

Структурированное ценообразование для взаимозаменяемых товаров

Н. К. Савельев¹, Ю. В. Дорн², В. В. Стрижов³

Аннотация: В работе исследуется задача поиска оптимальных, с точки зрения максимизации дохода, цен для группы взаимозаменяемых товаров (субститутов). Предлагается способ построения мультиагентной системы на основе исторических данных о покупках для моделирования распределения персональных уровней (private value) цен агентов. Преимуществами данной системы по сравнению с системами, моделирующими функцию спроса, является гибкость, возможность персонального ценообразования и более точное моделирование структуры связей между товарами. Разработан алгоритм дискретной оптимизации цен для произвольной выборки агентов. Для дальнейшего поиска оптимальных цен для всего множества агентов применяются алгоритмы обучения с подкреплением.

Ключевые слова: ценообразование; взаимозаменяемые товары; мультиагентная система; дискретная оптимизация; обучение с подкреплением.

DOI: 00.00000/000000000000000

1 Введение

В работе рассматривается задача поиска оптимальных, с точки зрения максимизации дохода цен, для группы взаимозаменяемых товаров (субститутов). Для субститутов существует прямое соотношение между ценой на один из них и спросом на другой, то есть снижение (повышение) цены одного товара вызывает уменьшение (увеличение) спроса на другой. Поэтому алгоритмы поиска оптимальной цены на один отдельный товар, например алгоритм онлайн обучения, описанный в [1] не могут быть

¹Московский физико-технический институт, savelev.nk@phystech.edu

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

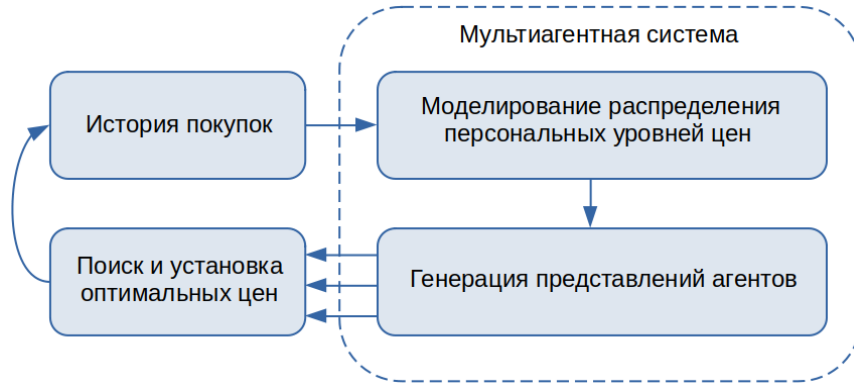
³Вычислительный центр имени А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, Московский физико-технический институт, strijov@phystech.edu

использованны для решения нашей задачи, либо же они должны быть доработаны и адаптированы для учета зависимостей между товарами.

Один из способов учета взаимосвязей между ценами на одни товары и спросом на другие - это моделирование совместной функции спроса, аргументом этой функции является вектор цен на данную группу товаров, а значением будет вектор спроса на соответствующие товары. Так, например, в работе [2] функция спроса моделируется с помощью оценки вектора эластичностей цен для группы товаров.

В данной работе предлагается уйти от моделирования совместной функции спроса и моделировать поведение покупателей. Каждый покупатель описывается вектором персональных уровней (private value) цен на некоторую группу товаров. Персональный уровень цены покупателя - это максимальное значение цены товара, при котором возможна покупка этого товара покупателем. Такой подход позволяет при произвольных ценах оценивать, какой из товаров предпочтет покупатель. То есть становится возможно моделирование выбора товара покупателем в зависимости от набора цен, установленных на группу товаров.

Другими словами, создается мультиагентная система, в которой каждый агент (покупатель) представляется одним вектором - вектором оценок персональных уровней цен, и на основе этого представления моделируется поведение агента, то есть его выбор товара при заданном векторе цен на всю группу товаров.



2 Постановка задачи

Дано:

- \mathcal{B} ($|\mathcal{B}| = k$) - множество взаимозаменяемых товаров;
- \mathcal{C} ($|\mathcal{C}| = n$) - множество клиентов;
- $\{\mathcal{J}_t\}_{t=1}^T$ - история заказов, где $\mathcal{J}_t = \{(d_{i,j,t}, p_{i,t})\}_{i \in \mathcal{B}, j \in \mathcal{C}}$, $d_{i,j,t}$ - количество товара i купленное пользователем j в момент времени t по цене $p_{i,t}$.

Фактический спрос $d_{i,j,t}$ при цене $p_{i,t}$ можно считать i -й компонентой реализации случайной многозначной функции спроса $d_j(p_t)$ клиента j , которая нам неизвестна.

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

Описанная задача делится на несколько подзадач.

Первая из них это задача построения мультиагентной системы на основе исторических данных о покупках товаров клиентами. Для построения такой системы требуется разработать способ моделирования персональных уровней цен агентов.

Вторая подзадача - это задача нахождения оптимального вектора цен для набора представлений агентов. Пусть $Q \in \mathbb{R}_+^{n \times k}$ - матрица представлений n агентов.

Введем целевую функцию дохода $Rev : \mathbb{R}^k \rightarrow \mathbb{R}$,

$$Rev(p) = \sum_{j=1}^k p_j \sum_{i=1}^n \mathcal{I}(Q, p, i, j), \quad (1)$$

$$\mathcal{I}(Q, p, i, j) = \begin{cases} 1, & \text{if } \mathcal{S} := \{s : q_{is} \geq p_s\} \neq \emptyset \text{ and } j = \operatorname{argmax}_{s \in \mathcal{S}} q_{is} \\ 0, & \text{otherwise} \end{cases}.$$

$\mathcal{I}(Q, p, i, j)$ - это индикаторная функция, она равна 1 в случае, когда i -й агент выбирает для покупки j -й товар по фиксированному вектору цен p .

Тогда искомым вектор цен является решением оптимизационной задачи

$$Rev(p) \rightarrow \max_p. \quad (2)$$

3 Построение мультиагентной системы

Большая часть клиентов имеют историю покупок только небольшой части товаров из исследуемой группы взаимозаменяемых товаров. Поэтому предлагается построить модель не просто моделирующую распределение представлений агентов, а еще и учитывающую информацию о покупках заданного агента.

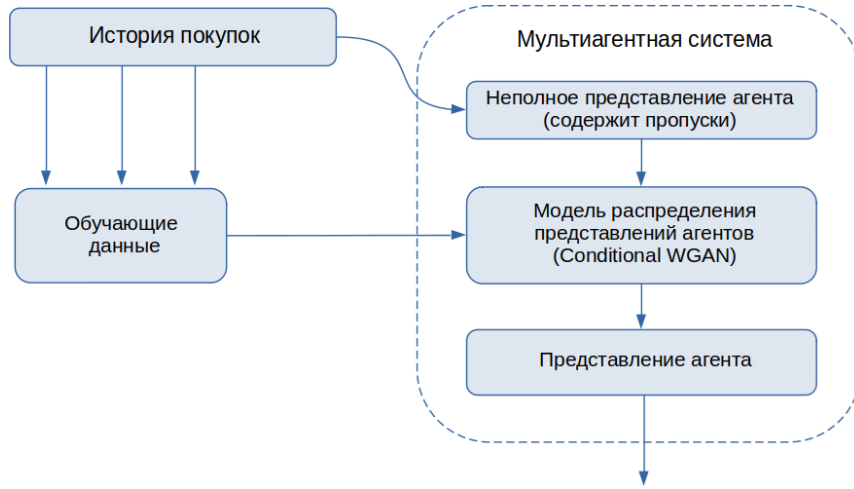
В качестве такой модели была выбрана модель Conditional WGAN (Wasserstein Generative Adversarial Network). Необходимость использования расстояния Вассерштейна обоснована тем, что распределение представлений агентов мультимодально, причем моды выражены сильно. Поэтому во время обучения распределения в генераторе и дискриминаторе могут слабо пересекаться, и для того чтобы ускорить или вообще сделать возможным обучение GAN, необходимо использовать расстояние Вассерштейна. В дополнение к этому, используется именно условный (conditional) GAN для того чтобы учесть информацию из истории покупок заданного агента.

В качестве условного вектора на вход Conditional GAN подается вектор неполного представления агента, построенный на основе истории покупок. Неполное представ-

ление агента - вектор $\hat{q} \in \mathbb{R}^k$ с компонентами:

$$\hat{q}_i = \max\{p_{i,t} : (d_{i,t}, p_{i,t}) \in \{\mathcal{J}_t\}_{t=1}^T\}, \quad \max\{\emptyset\} = Null.$$

Для обучения Conditional WGAN использовались неполные представления агентов, полученные в различные моменты времени истории покупок. Тем самым сеть была обучена дополнять недостающие персональные уровни цен в неполном представлении агента.



4 Поиск оптимального вектора цен

Пусть $Q \in \mathbb{R}_+^{n \times k}$ - матрица представлений n агентов.

Введем целевую функцию дохода $Rev : \mathbb{R}^k \rightarrow \mathbb{R}$,

$$Rev(p) = \sum_{j=1}^k p_j \sum_{i=1}^n \mathcal{I}(Q, p, i, j), \quad (3)$$

$$\mathcal{I}(Q, p, i, j) = \begin{cases} 1, & \text{if } \mathcal{S} := \{s : q_{is} \geq p_s\} \neq \emptyset \text{ and } j = \operatorname{argmax}_{s \in \mathcal{S}} q_{is} \\ 0, & \text{otherwise} \end{cases}.$$

$\mathcal{I}(Q, p, i, j)$ - это индикаторная функция, она равна 1 в случае, когда i -й агент выбирает для покупки j -й товар по фиксированному вектору цен p .

Тогда искомый вектор цен является решением оптимизационной задачи

$$Rev(p) \rightarrow \max_p. \quad (4)$$

Утверждение. Для любой матрицы $Q \in \mathbb{R}_+^{n \times k}$, найдется такой набор индексов i_1, \dots, i_k , что вектор $(q_{i_1 1}, \dots, q_{i_k k})^T$ является решением оптимизационной задачи (2).

Доказательство. Введем множества $Q_j = \{q_{ij}\}_{i=1}^n$, $j = 1, \dots, k$ и множество $P = \{p \in \mathbb{R}^k : \min Q_j \leq p_j \leq \max Q_j, j = 1, \dots, k\}$. Тогда $\sup F(p) = \sup_{p \in P} F(p)$.

Рассмотрим произвольный вектор $x \in P$. Постором для него вектор $y : y_j = \min_{q \in Q_j, q \geq x_j} q$.

Тогда $F(x) \leq F(y)$.

Следовательно, можно искать максимум $F(p)$ на конечном множестве точек вида $(q_{i_1 1}, \dots, q_{i_k k})^T$.

Пусть p^* - искомое решение оптимизационной задачи (2). Рассмотрим процедуру, позволяющую значительно уменьшить множество точек, внутри которого находится p^* .

Процедура clean:

1. Упорядочим элементы матрицы Q по убыванию.
2. Рассмотрим первый элемент получившегося упорядоченного набора - $q_{i_1 j_1}$.
 $p_{j_1}^* \leq q_{i_1 j_1} \Rightarrow I(Q, p, i_1, j) = 0$, $j \neq j_1$ ($q_{i_1 j} \leq q_{i_1 j_1}$) \Rightarrow можно убрать из рассмотрения все элементы вида $q_{i_1 j} : q_{i_1 j} \leq q_{i_1 j_1}$.
3. Рассматриваем следующий элемент обновленного упорядоченного набора, имеющий другой второй индекс - $q_{i_2 j_2}$, первый индекс этого элемента тоже будет другой, так как иначе он был бы удален на предыдущем шаге.
 $p_{j_2}^* \leq q_{i_2 j_2} \Rightarrow I(Q, p, i_2, j) = 0$, $q_{i_2 j} \leq q_{i_2 j_2} \Rightarrow$ можно убрать из рассмотрения все элементы вида $q_{i_2 j} : q_{i_2 j} \leq q_{i_2 j_2}$.
4. Аналогично повторяем шаг 3 для остальных индексов.

Тем самым была доказана следующая теорема.

Теорема. Пусть $Q \in \mathbb{R}_+^{n \times k}$, $n \geq k$. Тогда функционал (1) достигает максимума на векторе вида $(q_1, \dots, q_k)^T$, где q_j - это элемент j -го столбца матрицы Q , который не был удален процедурой clean.

Следствие (Необходимое условие единственности максимума).

Пусть $Q \in \mathbb{R}_+^{n \times k}$. Для того, чтобы точка максимума функционала (1) была единственна, необходимо выполнение неравенства $n \geq k$.

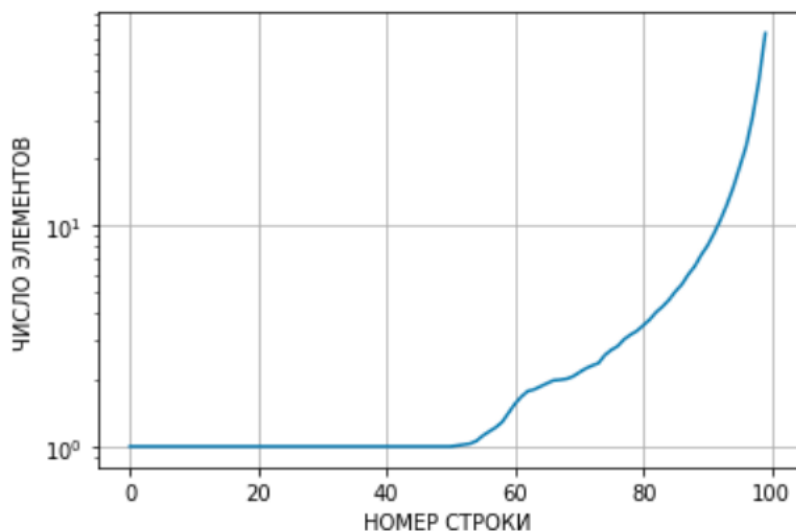
Доказательство. Пусть $n < k$. Применим к матрице Q процедуру clean. На каждом шаге процедуры clean рассматривается элемент матрицы Q с индексами, отличными от индексов элементов, рассмотренных на предыдущих шагах. Следовательно, будет сделано n шагов. Но тогда в матрице Q будет столбец, ни один элемент которого не рассматривался в ходе процедуры clean. А значит, все элементы этого столбца меньше рассмотренных и все они будут удалены. Обозначим индекс этого столбца, как \hat{j} .

Пусть p^* - точка максимума функционала (1). Тогда изменение компоненты $p_{\hat{j}}^*$ не изменяет значения функционала. Следовательно, точка максимума не единственна.

5 Вычислительный эксперимент

Первым этапом практических экспериментов стало исследование числа точек, остающихся в матрице представлений агентов после применения к ней процедуры clean.

В ходе эксперимента было сгенерировано 100 матриц представлений агентов размера 100 на 100. Элементы матриц являлись независимыми равномерно распределенными на отрезке $[0, 1]$ случайными величинами. С каждой матрицей была произведена процедура clean, далее строки матрицы были отсортированы по числу оставшихся элементов. Среднее число оставшихся элементов этих строк представлено на графике.



Были получены довольно интересные результаты. Зависимость числа остающихся в строке элементов от номера строки довольно необычна и интересна.

Продолжается работа над исследованием этой зависимости.

6 Заключение

(Осталось от шаблона, прошу не обращать внимания)

В работе рассматривалась задача оптимизации модели глубокого обучения с наперед заданной сложностью. Итоговый метод заключался в представлении модели глубокого обучения в виде гиперсети. Использовался байесовский подход. Были введены вероятностные предположения о параметрах моделей глубокого обучения. По результатам вычислительного эксперимента можно сделать вывод о том, что модели на основе гиперсети имеют меньшую точность классификации, чем обычные модели. Однако при использовании гиперсети снижаются вычислительные затраты и сохраняются свойства моделей при прореживании.

В дальнейшем планируется исследовать теоретические свойства гиперсетей, а также улучшить предложенные модели для построения сетей глубокого обучения с контролем сложности.

Список литературы

- [1] Sébastien Bubeck, Nikhil R. Devanur, Zhiyi Huang, and Rad Niazadeh. Online auctions and multi-scale online learning. *CoRR*, abs/1705.09700, 2017.
- [2] Ravi Ganti, Matyas Sustik, Quoc Tran, and Brian Seaman. Thompson sampling for dynamic pricing, 2018.

Список литературы

- [1] *Graves A.* Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems (NIPS 2011). P. 2348–2356.
- [2] *Ha D., Dai A. M., Le Q. V.* HyperNetworks // arXiv.org, 2016. arXiv:1609.09106 [cs.LG]. P. 1–29.
- [3] *Kuznetsov M. P., Tokmakova A. A., Strijov V. V.* Analytic and Stochastic Methods of Structure Parameter Estimation // Informatica, 2016. Vol. 27. P. 607–624.
- [4] *Bakhteev O. Yu., Strijov V. V.* Deep Learning Model Selection of Suboptimal Complexity // Automatics and Remote Control, 2018. Vol. 79. P. 1474–1488.
- [5] *Saxena S., Verbeek J.* Convolutional Neural Fabrics // Advances in Neural Information Processing Systems 29: 30th Annual Conference on Neural Information Processing Systems (NIPS 2016). P. 4053–4061.
- [6] *Xie S., Zheng H., Liu C., Lin L.* SNAS: Stochastic Neural Architecture Search // arXiv.org, 2019. arXiv:1812.09926 [cs.LG]. P. 1–17.
- [7] *Wu B., Dai X., Zhang P., Wang Y., Sun F., Wu Y., Tian Y., Vajda P., Jia Y., Keutzer K.* FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — IEEE, 2019. P. 10726–10734.
- [8] *Lorraine J., Duvenaud D.* Stochastic Hyperparameter Optimization through Hypernetworks // arXiv.org, 2018. arXiv:1802.09419 [cs.LG]. P. 1–9.
- [9] *LeCun Y., Cortes C., Burges C.* The MNIST dataset of handwritten digits, 1998. <http://yann.lecun.com/exdb/mnist/index.html>.