ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

СОВРЕМЕННЫЕ ПОДХОДЫ РЕКОМЕНДАЦИИ СЛЕДУЮЩЕЙ КОРЗИНЫ

Выпускная квалификационная работа
студента образовательной программы магистратуры
«Машинное обучение и высоконагруженные системы»

Работу выполнил:

Хыльма М.Д.

Научный руководитель:

Аспирант факультета компьютерных наук

Ананьева М.Е.

Москва 2023 г.

Оглавление

Постановка и описание задачи

Существует много методов для предсказания и рекомендации следующей корзины. Одни алгоритмы используют частотные характеристики и повторяющиеся паттерны, другие исследуют временные компоненты, стараясь учесть такие признаки как сезонность, интервал между покупками и так далее. Нельзя забывать и об алгоритмах, построенных на нейронных сетях, которые также способны показать хорошее качество.

Эффективность моделей и их качество работы сильно зависит от используемого набора данных и метрик. В ряде задач нейронные сети показывают отличные результаты, и качество рекомендации получается лучше, чем при использовании более простых моделей, основанных на частотных признаках. Однако бывают и обратные ситуации. Более того, учитывание временного контекста при составлении следующей корзины пользователя также способно повысить качество работы рекомендательной системы.

Учитывая всё вышесказанное, перед нами встают следующие задачи:

- Изучить современные подходы к рекомендации следующей корзины
- Провести ряд экспериментов и определить текущий SOTA для нашей задачи, сделать анализ полученных результатов
- Используя полученные знания, выдвинуть несколько гипотез по улучшению качества рекомендаций (например, добавление фактора времени между покупками и значения времени текущего контекста получения рекомендаций)
- Сравнить нашу реализацию с предыдущими SOTA алгоритмами, сделать выводы

Ожидаемые результаты

По итогу работы мы ожидаем сравнения имеющихся алгоритмов рекомендаций следующей корзины. Для сравнения будут использоваться несколько датасетов с разными свойствами и несколько метрик с целью получения более устойчивых результатов. По итогам этого сравнения будут сделаны выводы о текущем SOTA алгоритме и о причинах того, почему данный алгоритм дает лучшее качество.

После этого будут выдвинуты несколько предположений о способах улучшения качества рекомендаций. Выдвинутая гипотеза будет реализована и проверена на тех же наборах данных, которые использовались в первой части нашей работы и с тем же набором метрик. Благодаря такому подходу мы сможем наглядно увидеть, насколько успешной оказалась наша идея и какое качество мы получаем.

Обзор литературы

Задача предсказания следующей корзины пользователя является достаточно сложной. В статье [2] авторы отмечают данную задачу как «мультипользовательскую и мультизадачную». Её мультипользовательский аспект заключается в том, что мы не можем утверждать, будто все покупки совершаются одним единственным человеком. Это может быть сразу целая семья, каждый член которой заказывает товары, интересные только лишь ему, и вкусы членов семьи необязательно совпадают. А мультизадачность рекомендации следующей корзины выражается в том, что сами задачи могут быть разными: купить еду, купить предметы гигиены и тп.

Самыми простыми в данной задаче являются модели G-TopFreq, P-TopFreq и GP-TopFreq. Это алгоритмы, которые используют самые популярные товары для формирования следующей корзины. Разница между ними в том, что G-TopFreq рекомендует k самых популярных по всему набору данных, а P-TopFreq рекомендует k самых популярных из истории покупок данного пользователя. GP-TopFreq является комбинацией двух предыдущих алгоритмов: он рекомендует популярных товары из истории пользователя, и потом заполняет оставшиеся слоты популярными товарами среди всего датасета. Стоит отметить, что данный способ пользуется популярностью из-за своей простоты и часто используется как бейзлайн в прочих экспериментах. Существуют, однако, и более сложные модели.

Кроме того, стоит помнить и про временной контекст при составлении следующей корзины пользователя. Помнить об этом нужно по нескольким причинам. Во-первых, в ряде товаров очевидно будет присутствовать сезонность, например, новогодние украшения. Во-вторых, сам пользователь не покупает каждый раз один и тот же набор продуктов. Какие-то товары закупаются впрок, какие-то просто расходуются медленнее. Пользователь вполне может покупать каждый день бутылку молока, но не пакет соли. При этом и тот, и другой товары имеются в его истории покупок и могут быть для него одинаково важны. Нередко возникают ситуации, когда клиент совершил одноразовую покупку, нетипичную для него, и вряд ли планирует в ближайшее время снова этот товар покупать. Все описанные выше случаи нам хотелось бы грамотно учитывать.

Есть ряд интересных статей, которые предлагают способы решения описанных проблем. Учитывать временной эффект пытались авторы в своей работе [1]. Они вводили веса для товаров в зависимости от времени их последнего появления в корзине: чем раньше появлялся товар, тем меньше становился его вес. Это помогает избавиться от рекомендаций товаров, которые пользователю не интересны, однако теряется сезонность. Если товар покупается раз в неделю по субботам, то он будет к пятнице иметь уже маленький вес и не будет рекомендован. В статье [2] предлагается следующее: вводятся специальные параметры *recency aware user-wise popularity* и *recency window,* с помощью которых авторы учитывают временное смещение пользовательских предпочтений. Объединив данные параметры с популярностью товаров и использовав всё это в модели коллаборативной фильтрации, авторы получили модель, способную соревноваться с более сложными моделями и показывать хорошее качество. В работе [3] был использован совсем другой подход. Там был представлен основанный на нейронных сетях алгоритм, который моделировал поведение потребления клиента. С помощью бинарной классификации модель определяет, стоит ли рекомендовать товар в следующей корзине или нет. Интересную идею представили авторы в своей статье [4]. Там тоже пытались учитывать временной контекст для предсказания следующей корзины товаров. Авторы предложили сложную архитектуру, которая состояла из нескольких слоев, использовала взвешенные графы для выучивания взаимосвязей между товарами и комбинировала статистические и динамические атрибуты пользователей.

Как показано в приведенном выше обзоре существующей литературы, задача рекомендации следующей корзины остается актуальной до сих пор. Новые подходы и идеи постоянно появляются, и каждая новая модель демонстрирует лучшее качество, чем предыдущие. Поэтому задача обобщения самых популярных алгоритмов, их оценка и определение актуального на данный момент SOTA алгоритма является полезной и востребованной.

Разведочный анализ данных

TaFeng датасет

Данный набор данных содержит информацию о покупках в продуктовых магазинах в Китае в период с ноября 2000 по февраль 2001. Содержит 817741 строк и 9 признаков. Признак *transaction_dt* отвечает за дату совершения покупки и представляется в виде строки «год/месяц/день». *Customer_id*, *age_group*, *pin_code* содержат информацию о клиенте, а *product_subclass*, *product_id* - о товаре. Оставшиеся признаки *amount*, *asset*, *sales_price* являются числовыми и описывают уже сами покупки клиента.

Проведем разведочный анализ. Первым делом проверим наличие пропусков в данных:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 817741 entries, 0 to 817740
Data columns (total 9 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   TRANSACTION_DT    817741 non-null  object
 1   CUSTOMER_ID       817741 non-null  int64 
 2   AGE_GROUP         795379 non-null  object
 3   PIN_CODE          817741 non-null  object
 4   PRODUCT_SUBCLASS  817741 non-null  int64 
 5   PRODUCT_ID        817741 non-null  int64 
 6   AMOUNT            817741 non-null  int64 
 7   ASSET             817741 non-null  int64 
 8   SALES_PRICE       817741 non-null  int64 
dtypes: int64(6), object(3)
memory usage: 56.1+ MB
```

Видим, что имеется 22362 пропущенных значения в признаке *age_group*, но заполнить данные пропуски никак нельзя. Однако пропуски составляют менее 3% всех наших данных, поэтому наличие пропущенных значений не является критичным.

Помимо пропусков можем еще посмотреть на основные статистики нашего набора данных:

	CUSTOMER_ID	PRODUCT_SUBCLASS	PRODUCT_ID	AMOUNT	ASSET	SALES_PRICE
count	8.177410e+05	817741.000000	8.177410e+05	817741.000000	817741.000000	817741.000000
mean	1.406620e+06	284950.495933	4.461639e+12	1.381781	112.109848	131.875589
std	7.489784e+05	226390.701451	1.690093e+12	2.897473	603.661776	631.057633
min	1.069000e+03	100101.000000	2.000882e+07	1.000000	0.000000	1.000000
25%	9.692220e+05	110106.000000	4.710085e+12	1.000000	35.000000	42.000000
50%	1.587722e+06	130106.000000	4.710421e+12	1.000000	62.000000	76.000000
75%	1.854930e+06	520314.000000	4.712500e+12	1.000000	112.000000	132.000000
max	2.000200e+07	780510.000000	9.789580e+12	1200.000000	432000.000000	444000.000000

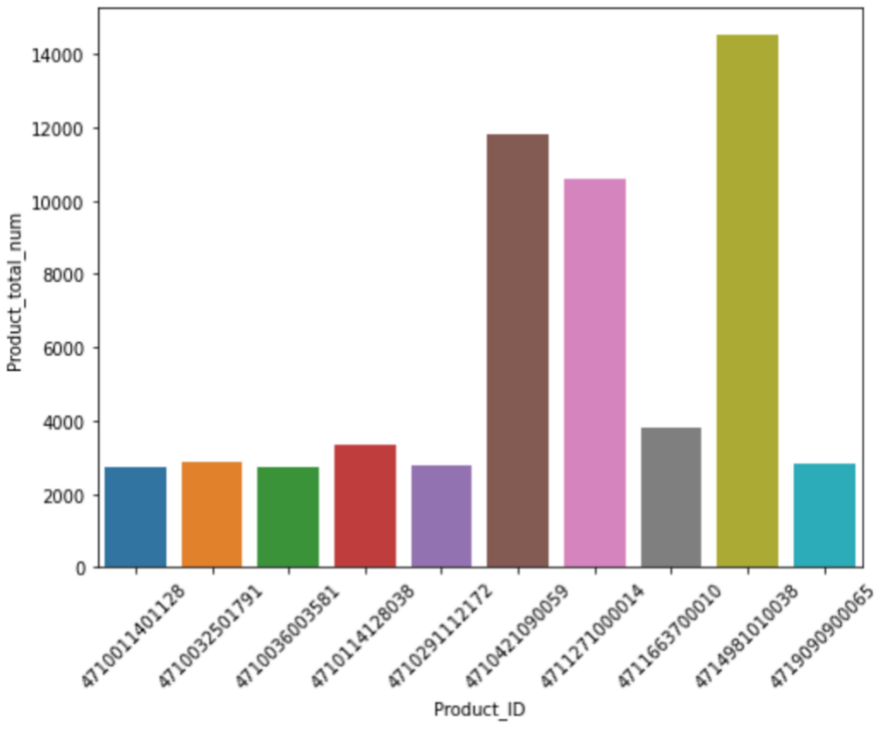
Ошибочных данных не наблюдается, однако заметны некоторые выбросы в данных. Это касается аномально высоких значений параметров amount, asset и sales_price. Если мы посмотрим детальнее на такие объекты, то увидим следующее:

	TRANSACTION_DT	CUSTOMER_ID	AGE_GROUP	PIN_CODE	PRODUCT_SUBCLASS	PRODUCT_ID	AMOUNT	ASSET	SALES_PRICE
271999	12/4/2000	2120829	25-29	Others	100508	4710421090059	1200	14400	14820
288806	12/4/2000	2123851	25-29	Others	100508	4710421090059	1200	14400	14820

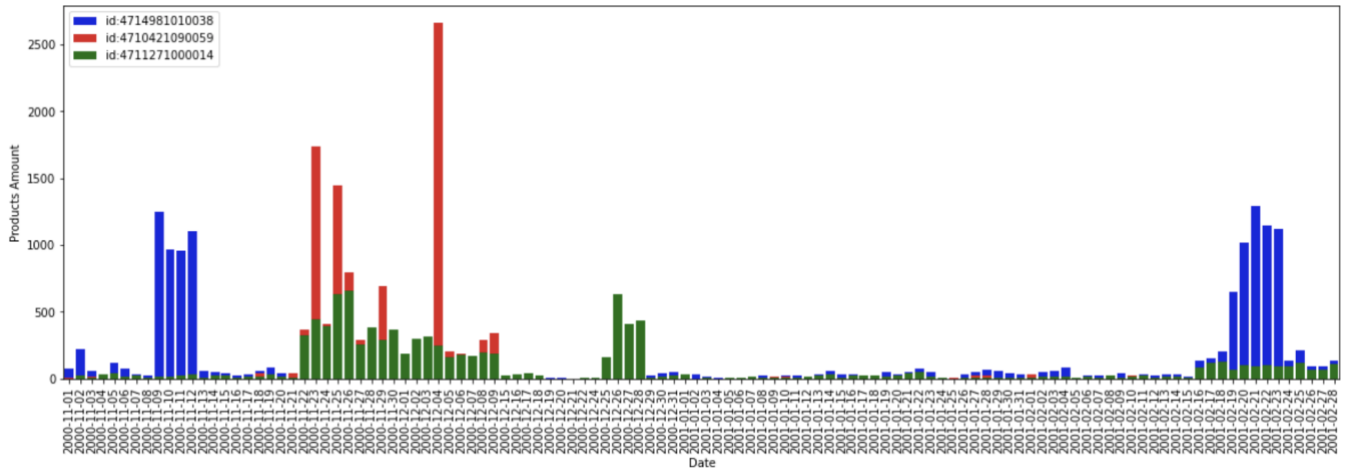
	TRANSACTION_DT	CUSTOMER_ID	AGE_GROUP	PIN_CODE	PRODUCT_SUBCLASS	PRODUCT_ID	AMOUNT	ASSET	SALES_PRICE
731970	2/17/2001	1622362	<25	221	100516	4711588210441	800	432000	444000

Видно, что объектов с аномально высоким значением *amount* два и различаются они только по признаку *customer_id*, а аномальные значения признаков *asset* и *sales_price* соответствуют одному объекту.

Теперь посмотрим на самые популярные товары, а также на их поведение. Сгруппировав по признаку product_id, можем получить общее количество купленного товара. Посмотрим на 10 самых популярных:



Самыми популярными тут являются товары 4714981010038, 4710421090059 и 4711271000014. Если посмотреть на их поведение в течение всего рассматриваемого периода, то мы получим следующий график:



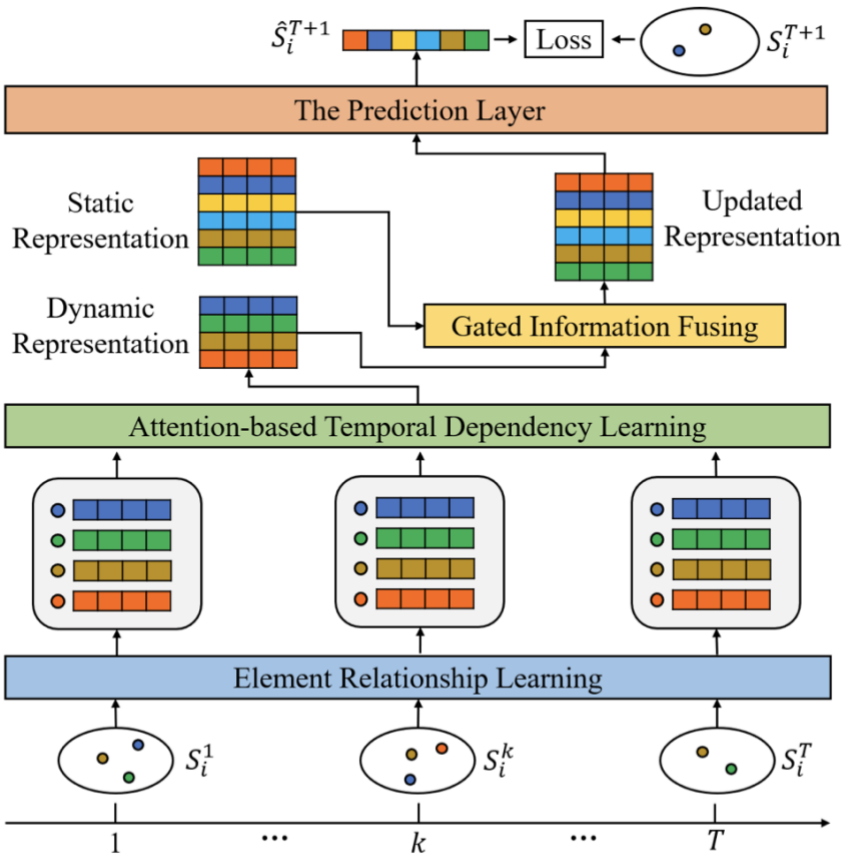
Видим, что *product_id* 4714981010038 чаще всего покупают с 10 по 13 ноября и с 19 по 23 февраля, в то время как основные продажи *product_id* 4710421090059 и 4711271000014 приходятся на конец ноября-начало декабря.

Последнее, что мы можем посмотреть – это средний размер корзины покупателей. Сгруппируем данные по *customer_id* и *transaction_dt*, посчитаем средний размер корзины по каждому клиенту. Воспользуемся методом *describe* из библиотеки *pandas* и получим основные статистики по размеру корзин покупателей:

	Mean_basket_size
count	32266.000000
mean	7.676456
std	6.603849
min	1.000000
25%	3.200000
50%	6.000000
75%	10.000000
max	82.000000

Описание baseline-модели

В качестве бейзлайна мы будем использовать модель DNNTSP. Данная модель основана на использовании нейронных сетей и графов и состоит из трех основных блоков: element relationship learning, attention-based temporal dependency learning и gated information fusing.



Первая компонента модели используется для выучивания взаимосвязей элементов в наборе. Для этого сначала создается взвешенный граф, затем информация распространяется среди элементов динамического графа и, в конце, представление каждого элемента обновляется с помощью полученной информации.

Второй блок модели учит временную зависимость каждого элемента в разных наборах: берется представление элемента и используется механизм внимания, чтобы выучить временную зависимость наборов и элементов в прошлом. После этого историческая информация передается в следующую, последнюю компоненту модели.

Третья компонента объединяет статическое и динамическое представление с помощью gated updating mechanism. Благодаря этой компоненте, объединив всю

имеющуюся информацию, данная модель может более качественно выдавать предсказания следующей корзины.

Описанная выше архитектура показывает хорошее качество и способно превзойти такие алгоритмы, как Personal Top, TIFUKNN, DREAM, Sets2Sets и другие, поэтому мы будем использовать данную модель в качестве бейзлайна. Все наши последующие эксперименты будем сравнивать с данной моделью.

Описание архитектуры проекта

У проекта планируется следующая архитектура:

- Подготовка данных для работы с ними

Мы возьмем четыре популярных датасета для задачи рекомендации следующей корзины. Это будут датасеты TaFeng, DC, TaoBao, TMS. После проведения разведочного анализа мы предобработаем все четыре датасета и подготовим таким образом, чтобы они были пригодны для обучения моделей.

- Обучение baseline-модели и оценка качества работы модели

В качестве бейзлайна мы возьмем DNNTSP, архитектура которой уже описана выше. Мы обучим модель на всех датасетах и посчитаем на каждом датасете значения метрик phr, ndcg, recall. С этими значениями мы и будем в дальнейшем сравнивать другие модели.

- Проведение экспериментов

Планируется провести ряд экспериментов с разными моделями с целью оценить, как хорошо в тех или иных кейсах работают наиболее популярные модели исследуемой нами задачи. Кроме того, планируется эксперимент по внесению изменений в архитектуру DNNTSP, что может привести к улучшению качества её работы.

- Валидация результатов экспериментов, сравнение качества улучшенной модели с остальными

В конце работы, мы сравним качества используемых в данной работе моделей на валидационной части датасетов и сделаем вывод о том, удалось ли нам добиться улучшения качества работы DNNTSP модели после внесения в её архитектуру изменений.

Описание плана экспериментов с моделями

В данной работе план экспериментов состоит из двух частей: эксперименты с моделями и эксперименты с способами улучшения модели DNNTSP.

В первой части экспериментов планируется взять наиболее популярные модели для задачи рекомендации следующей корзины: TopFreq, Personal TopFreq, TIFUKNN, DNNTSP и ETGNN. Будем проверять качество их работы на датасетах TaFeng, DC, TaoBao, TMS с помощью метрик phr, ndcg, recall. Нам будет интересно посмотреть какое качество демонстрируют модели с разными архитектурами на разных по своей структуре наборах данных.

Во второй части экспериментов мы возьмем за основу модель DNNTSP и попытаемся улучшить её качество. Мы попробуем перенести идею, заложенную в модели TiSASRec для задачи рекомендации следующего элемента последовательности, в модель DNNTSP для рекомендации следующей корзины.

Список литературы

1. Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling Personalized Item Frequency Information for Next-basket Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’20), July 25–30, 2020, Virtual Event, China. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401066
2. Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2020. Recency Aware Collaborative Filtering for Next Basket Recommendation. In Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP ’20), July 14–17, 2020, Genoa, Italy. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340631.3394850
3. Mozhdeh Ariannezhad, Sami Jullien, Ming Li, Min Fang, Sebastian Schelter, and Maarten de Rijke. 2022. ReCANet: A Repeat Consumption-Aware Neural Network for Next Basket Recommendation in Grocery Shopping . In Proceed- ings of the 45th International ACM SIGIR Conference on Research and Devel- opment in Information Retrieval (SIGIR ’22), July 11–15, 2022, Madrid, Spain. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531708
4. Le Yu , Leilei Sun , Bowen Du , Chuanren Liu , Hui Xiong , Weifeng Lv . 2020. Predicting Temporal Sets with Deep Neural Networks . In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’20), August 23–27, 2020, Virtual Event, CA, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403152