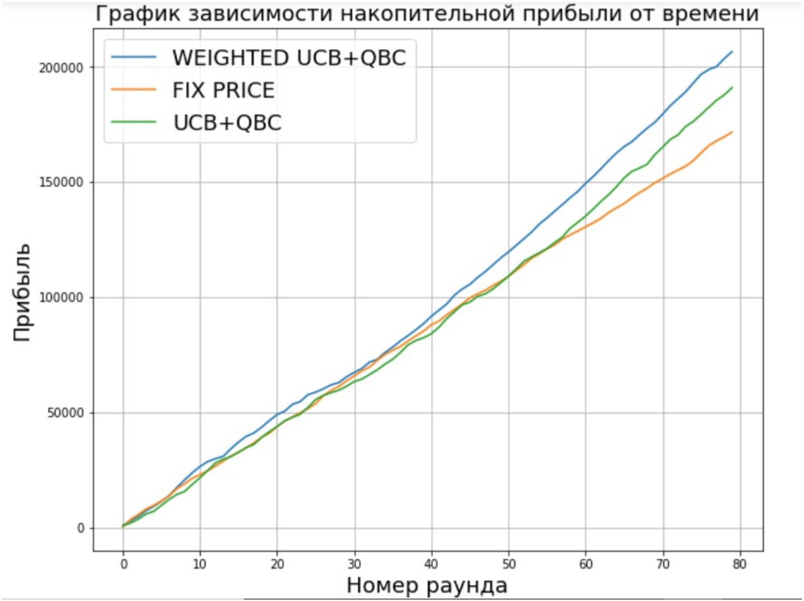
Критерии качества

- прибыль, полученная с продаж страховки,
- время, за которое каждый из алгоритмов нашел оптимальную цену.

Данные

Данные поступают онлайн с продажи страховок по каналам, потери внутри которых несущественны для компании.

Вычислительный эксперимент: сравнение алгоритмов



Выводы

В результате экспериментов мы получили:

- предложенный алгоритм WEIGHTED UCB+QBC увеличил прибыль с продаж страховки на 20% по сравнению с политикой фиксированной цены,
- WEIGHTED UCB+QBC быстрее находит оптимальную цену по сравнению с алгоритмом UCB+QBC.

Алгоритм	Число раундов
UCB+QBC	56 ± 4
WEIGHTED UCB+QBC	34 ± 5

Выносится на защиту

- алгоритм WEIGHTED UCB+QBC,
- результаты экспериментов.