«Московский физико-технический институт (национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Кафедра «Интеллектуальные системы»

Гунаев Руслан Гуламович

Онлайн ценообразование с помощью структурированных многоруких бандитов

03.03.01 — Прикладные математика и физика

Выпускная квалификационная работа Бакалавра

Научный руководитель:
Рудаков К. В

Научный консультант:
Дорн Ю. В

Москва
2021

Аннотация

Большинство онлайн-рынков характеризуются конкурентной средой. Из-за сложности таких рынков трудно разработать эффективные стратегии ценообразования. В данной работе предложен алгоритм WEIGHTED UCB+QBC для решения задачи ценообразования в страховании. Предложенный алгоритм является усовершенствованием алгоритма UCB. Модификация заключатеся в использовании активного обучения, которое позволяет использовать различные параметризации неизвестной функции спроса, делая наиболее правильный отбор точек в процессе работы алгоритма. Это позволяет сокращать траты во время проведения экспериментов. Также мы используем весовые функции, позволяющие отдавать предпочтение только хорошим параметризациям, также учитывать риски использования цен, сильно удаленных от уже проверенных. Главная цель алгоритма – как можно быстрее находить оптимальную цену для продажи страховки. В результате экспериментов на данных компании Тинькофф мы получили увеличение прибыли на 20% по сравнению с политикой фиксированных цен.

Содержание

1 Введение

Динамическое ценообразование является фундаментальной проблемой и имеет различные приложения в сфере финансовых услуг, доставке товаров, такси, страховании. С развитием интернета, а также с распространением вируса по всему миру, актуальность этой задачи вышла на новый уровень. Крупнейшие технологические компании нуждаются в новых методах, позволяющих как можно быстрее решать эту задачу. Решая задачу динамического ценообразования, онлайн-магазины имеют возможность постоянно менять цены, тем самым находя оптимальную как для себя с точки зрения выгоды, так и для покупателей; онлайн-такси могут правильно регулировать спрос и предложение, тем самым осуществляя как можно больше заказов, то же касается и доставок. На конкурентном рынке установка больших цен может привести к полному отсутствию спроса, в противоположном случае магазин рискует уйти в большой минус [10]. Решая задачу, продавец предлагает цены конкретного продукта последовательно к потоку потенциальных покупателей и наблюдает за успехом или неудачей в каждой продаже. Предполагается, что характеристики покупателей идентичны и могут быть описаны моделью спроса $Q(x)$ – вероятностью, что человек купит товар по цене x [3]. Цель – максимизировать общую прибыль на горизонте времени длиной T [9]. На практике модель спроса неизвестна продавцу и должна быть изучена онлайн путем продажи товаров. Таким образом, на каждом шаге продавец при выборе цены сталкивается с проблемой изучения модели спроса или же продажи продукта по цене, имеющей лучшую историю [8].

Можно выделить типичные подходы в решении задачи динамического ценообразования [7]:

1. сбор информации о продажах товара по разным стоимостям, с целью получения эластичности,

2. построение модели спроса от цены,

3. нахождение оптимальной цены.

Такой подход никак не учитывает изменения на рынке, появление конкурентов, товаров-заменителей. В данном подходе цена подбирается один раз и дальше товар продается по фиксированной цене. На практике же эластичность может меняться, более того модель спроса может сильно отличаться от действительности. В этой работе мы предлагаем другой подход, основанный на многоруких бандитах и активном обучении [5, 15]. Мы не пытаемся найти правильную модель спроса, вместо этого мы

используем различные аппроксимации [2], от которых достаточно лишь того, чтобы они верно указывали на оптимальную цену. Предлагается:

1. Алгоритм UCB+QBC, который является улучшением алгоритма UCB с помощью несогласия в комитете(QBC) [6, 13, 17]. Благодаря этому методу алгоритм ищет не просто оптимальные с точки зрения формулы цены, а отбирает их так, чтобы расхождение в аппроксимациях было максимально большое. Таким образом, изучаются точки, позволяющие наиболее быстро находить верную аппроксимацию, а значит и оптимальную цену.

2. Алгоритм WEIGHTED UCB+QBC. Этот алгоритм является модификацией первого с помощью весовых функций. Здесь мы хотим учитывать две очень важные эвристики: при выборе цены нужно отдавать предпочтение наилучшим аппроксимациям, нельзя тестировать заведомо плохие точки, ведь тогда мы потеряем деньги на эксперименте.

Подобные методы решения задачи динамического ценообразования активно используются в компании Яндекс, именно поэтому было решено изучать именно это направление и попытаться его развить. Предложенный подход способен адаптироваться к изменениям, изменяя аппроксимации спроса. Также данный алгоритм в силу использования активного обучения быстрее чем UCB находит оптимальную цену.

Проблема динамического ценообразования совместно с многорукими бандитами исследовалась как от отдельная проблема еще в 1974 г. Ротшильдом [14]. Математическая абстракция многоруких бандитов в своей базовой форме включает N независимых ручек и одного игрока. Каждая ручка при игре выдает независимые и одинаково распределенные награды, полученные из распределения с неизвестным средним θ_i. Каждый раз игрок выбирает одну ручку для игры, стремясь максимизировать ожидаемую сумму награды, полученную за горизонт T [16].

Существует несколько основных алгоритмов, позволяющих быстро находить оптимальные(приносящие наибольшую награду) ручки. Первый из них ε-greedy алгоритм [16]. Самый простой алгоритм, который на каждом шаге с вероятностью ε выбирает ручку с наибольшим средним выигрышом, а в остальных случаях выбирает ручку случайно, таким образом, во время работы алгоритма будет изучена каждая ручка. Но как показывает практика, плохо быть жадным, а быть жадным бандитом плохо вдвойне. Поэтому в реальных задачах применяются два других алгоритма: UCB [1, 16] и семплирование Томпсона [4, 11].

В статье [12] предлагается алгоритм, использующий в своей основе семплирование Томпсона. Для того, чтобы решить проблему изменяющейся эластичности, предложенный алгоритм на каждой итерации будет обновлять эластичность, семплируя ее из некоторого распределения.

2 Постановка задачи

2.1 Задача динамического ценообразования

Формально задачу динамического ценообразования можно поставить следующим образом: требуется найти последовательность цен $\hat{\mathbf{X}}(T) = (x_1, x_2, \ldots, x_T)$ такую, что прибыль $r(x)$ будет максимальна. Существует 4 варианта постановки:

1. $$x^* = \arg\max_{x(T)} \mathsf{E}[r(x(T))].$$

   В этой постановке нам необходимо найти оптимальную цену в конкретный момент времени T.

2. $$x^* = \arg\max_{x(t)} \mathsf{E}[r(x(t))].$$

   Необходимо найти оптимальную цену за наименьшее время.

3. $$\hat{\mathbf{X}}(T) = \arg\max_{\mathbf{X}(T)} \sum_{t=1}^{T} \mathsf{E}[r(x_t)].$$

   Здесь мы хотим оптимизировать всю траекторию цен.

4. Найти зависимость $r(x)$.

Первые две постановки нам не подходят, потому что во время тестирования алгоритма мы будем тестировать неоптимальные цены, тем самым теряя деньги компании, поэтому в рамках данной работы мы сконцентрируемся на 3 постановке.

Также необходимо учитывать бизнес-ограничения.

1. Нельзя менять цену слишком резко $\dfrac{x_{i+1}}{x_i} \sim 1$ для любого момента времени i.

2. Экспертные ограничения на цену $x_{\min} \leqslant x_i \leqslant x_{\max}$ для любого момента времени i.

3. Каждый эксперимент стоит денег.

Таким образом, можем записать итоговую постановку задачи. Найти $\hat{\mathbf{X}}(T)$ такую, что

$$\hat{\mathbf{X}}(T) = \arg\max_{\mathbf{X}(T)} \sum_{t=1}^{T} \mathsf{E}[r(x_t)] \\ s.t\ x_{\min} \leqslant x_t \leqslant x_{\max}\ \forall t \leqslant T .$$

В текущей постановке выполнены все три ограничения: в процессе эксперимента цена подбирается так, чтобы максимизировать общую прибыль, оптимальная цена лежит в нужном диапазоне, внутри которого разброс цен небольшой.

2.2 Функция спроса

Теперь добавим специфику. Нашу целевую функцию прибыли можно представить в виде

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

где $Q(x)$ – число проданных страховок по цене x, иначе говоря спрос. Проблема заключается в том, что мы не знаем настоящую зависимость спроса от цены, поэтому предлагается использовать различные ее параметризации:

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

Эти параметризации не обязаны точно аппроксимировать функцию спроса, основная задача заключается в том, чтобы в результате работы алгоритма обновлять параметры так, чтобы аппроксимации верно указывали на оптимальную цену.

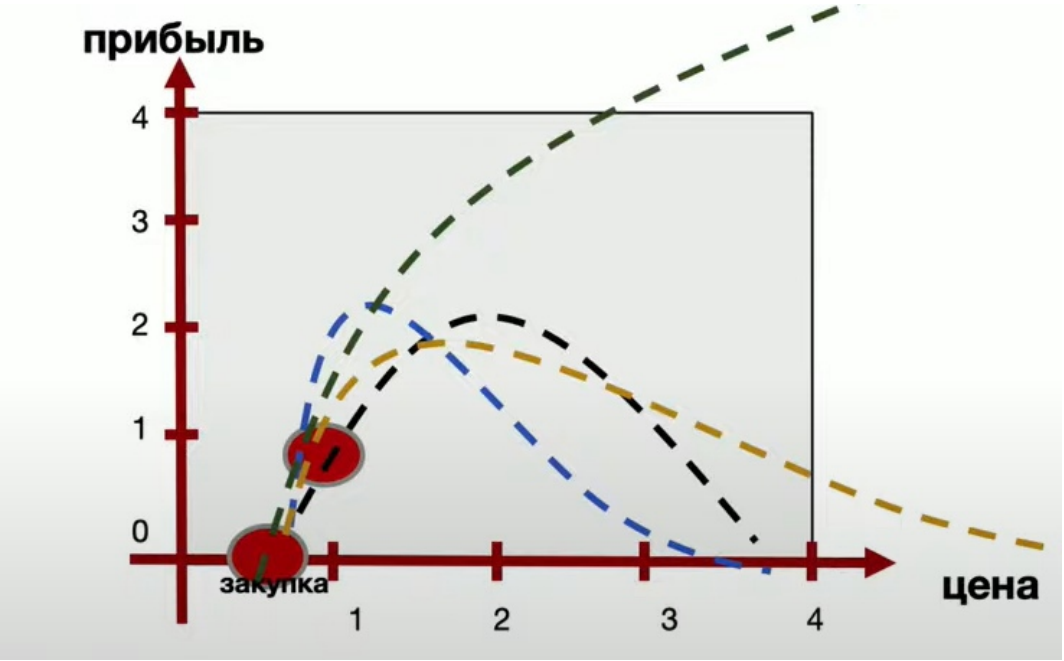

Рис. 1: Пример аппроксимации по двум точкам.

3 Многорукие бандиты

3.1 Алгоритм UCB

Положим $n_{i,t}$ – количество раз, когда была сыграна ручка i до момента времени t. r_t – награда, которую мы получаем в момент времени t. $I_t \in \{1, 2, \ldots, N\}$ – выбранная ручка в момент времени t. Эмпирическая оценка награды ручки i в момент t:

$$\hat{\mu}_{i,t} = \frac{\sum_{s=1:I_s=i}^{l} r_s}{n_{i,t}}.$$

Регрет задается следующим образом

$$R(T) = T\mu^\star - \sum_{t=0}^{T-1} \mathsf{E}\left[r^t\right] \tag{1}$$

UCB присваивает каждой ручке в каждый момент времени следующее значение:

$$\mathrm{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}}$$

Algorithm 1: UCB algorithm

Data: N arms, number of rounds $T \geq N$
for $t = 1 \ldots N$ **do**
  play arm t
end
for $t = N + 1 \ldots T$ **do**
  play arm
$$I_t = \arg\max_{i \in \{1 \ldots N\}} \mathrm{UCB}_{i,t-1}$$
end

Theorem 3.1 (Верхняя оценка ожидаемого регрета UCB алгоритма)**.** *Пусть $R(T)$ – регрет UCB алгоритма для некоторого многорукого бандита, тогда для любого T верна верхняя оценка*

$$\mathsf{E}[R(T, \Theta)] \leq \sum_{i:\mu_i<\mu^*} \frac{4 \ln T}{\Delta_i} + 8\Delta_i, \;\; \Delta_i = \mu^* - \mu_i.$$

Доказательство. Есть более фундаментальная причина выбора $\sqrt{\frac{\ln t}{n_{i,t}}}$. Эта верхняя оценка вытекает из неравенства Чернова-Хоффдинга. Для каждой ручки верно

$$|\hat{\mu}_{i,t} - \mu_i| < \sqrt{\frac{\ln t}{n_{i,t}}}$$

с вероятностью не меньше $1 - 2/t^2$. Их этого получаем два важных неравенства:

1. Нижняя граница для $\mathbf{UCB}_{i,t}$. С вероятностью не меньше $1 - 2/t^2$,

$$\text{UCB}_{i,t} > \mu_i$$

2. Верхняя граница для $\hat{\mu}_{i,t}$ с большим числом семплов. При $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$, с вероятностью не меньшей $1 - 2/t^2$ верно,

$$\hat{\mu}_{i,t} < \mu_i + \frac{\Delta_i}{2}$$

1 показывает, что значение UCB, вероятно, равно истинному вознаграждению: в этом смысле алгоритм UCB оптимистичен. 2 – что при наличии достаточного количества (а именно, по крайней мере, $\frac{4 \ln t}{\Delta_i^2}$) семплов оценка вознаграждения, вероятно, не превышает истинное вознаграждение более чем на $\Delta_i/2$. Эти ограничения показывают, что алгоритм быстро находит субоптимальную ручку.

Lemma 3.2. *В любой момент времени t, если субоптимальная ручка i ($\mu_i < \mu^*$) была сыграна $n_{i,l} \geq \frac{4 \ln t}{\Delta_i^2}$ раз, тогда $UCB_{i,t} < UCB_{I^*,t}$ с вероятностью $1 - 4/t^2$. Это значит, что любого t,*

$$P\left(I_{t+1} = i \mid n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}\right) \leq \frac{4}{t^2}$$

Доказательство.

$$\begin{aligned}
\text{UCB}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} &\leq \hat{\mu}_{i,t} + \frac{\Delta_i}{2} && \text{при } n_{i,t} \geq \frac{4 \ln L}{\Delta_i^2} \\
&< \left(\mu_i + \frac{\Delta_i}{2}\right) + \frac{\Delta_i}{2} \\
&= \mu^* && \text{при } \Delta_i := \mu^* - \mu_i \\
&< \text{UCB}_{i^*,t}
\end{aligned}$$

□

Lemma 3.3. *Пусть $n_{i,T}$ – количество раз, когда ручка i была выбрана алгоритмом . Тогда для любой ручки с $\mu_i < \mu^*$,*

$$\mathsf{E}\left[n_{i,T}\right] \leq \frac{4\ln T}{\Delta_i} + 8$$

Доказательство. Для любой ручки i ожидаемое число раз, когда она была сыграна

$$\begin{aligned}
\mathsf{E}\left[n_{i,T}\right] &= 1 + \mathsf{E}\left[\sum_{t=N}^{T} \mathsf{I}\left(I_{t+1} = i\right)\right] \\
&= 1 + \mathsf{E}\left[\sum_{t=N}^{T} \mathsf{I}\left(I_{t+1} = i, n_{i,t} < \frac{4\ln t}{\Delta_i^2}\right)\right] + \mathsf{E}\left[\sum_{t=N}^{T} \mathsf{I}\left(I_{t+1} = i, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right)\right. \\
&\leq \frac{4\ln T}{\Delta_i^2} + \mathsf{E}\left[\sum_{t=N}^{T} \mathsf{I}\left(I_{l+1} = i, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right)\right] \\
&= \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} P\left(I_{l+1} = i, n_{i,l} \geq \frac{4\ln t}{\Delta_i^2}\right) \\
&= \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} P\left(I_{t+1} = i \mid n_{i,t} \geq \frac{4\ln T}{\Delta_i^2}\right) P\left(n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right) \\
&\leq \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \frac{4}{t^2} \\
&\leq \frac{4\ln T}{\Delta_i^2} + 8
\end{aligned}$$

□

Тогда пользуясь леммами, итоговый ожидаемый регрет до времени T:

$$\mathsf{E}[R(T,\Theta)] = \sum_{i:\mu_i<\mu^*} \mathsf{E}\left[n_{i,T}\right]\Delta_i \leq \sum_{i:\mu_i<\mu^*} \frac{4\ln T}{\Delta_i} + 8\Delta_i$$

□

3.2 Активное обучение

В силу того, что во время проведения эксперимента мы не можем рисковать, проверяя плохие цены, в данной работе предлагается использовать активное обучение. Цель активного обучения заключается в том, чтобы достичь как можно лучшего качества, использовав при этом как можно меньше примеров. Предлагается использовать несогласие в комитете. 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.

3.3 UCB+QBC

В работе предлагается комбинация UCB и активного обучения. Теперь вместо обычного UCB, будет выбрана ручка, максимизирующая функционал:

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

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

Первое слагаемое соответствует части UCB, здесь происходит усреднение всех моделей. Второе слагаемое – QBC, в качестве расхождения выбрана выборочная дисперсия.

3.4 WEIGHTED UCB+QBC

Во время проведения базовых экспериментов первого алгоритма мы столкнулись с проблемой, что все модели одинаково учитываются при принятии решения. Для меньшего использования неизвестных точек придумано две эвристики:

1. мы хотим отдавать большее предпочтение тем параметризациям, оптимум которых близок к настоящему оптимуму функции спроса;
2. мы не хотим работать с параметризациями, показывающие хоть и верный оптимум, но который находится вне экспертных ограничений.

В новом алгоритме предлагается не просто усреднять все модели, а делать это взвешенно.

Теперь ручка будет выбираться по новой формуле

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

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

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:

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

4 Вычислительный эксперимент

Вычислительный эксперимент был проведен внутри компании Тинькофф. В качестве товара была выбрана страховка. Для начала требовалось найти сегмент, внутри которого потери на экспериментах не существенны для компании. Также алгоритм UCB работает с конечным числом ручек, поэтому требовалось правильно структурировать цены. Помимо этого изменение цены с 10 рублей до 1010 более существенно, чем изменение с 10000 рублей до 11000. Ниже приведено распределение цен по ручкам.

Нижняя граница	Верхняя граница	Среднее
299	301	300
301	303	302
303	305	304
305	307	306
307	309	308
309	311	310
311	315	313
315	317	316
317	321	319
321	323	322
323	327	325
327	333	330
333	337	335
337	345	341
341	357	347

Таблица 1: Распределение цен по ручкам

Эксперимент представляет из себя ABC-тестирование. Первая группа видит цены согласно политике фиксированных цен, страховка продается за 310 рублей. Вторая группа соответствует алгоритму UCB+QBC. Третья – WEIGHTED UCB+QBC. До проведения эксперимента необходимо понять, что средняя ежедневная прибыль в каждой группе одинаковая, чтобы исключить дополнительные факторы.

Номер группы	1	2	3
Средняя прибыль	13106	12898	13367

Таблица 2: Средняя ежедвневная прибыль в группах

Целью эксперимента является сравнение накопительной прибыли для трех подходов: политика фиксированных цен, алгоритм UCB+QBC, алгоритм WEIGHTED UCB+QBC. Ожидаем получить наибольшую прибыль для третьей группы клиентов. Критерием качества являются накопительная прибыль и время, за которое алгоритмы находят оптимальную цену.

Итоговый экспреримент длился 3 недели, цены менялись 4 раза за день.

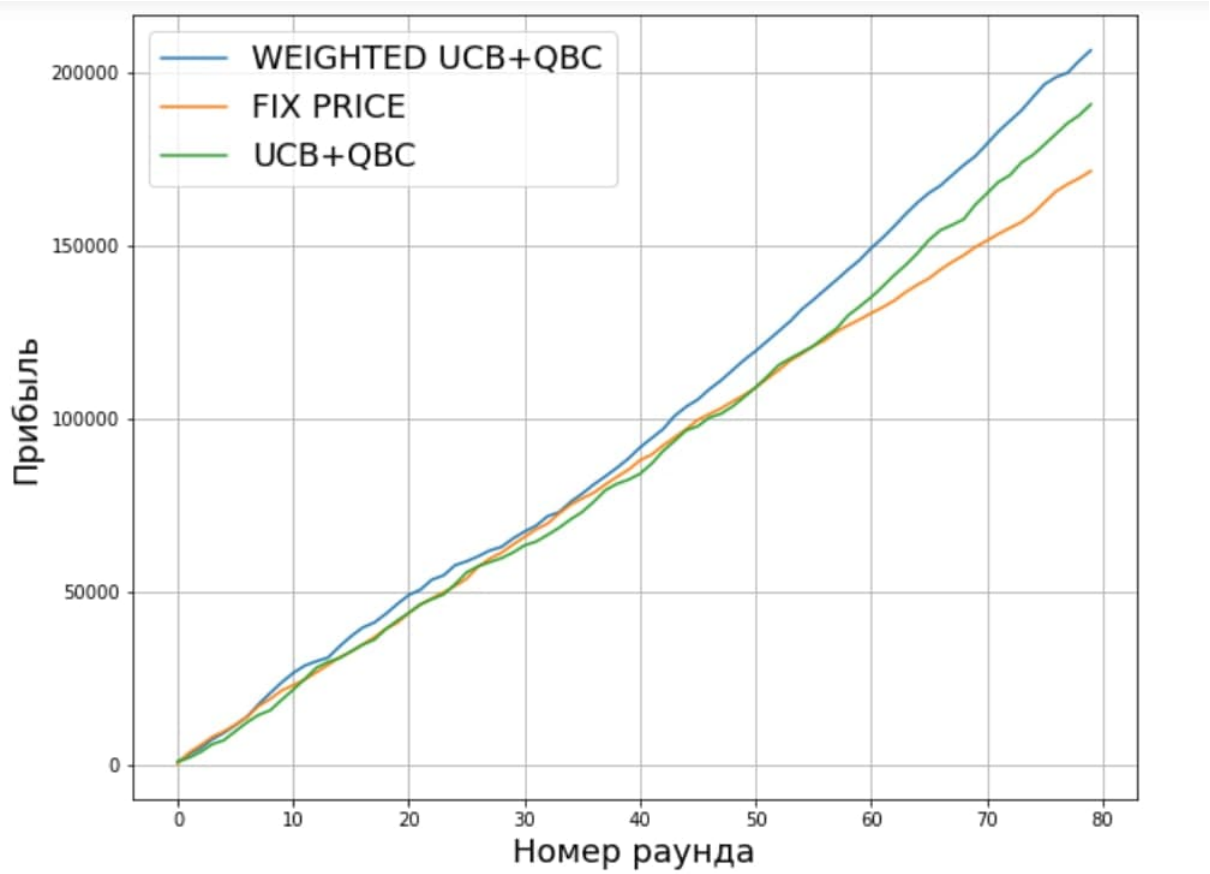

Рис. 2: График зависимости накопительной прибыли от номера раунда

По результатам данного эксперимента мы получили увеличение на 20% прибыли для алгоритма WEIGHTED UCB+QBC по сравнению с политикой фиксированных, которая работает на данный момент.

Теперь, зная спрос на каждую цену, мы можем провести второй эксперимент, чтобы узнать время, за которое алгоритмы находят оптимальную цену.

Проведем синтетический эксперимент: Если алгоритм выбирает определенную ручку, то случайно выбираем спрос из исторических данных.

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

Рис. 3: Результаты второго эксперимента

WEIGHTED UCB+QBC существенно быстрее находит оптимальную цену по сравнению с алгоритмом UCB+QBC.

5 Заключение

В качестве темы для исследования в рамках бакалаврской работы я выбрал задачу динамического ценообразования. С развитием онлайн сервисов эта задача становится одной из самых актуальных, так как цель любого бизнеса – заработок денег. Существует множество подходов в решении этой задачи, я остановился на том методе, который активно применяется в компании Яндекс в различных сервисах.

В данной работе предложен алгоритм WEIGHTED UCB+QBC, который показал значительное улучшение в качестве по сравнению с политикой, которая применяется на данный момент в компании Тинькофф в страховании. Помимо этого алгоритм показал отличную скорость в нахождении оптимальной цены.

В дальнейшем планируется несколько направлений работы:

1. нахождение дополнительных важных эвристик при подборе цены,
2. изменение весовых функций, способных адаптировать эти эвристики на практике,
3. использование семплирования Томпсона вместо алгоритма UCB, так как данный метод имеет более высокие скорости сходимости,
4. доказательство оптимальности построенного метода,
5. внедрение построенной модели в продакшн компании.

Список литературы

[1] Eric Moulines Aurélien Garivier. On upper-confidence bound policies for non-stationary bandit problems. 2008.

[2] Archer B. Demand forecasting and estimation. 1987.

[3] J. Buchanan. The demand and supply of public goods. 1968.

[4] Abbas Kazerouni Ian Osband Zheng Wen Daniel Russo, Benjamin Van Roy. A tutorial on thompson sampling. 2017.

[5] Michael I. Jordan David A. Cohn, Zoubin Ghahramani. Active learning with statistical models. *Journal of Articial Intelligence Research 4*, pages 129–145, 1996.

[6] H. Sompolinsky H. S. Seung, M. Opper. Query by committee. *Proceedings of the fifth annual workshop on Computational learning theory*, 1992.

[7] Keskin N Bora Harrison, J Michael and Assaf Zeevi. Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science*, pages 570—-586, 2012.

[8] Thomas Keil Shaker A. Zahra Juha Uotila, Markku Maula. Exploration, exploitation, and financial performance: analysis of sp 500 corporations. *Strategic Management*, pages 221–231, 2009.

[9] Phillips Lewic and Robert. Pricing and revenue optimization. stanford university press,. 2005.

[10] Yishay Mansour and Aleksandrs Slivkins. Competing bandits: Learning under competition. 2018.

[11] Lihong Li Olivier Chapelle. An empirical evaluation of thompson sampling. 2011.

[12] Quoc Tran Ravi Ganti, Matyas Sustik and Brian Seaman. Thompson sampling for dynamic pricing. 2018.

[13] Ross D. King Robert Burbidge, Jem J. Rowland. Active learning for regression based on query by committee. 2007.

[14] M. Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 1974.

[15] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pages 102–112, 2012.

[16] Aleksandrs Slivkins. Introduction to multi-armed banditsn. *Mathematical Programming Computation*, 2017.

[17] Eli Shamir Yoav Freund, H. Sebastian Seung. Selective sampling using the query by committee algorithm. 1997.