Динамическое ценообразование с помощью томсоновского сэмплирования

А. А. Харь[1], **Ю. В. Дорн**[2], **В. В. Стрижов**[3]

Аннотация: В этой работе исследуется задача динамического ценообразования. Диамическое ценообразование заключается в изменении цен на товары на регулярной основе и использует результаты предыдущих ценовых решений для обновления цен. Наиболее популярные подходы к динамическому ценообразованию используют подход пассивного обучения, при котором алгоритм использует исторические данные для приближения значений различных неизвестных параметров задачи, и использует обновленные параметры для создания новых цен. Был предложен алгоритм активного обучения, использующий томсоновское сэмплирование, для более эффективного и точного приближения значений неизвестных параметров. Доказана теорема о сходимости этого алгоритма.

Ключевые слова: многорукие бандиты, томсоновское сэмплирование, доход, спрос, эластичность, пассивные алгоритмы, активные алгоритмы.

DOI: 00.00000/00000000000000

[1]Московский физико-технический институт, khar.aa@phystech.edu

[2]Московский физико-технический институт, dornyv@yandex.ru

[3]Вычислительный центр имени А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, Московский физико-технический институт, strijov@phystech.edu

Содержание

1 Введение

В последние годы интернет позволил многим отраслям промышленности выйти в онлайн. Коммерция, предполагающая продажу товаров как физических, так и цифровых, через интернет, является одной из наиболее быстро развивающихся отраслей в наше время. В настоящее время существует много гигантских онлайн продавцов, таких как Walmart.com, Amazon.com, Jet.com, Target.com, в связи с чем произошло перемещение коммерции с офлайн среды на онлайн. В то время как крупнейшие обычные магазины ограничены в пространстве и, как правило, хранять порядка 100 тысяч самых популярных товаров, крупнейшие розничные сети электронной коммерции менее ограничены в пространстве и, как правило, могут позволить себе наличие порядка 400 миллионов товаров. Розничные провавцы электронной комерции извлекли выгоду из стремительного роста онлайн активности. Теперь нетрудно изучать поведение клиентов в интернете, например, какие товары клиент просматривал на сайте, какие товары в целом популярны, какие товары клиент покупал в прошлом, и в дальнейшем использовать эту персонализированную информацию в будущем.

Одна из самыхважных проблем, которую должен решить каждый розничный продавец – это правильная оценка цены товара. Установка слишком высоких или слишком низзких цен может привести к ухудшению качества обслуживания клиентов и существенно повлиять на доход продавца. Размер каталога товароа, хранящихся в обычном магазине, очень мал, и из-за сложности изменения цен ценообразование на товары в обычных магазинах является простой задачей, которая может быть решена путем проведения элементарного анализа данных в Excel. Однако наличие большого количества товаров на сайте электронной комерции, возможность собирать большие объемы данных о поведении пользователей, а также относительная легкость, с которой можно изменять цены на товары, создают уже непростую, интересную задачу ценообразования. В обычных магазинах цены на товары, как правило, не меняются. Однако такая статическая ценовая политика не работает в электронной коммерции. Статическое ценообразование никак не моделирует кривую спроса/цены и не позволяет фирме реагировать на изменения цен у конкурентов. Более того статическое ценообразование не может в полной мере использовать возможности вычисений, данных и простоту реализации изменений цен в онлайн-мире. Поэтому динамическое ценообразование является предпочтительной ценовой политикой в электронной коммерции.

В этой работе мы изучим алгоритмы динамического ценообразования. Диманическое ценообразование в электронной коммерции – сложная задача из-за своих масштабов и непредсказуемого поведения покупателей. Непонятно, как можно использовать данные, собранные в прошлом, для того, чтобы выставить боеле выгодные для продавца цены. Типичный подход к динамическому ценообразованию состоит из следующих шагов:

- сбор исторических данных о цене товара и спросе на этот товар в разных ценовых точках;
- создание статистической модели спроса как функции от цены и оценка параметров модели, использующая исторические данные;
- оптимизация некоторой метрики, представляющей интерес продавца (например, доход) используя функцию спроса, с целью получить новую оптимальную цену;
- применение полученной цены к товару и повторение вышеописанного.

Мы классифицируем эти подходы к динамическому ценообразованию как пассивные алгоритмы. Такие подходы близоруки и пытаются оптимизировать метрику в краткосрочной перспективе, не прилагая активных усилий для изучения функции спроса. Показано, что такие подходы приводят к неполному обучению [5], а также имеют низкую производительность и в долгосрочной перспективе теряют доход. В этой работе мы покажем, как можно избежать этой проблемы.

1. Мы предлагаем алгоритм активного обучения для динамического ценообразования. В то время как подходы динамического ценообразования, основанные на пассивном обучении, рассматривают проблему оптимизации, заключающуюся в оптимизации интересующей метрики на конечном горизонте T, мы предлагаем алгоритм под названием MAX-REV-TS, который решает проблему динамического ценообразования как проблему оптимизации в условиях неопределенности.

2. Мы начнем с объяснения архитектуры существующей системы динамического ценообразования. Эта система делает точные параметрические предположения относительно функции спроса и регулярно максимизирует интересующий показательно с уловиях ограничений. В данной работе мы будем рассматривать функцию дохода в качестве интересующей нас метрики. Параметры функции спроса регулярно обновляются. Поскольку эта система основана на пассивном обучении, мы будем называть эту систему MAX-REV-PASSIVE.

3. Мы представляем новую систему, MAX-REV-TS, которая похожа на MAX-REV-PASSIVE, но использует алгоритм многорукого бандита – томсоновское сэмплирование (Thompson Sampling). MAX-REV-TS использует ту же параметрическую формулу для функции спроса, что и MAX-REV-PASSIVE, и максимизирует доход на фиксированном горизонте T. Томсоновское сэмплирование (TS) работает следующим образом: к неизвестным параметрам модели функции спроса применяется байесовское априорное значение, и с использованием правила Байеса вычисляется апостериорное распределение. Затем томсоновское сэмплирование генерирует выборку параметра из апостериорного распределения и генерирует новый набор цен, максимизируя функцию вознаграждения для следующего временного шага, используя полученные параметры. Этот процесс повторяется до конца фиксированного горизонта T.

4. Обычно алгоритмы многоруких бандитов применяются в случаях, когда существует конечное количество фиксированных

ручек [7], мы же продемонстрируем практическое применение более сложной задачи оптимизации многоруких бандитов, имеющих бесконечной число ручек.

2 Постановка задачи

Дано:

- $\mathcal{B}$ ($|\mathcal{B}| = n$) - множество товаров;
- Период времени T - горизонт планирования;
- $\{\mathcal{J}_t\}_{t=1}^{T}$ - история заказов, где $\mathcal{J}_t = \{(d_{i,t}, p_{i,t})\}_{i\in\mathcal{B}}$,
  $d_{i,t}$ - количество товара i купленное в период времени t по цене $p_{i,t}$.

Фактический *агрегированный* спрос $d_{i,t}$ при цене $p_{i,t}$ можно считать i-й компонентой реализации случайной многозначной функции *агрегированного* спроса $\mathcal{D}_t(p_{1,t}, \dots, p_{n,t})$, которая нам неизвестна.

Задача: Построить алгоритм ценообразования, максимизирующий ожидаемый доход за период времени T.

Замечание: После каждой установки цен мы наблюдаем фактический спрос (покупки), эти наблюдения алгоритм может использовать во время работы.

Введем обозначения:

- $\mathbf{p}_t = (p_{1,t}, \dots, p_{n,t})$ - вектор цен;
- $R_t(\mathbf{p}_t) = \sum_{i\in\mathcal{B}} \mathcal{D}_{i,t}(\mathbf{p}_t) \cdot p_{i,t}$ - случайная функция прибыли при ценах $\mathbf{p}_t$;
- $\mathbf{p}^* = \arg\max\limits_{\mathbf{p}} \sum\limits_{t=1}^{T} \mathtt{E} R_t(\mathbf{p})$ - оптимальные постоянные цены при известном распределении $\mathcal{D}(\mathbf{p}_t)$.

Задача: Построить алгоритм подбора цен $\mathbf{p}_t$, максимизирующий выражение:

$$\sum_{t=1}^{T} \mathtt{E} R_t(\mathbf{p}_t)$$

при условии, что распределение $\mathcal{D}(\mathbf{p}_t)$ изначально неизвестно.

3 Архитектура алгоритма динамического ценообразования

Любой алгоритм динамического ценообразования должен каким-либо образом моделировать функцию спроса, а затем использовать эту смоделированную функцию для рассчета оптимальных цен, которые будут применяться каждый день.

Стандартный подход к моделированию спроса в задачах ценообразования содержит предположение о том, что функция спроса принадлежит некоторому параметрическому семейству, неизвестные параметры оцениваются, используя статистические методы. Исследовано множество параметрических моделей, таких как линейные модели, лог-линейные, модели постоянной эластичности и логит-модели. Во всех этих моделях считается, что функция спроса постоянна. В действительности же она таковой не является, поэтому необходимо моделировать функцию спроса более точно.

Обычно функцией спроса является модель постоянной эластичности, которая моделирует спрос на товар i следующим образом:

$$d_i(p_i) = f_i\Big(\frac{p_i}{p_{0,i}}\Big)^{\gamma_i^*}, \tag{1}$$

где $d_i(p_i)$ – спрос на товар i по цене p_i, f_i – базовый спрос на товар i по цене $p_{0,i}$,
$\gamma_i^* < 0$ – эластичность товара i.

Для того, чтобы учитывать возможные изменения в функции спроса, мы смоделируем функцию спроса на товар i в момент времени t следующим образом:

$$d_{i,t}(p_{i,t}) = f_{i,t}\Big(\frac{p_{i,t}}{p_{i,t-1}}\Big)^{\gamma_i^*}, \tag{2}$$

где $f_{i,t}$ – базовый прогноз спроса товара i в момент времени t, если цена товара составляет $p_{i,t-1}$, γ_i^* – эластичность товара i. Эластичность товара γ_i^* показывает, как спрос товара i меняется относительно его цены.

Для прогнозирования базового спроса $f_{i,t}$ для товара i в момент времени t мы будем использовать спрос на товар i в прошлом в разных ценовых точках
$((p_{i,1}, d_{i,1}), \ldots, (p_{i,t-1}, d_{i,t-1}))$. Для прогнозирования можно использовать, например, модель для временных рядов ARMA. В данной работе достаточно рассматривать это как некий черный ящик, и можно предположить, что прогнозы базового спроса у нас есть.

Если $p_{i,t}$ близко к $p_{i,t-1}$, то можно следующим образом аппроксимировать функции спроса:

$$d_{i,t}(p_{i,t}) \approx f_{i,t} + (p_{i,t} - p_{i,t-1}) \frac{f_{i,t}\gamma_i^*}{p_{i,t-1}} \tag{3}$$

С такой аппроксимацией оценка эластичности может быть сведена к задаче линейной регрессии, которую можно решать, используя, к примеру метод наименьших квадратов. Вместо того, чтобы смотреть все данные, мы сосредоточимся только на последних (за последние 2-3 месяца).

Оценить эластичность гораздо сложнее, чем прогнозировать базовый спрос. Это связано с тем, что если товар имеет k различных цен в последние месяцы, то оценивая эластичноть этого товара, мы используем k точек. Часто бывает, что k меньше 5 для многих элементов, в связи с чем оценка эластичности усложняется, так как данные получаются разреженными. В таких случаях предлагается сгруппировать товары с одинаковой эластичностью и оценивать эластичность таких товаров вместе.

В реальных задачах всегда есть некие ограничения на выставляемые цены, то есть минимальная цена, ниже которой мы устанавливать не имеем права, и максимальная – выше которой мы устанавливать не имеем права. Эти органичения могут меняться со временем, но они всегда известны алгоритму. В данной работе нам будет достаточно считать, что эти ограничения всегда являются допустимым, выпуклым набором и достаточно просты, чтобы была возможна оптимизация.

Здесь и далее мы будем рассматривать ситуацию, когда покупатель положил в корзину $\mathcal{B}$ некие товары. Прибыль от товара i из корзины $\mathcal{B}$ может быть оценена следующим образом:

$$Rev_{i,t}(p_{i,t}) = p_{i,t} \cdot d_{i,t}(p_{i,t}) \tag{4}$$

$$\approx p_{i,t}\Big(f_{i,t} + (p_{i,t} - p_{i,t-1})\frac{f_{i,t}\gamma_i^*}{p_{i,t-1}}\Big) \tag{5}$$

$$= \frac{p_{i,t}^2 f_{i,t}\gamma_i^*}{p_{i,t-1}} + p_{i,t}(f_{i,t} - f_{i,t}\gamma_i^*), \tag{6}$$

где первая строка – определение прибыли, вторая строка – соответствует линейной аппроксимации функции спроса.

Пусть $\mathbf{p} = [p_1, \ldots, p_n]$ – вектор цен на товары в корзине $\mathcal{B}$. Используя оценки на эластичность γ_i^*, базовый спрос $f_{i,t}$, уже полученные нами, решаем оптимизационную задачу, максимизируя суммарную прибыль за все товары в корзине:

$$\mathbf{p}_t = \arg\max_{\mathbf{p}} \sum_{i \in \mathcal{B}} \frac{p_{i,t}^2 f_{i,t}\gamma_i^*}{p_{i,t-1}} + p_{i,t}(f_{i,t} - f_{i,t}\gamma_i^*) \tag{7}$$

$$subject\ to: \ \mathbf{p} = [p_1, \ldots, p_n] \in \mathcal{C}_t$$

Так как эластичность γ_i^* – отрицательна, то эта задача является задачей выпуклой оптимизации. Соответственно эту задачу можно решить при помощи стандартных методов оптимизации [3]. Мы назовем задачу оптимизации (7) задачей оптимизации MAX-REV, посколько она пытается максимизировать доход корзины при некоторых ограничениях. Соответственно, когда MAX-REV используется вместе с методами пассивной оценки эластичности (например, метод наименьших квадратов), то мы называем этот алгоритм MAX-REV-PASSIVE.

Algorithm 1 Архитектура алгоритма динамического ценообразования.

input Множество товаров $\mathcal{B}$, период времени T, в течении которого хотим максимизировать прибыль.

For $t = 1, \ldots, T$:

- $\forall i \in \mathcal{B}$ рассчитываем базовый прогноз спроса $f_{i,t}$;
- $\forall i \in \mathcal{B}$ рассчитываем эластичность γ_i^*;
- решаем оптимизационную задачу (7), получаем цены $\mathbf{p}_t$;
- применяем полученные цены, получаем реальную прибыль R_t.

End for

4 Решение оптимизационной задачи в условиях неопределенности

Пассивный алгоритм динамического ценообразования MAX-REV-PASSIVE пассивно оценивает прогнозы спроса и эластичность, впоследствии используя их в оптимизационной задаче MAX-REV, максимизирующей суммарную прибыль. Пассивные алгоритмы динамического ценообразования являются алгоритмами неполного обучения и не обеспечивают максимальную прибыль в долгосрочной перспективе. На самом деле оценить прогнозы спроса и эластичность очень непросто. Если цена не меняется часто (что характерно большинству задач ценообразования), то становится очень трудно оценить эластичность цены, что, в свою очередь, приводит к еще меньшему количеству изменений цен и впоследствии неоптимальным ценам.

Один из подходов, учитывающий неточности в эластичности, заключается в моделировании оптимизационной задачи MAX-REV как проблемы робастной оптимизации [1]. Параметр эластичности рассматривается как неопределенная величина, и мы пытаемся максимизировать доход даже при самом плохом значении эластичности. Тогда проблема MAX-REV выглядит следующим образом:

$$\mathbf{p}_t = \arg\max_{\mathbf{p}} \min_{\gamma^* \in \mathcal{G}} \frac{p_{i,t}^2 f_{i,t} \gamma_i^*}{p_{i,t-1}} + p_{i,t}(f_{i,t} - f_{i,t}\gamma_i^*), \tag{8}$$

$$subject\ to: \ \mathbf{p} = [p_1, \ldots, p_n] \in \mathcal{C}_t$$

где $\mathcal{G} \subset \mathbb{R}^{\mathcal{B}}$ – множество всевозможных значений эластичности. Такая формулировка заменяет точечную оценку параметра эластичности γ^*. Но у такого подхода есть свои недостатки: обычно непонятно, каким образом учитывать обратную связь при обновлении множества $\mathcal{G}$.

Альтернативный подход состоит в рассмотрении задачи максимизации прибыли MAX-REV как задачи оптимизации в условиях неопределенности. Таким образом, мы имеем возможность обновлять параметры, используя обратную связь (новую информацию).

Принятие решений в условиях неопределенности – это область задач, в которых решается, как принимать решение для максимизации некоторой величины, когда существует неопределенность в некоторых переменных целевого функционала. У этих задач огромный спектр применения: онлайн-рекомендации, планирования маршрутов, интернет-реклама. Решение таких задач подразумевает некие исследования, в ходу которых мы принимаем возможно неоптимальные решения с целью улучшения наших оценок. Таким образом, принятие неоптимальных решений помогает собрать больше разнообразной информации, но оно также мешает оптимизации. Поэтому перед нами появляется проблема, известная как компромисс между исследованием и эксплуатацией (exploration-exploitation trade-off).

Самая популярная модель для решения такой проблемы – многорукий бандит, которая выглядит следующим образом: есть k ручек, в каждый момент времени агент может дергать за выбранную им ручку и получать выигрыш, который генерируется путем сэмплирования из некоего (агенту неизвестного) распределения, которое зависит от выбранной ручки. Таким образом при выборе конкретной ручки a агент не получает никакой информации об остальных ручках. Цель агента – получить максимальный выигрыш за T раундов.

Простейшие алгоритмы решения этой задачи предлагают дернуть каждую ручку один (или несколько) раз, затем выбрать ручку, средний выигрыш которой получится максимальным и продолжить дергать только за нее. Очевидно, что такое жадное решение не является оптимальным.

Другое простое решение называется ε-жадным алгоритмом. Суть этого алгоритма состоит в том, что агент выбирает ручку с максимальным средним выигрышем на данный момент с вероятностью $1 - \varepsilon$, и случайную ручку с вероятностью ε. Таким образом первый случай соответствует эксплуатации, а второй – исследованию. В случае, когда $\varepsilon = 0$, этот алгоритм сводится к обычному жадному алгоритму, в случае, когда $\varepsilon = 1$, алгоритм только исследует ручки, по сути выбирая на каждом шаге случайную ручку. Однако при грамотном выборе параметра ε можно получить неплохой результат.

Производительность таких алгоритмов измеряется с точки зрения потерь (regret). Потери определяются как разница между прибылью агента и прибылью некоего оракула, который знает какая ручка оптимальна и дергает только за нее, то есть всегда действует оптимальным обазом. Алгоритм называется согласованным, если средние потери алгоритма стремятся к 0. Хотя ε-жадные алгоритмы и могут показывать хороший результат при грамотной выборе параметра ε, есть алгоритмы получше, два из которых мы далее рассмотрим.

Обобщением проблемы многорукого бандита является проблема оптимизации бандита, в которой решают задачу оптимизации с некими неизвестными параметрами. К примеру, задача линейной оптимизации выглядит следующим образом:

$$\max_{\mathbf{x} \in \mathcal{C}} \mathbf{x}^T \theta^*, \tag{9}$$

где $\mathcal{C}$ – известное множество, θ^* – неизвестный вектор. Игра также состоит из раундов, в каждом из которых агент выбирает $\mathbf{x}$, и видит зашумленную оценку целевого функционала $\mathbf{x}^T \theta^*$ [4].

Алгоритм UCB (upper confidence bound), дословно алгоритм верхней доверительной границы действует следующим образом: он высчитывает верхние доверительные границы для каждой из ручек и выбирает ту из них, которая имеет максимальную верхнюю доверительную границу. Алгоритм UCB является оптимальным алгоритмом, но в некоторых задачах точное вычисление верхних доверительных границ может быть затруднено.

Альтернативным вариантом алгоритму UCB является алгоритм томсоновского сэмплирования (Thompson Sampling) [6]. Алгоритм томсоновского сэмплирования является байесовским алгоритмом, который задает априорное распределение, а далее на каждом шаге это распределение обновляет при помощи правил Байеса. По сути алгоритм томсоновского сэмплирования выбирает ручку с такой вероятностью, с которой эта ручка является оптимальной. Этот алгоритм зачастую проще реализовывать, чем алгоритм UCB. Это связано с тем, что в алгоритме томсоновского сэмплирования нет явного построения доверительных интервалов для неопределенных параметров. Именно поэтому алгоритм томсоновского сэмплирования зачастую выбирают для решения такого рода задач.

5 Динамическое ценообразование как задача о многоруких бандитах и томсоновское сэмплирование

Для того, чтобы применить томсоновское сэмплирование к задаче MAX-REV, необходимо ее переформулировать в терминах многоруких бандитов. "Бандитская"формулировка моделирует неопределенность истинных значений эластичности, но предполагает, что базовый спрос мы знаем. То есть как ранее говорилось, мы будем считать, что истинный базовый спрос мы получаем с помощью некоего оракула. Мы так делаем потому что прогноз спроса оценить гораздо легче, по сравнению с эластичностью цен, поскольку в прогнозе спроса нет проблемы разреженности данных. Таким образом, в Алгоритме 2 показано, каким образом мы это осуществляем.

Algorithm 2 Задача MAX-REV как задача о многоруких бандитах.

0: **input** Множество товаров $\mathcal{B}$, период времени T, в течении которого хотим максимизировать прибыль.
Оракулом выбирается вектор $\boldsymbol{\gamma}^* \preceq 0$, агент его не знает.
For $t = 1, \dots, T$:
- агент "играет"вектор $\mathbf{p}_t$ для множества товаров $\mathcal{B}$;
- агент получает стохастический выигрыш R_t, такой, что $\mathtt{E}[R_t] = Rev_t(\mathbf{p}_t, \gamma^*, f_{1,t}, \dots, f_{n,t})$, выражение для Rev_t в (12).

End for

Томсоновское сэмплирование начинается с априорного распределения для неизвестных параметров, в нашем случае это параметр эластичности γ^*. Каждый раунд мы получаем реальный доход от множества товаров R_t. Тогда вероятностная модель задается следующими уравнениями:

$$\Pi_0(\boldsymbol{\gamma}^*) = N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{10}$$

$$l(R_t; Rev_t, \gamma^*) = N(R_t; Rev_t, \sigma^2) \tag{11}$$

$$Rev_t(\mathbf{p};\boldsymbol{\gamma}^*,\mathbf{f}) = \sum_{i\in\mathcal{B}} \frac{p_i^2 f_{i,t}\gamma_i^*}{p_{i,t-1}} + p_i(f_{i,t} - f_{i,t}\gamma_i^*) \tag{12}$$

$$\Pi_t(\boldsymbol{\gamma}^*) \propto \Pi_{t-1}(\boldsymbol{\gamma}^*) N(R_t; Rev_t, \sigma^2) \tag{13}$$

Выражение (13) – получение апостериорного распределения параметра γ^* с использованием байесовского обновления.
Введем обозначения:

$$\theta_{ti} = \frac{p_i^2 f_{i,t}}{p_{i,t-1}} - p_i f_{i,t} \tag{14}$$

$$\bar{R}_t = \sum_{i\in\mathcal{B}} p_i f_{i,t} \tag{15}$$

Тогда $Rev_t = \sum_{i\in\mathcal{B}} \frac{p_{i,t}^2 f_{i,t}\gamma_i^*}{p_{i,t-1}} - p_{i,t}f_{i,t}\gamma_i^* + p_{i,t}f_{i,t} = \boldsymbol{\gamma}^{*T}\boldsymbol{\theta}_t + \bar{R}_t$.
$N(R_t, Rev_t, \sigma^2) = N(R_t, \boldsymbol{\gamma}^{*T}\boldsymbol{\theta}_t + \bar{R}_t, \sigma^2) \propto$
$\propto \exp\Big(-\frac{1}{\sigma^2}\big(R_t - \bar{R}_t - \boldsymbol{\gamma}_*^T\boldsymbol{\theta}_t\big)^2\Big) \propto \exp\big(-(\boldsymbol{\gamma}_* - \boldsymbol{\beta}_t)^T\mathbf{M}_t^{-1}(\boldsymbol{\gamma}_* - \boldsymbol{\beta}_t)\big)$,
где

$$\mathbf{M_t}^{-1}\boldsymbol{\beta}_t = \frac{R_t - \bar{R}_t}{\sigma^2}\boldsymbol{\theta}_t \Rightarrow \mathbf{M}_t^{-1} = \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2} + \lambda\mathbf{I} \tag{16}$$

$$\Pi_t(\boldsymbol{\gamma}^*) \propto N(\boldsymbol{\gamma}^*; \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}) N(\boldsymbol{\gamma}; \boldsymbol{\beta}_t, \mathbf{M}_t) \propto N(\boldsymbol{\gamma}^*; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \tag{17}$$

где

$$\boldsymbol{\mu}_t = \big(\boldsymbol{\Sigma}_{t-1}^{-1} + \mathbf{M}_t^{-1}\big)^{-1}\big(\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\mu}_{t-1} + \frac{R_t - \bar{R}_t}{\sigma^2}\boldsymbol{\theta}_t\big), \tag{18}$$

$$\boldsymbol{\Sigma}_t = \big(\boldsymbol{\Sigma}_{t-1}^{-1} + \mathbf{M}_t^{-1}\big)^{-1} \tag{19}$$

$\lambda > 0$ – малая положительная константа, которая была добавлена к матрице $\mathbf{M}_t$, чтобы она была обратимой.

Algorithm 3 MAX-REV-TS – активный алгоритм динамического ценообразования.

0: **input** Множество товаров $\mathcal{B}$, период времени T, в течении которого хотим максимизировать прибыль.

• Оракулом выбирается вектор $\boldsymbol{\gamma}^* \preceq 0$, агент его не знает.

• Инициализируется априорное распределение $\Pi_0(\boldsymbol{\gamma}^*)$

For $t = 1, \dots, T$:

• Сэмплируем $\boldsymbol{\gamma}_t \sim \Pi_{t-1}$ до тех пор, пока все компоненты $\boldsymbol{\gamma}_t$ не будут
отрицательны;

• Используя прогноз спроса, получаем $f_{i,t}$ для всех $i \in \mathcal{B}$;

• Решаем оптимизационную задачу $\mathbf{p}_t = \arg\max_{\mathbf{p}} \sum_{i \in \mathcal{B}} \dfrac{p_i^2 f_{i,t} \gamma_i^*}{p_{i,t-1}} +$ $p_i(f_{i,t} - f_{i,t}\gamma_i^*)$;

• Применяем цены $\mathbf{p}_t$, получаем прибыль R_t;

• Изменяем распределение $\Pi_t(\gamma^*)$ по формулам (18), (19).

End for

Выражение (10) задает априорное распределение на вектор эластичностей, выражение (11) – правдоподобие для наблюдаемого дохода. Использование такого априорного распределения приводит к очень простым апостериорным обновлениям. Более того, в нашей реализации TS мы отклоняем неотрицательные вектора эластичности. Можно было бы применить более подходящие распределения как для эластичности, так и для дохода, которые отражают тот факт, что эти параметры принимают только ограниченный набор значений (например, логнормальное распределение может быть более подходящим). Однако с такими распределениями, как правило, трудно получить обновления, и приходится прибегать к методам приближенной выборки [2]. Псевдокод для MAX-REV-TS показан в Алгоритме 3.

Теорема. Пусть $\boldsymbol{\gamma}^*$ – истинный вектор эластичностей. Тогда $||\mathtt{E}(\boldsymbol{\gamma}^* - \boldsymbol{\mu}_t)|| \to 0$, при $t \to +\infty$.

Доказательство.

$$\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2} = \sum_{i=1}^{t}\frac{\boldsymbol{\theta}_i\boldsymbol{\theta}_i^T}{\sigma^2};$$

Рассмотрим такой раунд τ, что матрица $\boldsymbol{\Sigma}_\tau^{-1} = \sum\limits_{i=1}^{\tau}\frac{\boldsymbol{\theta}_i\boldsymbol{\theta}_i^T}{\sigma^2}$ невырождена.

Тогда $\forall\ t > \tau$ выполнено:

$$\mathtt{E}(\boldsymbol{\gamma}^* - \boldsymbol{\mu}_t) = \boldsymbol{\gamma}^* - \mathtt{E}[\boldsymbol{\mu}_t] = \boldsymbol{\gamma}^* - \mathtt{E}\Big[\big(\boldsymbol{\Sigma}_{t-1}^{-1} + \mathbf{M}_t^{-1}\big)^{-1}\Big(\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\mu}_{t-1} + \frac{R_t - \bar{R}_t}{\sigma^2}\boldsymbol{\theta}_t\Big)\Big] =$$

$$= \boldsymbol{\gamma}^* - \big(\boldsymbol{\Sigma}_{t-1}^{-1} + \mathbf{M}_t^{-1}\big)^{-1}\Big(\boldsymbol{\Sigma}_{t-1}^{-1}\mathtt{E}[\boldsymbol{\mu}_{t-1}] + \frac{\boldsymbol{\theta}_t}{\sigma^2}\mathtt{E}[R_t] - \frac{\bar{R}_t\boldsymbol{\theta}_t}{\sigma^2}\Big) =$$

$$= \boldsymbol{\gamma}^* - \Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2} + \lambda\mathbf{I}\Big)^{-1}\Big(\boldsymbol{\Sigma}_{t-1}^{-1}\mathtt{E}[\boldsymbol{\mu}_{t-1}] + \frac{\boldsymbol{\theta}_t}{\sigma^2}(\boldsymbol{\theta}_t^T\boldsymbol{\gamma}^* + \bar{R}_t - \bar{R}_t)\Big) =$$

$$= \boldsymbol{\gamma}^* - \Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\Big)^{-1}\Big(\Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\Big)\boldsymbol{\gamma}^* + \boldsymbol{\Sigma}_{t-1}^{-1}\big(\mathtt{E}[\boldsymbol{\mu}_{t-1}] - \boldsymbol{\gamma}^*\big)\Big) =$$

$$= \boldsymbol{\gamma}^* - \boldsymbol{\gamma}^* - \Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\Big)^{-1}\boldsymbol{\Sigma}_{t-1}^{-1}\big(\mathtt{E}[\boldsymbol{\mu}_{t-1}] - \boldsymbol{\gamma}^*\big)\big) =$$

$$= -\Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\Big)^{-1}\boldsymbol{\Sigma}_{t-1}^{-1}\big(\mathtt{E}[\boldsymbol{\mu}_{t-1}] - \boldsymbol{\gamma}^*\big)\big) = \Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\Big)^{-1}\boldsymbol{\Sigma}_{t-1}^{-1}\big(\boldsymbol{\gamma}^* - \mathtt{E}[\boldsymbol{\mu}_{t-1}]\big)\big).$$

$$\Big(\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\Big)^{-1}\boldsymbol{\Sigma}_{t-1}^{-1} = \Big(\sum_{i=1}^{t}\frac{\boldsymbol{\theta}_i\boldsymbol{\theta}_i^T}{\sigma^2}\Big)^{-1}\sum_{i=1}^{t-1}\frac{\boldsymbol{\theta}_i\boldsymbol{\theta}_i^T}{\sigma^2} = \big(\sum_{i=1}^{t}\boldsymbol{\theta}_i\boldsymbol{\theta}_i^T\big)^{-1}\sum_{i=1}^{t-1}\boldsymbol{\theta}_i\boldsymbol{\theta}_i^T.$$

Обозначим $\mathbf{x}_t = \boldsymbol{\gamma}^* - \mathtt{E}[\boldsymbol{\mu}_t],\ \mathbf{x}_{t-1} = \boldsymbol{\gamma}^* - \mathtt{E}[\boldsymbol{\mu}_{t-1}],\ \mathbf{A}_t = \sum\limits_{i=1}^{t} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^T,\ \mathbf{A}_{t-1} = \sum\limits_{i=1}^{t-1} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^T$.

Тогда $\mathbf{x}_t = \mathbf{A}_t^{-1}\mathbf{A}_{t-1}\mathbf{x}_{t-1} \Rightarrow \mathbf{A}_t\mathbf{x}_t = \mathbf{A}_{t-1}\mathbf{x}_{t-1}$.

$c = ||\mathbf{A}_t\mathbf{x}_t|| \geqslant \lambda_{min}^{(t)}||\mathbf{x}_t||$, где $\lambda_{min}^{(t)}$ – минимальное собственное число матрицы $\mathbf{A}_t$,

c – константа $\Rightarrow$

$$||\mathbf{x}_t|| \leqslant \frac{c}{\lambda_{min}^{(t)}}. \tag{20}$$

$\mathbf{A}_t = \mathbf{A}_{t-1} + \dfrac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}$;

$\forall\ \mathbf{x}:\ ||\mathbf{x}|| = 1 \hookrightarrow\ \mathbf{A}_t\mathbf{x} = \mathbf{A}_{t-1}\mathbf{x} + \dfrac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\mathbf{x}$.

Рассмотрим скалярное произведение $\mathbf{x}^T\mathbf{A}_t\mathbf{x}$:

$\mathbf{x}^T\mathbf{A}_t\mathbf{x} = \mathbf{x}^T\mathbf{A}_{t-1}\mathbf{x} + \mathbf{x}^T\dfrac{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^T}{\sigma^2}\mathbf{x} \geqslant \lambda_{min}^{(t-1)} + \dfrac{(\boldsymbol{\theta}_t^T\mathbf{x})^2}{\sigma^2}\ \Rightarrow$

$||\mathbf{A}_t\mathbf{x}|| \geqslant \mathbf{x}^T\mathbf{A}_t\mathbf{x} \geqslant \lambda_{min}^{(t-1)} \quad \forall\ \mathbf{x}:\ ||\mathbf{x}|| = 1 \Rightarrow$

$$\lambda_{min}^{(t)} \geqslant \lambda_{min}^{(t-1)}, \tag{21}$$

причем равенство достигается только в случае, когда вектор $\boldsymbol{\theta}_t$ перпендикулярен $\mathbf{x}_{min}^{(t-1)}$ – собственному вектору матрицы $\mathbf{A}_{t-1}$, соответствующему собственному числу $\lambda_{min}^{(t-1)}$.

Тогда в силу (20), (21) получаем, что $\lambda_{min}^{(t)} \to +\infty$ при $t \to +\infty\ \Rightarrow$
$||\mathbf{x}_t|| = ||\boldsymbol{\gamma}^* - \mathtt{E}[\boldsymbol{\mu}_t]|| \to 0$ при $t \to +\infty$.

6 Вычислительный эксперимент

Мы использовали синтетические данные для проверки алгоритма. Для этого мы создаем 100 товаров и моделируем поведение 1000 покупателей. Делаем мы это следующим образом:

- Для каждого покупателя мы создаем вектор размера 100 и заполняем его числами, полученными из распределения $N(1, 0.3)$. Эти числа означают максимальную цену, по которой бы клиент купил определенный товар.

- Когда алгоритм в очередной раз выдает нам оптимальные цены, нам нужно их применить к покупателям и вычислить доход. Для начала мы создаем список доступных товаров для каждого из покупателей (список тех товаров, цена на которые не выше собственного ограничения покупателя). Затем, если список доступных товаров пуст, то это означает, что этот клиент ничего не купит, если же список не пуст, то покупатель покупает тот товар, у которого максимально его собственное ограничение цены (выбирает самый ценный для себя из доступных).

- Подсчитывается сумма товаров, которые купили бы клиенты.

Наш алгоритм работал при $T = 100$, после чего мы построили график зависимости дохода от номер итерации. На рисунке 1 видно, что доход флуктуирует, но видна явная тенденция роста, что не может не радовать.

Конечно тестирование на реальных данных дает гораздо более правдоподобный результат. Пока, к сожалению, нам не предоставилось возможности протестировать наш алгоритм на каки-либо реальных покупателях.

Поэтому пока мы моделировали поведение клиентов, чтобы показать то, что алгоритм действительно неплохо выполняет поставленную перед ним задачу максимизации дохода.

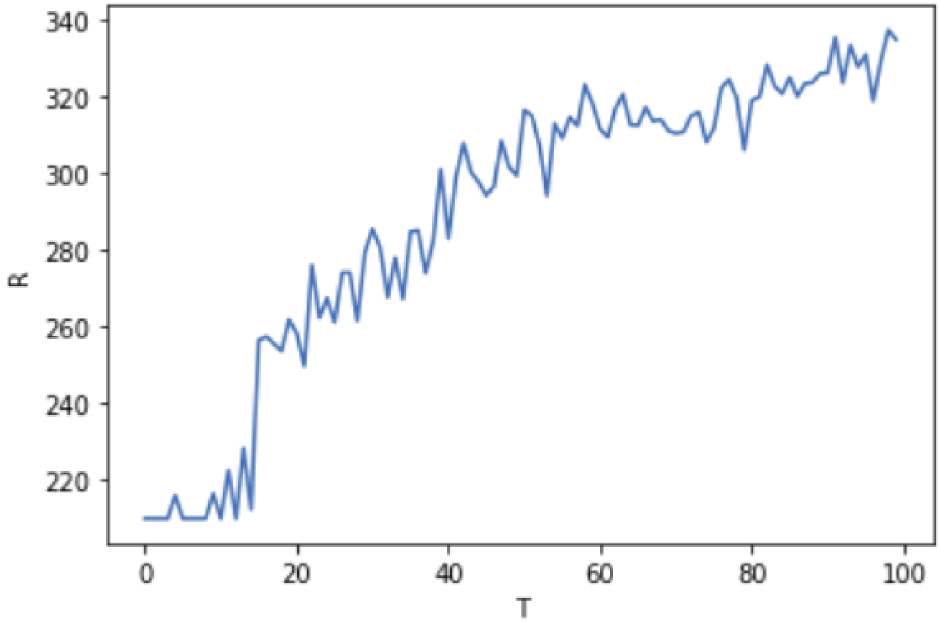

Рис. 1: График зависимости дохода от времени.

7 Заключение

В данной работе рассматривалась задача динамического образования как оптимизационная задача с целью максимизации прибыли.

Было предложено моделирование функции спроса, выражение через нее и другие параметры функции дохода. Был предложен алгоритм, который максимизировал эту функцию дохода, используя для оценки неизвестного параметра эластичности томсоновское сэмплирование.

Была доказана теорема о сходимости этого алгоритма.

Проведено экспериментальное исследование работы алгоритма на синтетических данных.

Список литературы

[1] A. Ben-Tal, L.E. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[3] S. Boyd, S.P. Boyd, L. Vandenberghe, and Cambridge University Press. *Convex Optimization*. Number ч. 1 in Berichte über verteilte messysteme. Cambridge University Press, 2004.

[4] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721, 2012.

[5] J. Michael Harrison, N. Bora Keskin, and Assaf Zeevi. Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science*, 58(3):570–586, 2012.

[6] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, and Ian Osband. A tutorial on thompson sampling. *CoRR*, abs/1707.02038, 2017.

[7] Parikshit Shah, Ming Yang, Sachidanand Alle, Adwait Ratnaparkhi, Ben Shahshahani, and Rohit Chandra. A practical exploration system for search advertising. page 1625–1631, 2017.