ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Факультет компьютерных наук

ГРАФОВАЯ РЕКОМЕНДАТЕЛЬНАЯ МОДЕЛЬ ДЛЯ РЕКОМЕНДАЦИИ СЛЕДУЮЩЕЙ КОРЗИНЫ НА ОСНОВЕ МЕХАНИЗМА ВНИМАНИЯ

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.

Аннотация

Задача рекомендации следующей корзины отличается от прогнозирования временных рядов или предсказания следующего элемента последовательности тем, что требуется предсказать не один элемент, а некоторое множество. В виду сложности такой задачи, она всё еще остается актуальной, несмотря на большое количество уже имеющихся решений. Целью данной работы было улучшение качества рекомендации существующей модели путем доработки её архитектуры добавлением в неё новой идеи.

Для дальнейшей работы была выбрана модель DNNTSP, основанная на нейронных сетях. Была выдвинута гипотеза по улучшению её качества рекомендации. Идея заключается в том, чтобы добавить в модель в явном виде информацию о временных интервалах между корзинами.

Обновленный вариант DNNTSP, которая теперь учитывает временные интервалы между корзинами, был обучен на трех популярных наборах данных. После этого мы сравнили метрики качества новой модели с базовой версией DNNTSP.

Полученные результаты демонстрируют повышение качества работы новой модели. Величина прироста качества варьируется в зависимости от датасета, однако можно утверждать, что учитывание временных интервалов позволяет модели повысить качество рекомендации.

Оглавление

Введение *4*

Глава 1 ..... *6*

1.1 Теоретическое введение **6**

1.2 Обзор существующей литературы **6**

1.3 Описание наборов данных **9**

- TaFeng 9
- Dunnhumby-Carbo 9
- TaoBao 9

1.4 Описание используемых метрик качества **10**

1.5 Постановка задачи и ожидаемые результаты **11**

Глава 2 ..... *12*

2.1 Описание архитектуры DNNTSP **12**

2.2 Сравнение качества работы модели DNNTSP с другими популярными моделями **14**

2.3 Описание идеи по улучшению качества работы модели **16**

2.4 Реализация идеи **17**

Глава 3 ..... *19*

3.1 Сравнение качества новой модели с базовой **19**

Заключение *20*

Список литературы *21*

Введение

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. 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.

В представленной работе будут изучаться методы решения задачи рекомендации следующей корзины. Особое внимание будет уделено модели DNNTSP, поскольку именно эту модель мы возьмем за основу и будем пытаться улучшить её качество. В процессе изучения существующих методов решения задачи мы рассчитываем выдвинуть гипотезу по способу повышения качества работы существующей модели и реализовать данную идею. Основной целью данной работы является улучшение качества рекомендации существующей модели путем добавления в её архитектуру новой идеи.

В первой главе приводится теоретическое введение в задачу рекомендации следующей корзины, обзор существующей литературы по данной тематике, описание используемых в данной работе наборов данных и используемых метрик качества, а также формулируется постановка задачи.

Вторая глава целиком посвящена модели DNNTSP, а также идее, с помощью которой мы собираемся повысить качество её работы. Сначала будет дано описание архитектуры DNNTSP, и приведены результаты её работы в сравнении с другими популярными моделями. Затем мы перейдем к описанию идеи, которая может

повысить качество работы. В конце главы будет описана реализация идеи и её интеграция в базовую модель.

В третьей главе приводятся эксперименты с моделями. Мы обучим базовую DNNTSP и её новую версию на трех разных наборах данных и сравним качество их работы с помощью трех основных метрик, которые используются в задачах такого типа.

Глава 1

1.1 Теоретическое введение

Как уже было сказано выше, задача рекомендации следующей корзины заключается в том, чтобы по имеющимся историческим данным о предыдущих покупках пользователя предсказать его следующую корзину, то есть некий набор товаров, который пользователь купит при следующем своем походе в магазин.

Формализуем изучаемую задачу. Пусть имеется набор пользователей $U = \{u_1, u_2, \dots, u_n\}$ и набор товаров $I = \{i_1, i_2, \dots, i_m\}$, где n и m – количество пользователей и товаров соответственно. По каждому пользователю u_i имеется последовательность $S_i = \{s_i^1, s_i^2, \dots, s_i^k\}$, которая представляет собой последовательность корзин данного пользователя, то есть последовательность наборов товаров, купленных пользователем. Каждый набор товаров s_i^j состоит из элементов множества I. Тогда задачу предсказания следующей корзины можно представить в следующем виде:

$S_i^{k+1} = f(s_i^1, s_i^2, \dots, s_i^k)$, где f – это некоторая модель. Модель получает на вход k предыдущих корзин пользователя и $k + 1$ предсказывает корзину.

1.2 Обзор существующей 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.

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 хотелось бы грамотно учитывать.

Есть ряд интересных моделей из семейства KNN моделей, которые предлагают способы решения описанных проблем. Учитывать временной эффект пытались Haoji Hu, Xiangnan He, Jinyang Gao и Zhi-li Zhang в своей модели TIFUKNN [1]. Авторы 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], посвященной модели UP-CF, Guglielmo Faggioli, Mirko Polato и Fabio Aiolli предлагают 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. Недостатком же данных моделей является тот факт, что предсказывать они могут лишь элементы, которые до этого уже встречались в истории пользователя.

Есть также класс моделей, основанных на нейронных сетях. Одними из первых таких моделей являются DREAM [7] и Sets2Sets [8]. В первой модели авторы используют рекуррентные нейронные сети для выучивания признаков, которые отражают особенности и общие паттерны поведения пользователей. А в основу Set2Set положена энкодер-декодер архитектура и слой внимания для использования информации о частоте появления элементов в корзине. Существуют и более современные подходы. В работе [3] Mozhdeh Ariannezhad и Sami Jullien использовали совсем другой подход и представили модель ReCANet. Основанный на нейронных сетях алгоритм ымоделировал поведение потребления клиента. С помощью бинарной классификации модель определяет, стоит ли рекомендовать товар в следующей корзине или нет. Интересная идея представлена в статье [4]. Le Yu и Leilei Su предложили сложную архитектуру, которая состояла из нескольких слоев, использовала взвешенные графы для выучивания взаимосвязей между товарами и комбинировала статические и динамические представления элементов. Данная модель способна превзойти по качеству остальные модели из класса нейронных сетей, что и будет продемонстрированно далее.

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. Мы можем воспользоваться обилием использующихся идей и попытаться улучшить качество работы существующей модели, добавив в неё новую идею.

1.3 Описание наборов данных

Обучать модели и оценивать качество их работы будем на трех популярных датасетах, которые часто используются в задаче рекомендации следующей корзины: TaFeng, Dunnhumby-Carbo и TaoBao. Каждый из датасетов содержит информацию о дате покупки, id купленного товара и id пользователя, купившего данный товар. Мы будем считать товары, купленные в один день, элементами одной корзины. В каждом датасете мы ограничивали набор товаров, оставляя лишь наиболее популярные. Кроме того, пользователи, у которых в истории покупок было менее 4 корзин, отбрасывались.

Датасет	Количество пользователей	Количество товаров	К/П	Т/К	Доля новых товаров в следующей корзине
TaFeng	9841	4935	7.6	5.7	0.77
DC	9000	4000	12.7	7.8	0.54
TaoBao	28337	800	5.8	1.1	0.78

- TaFeng

Набор данных содержит информацию о покупках в продуктовых магазинах в Китае в период с ноября 2000 по февраль 2001

- Dunnhumby-Carbo

Данный датасет содержит информацию о транзакциях домохозяйств, которые регулярно совершают покупки в розничной сети. Для работы были взяты транзакции за первые три месяца.

- TaoBao

Датасет собирается по данным с китайской торговой онлайн платформы и содержит информацию о четырех видах взаимодействия пользователя с товаром: переход на страницу с товаром, добавление товара в корзину, покупка товара и добавления товара в избранное. В данной работе оставили информацию лишь о купленных товарах, все прочие виды взаимодействия были удалены.

1.4 Описание используемых метрик качества

Самыми популярными метриками качества для задачи рекомендации следующей корзины являются следующие три метрики: phr@k, ndcg@k и recall@k. PHR сразу считается по всем пользователям в выборке, а ndcg и recall считаются по каждому пользователю отдельно, а затем усредняются. Распишем чуть подробнее про каждую из метрик.

Personal Hit Ratio отражает долю пользователей, которым мы верно предсказали хотя бы один элемент следующей корзины.

$$\text{PHR@K} = \frac{\sum_i^N \text{I}\left(|S_{pred} \cap S_{true}|\right)}{N}$$

, где I – индикаторная функция, принимающая 0 если пересечение равно пустому множеству, и 1 иначе.

Normalized Discounted Cumulative Gain используется для оценки качества ранжирования элементов.

$$\text{NDCG@K} = \frac{\sum_{k=1}^{K} \frac{\text{I}\left(P_i^k, S_i\right)}{log_2(k+1)}}{\sum_{k=1}^{\min(K,|S_i|)} \frac{1}{log_2(k+1)}}$$

, где P_i – предсказанная следующая корзина

, S_i – истинная следующая корзина

, I – индикаторная функция, принимающая 1 если P_i^k находится в S_i и 0 – иначе.

Recall демонстрирует насколько хорошо модель предсказывает все правильные элементы. Иными словами, сколько элементов из списка релевантных модель действительно предсказала.

$$\text{Recall@}K = \frac{|S_{pred} \cap S_{true}|}{|S_{true}|}$$

Каждая из метрик оценивает то или иное качество модели. Использование трех описанных выше метрик для нас будет достаточно для качественного и разностороннего сравнения работы моделей.

1.5 Постановка задачи и ожидаемые результаты

В данной работе исследуется задача рекомендации следующей корзины пользователя. Основной акцент делается на улучшении качества рекомендации существующей популярной модели DNNTSP, основанной на нейронных сетях. Повысить качество работы указанной модели мы собираемся путем добавления в её архитектуру новой идеи. Учитывая всё вышесказанное, перед нами встают следующие задачи:

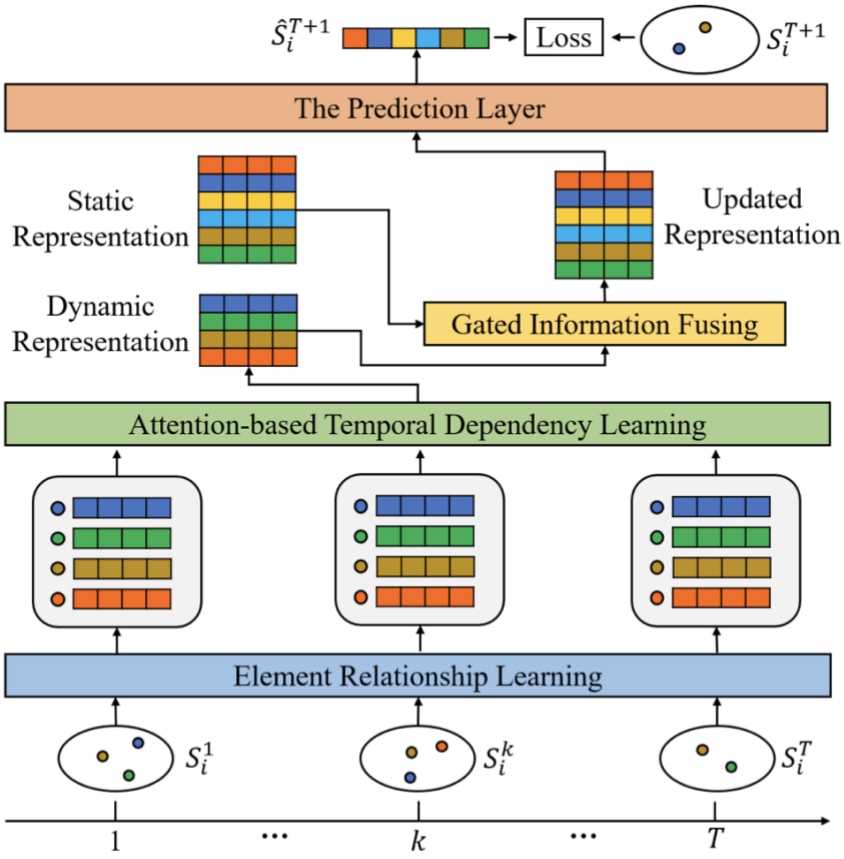
- Изучить современные подходы к рекомендации следующей корзины
- Выдвинуть гипотезу по способу улучшения качества рекомендаций
- Реализовать выдвинутую гипотезу и сравнить качество новой модели с базовой моделью DNNTSP

По итогу работы мы ожидаем получить новую модель для решения задачи рекомендации следующей корзины. Благодаря добавлению новой идеи данная модель будет показывать по метрикам лучшее качество, чем исходная версия модели, которая была взята нами за основу.

Глава 2

2.1 Описание архитектуры DNNTSP

В качестве основы для дальнейшего исследования мы будем использовать модель DNNTSP. Данная модель основана на использовании нейронных сетей и состоит из трех основных блоков: element relationship learning, attention-based temporal dependency learning и gated information fusing. Общая схема модели представлена на Рисунке 1. Рассмотрим каждый блок детальнее.



Первая компонента модели используется для выучивания взаимосвязей элементов в корзинах пользователя. Благодаря использованию взвешенных графов удается выучить взаимодействие элементов не только в рамках одной корзины, но и среди всех корзин пользователя, а также сохранить информацию о взаимодействии элементов. С помощью свертки информация распространяется между элементами в каждом из графов, тем самым преставление каждого элемента обновляется информацией о взаимодействии данного элемента с другими элементами.

В данной работе уже было упомянуто, что разные товары могут встречаться в корзинах в разной частой, что представляет собой дополнительную сложность. Второй блок модели DNNTSP используется как раз для того, чтобы поймать основные частотные паттерны товаров. Attention-based Temporal Dependency Learning получает на вход последовательность из представлений элементов, а затем с помощью механизма внимания учит временную зависимость элементов в наборах.

И статическое представление элементов, которое представляет собой просто эмбеддинг, и динамическое представление, полученное на выходе из предыдущего слоя, затем объединяются вместе в третьей компоненте модели. Благодаря этой компоненте, объединив всю имеющуюся информацию, модель может более качественно выдавать предсказания следующей корзины. Кроме этого, данный блок выполняет еще одну роль. Пользователи могут иметь схожие шаблоны поведения при совершении покупок. Gated Information Fusing блок, помимо объединения статической и динамической информации элементов, также пытается уловить эти скрытые паттерны, ведь их использование также может повысить качество работы рекомендательной модели.

Таким образом, мы можем заметить, что описанная нами модель использует сразу несколько приемов. Взвешенный динамический граф и свертка позволяют сохранить как можно больше изначальной информации о взаимодействие элементов и распространить эту информацию по всем наборам элементов пользователя. Механизм внимания позволяет выучить временные паттерны, а последующее объединение статического и динамического представлений элементов не только способствует улучшению качества представлений элементов, но и позволяет поймать скрытые паттерны поведения, общие для пользователей, и это также позитивно сказывается на качестве рекомендаций.

2.2 Сравнение качества работы модели DNNTSP с другими популярными моделями

Описанная выше архитектура показывает хорошее качество и способно превзойти такие алгоритмы, как Personal Top, TIFUKNN, DREAM, Sets2Sets и другие. Хорошие сравнения были сделаны в статьях [4] и [5]. В статье [4] сами авторы модели DNNTSP сравнили её качество с другими популярными моделями на нескольких датасетах. Результаты их работы отражены в таблице 1.

Datasets	Methods	K=10			K=20			K=30			K=40		
		Recall	**NDCG**	**PHR**	**Recall**	**NDCG**	**PHR**	**Recall**	**NDCG**	**PHR**	**Recall**	**NDCG**	**PHR**
TaFeng	Top	0.1025	0.0974	0.3047	0.1227	0.1033	0.3682	0.1446	0.1104	0.4256	0.1561	0.1140	0.4474
	PersonalTop	0.1214	0.1128	0.3763	0.1675	0.1280	0.4713	0.1882	0.1336	0.5063	0.2022	0.1398	0.5292
	ElementTransfer	0.0613	0.0644	0.2255	0.0721	0.0670	0.2519	0.0765	0.0676	0.2590	0.0799	0.0687	0.2671
	DREAM	0.1174	0.1047	0.3088	0.1489	0.1143	0.3814	0.1719	0.1215	0.4383	0.1885	0.1265	0.4738
	Sets2Sets	0.1427	0.1270	0.4347	0.2109	0.1489	0.5500	0.2503	0.1616	0.6044	0.2787	0.1700	0.6379
	DNNTSP	**0.1752**	**0.1517**	**0.4789**	**0.2391**	**0.1720**	**0.5861**	**0.2719**	**0.1827**	**0.6313**	**0.2958**	**0.1903**	**0.6607**
DC	Top	0.1618	0.0880	0.2274	0.2475	0.1116	0.3289	0.3204	0.1288	0.4143	0.3940	0.1448	0.4997
	PersonalTop	0.4104	0.3174	0.5031	0.4293	0.3270	0.5258	0.4499	0.3318	0.5496	0.4747	0.3332	0.5785
	ElementTransfer	0.1930	0.1734	0.2546	0.2280	0.1816	0.3017	0.2589	0.1929	0.3417	0.2872	0.1955	0.3783
	DREAM	0.2857	0.1947	0.3705	0.3972	0.2260	0.4964	0.4588	0.2407	0.5613	0.5129	0.2524	0.6184
	Sets2Sets	0.4488	0.3136	0.5458	0.5143	0.3319	0.6162	0.5499	0.3405	0.6517	0.6017	0.3516	0.7005
	DNNTSP	**0.4615**	**0.3260**	**0.5624**	**0.5350**	**0.3464**	**0.6339**	**0.5839**	**0.3578**	**0.6833**	**0.6239**	**0.3665**	**0.7205**
TaoBao	Top	0.1567	0.0784	0.1613	0.2494	0.1019	0.2545	0.3166	0.1164	0.3220	0.3679	0.1264	0.3745
	PersonalTop	0.2190	0.1535	0.2230	0.2260	0.1554	0.2306	0.2354	0.1575	0.2402	0.2433	0.1590	0.2484
	ElementTransfer	0.1190	0.1153	0.1217	0.1253	0.1166	0.1284	0.1389	0.1197	0.1427	0.1476	0.1214	0.1516
	DREAM	0.2431	0.1406	0.2491	0.3416	0.1657	0.3483	0.4060	0.1796	0.4129	0.4532	0.1889	0.4606
	Sets2Sets	0.2811	0.1495	0.2868	0.3649	0.1710	0.3713	0.4267	0.1842	0.4327	0.4672	0.1922	0.4739
	DNNTSP	**0.3035**	**0.1841**	**0.3095**	**0.3811**	**0.2039**	**0.3873**	**0.4347**	**0.2154**	**0.4406**	**0.4776**	**0.2238**	**0.4843**
TMS	Top	0.2627	0.1627	0.4619	0.3902	0.2017	0.6243	0.4869	0.2269	0.7222	0.5605	0.2448	0.8007
	PersonalTop	0.4508	0.3464	0.6440	0.5274	0.3721	0.7146	0.5453	0.3765	0.7339	0.5495	0.3771	0.7374
	ElementTransfer	0.3292	0.2984	0.4752	0.3385	0.3038	0.4828	0.3410	0.3034	0.4863	0.3423	0.3036	0.4889
	DREAM	0.3893	0.3039	0.6090	0.4962	0.3379	0.7279	0.5677	0.3570	0.794	0.6155	0.3690	0.8315
	Sets2Sets	**0.4748**	**0.3782**	**0.6933**	0.5601	**0.4061**	0.7594	0.6145	**0.4204**	0.8131	0.6627	**0.4321**	0.8570
	DNNTSP	0.4693	0.3473	0.6825	**0.5826**	0.3839	**0.7880**	**0.6440**	0.4000	**0.8439**	**0.6840**	0.4097	**0.8748**

Таблица 1.

Для сравнения качества использовались метрики recall, NDCG, PHR для topK равному 10, 20, 30 и 40. Жирным шрифтом выделены лучшие значения метрик. Как можно заметить, в трех датасетах из четырех модель DNNTSP демонстрирует значимое улучшение качества.

Похожие эксперименты проделали в статье [5]. В данной статье авторы исследовали существующие модели для решения задачи рекомендации следующей корзины с целью установления текущего SOTA метода. Авторы использовали те же метрики качества, однако список изучаемых моделей значительно шире. Их результаты представлены в Таблице 2.

Size		10			20		
Dataset	Methods	Recall	NDCG	PHR	Recall	NDCG	PHR
TaFeng	G-TopFreq	0.0831 (0.0018)	0.0864 (0.0017)	0.2498 (0.0024)	0.1114 (0.0029)	0.0961 (0.0018)	0.3311 (0.0006)
	P-TopFreq	0.1069 (0.0023)	0.0955 (0.0019)	0.3473 (0.0033)	0.1395 (0.0026)	0.1096 (0.0019)	0.4329 (0.0038)
	GP-TopFreq	0.1211 (0.0031)	0.1015 (0.0023)	0.3691 (0.0043)	0.1693 (0.0031)	0.1208 (0.0022)	0.4834 (0.0040)
	UP-CF@r	0.1249 (0.0027)	0.1104 (0.0019)	0.3983 (0.0035)	0.1694 (0.0034)	0.1280 (0.0021)	0.4877 (0.0048)
	TIFUKNN	0.1251 (0.0033)	0.1016 (0.0014)	0.3852 (0.0029)	0.1817 (0.0037)	0.1232 (0.0016)	0.5043 (0.0035)
	Dream	0.1134 (0.0023)	0.1022 (0.0018)	0.3035 (0.0024)	0.1463 (0.0034)	0.1149 (0.0021)	0.3905 (0.0039)
	Beacon	0.1139 (0.0032)	0.1033 (0.0023)	0.3055 (0.0044)	0.1475 (0.0026)	0.1154 (0.0020)	0.4002 (0.0024)
	CLEA	0.1184 (0.0038)	0.1046 (0.0030)	0.3076 (0.0044)	0.1477 (0.0035)	0.1165 (0.0028)	0.3957 (0.0037)
	Sets2Sets	0.1349 (0.0034)	0.1124 (0.0025)	0.4080 (0.0058)	0.1904 (0.0021)	0.1342 (0.0020)	0.5261 (0.0041)
	DNNTSP	0.1526 (0.0034)*	0.1314 (0.0022)*	0.4460 (0.0044)*	0.2077 (0.0044)*	0.1532 (0.0025)*	0.5496 (0.0038)*
Dunnhumby	G-TopFreq	0.0982 (0.0009)	0.1050 (0.0010)	0.4621 (0.0033)	0.1264 (0.0006)	0.1192 (0.0008)	0.5314 (0.0030)
	P-TopFreq	0.2319 (0.0013)	0.2340 (0.0012)	0.6559 (0.0017)	0.3030 (0.0012)	0.2695 (0.0011)	0.7173 (0.0017)
	GP-TopFreq	0.2356 (0.0013)	0.2358 (0.0013)	0.6649 (0.0019)	0.3141 (0.0012)	0.2743 (0.0011)	0.7372 (0.0026)
	UP-CF@r	0.2428 (0.0006)	0.2468 (0.0012)	0.6747 (0.0024)	0.3185 (0.0009)	0.2848 (0.0013)	0.7339 (0.0014)
	TIFUKNN	0.2396 (0.0008)	0.2408 (0.0011)	0.6761 (0.0022)	0.3191 (0.0015)	0.2799 (0.0012)	0.7407 (0.0030)
	Dream	0.0950 (0.0008)	0.1036 (0.0008)	0.4586 (0.0040)	0.1300 (0.0013)	0.1205 (0.0011)	0.5395 (0.0033)
	Beacon	0.0995 (0.0009)	0.1066 (0.0011)	0.4700 (0.0037)	0.1354 (0.0009)	0.1241 (0.0010)	0.5506 (0.0025)
	CLEA	0.1552 (0.0010)	0.1732 (0.0010)	0.5541 (0.0038)	0.1866 (0.0012)	0.1864 (0.0007)	0.6273 (0.0029)
	Sets2Sets	0.1691 (0.0023)	0.1473 (0.0015)	0.5802 (0.0053)	0.2552 (0.0018)	0.1880 (0.0015)	0.6893 (0.0046)
	DNNTSP	0.2404 (0.0007)	0.2430 (0.0010)	0.6767 (0.0022)	0.3242 (0.0004)*	0.2839 (0.0007)	0.7427 (0.0016)
Instacart	G-TopFreq	0.0710 (0.0003)	0.0811 (0.0001)	0.4542 (0.0010)	0.0990 (0.0001)	0.0962 (0.0002)	0.5248 (0.0017)
	P-TopFreq	0.3260 (0.0008)	0.3378 (0.0007)	0.8449 (0.0018)	0.4307 (0.0008)	0.3939 (0.0002)	0.8957 (0.0019)
	GP-TopFreq	0.3269 (0.0008)	0.3383 (0.0007)	0.8463 (0.0018)	0.4354 (0.0007)	0.3961 (0.0003)	0.9011 (0.0017)
	UP-CF@r	0.3506 (0.0007)	0.3631 (0.0007)	0.8652 (0.0020)	0.4591 (0.0008)	0.4222 (0.0005)	0.9079 (0.0012)
	TIFUKNN	0.3601 (0.0015)*	0.3721 (0.0008)*	0.8642 (0.0005)	0.4709 (0.0015)*	0.4323 (0.0003)*	0.9097 (0.0011)
	Dream	0.0712 (0.0004)	0.0805 (0.0005)	0.4551 (0.0008)	0.0997 (0.0006)	0.0957 (0.0007)	0.5304 (0.0027)
	Beacon	0.0734 (0.0003)	0.0838 (0.0003)	0.4628 (0.0006)	0.1050 (0.0006)	0.1009 (0.0004)	0.5462 (0.0022)
	CLEA	0.1221 (0.0014)	0.1449 (0.0016)	0.5603 (0.0053)	0.1514 (0.0045)	0.1592 (0.0019)	0.6347 (0.0094)
	Sets2Sets	0.2125 (0.0013)	0.1923 (0.0019)	0.7185 (0.0040)	0.3077 (0.0040)	0.2402 (0.0020)	0.8284 (0.0040)
	DNNTSP	0.3330 (0.0003)	0.3412 (0.0004)	0.8525 (0.0011)	0.4423 (0.0005)	0.4000 (0.0005)	0.9042 (0.0003)

Таблица 2.

Желтым цветом выделены лучшие значения метрик. Интересно, что в наборах данных Dunnhumby и Instacart DNNTSP проигрывает более простой модели UP-CF@r. Тем не менее, проигрыш в качество не слишком большой. А среди моделей, основанных на нейронных сетях, модель показывает стабильно лучшее качество на всех датасетах.

Таким образом, модель DNNTSP показывает стабильно высокое качество рекомендации. В зависимости от датасетов, она может немного уступать более простым моделям, однако по-прежнему остается лучше в классе моделей, основанных на нейронных сетях. Именно поэтому мы остановили свой выбор на этой

модели. Попытаемся еще сильнее улучшить качество её работы, добавив в её архитектуру новую идею.

2.3 Описание идеи по улучшению качества работы модели

Обратимся к модели TiSASRec. Её описание дано в статье [6]. Данная модель используется в задачах рекомендации следующего элемента последовательности. Эта модель также основана на нейронных сетях и тоже использует механизм внимания в своей архитектуре. Ключевая особенность данной модели состоит в том, что авторы в явном используют временные интервалы между интеракциями пользователя с товарами. Эта информация передается в attention слой модели и там используется для получения представления элементов. Благодаря использованию матриц временных интервалов при получении представлений элементов, модель TiSASRec показывает SOTA-качество в задаче рекомендации следующего элемента последовательности, что подтверждается экспериментами в статье [6].

Мы можем интегрировать эту идею в модель DNNTSP. Использование матриц временных интервалов дало значимый прирост качества в задаче рекомендации следующего элемента последовательности, поэтому есть основания полагать, что это позволит повысить качество и в задаче рекомендации следующей корзины пользователя.

2.4 Реализация идеи

В первую очередь необходимо было определиться с форматом матриц временных интервалов. Просто подавать матрицу $k \times k$, где k – количество корзин в истории пользователя, мы не можем. Элементы не встречаются в каждой корзине, поэтому такая матрица не даст нам никакой информации. Вместо этого мы будем конструировать матрицу временных интервалов по каждому элементу.

- Создание матрицы временных интервалов для элемента из истории пользователя

Для пользователя u имеется набор k корзин $S_u = \{s_1^u, s_2^u, \dots, s_k^u\}$ и последовательность таймстемпов $T_u = \{t_1^u, t_2^u, \dots, t_k^u\}$, где t_j^u– таймстемп покупки s_j^u. Все элементы из s_j^u принадлежат $I_u \in I$.

Для каждого элемента i из I_u мы будем рассчитывать матрицу m_i^u:

$$m_i^u = \begin{bmatrix} d_{11}^u & d_{12}^u & \dots & d_{1k}^u \\ \dots & \dots & \dots & \dots \\ d_{k1}^u & d_{k2}^u & \dots & d_{kk}^u \end{bmatrix}, \qquad \text{где } d_{ij}^u = \left| t_j^u - t_i^u \right|,$$

Если элемент не встретился в корзине, то $d_{ij}^u = 0$.

Таким образом, для каждого элемента из истории покупок пользователя будет рассчитываться матрица временных интервалов, и тензор из таких матриц по каждому элементу каждого пользователя из батча потом будет подаваться на вход в слой модели, содержащий механизм внимания.

- Attention слой

Скажем еще пару слов о формате данных, которые поступают на вход Attention-based Temporal Dependency Learning компоненты модели. На вход в данный блок приходит тензор размерности N, где N = n1 + n2 + … + n_bs, ni – это количество элементов, имеющихся в истории покупок пользователя i, bs – количество пользователей в батче. Тензор состоит из преставлений элементов Cij, полученных на предыдущем слое модели с помощью взвешенных графов и свертки.

В базовой версии модели используется следующая формула для получения выходных данных:

$$Z_{ij} = softmax\left(\frac{(C_{ij}W_q)\cdot(C_{ij}W_k)^{\mathrm{T}}}{\sqrt{F''}} + M_i\right)\cdot(C_{ij}W_v)$$

, где W_q, W_k, W_v – обучаемые параметры

, M_i – маска внимания для предотвращения утечки информации из будущего

, Z_{ij} – обновленное представление элементов

В новой версии модели помимо Cij мы подаем в данный слой еще и тензор той же размерности N, который содержит матрицы интервалов между появлениями элемента в корзинах пользователя. Каждый временной интервал матрицы представлен в виде эмбеддинга той же размерности, что эмбеддинги элементов корзины. Чтобы обновлять представления элементов в том числе информацией о временных интервалах, перепишем формулу выше следующим образом:

$$S_{ij} = softmax\left(\frac{\left(C_{ij}W_q\right)\cdot\left(C_{ij}W_k\right)^{\mathrm{T}} + E_k^{time}\cdot\left(C_{ij}W_q\right)^{ext}}{\sqrt{F''}} + M_i\right)$$

$$Z_{ij} = S_{ij}\cdot\left(C_{ij}W_v\right) + S_{ij}^{ext}\cdot E_v^{time}$$

, где E_k^{time} и E_v^{time} – матрицы временных интервалов m_i^u, в которых каждый интервал d_{ij}^u представлен в виде эмбеддинга

, ext – добавление дополнительной размерности

Теперь при получении обновленного представления элементов мы в явном виде учитываем еще и информацию о временных интервалах между появлениями элемента в корзинах пользователя. Данный прием должен повысить качество рекомендаций.

Глава 3

3.1 Сравнение качества новой модели с базовой

В предыдущей главе мы обсудили архитектуру базовой версии DNNTSP, выдвинули идею по улучшению качества работы данной модели и реализовали её. Назовем нашу новую версию модели TiaDNNTSP (Time-interval aware DNNTSP). Обучим теперь новую модель на трех датасетах и сравним качество её работы с базовой версией модели. Получили следующие результаты:

	TaFeng											
	PHR_10	**PHR_20**	**PHR_30**	**PHR_40**	**NDCG_10**	**NDCG_20**	**NDCG_30**	**NDCG_40**	**Recall_10**	**Recall_20**	**Recall_30**	**Recall_40**
DNNTSP	0.4784	0.5901	0.6430	0.6673	0.1478	0.1709	0.1828	0.1902	0.1742	0.2437	0.2795	0.3030
TiaDNNTSP	0.4820	0.5937	0.6438	0.6673	0.1568	0.1783	0.1894	0.1970	0.1787	0.2455	0.2795	0.3033
	DC											
	PHR_10	**PHR_20**	**PHR_30**	**PHR_40**	**NDCG_10**	**NDCG_20**	**NDCG_30**	**NDCG_40**	**Recall_10**	**Recall_20**	**Recall_30**	**Recall_40**
DNNTSP	0.65	0.7261	0.7511	0.7739	0.2673	0.2794	0.2961	0.3099	0.2216	0.3097	0.3615	0.3999
TiaDNNTSP	0.6628	0.7261	0.7578	0.7789	0.2698	0.2803	0.2975	0.3105	0.2296	0.3097	0.3648	0.4012
	TaoBao											
	PHR_10	**PHR_20**	**PHR_30**	**PHR_40**	**NDCG_10**	**NDCG_20**	**NDCG_30**	**NDCG_40**	**Recall_10**	**Recall_20**	**Recall_30**	**Recall_40**
DNNTSP	0.3020	0.3801	0.4336	0.4779	0.1786	0.1984	0.2099	0.2183	0.2965	0.3744	0.4274	0.4706
TiaDNNTSP	0.3027	0.3802	0.4331	0.4776	0.1797	0.1993	0.2107	0.2191	0.2972	0.3744	0.4271	0.4703

Желтым цветом выделены ситуации, когда новая модель не смогла превзойти по качеству базовую. Нетрудно заметить, что лучше всего модель отработала на датасетах TaFeng и Dunnhumby. Мы наблюдаем прирост качества по всем метрикам кроме двух, где новая модель дает то же качество, что и базовая. Чуть хуже модель отработала датасете TaoBao, но и там заметен прирост качества в ряде метрик. Тем не менее, разница в качестве на разных наборах данных заметна. Выяснение причин такого поведения может являться одной из тем дальнейших исследований.

Кроме этого, можем обратить внимание на метрику ndcg@k. Примечательно, что данные метрики выше у новой модели во всех датасетах при всех значениях параметра k. Возможно это связано с тем, что TiaDNNTSP лучше сортирует первые элементы благодаря добавленному нами учету временных интервалов между появлениями элементов в корзинах пользователя.

Заключение

В данной работе исследовалась задача рекомендации следующей корзины пользователя. Целью работы была доработка архитектуры существующей модели для улучшения её качества рекомендации. Изучив существующие подходы к решению данной задачи, мы остановили свой выбор на модели DNNTSP, поскольку данная модель показывает лучшее качество среди класса моделей, основанных на нейронных сетях. Основная проблема моделей этого класса в том, что им сложно выучить и учитывать информацию, основанную на частоте появлений элементов. Мы постарались убрать этот недостаток, доработав архитектуру модели и добавив в неё новую идею.

Идея, с помощью которой мы собирались улучшить качество работы базовой DNNTSP заключалась в том, чтобы в явном виде передавать в модель информацию о временных интервалах между появлениями элементов в истории пользователя. Данный подход уже был использован в модели для задачи рекомендации следующего элемента последовательности, и там он показал значительный прирост качества.

Мы реализовали предложенную нами идею и получили TiaDNNTSP: Time-interval aware DNNTSP, которая теперь в явном виде учитывает информацию об интервалах между появлениями элементов в корзинах и использует её для обновления представлений элементов.

Мы обучили новую модель на трех разных датасетах и использовали три популярные метрики качества для сравнения результатов работы новой модели и базовой версии DNNTSP. Полученные результаты показывают прирост в качестве рекомендации новой модели. Величина этого прироста варьируется в зависимости от датасета, однако мы можем заключить, что реализованная нами идея действительно улучшает рекомендательную способность алгоритма.

В дальнейших исследованиях может быть детально изучена причина, по которой прирост качества варьируется в зависимости от датасета, и какие особенности датасета влияют на это. Кроме того, требуются дальнейшие эксперименты с полученной моделью и оптимизация процесса получения матриц временных интервалов для элементов с целью повышения её скорости работы.

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
5. Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. 2023. A Next Basket Recommendation Reality Check. ACM Trans. Inf. Syst. 1, 1 (March 2023), 29 pages.
6. Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM ’20), Feb- ruary 3–7, 2020,

Houston, TX, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371786

7. F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, “A dynamic recurrent model for next basket recommendation,” in SIGIR, 2016, pp. 729–732
8. H. Hu and X. He, “Sets2Sets: Learning from sequential sets with neural networks,” in SIGKDD, 2019, pp. 1491–1499.