ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК

Андреевский Александр Константинович

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

# Динамические двухуровневые рекомендательные системы на основе графа знаний
# Dynamic hierarchical recommender systems based on knowledge graph

по направлению подготовки 01.04.02 Прикладная математика и информатика
образовательная программа «Финансовые технологии и анализ данных»

Научный руководитель
Приглашенный преподаватель, аспирант

Ананьева М. Е.

Студент
Андреевский А. К.

Москва, 2022

# Оглавление

# Глава 1

# Введение

Развитие информационных технологий принесло с собой новые, ранее неизвестные, вызовы. Ограниченный ресурс человеческого внимания становится не способен справляться с массивами информации, с которыми ежедневно сталкивается каждый из нас. Проблему ранжирования информации в частности помогают решать рекомендательные системы.

Это направление возникло не столь давно, однако успело пройти через череду трансформаций, как, впрочем, и другие области машинного обучения. Изначально в основу рекомендательных систем было заложено предположение о том, что пользователи выбирают товары, похожие на те, с которыми они взаимодействовали ранее. Задача, таким образом, состояла в поиске похожих товаров по истории пользовательских взаимодействий. Затем на первый план вышла идея использования эмбеддингов - векторных представлений пользователей и товаров, позволяющих выявлять высокоуровневые предпочтения. В этом подходе задача рекомендации решается вычислением попарной близости векторов пользователей и товаров и рекомендацией наиболее близких из них. Среди разнообразных способов создания эмбеддингов наибольший интерес представляют модели глубокого обучения, позволяющие учитывать сложные нелинейные связи.

Однако моделирование предпочтений на основе исторических взаимодействий недостаточно хорошо позволяет отделять коллаборативный сигнал от шума и не учитывает характерные особенности пользователей и объектов. В качестве решения этой проблемы большое распространение получили графовые модели. Так, заметим, что задачу рекомендации в базовом варианте можно сформулировать в виде задачи предсказания ребер (link prediction) на симметричном двудольном графе с вершинами «пользователи – товары». Сочетание графового представления с нейронными сетями привело к развитию графовых нейронных сетей (GNN). Процесс обучения GNN состоит из двух этапов: итеративной агрегации информации по соседним вершинам и обновления эмбеддинга каждой вершины.

Среди преимуществ GNN можно назвать сохранение топологической структуры графа, учет нетривиальных, нелинейных взаимосвязей, относительно простое обогащение эмбеддингов дополнительными признаками. В частности, можно использовать граф знаний для добавления справочной информации об отношениях между вершинами [1], социальный граф для извлечения информации об отношениях между пользователями [2], динамический граф для учета динамического характера взаимодействий и изменяющихся во времени предпочтений [3], иерархический граф для учета

краткосрочных и долгосрочных интересов [4] и т.д.

В данной работе исследуется динамическая двухуровневая рекомендательная система на основе графа знаний. Детально рассматривается архитектура модели KDA, позволяющая добиться более качественных рекомендаций за счет механизма внимания, агрегации взаимосвязей между товарами и обработки временных последовательностей.

Ценность исследования заключается в следующих практических результатах:

1. Предложена модификация существующей архитектуры модели KDA, использующая рекуррентную модель для обработки последовательности динамических эмбеддингов

2. Проведено ablation study – исследование отдельных компонентов модели, подбор оптимальных параметров

3. Проведено сравнение нескольких методов агрегации эмбеддингов

4. Реализовано ансамблирование нескольких графовых моделей для создания более точных рекомендаций

5. Выполнен обзор распространенных графовых нейронных моделей

Исследование состоит из нескольких частей. В первом разделе описана формальная постановка задачи рекомендаций на графах, представлен критический обзор релевантной литературы, с классификацией существующих моделей и обсуждением существующих сложностей и нерешенных проблем. Второй раздел содержит описание архитектуры KDA, метрик качества рекомендаций и используемого при обучении датасета. Следующий раздел содержит отчет о поставленных экспериментах и их результатах. Наконец, заключительная часть подводит итоги работы и предлагает несколько потенциальных направлений будущих исследований.

# Глава 2

# Обзор литературы

## 2.1 Постановка задачи рекомендации на графе

Предварим обзор актуальных научных работ формализацией задачи и введением обозначений, к которым мы будем возвращаться в ходе исследования. Рассмотрим множество пользователей $u \in \mathcal{U}$ и множество объектов $i \in \mathcal{I}$. В общем виде задача рекомендательной системы сводится к тому, чтобы оценить интерес пользователя к определенному объекту, что можно интерпретировать как расстояние между векторными представлениями пользователя и объекта:

$$y_{u,i} = f(e_u, e_i),$$

где $f(.)$ - функция для оценки близости векторов, $e_u$ и $e_i$ - выученные эмбеддинги пользователей и объектов, $y$ - вероятность взаимодействия между $u$ и $i$. За меру близости обычно принимают скалярное произведение векторов, различные меры векторного расстояния (косинусное, евклидово и другие).

Как упоминалось ранее, взаимодействия между пользователями и объектами удобно моделировать на графах. Опишем эту процедуру более формально. Двудольный граф $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ представляет собой множество вершин $\mathcal{V}$ двух типов - пользователи и объекты - и множество ребер $\mathcal{E}$, обозначающих взаимодействия пользователей с объектами. Схематическое изображение графа представлено на рисунке 2.1. Отдельно введем понятие множества соседних вершин $\mathcal{N}(v) = u \in \mathcal{V} | (v, u) \in \mathcal{E}$ и множества взаимодействий пользователей с объектами $\mathcal{R} = \{r_{u,i} | \exists\, e_{ij} \in \mathcal{E}\}$. Под взаимодействием, в зависимости от задачи, понимают покупку, добавление в корзину, в избранное, и т.д.

## 2.2 Графовые нейронные сети

Итак, после краткого введения в предметную область, обратимся к актуальным направлениям научных исследований в области рекомендательных систем на графах. Последовательно рассмотрим принципы работы графовых нейронных сетей (GNN) и их разновидности, способы использования графа знаний в качестве внешней информации, динамические графы, учитывающие темпоральную структуру пользовательских предпочтений. Наконец, обратимся непосредственно к моделям, близким по своей архитектуре к модели KDA, лежащей в основе данного исследования.

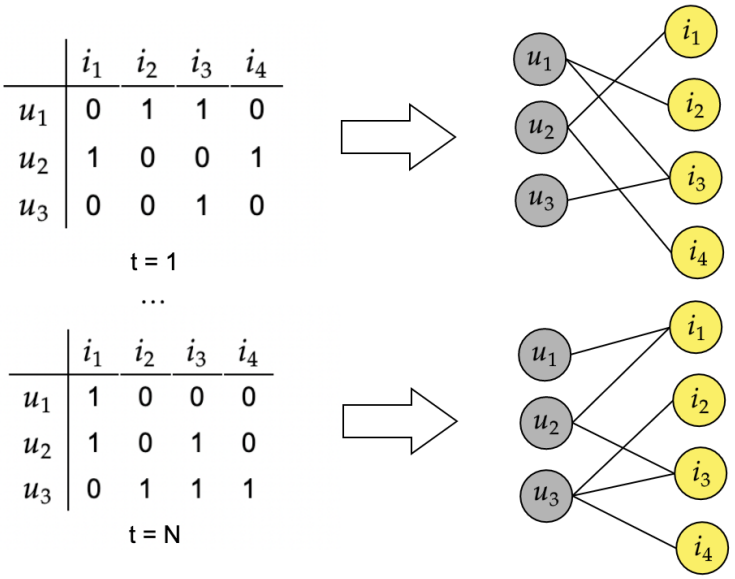


Рис. 2.1: Пример построения двудольного графа по истории взаимодействий

Начнем с краткого обзора графовых нейронных сетей. В самом общем смысле в основе GNN заложен принцип агрегации информации и ее итеративного распространения [5]. Иными словами, на первом этапе векторное представление заданной вершины итеративно обновляется на основе агрегации эмбедингов ее соседних вершин. Затем информация последовательно распространяется по графу (message passing), от вершины к вершине, что позволяет учитывать не только характеристики заданной вершины, но и характеристики вершин в окрестности ее соседей (multi-hop neighbours). Процесс агрегации информации и обновления в общем виде выражается как

$$AGG : n_v = Aggregator(\{e_u, u \in \mathcal{N}_v\}), \quad UPDATE : n_v = Updater(e_v, n_v),$$

где $n_v$ обозначает множество соседних вершин вершины $v$, $e_v$ обозначает скрытое представление вершины $v$. Иллюстрация к описанным выше принципам работы GNN приводится на рисунке 2.2 [5].

Представляется удобным способ классификации моделей по выбранным функциям агрегации и обновления. Так, существуют различные способы получения информации по множеству соседних вершин. Может использоваться множество всех соседей [6], однако это делает вычисления затратными по времени и памяти, особенно в случае вершин с высокой степенью связности (например, наиболее популярных товаров) [7]. Для решения этой проблемы предлагается сэмплировать соседей путем случайного блуждания в окрестности вершины и выбора наиболее посещаемых из них [8, 9, 10], что позволяет учитывать топологические свойства графа. В случае гетерогенного графа, обладающего различными типами вершин и ребер, авторы

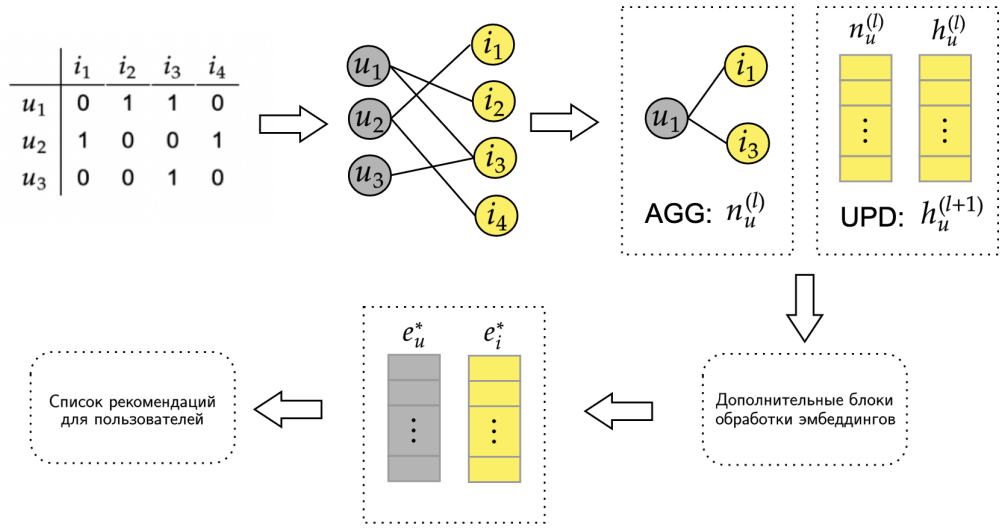


Рис. 2.2: Общий принцип устройства графовых нейронных сетей для задачи рекомендации

модели HetGNN [11] предлагают делить соседние вершины на подмножества в зависимости от их типа. Затем функция агрегации с механизмом внимания по отдельности обрабатывает каждый тип вершин и отношений.

Для решения описанной ранее проблемы несбалансированности (также встречающейся в литературе под названием «popularity bias») может использоваться нормировка на степень вершин, как в модели GGNN [12]:

$$AGG : n_v^l = \frac{1}{|\mathcal{N}_v|} \sum_{j \in \mathcal{N}_v} h_j^l,$$

где $|\mathcal{N}_v|$ - количество соседей вершины $v$. Или нормировка эмбеддинга вершины на его L2-норму, как в модели TASRec [13]:

$$e(n_v^l) = \frac{e(n_v^l)}{||e(n_v^l)||_2}.$$

Отдельной задачей является importance pooling - оценка влияния инцидентных вершин. Авторы модели GAT предлагают использовать для этого механизм внимания:

$$AGG : n_v^l = \sum \alpha_{v,i} h_i^l, \quad \alpha_{v,i} = \frac{exp(Att(a^\top [Wh_v^l \oplus Wh_i^l]))}{\sum_{j \in \mathcal{N}_v} exp(Att(a^\top [Wh_v^l \oplus Wh_j^l]))},$$

где $a$ - выучиваемый параметр модели, $W$ - взвешивающая матрица, $Att$ - функция внимания (в случае GAT была выбрана функция $LeakyReLU$).

Данный подход нашел широкое распространение в исследованиях, в частности, в модели SASRec [14]. SASRec использует механизм self-attention, чтобы оценивать релевантность товаров из покупательской корзины пользователя и учитывать их взаимное влияние.

Стоит отметить что механизм внимания не лишен некоторых недостатков. В частности, авторы модели Locker [15], отмечают, что механизму внимания зачастую не удается корректно оценить вклад краткосрочных пользовательских интересов. В модели делается попытка разделить моделирование долгосрочных и краткосрочных семантических связей путем использования отдельных энкодеров для обработки каждого типа взаимодействий. Кроме того, как отмечается в статье [16], механизм внимания распределяет веса по всем соседям, тем самым может дисконтироваться влияние локальных сообществ. Чтобы не утратить в результате этого коллаборативный сигнал, в модели GraphSAIL предлагается комбинировать локальную и глобальную структуру графа с контролем на метрику близости между соседними вершинами [17].

После непосредственного создания эмбеддингов следует этап оптимизации. В работах, посвященных рекомендательным системам на графах, для этого часто используется попарная функция потерь BPR-Loss (Bayessian Personalized Ranking). Оптимизация BPR происходит по тройкам $\mathcal{O} = (i, n, p)$, где $i$ - объект выборки, $n$ - отрицательный (нерелевантный) объект по отношению к $i$ (не связанный с ним ребром), $p$ - положительный (релевантный) объект [18]. Функция потерь записывается следующим образом:

$$\mathcal{L}_{BPR} = \sum_{i,n,p} -ln\sigma\Big(e(i,n) - e(i,p)\Big),$$

где $\sigma$ - сигмоидная функция, $e(.)$ - эмбеддинг объектов. В ходе оптимизации модель учится отличать положительные и отрицательные объекты выборки, тем самым оптимизируя качество представлений.

## 2.3 Граф знаний

Актуальным направлением исследований является использование дополнительной внешней информации об объектах, закодированной в графе знаний. Использование графа знаний позволяет работать с проблемой «холодного старта», в частности, в ходе индуктивного обучения, при работе с вершинами, не встречавшимися в обучающей выборке, с проблемой масштабирования и интерпретации рекомендаций [5]. К недостаткам графовых моделей, основанных на графах знаний, можно отнести сложность с уравновешиванием информации по взаимодействиям и информации по графу знаний - авторы отмечают смещенность, которая не позволяет прийти к динамическому равновесию двух источников информации, что снижает качество рекомендаций. Кроме того, существуют проблемы с извлечением сложных взаимосвязей между пользователями и отдельными аттрибутами объектов. Над решением этой проблемы активно работают существующие мультимодальные модели.

Граф знаний представляет собой граф вида $\mathcal{G}_{KG} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{O}\}$, где $\mathcal{E}$ - множество сущностей (entities), $\mathcal{O}$ - множество отношений (relations). Тройкой (h,r,t) закодировано наличие связи r (relation) между

вершиной h (head) и вершиной t (tail). Примерами сущностей могут служить наименования торговых марок, ценовые сегменты, товарные характеристики. Отношения описывают тип связи, существующий между объектами. Например, сущность «чай» может быть связана с сущностью «кофе» отношением «товары-субституты».

Зачастую, чтобы обогатить историю взаимодействий пользователей с внешней информацией, граф знаний интегрируют в основной графом путем добавления в граф знаний вершин пользователей в качестве отдельной сущности [19, 20]. Это позволяет обогатить знания об исторических патернах взаимодействий пользователей с товарами с дополнительными семантическими характеристиками объектов и повышает интепретируемость рекомендаций. Более того, такая связь позволяет сделать рекомендации более релевантными. Близкие по семантическим свойствам объекты, связанные отношением в графе знаний, смогут быть рекомендованы пользователю, который ранее с ними не взаимодействовал с ними напрямую, но был связан косвенно (например, фильмы одного режиссера или товары одного производителя).

Так, в модели RippleNet эмбеддинг пользователя предлагается формировать на основе множества сущностей (ripple set), связанных отношениями с объектами, с которыми взаимодействовал пользователь. Инцидентные вершины, образующие ripple set, набираются методом k-hop neighbours. Глубина множества $k$ вынесена в параметр модели, что позволяет учитывать как ближайших, локальных соседей, так и удаленных от целевой вершины соседей. Например, для просмотренного фильма соседней вершиной первого порядка будет являться режиссер, а вершиной второго порядка - другой фильм, снятый этим режиссером. За счет последовательного обхода вершин удается моделировать распространение предпочтений по графу и учитывать потенциальные интересы пользователей, которые скрыты за историей его взаимодействий. Также посредством того, что для каждой рекомендации можно проследить «путь» от пользователя к товару, рекомендации становятся интерпретируемыми, раскрывая интересы пользователей. Это является частью отдельного направления исследования по созданию интрепретируемых рекомендаций на графах знаний - meta-path based learning и в частности используется в модели KGAT [21].

Ранее нами также поднимался вопрос моделирования мультимодальности графов. При этом необходимо сделать поправку на то, что степень заинтересованности пользователей варьируется в зависимости от типа взаимодействия (клик / добавление в избранное / покупка). Эта особенность лежит в основе модели GES-SASRec [22]. Каждое отношение обрабатывается по отдельности и хранится в отдельном векторе, которые затем подаются на вход в сверточный слой. Полученные «сглаженные» эмбеддинги подаются на вход модели SASRec.

Модель MKGAT [23] также учитывает свойство мультимодальности графов знаний. В качестве внешней информации используется несколько источников: текстовые описания товара (эмбеддинги Word2Vec), изображения

товара (эмбеддинги из ResNet, обученной на ImageNet). Они учитываются в дополнение к структурному графу, построенному на взаимодействиях пользователей с товарами. Эмбеддинги строятся по отдельности, затем конкатенируются и на финальном шаге подаются в attention-слой. Модель KHGT [24] учитывает мультимодальность графа при помощи основанного на механизме внимания энкодера, который обучается отличать взаимодействия разных типов и зависимости между ними.

Существует два основных подхода к созданию эмбеддинга графа знаний: геометрический и семантический. Первый подход основан на предпосылке о асимметричности отношений между сущностями и по построению учитывает направление исходящих связей. Тройки $(h, r, t)$ отображаются в новое векторное гиперпространство, а качество отображения оценивается по расстоянию между $h$ и $t$. Для оценки схожести между $h$ и $t$ используются 3 основных метода: TransE [25], TransH [26] и TransR [27].

В этом исследовании выбран второй подход, в ходе которого все отношения рассматриваются как симметричные, а для оценки сходства между тройками используется их скалярное произведение.

## 2.4 Динамические графы

Для того, чтобы учесть эволюцию предпочтений во времени, могут использоваться динамические графы. Динамический граф представляет собой граф вида $G = \{\mathcal{V}, \mathcal{E}, \mathcal{T}\}$, где $\mathcal{V}$ - множество вершин, $\mathcal{E}$ - множество ребер, $\mathcal{T}$ - множество временных отрезков (timestamps). Тройка $(v_i, v_j, t)$ описывает отрезок времени, в который между вершинами $v_i$ и $v_j$ возникло ребро.

Для обработки последовательности эмбеддингов с темпоральными признаками на практике успешно применяются рекуррентные модели. В частности, как показала модель TimeLSTM [28], с этой задачей хорошо справляется LSTM. Применительно к графовым эмбеддингам, авторам модели DGRN удалось добиться прироста качества при помощи использования GRU. В DGRN использовалась и другая распространенная техника - сэмплирование подграфов. В момент времени $t_k$ для пользователя $u$ выбиралось множество $\mathcal{N}_u$ из $n$ товаров, с которыми он взаимодействовал в период $t_0, t_1, \ldots, t_k$. Затем, для каждого товара из $\mathcal{N}_u$ сэмплировалось множество $\mathcal{N}_i$ из $i$ пользователей, которые с ними взаимодействовали. Полученные множества товаров и пользователей образовывали подграфы $\mathcal{G}_u(t_k)$ для каждого момента времени $t_k$, по которым формировались эмбеддинги. Похожая архитектура была реализована в модели STEN [29], однако вместо графа знаний использовался социальный граф, построенный на пользовательских признаках.

Среди моделей, учитывающих динамику предпочтений при построение рекомендательной системы на графах, стоит упомянуть следующие:

- RetaGNN [30]. В рамках модели основной акцент сделан на моделировании типов отношений между вершинами. Так, при распространении

информации по графу веса соседних вершин формируются в зависимости от типа отношений и его влияния, оцененного при помощи механизма внимания.

- SURGE [31]. Особенностью модели является кластеризация похожих товаров по истории пользователя в подграфы, по которым выучиваются эмбеддинги. Фактически, это можно расценивать как создание искуственного графа знаний, где отношением выступает релевантность товаров в пользовательской корзине - более близкие товары попадают в один кластер. При рекомендации выбираются те товары, которые находятся в близких кластерах. Недостатком такого подхода является высокая вычислительная сложность, с которой можно столкнуться при создании подграфов на длинных последовательностях покупок.

- PTGCN [32]. Модель PTGCN учитывает позицию товара в последовательности при формировании динамических эмбеддингов с использованием механизма внимания на этапе агрегации. Отдельно стоит отметить, что для учета высокоуровневых связей между пользователями и товарами их эмбеддинги пропускаются через многослойную сверточную сеть.

- DRL-SRe [33] и TGSRec [34]. Данные модели будет детально разобраны далее.

- Chorus [35]. Авторы предлагают создавать эмбеддинги двух типов: базовые, созданные на основе истории взаимодействий, и «относительные», учитывающие динамику отношений между товарами. Полученные эмбеддинги объединяются в единый динамический эмбеддинг таким образом, чтобы учитывать временной интервал между взаимодействиями и интенсивность взаимосвязей товаров.

- KATRec [36]. Модель KATRec объединяет информацию по графу знаний и кратко- и долгосрочные предпочтения пользователей. Так же, как и предыдущая модель, KATRec использует два вида эмбеддингов: динамические пользовательские и основанные на отношениях товарные.

- IPAKG [37]. IPAKG использует эмбеддинг графа знаний для выявления пользовательских предпочтений и моделирует динамику пользовательских предпочтений и взаимосвязей между товарами при помощи рекуррентной нейронной сети, в которую подаются историческая последовательность пользовательских взаимодействий. Степень релевантности товара для пользователя оценивается при помощи функции интенсивности, основанной на механизме внимания

В силу сходства архитектур и подходов, мы отдельно рассмотрим модели DRL-SRe и TGSRec. DRL-SRe представляет собой двухуровневую динамическую модель для предсказания следующей покупки пользователя.

На первом уровне для каждого периода времени строится отдельный подграф, по которому обучаются эмбеддинги товаров и пользователей. На втором уровне эмбеддинги обрабатываются GRU, в результате чего на выходе получаются эмбеддинги пользователей и товаров $H_u = [h_u^1; h_u^2, \ldots, h_u^T]$, $H_i = [h_i^1; h_i^2, \ldots, h_i^T]$. Затем они конкатенируются и подаются на вход в MLP, после чего рассчитываются вероятности взаимодействий:

$$\hat{y}_{ui} = \sigma\Big(MLP(h_u^T \oplus h_i^T \oplus e_u \oplus e_i; \Theta^{MLP})\Big),$$

где $\sigma$ - сигмоида, $\Theta$ - параметры MLP, $(h_u^T, h_i^T)$ - скрытые состояния пользователя и товара в момент времени $T$ (из последнего слоя модели), $e_u, e_i$ - финальные эмбеддинги пользователя и товара. В качестве промежуточной функции активации используется ReLU. При обучении минимизируется функция потерь, основанная на кросс-энтропии:

$$\mathcal{L}_c = -\big(y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})\big).$$

В отличие от KDA, в DRL-SRe не используется внешняя информация в виде графа знаний, граф строится исключительно на истории взаимодействий пользователей.

В свою очередь, в модели TGSRec используется несколько способов обработки динамических паттернов. Для вершин двудольного графа (пользователей и товаров) создаются долгосрочные представления, для ребер - непрерывные по времени эмбеддинги, которые строятся отображением из скалярных временных моментов в непрерывные векторы $\Phi(T) : T \to \mathbb{R}^{d_T}, T \in \mathbb{R}^+$. Для отображения выбрана следующая функция:

$$\Phi(t) = \sqrt{\frac{1}{d_T}}[cos(w_1 t), sin(w_1 t), \ldots, cos(w_{d_T} t), sin(w_{d_T} t)]^\top,$$

где $d_T$ - размерность эмбеддинга, $w = [w_1, \ldots, w_{d_T}]^\top$ - выучиваемые параметры модели. В рамках этого метода близость, или корреляция, между тройками взаимодействий $(u, i, t_1)$ и $(u, j, t_2)$ вычисляется как скалярное произведение соответствующих эмбеддингов, $\Phi(t_1) \cdot \Phi(t_2)$. Другим нововведением модели можно считать применение блока, агрегирующего информацию по темпоральному эмбеддингу и эмбеддингам пользователей и товаров, и применяющих к нему слой коллаборативного внимания, оценивающего важность взаимодействий. Эмбеддинги пользователей и товаров получаются в результате конкатенации их темпоральных эмбеддингов с отображением момента времени в непрерывное векторное пространство:

$$h_u^{(l-1)}(t_s) = e_u^{l-1}(t_s) \oplus \Phi(t_s), h_i^{(l-1)}(t_s) = e_i^{l-1}(t_s) \oplus \Phi(t_s),$$

где $l$ - слой модели, $e(t_s)$ - темпоральный эмбеддинг по состоянию на момент времени $t_s$. Затем для целевого пользователя $u$ каждая пара «товар-момент

времени» взвешивается через механизм внимания:

$$\pi_t^u(i, t_s) = \frac{exp\Big(\pi_t^u(i, t_s)\Big)}{\sum_{(i', t_s' \in \mathcal{N}_u(t))} exp\Big(\pi_t^u(i', t_s')\Big)}$$

Для обновления эмбеддинга целевой вершины используется взвешенная сумма эмбеддингов соседних вершин, где весом выступает полученный attention score. После того, как эмбеддинги обрабатываются механизмом внимания, они пропускаются через сеть прямого распространения (FFN, feed forward network):

$$e_u = FFN\big(e_{\mathcal{N}_u}(t) \oplus h_u(t)\big),$$
$$e_i = FFN\big(e_{\mathcal{N}_i}(t) \oplus h_i(t)\big)$$

Полученные финальные представления подаются на вход в скоринговую функцию:

$$Score(u, i, t) = e_u(t) \cdot e_i(t)$$

В результате, рекомендации могут быть получены для любого выбранного момента времени $t$.

Необходимо подчеркнуть, что независимо от сходства архитектуры и подхода к решению задачи рекомендаций, между описанными моделями и KDA существуют различия в части непосредственного устройства блоков обработки истории взаимодействий, создания эмбеддинга графа знаний и вычисления релевантности товаров. Эти особенности явным образом описаны в следующем разделе.

# Глава 3

# Методология

Архитектура модели KDA, которая используется в исследовании, схематично представлена на рисунке 3.1.

В качестве датасета используется массив транзакций пользователей за период $S_u^T = t_o, t_1, \ldots, t_n$. Ставится задача предсказать товар, купленный пользователем в следующий момент времени $t_{n+1}$.

На первом уровне формируются векторные представления пользователей и товаров. Отношения между товарами моделируются в виде графа знаний, для каждого типа отношений создается отдельный динамический эмбеддинг. Признаки товаров, такие как принадлежность к категории, также учитываются при построении векторных представлений. При помощи механизма внимания оценивается величина взаимного влияния товаров. Оптимизация эмбеддингов товаров и генерируемых рекомендаций происходит одновременно путем минимизации составной функции потерь $\mathcal{L} = \mathcal{L}_{rec} + \gamma\mathcal{L}_{kg}$, где $\gamma$ - гиперпараметр, отвечающий за вес функции потерь по графу знаний. При создании эмбеддингов пользователей учитывается динамическая структура их предпочтений, выучиваемая по массиву транзакций за период. Последовательность исторических эмбеддингов пользователя обрабатывается рекуррентной моделью, позволяющей дополнительно учитывать эволюцию пользовательских интересов во времени.

На втором уровне модели скалярное произведение полученных эмбеддингов пользователей и товаров служит для оценки вероятности следующей покупки, позволяя формировать список рекомендаций на временной срез $t_{n+1}$ и ранжировать его в зависимости от выбранной длины рекомендаций.

В следующих разделах мы более подробно остановимся на двух уровнях архитектуры модели, опишем датасет, на котором ставятся эксперименты, и метрики качества рекомендаций.

## 3.1 Уровень создания эмбеддингов

На первом уровне агрегируется информация по транзакционному компоненту и отношениям, содержащимся в графе знаний. Данная задача решается через поэтапное формирование пользовательских и товарных эмбеддингов.

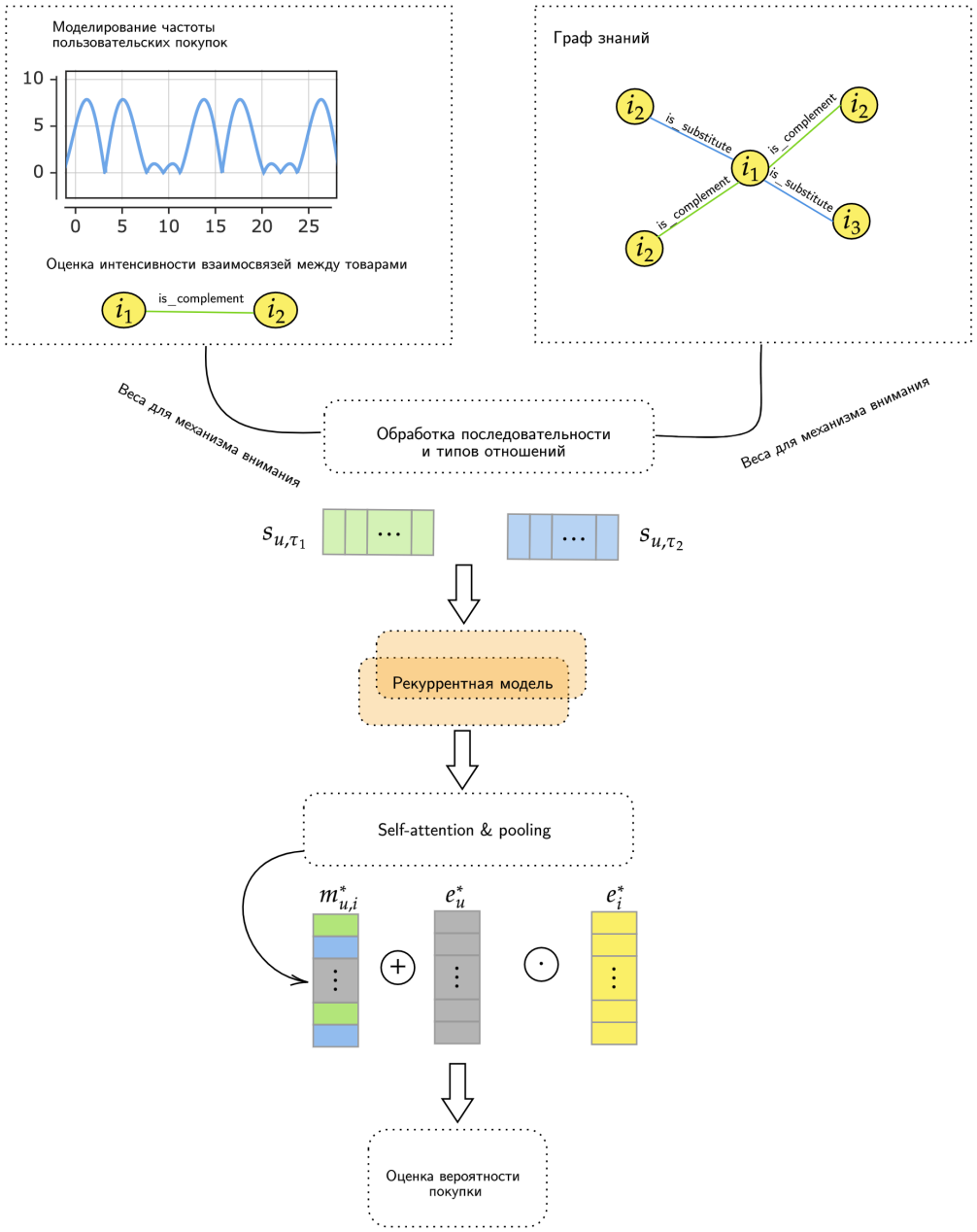


Рис. 3.1: Схема архитектуры модели KDA

### 3.1.1 Эмбеддинги товаров

Создание эмбеддингов товаров происходит на основе обработки симметричного графа знаний $\mathcal{G}_{KG} = (h, r, t)$, где $(h, t) \in \mathcal{I}$ - сущности, $r \in \mathcal{R}$ - отношение между ними. Граф знаний является симметричным в силу того, что отношения между товарами можно считать ненаправленными – товары A и B могут служить субститутами и комплементами друг для друга так же, как B и A. Поэтому в качестве функции близости между тройками $h, r, t$ выбирается симметричная мера $Rel_r$, удовлетворяющая условию $Rel_r(h, t) = Rel_r(t, h)$. На ее роль выбрана семантическая функция близости DistMult [38]:

$$Rel_r(h, t) = e_{i,h}^{\top} diag(r) e_{i,t},$$

где $diag(r)$ означает диагональную матрицу, заполненную по диагонали $r$.

Функция потерь выбрана таким образом, чтобы наблюдаемые отношения получали большие веса, чем ненаблюдаемые:

$$\mathcal{L}_{rel} = - \sum_{(h,r,t) \in \mathcal{R}} \log \Big( Rel_r(h, t) - Rel_r(\widetilde{h}, \widetilde{t}) \Big)$$

Для каждой тройки случайным образом сэмплируется $h$ или $t$, формируя негативные примеры $(\widetilde{h}, r, \widetilde{t}) \notin \mathcal{R}$.

### 3.1.2 Эмбеддинги пользователей

Эмбеддинги пользователей строятся таким образом, чтобы учитывать их взаимодействия с товарами и динамику их предпочтений. Для этого на первом уровне формируются динамические исторические представления $s_{i,\tau}$ для товара $i$ по последовательности транзакций $S_u^T$ для каждого типа отношений $\tau$:

$$s_{i,\tau} = \sum_{(j,t_j) \in S_u^T} INT(i, j, \tau) f_\tau(\Delta t_n) e_j,$$

где $e_j$ - эмбеддинг товара $j$, $INT(i, j, \tau) \in [0, 1]$ - вес, оценивающий релевантность товаров $i$ и $j$, $f_\tau(\Delta t_n) \in [0, 1]$ - функция temporal decay, присваивающая большие веса более поздним взаимодействиям. Временной интервал между приобретением целевого товара и товара $j$ $\Delta t = T - t_j$ для удобства нормируется при помощи логарифмической функции

$$\Delta t = max(0, \log_2(\Delta t / 60)).$$

Остановимся подробнее на подходе к оценке релевантности товаров и эволюции взаимосвязей между товарами. Для оценки релевантности товаров $i$ и $j$ используется следующая функция:

$$INT(i, j, \tau) = \frac{exp\big(Rel_r(i, j)\big)}{\sum_{j' \in S_u^T} \big(Rel_r(i, j')\big)}.$$

В результате семантически близкие товары получают большие веса, чем нерелевантные товары. Это позволяет более полноценно использовать потенциал графа знаний. Метрика близости между товарами $Rel_r$ рассматривается в следующем разделе. Для того, чтобы учитывать динамику пользовательских покупок, используется функция temporal decay.

Чтобы получить распределение времени транзакций, в модели предлагается использовать трансформацию Фурье. Принимая момент $N$ за основу, DFT отображает ограниченную последовательность равномерных точек непрерывной функции $f(t)$ в последовательность равной длины:

$$w_k = \frac{2\pi}{N}k, \quad k = 0, 1, \ldots, N-1.$$

$F[w_k]$ содержит информацию об амплитуде и фазе синусоиды частоты $w_k$ для функции $f(t)$. Таким образом, непрерывная функция времени трансформируется в дискретную функцию с определенной частотой, результатом которой является частотный эмбеддинг.

В модели KDA для каждого типа отношений $\tau$ строится частотный эмбеддинг $F_\tau$, инициализируемый случайным образом. Таким образом, для выбранного временного интервала $\Delta t_n$ значение векторного представления рассчитывается следующим образом:

$$f_\tau(\Delta t_n) = \frac{1}{N}\sum_{k=0}^{N-1}\left(F_\tau[w_k]e^{jw_k\Delta t_n}\right).$$

Полученные динамические эмбеддинги обрабатываются механизмом внимания, что позволяет выявить взаимное влияние между отношениями. В общем виде механизм внимания представим следующим образом:

$$Att(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d}})V,$$

где $Q$ - запрос, $K$ и $V$ - ключ и соответствующее ему значение, $d$ - размерность пространства. В модели KDA используется механизм self-attention, при котором все $Q, K, V$ формируются для одного целевого объекта. Под объектами понимаются динамические эмбеддинги $s_{i,\tau}$, которые составляют общий эмбеддинг $S_i = \left(s^T_{i,\tau_1}, s^T_{i,\tau_2}, \ldots, s^T_{i,\tau_M}\right)$ для всех рассматриваемых отношений $M$. Таким образом, тройка $Q, K, V$ зависит от эмбеддинга $S_i$:

$$Q = S_i W^Q, K = S_i W^K, V = S_i W^V,$$

где $W^Q, W^K, W^V$ - матрицы весов. Векторное представление с механизмом внимания $A_i$ записывается как

$$A_i = Att(Q, K, V).$$

В модели применяется несколько self-attention слоев, поверх которых накладывает сеть прямого распространения (FFN) с нелинейностью:

$$FFN(A_i) = ReLU(A_i W_1 + b_1)W_2 + b_2.$$

Для борьбы с переобучением и решения проблемы с исчезающим градиентом используются стандартные техники глубокого обучения: dropout, layer norm, residual connections.

Пропустив вектора через $K$ self-attention слоев, на выходе образуется последовательность динамических эмбеддингов $S_i^K$. Чтобы получить из них финальные эмбеддинги, используется один из методов агрегации (pooling-слой): усреднение, взятие максимума, суммирование. Эффективность каждого из перечисленных способов проверяется экспериментальным путем. После агрегации формируется финальный исторический эмбеддинг $m_{u,i}$ пользователей и товаров.

## 3.2 Уровень генерации рекомендаций и ранжирования кандидатов

На выходе первого уровня модель получает три вида эмбеддингов, содержащие в себе динамические и коллаборативные сигналы, построенные по историческим взаимодействиям. Это темпоральное представление $m_{u,i}$ динамики пользователей и товаров по временной последовательности $S_u^T$, обработанное рекуррентной моделью (GRU); эмбеддинг пользователя $e_u$, агрегирующий долгосрочные предпочтения, и эмбеддинг товара $e_i$. Далее для каждого из товаров-кандидатов рассчитывается метрика схожести путем скалярного произведения соответствующих эмбеддингов:

$$\hat{y}_{u,i} = (e_u + m_{u,i})e_i^\top + b_i,$$

где $b_i$ - параметр, оценивающий смещенность предпочтений пользователя по отношению к товару $i$. Затем кандидаты ранжируются в зависимости от полученной вероятности покупки, после чего топ-k из них служат в качестве рекомендации.

Оптимизация задачи создания эмбеддингов пользователей и товаров и задачи рекомендации происходит одновременно. В качестве функции потерь для задачи рекомендации выбран BPR Loss:

$$\mathcal{L}_{rec} = -\sum_{u \in \mathcal{U}} \sum_{i=2}^{N_u} \log \sigma(\hat{y}_{u,i} - \hat{y}_{u,\widetilde{i}}),$$

где $\widetilde{i}$ - случайно выбранный товар, служащий негативным примером, $\sigma$ - сигмоида.

В то же время общая функция потерь выглядит следующим образом:

$$\mathcal{L} = \mathcal{L}_{rec} + \gamma \mathcal{L}_{rel},$$

где $\gamma$ - гиперпараметр, отвечающий за вес, присваиваемый задаче создания эмбеддинга графа знаний.

## 3.3 Датасеты

В рамках исследования используются датасеты Ta-Feng Grocery[1] и TTRS: Tinkoff Transactions Recommender System benchmark [39]. Описательные статистики датасетов приведены в Таблице 3.1.

Датасет Ta-Feng Grocery содержит историю транзакций покупателей за 4 полных месяца, с 1 ноября 2000 г. по 28 февраля 2001 г. Каждая транзакция представляет собой корзину из приобретенных товаров, отсортированных в хронологическом порядке. Выбор этого датасета мотивирован его использованием в исследованиях по предсказанию следующей корзины пользователя [40, 41]. К числу ограничений датасета относится сравнительно небольшое число признаков (как продуктов, так и пользователей), наличие сезонных колебаний (в частности, всплеск покупательской активности в период рождественских праздников), высокая разреженность, низкая гранулярность (доступна лишь информация о дате совершения покупки, однако порядок приобретенных товаров неизвестен).

Датасет TTRS представляет собой анонимизированный массив финансовых транзакций. Как отмечают создатели датасета, на текущий момент он не имеет аналогов и представляет большой интерес для использования в задачах рекомендации. На текущий момент датасет не находится в открытом доступе, право на его использование представляется по запросу. В ходе исследования используется его сокращенная версия, охватывающая период с 1 января 2020 г. по 29 февраля 2020 г.

Для ослабления проблемы разреженности матриц в ходе экспериментов использовались лишь те пользователи, которые сделали не менее 10 покупок, и те товары, которые были куплены не менее 10 раз. На основе каждого датасета был составлен симметричный граф знаний по товарам, включающий в себя 2 вида отношений: is_complement и is_substitute. Отношение is_complement существует между комплементарными товарами - товарами, которые вместе встречаются в одной корзине покупателей (для каждого товара подбирается 5 наиболее частых парных товаров по обучающей выборке). Отношение is_substitute наблюдается между товарами из одной категории, которые можно рассматривать в виде взаимозаменяемых (длина списка товаров-субститутов ограничена 10).

В рамках модели не используются пользовательские признаки, исключительно коллаборативный сигнал по их взаимодействиям с товарами. В качестве признака товара используется его категория.

Разделение на обучающую, валидационную и тестовую выборку происходит по принципу leave-one-out. Последний товар, приобретенный пользователем, попадает в тестовую выборку в качестве целевого. Предпоследний

[1] https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset

по времени товар попадает в валидационную выборку, служащую для предварительной оценки качества модели и подбора гиперпараметров. Наконец, оставшиеся товары формируют обучающую выборку.

| Датасет | Ta-Feng | TTRS |
|---|---|---|
| число покупателей | 19 360 | 20 464 |
| число товаров | 10 561 | 1 354 |
| число взаимодействий | 686 390 | 730 933 |
| число троек (h,r,t) | 139 406 | 36 012 |

Таблица 3.1: Характеристики датасетов

## 3.4 Метрики качества

В данном исследовании ставится задача рекомендации товаров для следующей корзины пользователя. Для каждого пользователя выбирается последний приобретенный товар (ground truth item) из тестовой выборки, в которую входят 24 последних дня из периода. К купленному товару добавляются 99 случайных товаров в качестве негативных примеров, при этом контролируется то, что ни один из них пользователь не приобретал ранее (ни в обучающей, ни в тестовой выборке). Затем каждому из 100 товаров модель присваивает вероятность покупки. По этим вероятностям рассчитывается позиция приобретенного товара. Затем, в зависимости от выбранного параметра $k$, рассчитываются метрики качества рекомендаций. Так, при $k = 5$ рекомендация будет считаться успешной, если купленный товар попадет в топ-5 рекомендаций.

Для оценки качества используются метрики HR@k и NDCG@k [42, 43]. Метрика HR@k (hit ratio) показывает покрытие рекомендаций - соотношение случаев, в которых истинный товар попал в топ-$k$ рекомендаций, к числу всех рекомендаций. При выбранной постановке задачи метрика HR@k может считаться альтернативой Recall@k. Метрика NDCG@k (normalized discounted cumulative gain) оценивает качество ранжирования рекомендаций, присваивая большие веса товарам, попавшим в топ. Формулы расчета метрик приведены ниже:

$$HR@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} I(r_u \leq k),$$

$$NDCG@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{I(r_u \leq k)}{\log_2(r_u + 1)},$$

где $\mathcal{U}$ - множество покупателей, $I(\cdot)$ - индикатор того, что приобретенный товар попал в топ-k рекомендаций, $r_u$ - позиция приобретенного товара в списке рекомендаций, $r_u \in [1, 100]$. Параметр $k$ отвечает за длину списка рекомендаций. В качестве $k$ выбраны значения $5, 10, 15$, отражающие возможное число товаров в покупательской корзине.

# Глава 4

# Эксперименты

В рамках данного исследования ставятся следующие эксперименты:

1. Сравнение рекуррентных моделей, используемых на первом уровне
2. Сравнение способов агрегации динамических эмбеддингов
3. Сравнение с другими графовыми моделями
4. Использование блендинга моделей на втором уровне
5. Сравнение оптимальной конфигурации модели с baseline

В последующих разделах мы подробно остановимся на каждой поставленной задаче.

Логи экспериментов и код модели доступны в репозитории[1]. Во всех экспериментах в качестве оптимизатора использовался Adam, размер батча составлял 256, количество эпох было ограничено 10 (как показывают предварительные запуски модели, этого достаточно для обучения), длина эмбеддингов составляет 40, размер скрытого слоя внимания составляет 10, максимальная длина исторической последовательности транзакций ограничена 20 товарами. Все запуски производились на NVIDIA Tesla V100 GPU.

## 4.1 Сравнение рекуррентных моделей, используемых на первом уровне

В данном эксперименты сравниваются рекуррентные модели первого уровня, используемые для обработки последовательности динамических эмбеддингов: LSTM и GRU. Модели сравниваются по метрикам качества. Результаты эксперимента приводятся в Таблице 4.1.

По результатам эксперимента самое высокое качество по основным метрикам показала основная конфигурация модели KDA. Модель KDA в сочетании с GRU показывает практически то же качество, при этом превосходит по качеству LSTM и требует меньше времени для обучения в силу более простой архитектуры.

[1] https://github.com/alexandreevsky/gnn_recsys

| Ta-Feng Dataset | | | | | | |
|---|---|---|---|---|---|---|
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| KDA | 0.1149 | 0.0808 | 0.1559 | 0.0940 | 0.1849 | 0.1016 |
| $KDA_{LSTM}$ | **0.1347** | **0.0983** | **0.1733** | **0.1109** | **0.1981** | **0.1174** |
| $KDA_{GRU}$ | 0.1306 | 0.0938 | 0.1722 | 0.1071 | 0.1979 | 0.1140 |
| TTRS Dataset | | | | | | |
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| KDA | **0.8039** | **0.7778** | **0.8331** | **0.7872** | **0.8512** | **0.7920** |
| $KDA_{LSTM}$ | 0.7988 | 0.7725 | 0.8282 | 0.7821 | 0.8455 | 0.7866 |
| $KDA_{GRU}$ | 0.8036 | 0.7738 | 0.8307 | 0.7826 | 0.8487 | 0.7873 |

Наилучшее качество в разрезе по выбранным метрикам выделено жирным шрифтом.

Таблица 4.1: Сравнение рекуррентных моделей

## 4.2 Сравнение способов агрегации динамических эмбеддингов

Чтобы агрегировать информацию по всем временным срезам, на первом уровне эмбеддинги обрабатываются рекуррентной моделью. В рамках этого эксперимента предлагается сравнить несколько способов агрегации. Среди них - усреднение, взятие максимума, суммирование:

$$
\begin{aligned}
Mean : m_{u,i} &= \frac{1}{M}\sum_{m=1}^{M} s^{(K)}_{i,\tau_m}, \\
Max : m_{u,i} &= max\left(s^{(K)}_{i,\tau_1}, s^{(K)}_{i,\tau_2}, \ldots, s^{(K)}_{i,\tau_M}\right), \\
Sum : m_{u,i} &= \sum_{m=1}^{M} s^{(K)}_{i,\tau_m}.
\end{aligned}
$$

| TTRS Dataset | | | | | | |
|---|---|---|---|---|---|---|
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| $KDA_{GRU+Sum}$ | **0.8039** | **0.7785** | 0.8303 | **0.7870** | 0.8476 | **0.7916** |
| $KDA_{GRU+Max}$ | 0.8036 | 0.7746 | **0.8308** | 0.7835 | 0.8482 | 0.7881 |
| $KDA_{GRU+Mean}$ | 0.8036 | 0.7738 | 0.8307 | 0.7826 | **0.8487** | 0.7873 |

Таблица 4.2: Сравнение способов агрегации исторических эмбеддингов

По результатам экспериментов наиболее удачным способом агрегирования исторических эмбеддингов является их суммирование.

## 4.3 Сравнение с другими графовыми моделями

Сравним качество на датасетах с графовыми нейронными моделями, предсказывающими следующие взаимодействия пользователей. Часть из них можно считать прямыми аналогами KDA - эти модели также построены по двухуровневой системе с использованием графа знаний и учетом динамики предпочтений (SLRC+ [44], Chorus [35]). Модель TiSASRec [45] формирует динамические эмбеддинги с использованием механизма внимания, однако не использует граф знаний. Сравнение с TiSASRec позволит определить ценность той составляющей, который вносит граф знаний в моделирование пользовательских взаимодействий. Модели SASRec [14] и Caser [46] используют только коллаборативный сигнал, не принимая во внимание динамический компонент.

| Ta-Feng Dataset | | | | | | |
|---|---|---|---|---|---|---|
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| SASRec | 0.0632 | 0.0456 | 0.0849 | 0.0526 | 0.1012 | 0.0569 |
| CASER | 0.0525 | 0.0423 | 0.0686 | 0.0475 | 0.0806 | 0.0507 |
| TiSASRec | 0.0584 | 0.0430 | 0.0846 | 0.0514 | 0.1030 | 0.0562 |
| *SLRC+* | 0.1110 | 0.0785 | 0.1497 | 0.0911 | 0.1710 | 0.0967 |
| KDA | 0.1149 | 0.0808 | 0.1559 | 0.0940 | 0.1849 | 0.1016 |
| $KDA_{LSTM}$ | **0.1347** | **0.0983** | **0.1733** | **0.1109** | **0.1981** | **0.1174** |
| TTRS Dataset | | | | | | |
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| SASRec | 0.6801 | 0.5906 | 0.7522 | 0.6140 | 0.7892 | 0.6238 |
| CASER | 0.5859 | 0.4652 | 0.6868 | 0.4980 | 0.7367 | 0.5112 |
| TiSASRec | 0.6924 | 0.6004 | 0.7593 | 0.6220 | 0.7949 | 0.6314 |
| *SLRC+* | 0.7930 | 0.7635 | 0.8225 | 0.7730 | 0.8415 | 0.7781 |
| Chorus | 0.6131 | 0.4912 | 0.7056 | 0.5213 | 0.7529 | 0.5339 |
| KDA | **0.8039** | **0.7778** | **0.8331** | **0.7872** | **0.8512** | **0.7920** |
| $KDA_{GRU}$ | 0.8036 | 0.7738 | 0.8307 | 0.7826 | 0.8487 | 0.7873 |

Таблица 4.3: Сравнение с бенчмарком

По результатам экспериментов KDA с рекуррентной моделью GRU лишь незначительно уступает стандартной конфигурации KDA. Обе версии архитектуры позволяют достичь наибольшего качества предсказания на рассматриваемых датасетах. Модели, использующие граф знаний, показывают лучшее качество, чем стандартные модели, при этом архитектуры с добавлением темпоральных признаков превосходят остальные по своему качеству.

## 4.4 Блендинг предсказаний моделей на втором уровне

В ходе следующего эксперимента извлечем предсказания 3 предобученных моделей с наилучшим результатом на тестовой выборке, усредним и отранжируем. В результате, мы получим агрегированные рекомендации самых

эффективных моделей, которые теоретически в силу усреднения будут обладать меньшей дисперсией. Далее оценим их качество по тестовой выборке по выбранным метрикам: HR@k и NDCG@k. Результаты сравнения приведены в Таблице 4.4.

| TTRS Dataset | | | | | | |
|---|---|---|---|---|---|---|
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| $SLRC+$ | 0.7930 | 0.7635 | 0.8225 | 0.7730 | 0.8415 | 0.7781 |
| TiSASRec | 0.6924 | 0.6004 | 0.7593 | 0.6220 | 0.7949 | 0.6314 |
| $KDA_{GRU}$ | 0.8036 | 0.7738 | 0.8307 | 0.4798 | 0.8487 | 0.7873 |
| Blending | 0.7361 | 0.8140 | 0.7622 | 0.7578 | 0.8072 | 0.7716 |

Таблица 4.4: Результаты блендинга предсказаний моделей

Как показывает эксперимент, за счет блендинга предсказаний моделей удалось лишь незначительно улучшить качество рекомендаций.

## 4.5 Сравнение оптимальной конфигурации модели с baseline

Как утверждают авторы статьи [47], значительная часть рекомендательных моделей не проходит проверку на воспроизводимость результатов, в действительности уступая по качеству примитивнымуступая по качеству простейшим, «наивным» подходам. Проверим, имеет ли место такой эффект в нашем исследовании. Для этого сравним качество модели со стандартными бейзлайнами: Top Popular, Top Personal. Алгоритм Top Popular [48] предлагает рекомендовать пользователям список из наиболее популярных глобально товаров - место товара в списке определяется частотой его приобретения. Метод Top Personal [49] предлагает каждому пользователю список из товаров, с которыми он ранее взаимодействовал, отранжированных по мере убывания частоты взаимодействий. В качестве меры близости используется косинусная близость. Результаты сравнения приведены в Таблице 4.5.

По итогам эксперимента можно утверждать, что на датасете TTRS и Ta-Feng модель KDA показывает значительно лучшее качество, чем baseline-модели. При этом baseline-модели на датасете Ta-Feng показывают качество, сопоставимое с моделями SASRec и Caser.

| Ta-Feng Dataset | | | | | | |
|---|---|---|---|---|---|---|
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| Top Popular | 0.0454 | 0.0374 | 0.0555 | 0.0407 | 0.0729 | 0.0452 |
| Top Personal | 0.0499 | 0.033 | 0.0749 | 0.0411 | 0.0897 | 0.045 |
| $KDA_{GRU}$ | — | — | — | — | — | — |
| TTRS Dataset | | | | | | |
| Модель | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@15 | NDCG@15 |
| Top Popular | 0.3458 | 0.2428 | 0.4506 | 0.2766 | 0.5158 | 0.2937 |
| Top Personal | 0.6614 | 0.509 | 0.75 | 0.538 | 0.779 | 0.5457 |
| $KDA_{GRU}$ | **0.8036** | **0.7738** | **0.8307** | **0.7826** | **0.8487** | **0.7873** |

Таблица 4.5: Сравнение с baseline

# Глава 5

# Заключение

В рамках исследования были рассмотрены различные подходы к решению задачи рекомендаций по последовательности исторических взаимодействий. Были подробно описаны принципы существующих графовых нейронных моделей, в том числе представлены примеры моделей, использующих граф знаний для обогащения дополнительными сведениями о взаимном влиянии объектов и динамические эмбеддинги для учета вариативности пользовательских предпочтений. В практической части исследования были поставлены эксперименты по модификации архитектуры модели KDA, демонстрирующей SOTA-результаты на публичных датасетах. Были опробованы несколько подходов к обработке эмбеддингов, в частности, несколько способов агрегации и использование рекуррентных моделей для обработки последовательностей. Результаты показывают, что модель показывает сравнительно высокие результаты на транзакционных данных, выучивает основные семантические закономерности на графе знаний и темпоральные предпочтения пользователей по доступной истории взаимодействий. Эти выводы также подтверждаются сравнением с наивными подходами к рекомендации, выступающим в роли sanity check.

Возможным направлением для дальнейших исследований является развитие методов учета динамики пользовательских предпочтений, использование более совершенных методов моделирования временных последовательностей и оценки релевантности объектов.

# Список литературы


[1] Qingyu Guo et al. “A Survey on Knowledge Graph–Based Recommender Systems”. *ArXiv* 2003.00911 (2020).

[2] Changhao Song, Bo Wang, Qinxue Jiang, Yehua Zhang, Ruifang He, and Yuexian Hou. “Social Recommendation with Implicit Social Influence”. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).

[3] Mengqi Zhang, Shu Wu, Xueli Yu, and Liang Wang. “Dynamic Graph Neural Networks for Sequential Recommendation”. *ArXiv* 2104.07368 (2022).

[4] Chong Li, Kunyang Jia, Dan Shen, C.-J. Richard Shi, and Hongxia Yang. “Hierarchical Representation Learning for Bipartite Graphs”. *IJCAI*. 2019.

[5] Shiwen Wu, Wentao Zhang, Fei Sun, and Bin Cui. “Graph Neural Networks in Recommender Systems: A Survey”. *ACM Computing Surveys (CSUR)* (2022).

[6] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio’, and Yoshua Bengio. “Graph Attention Networks”. *ArXiv* 1710.10903 (2018).

[7] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam M. Shroff. “NISER: Normalized Item and Session Representations with Graph Neural Networks”. *ArXiv* 1909.04276 (2019).

[8] William L. Hamilton, Zhitao Ying, and Jure Leskovec. “Inductive Representation Learning on Large Graphs”. *NIPS*. 2017.

[9] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. “Graph Convolutional Neural Networks for Web-Scale Recommender Systems”. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018).

[10] Hongwei Wang et al. “RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems”. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (2018).

[11] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and N. Chawla. “Heterogeneous Graph Neural Network”. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

[12] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. “Gated Graph Sequence Neural Networks”. *CoRR* 1511.05493 (2016).

[13] Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, and Xiaoling Wang. “Temporal Augmented Graph Neural Networks for Session-Based Recommendations”. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).

[14] Wang-Cheng Kang and Julian McAuley. “Self-Attentive Sequential Recommendation”. *2018 IEEE International Conference on Data Mining (ICDM)* (2018), pp. 197–206.

[15] Zhankui He, Handong Zhao, Zhe Lin, Zhaowen Wang, Ajinkya Kale, and Julian McAuley. “Locker: Locally Constrained Self-Attentive Sequential Recommendation”. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021).

[16] Chengfeng Xu et al. “Graph Contextualized Self-Attention Network for Session-based Recommendation”. *IJCAI*. 2019.

[17] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark J. Coates. “GraphSAIL: Graph Structure Aware Incremental Learning for Recommender Systems”. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020).

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. “BPR: Bayesian Personalized Ranking from Implicit Feedback”. *ArXiv* 1205.2618 (2009).

[19] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. “CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems”. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).

[20] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. “Relation-Aware Graph Convolutional Networks for Agent-Initiated Social E-Commerce Recommendation”. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019).

[21] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. “KGAT: Knowledge Graph Attention Network for Recommendation”. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

[22] Tianyu Zhu, Leilei Sun, and Guoqing Chen. “Graph-based Embedding Smoothing for Sequential Recommendation”. *IEEE Transactions on Knowledge and Data Engineering* PP (2021).

[23] Rui Sun et al. “Multi-modal Knowledge Graphs for Recommender Systems”. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020).

[24] Lianghao Xia et al. “Knowledge-Enhanced Hierarchical Graph Transformer Network for Multi-Behavior Recommendation”. *ArXiv* 2110.04000 (2021).

[25] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. “Translating Embeddings for Modeling Multi-relational Data”. *NIPS*. 2013.

[26] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. “Knowledge Graph Embedding by Translating on Hyperplanes”. *AAAI*. 2014.

[27] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. “Learning Entity and Relation Embeddings for Knowledge Graph Completion”. *AAAI*. 2015.

[28] Y. Zhu et al. “What to Do Next: Modeling User Behaviors by Time-LSTM”. *IJCAI*. 2017.

[29] Yunzhe Li et al. “Extracting Attentive Social Temporal Excitation for Sequential Recommendation”. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021).

[30] Cheng-Mao Hsu and Cheng te Li. “RetaGNN: Relational Temporal Attentive Graph Neural Networks for Holistic Sequential Recommendation”. *Proceedings of the Web Conference 2021* (2021).

[31] Jianxin Chang et al. “Sequential Recommendation with Graph Neural Networks”. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).

[32] Liwei Huang, Yutao Ma, Yanbo Liu, Shuliang Wang, and Deyi Li. “Position–enhanced and Time-aware Graph Convolutional Network for Sequential Recommendations”. *ACM Transactions on Information Systems* (2022).

[33] Zeyuan Chen, Wei Zhang, Junchi Yan, Gang Wang, and Jianyong Wang. “Learning Dual Dynamic Representations on Time-Sliced User-Item Interaction Graphs for Sequential Recommendation”. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021).

[34] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S. Yu. “Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer”. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021).

[35] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. “Make It a Chorus: Knowledge- and Time-aware Item Modeling for Sequential Recommendation”. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).

[36] Mehrnaz Amjadi, Danial Mohseni-Taheri, and Theja Tulabandhula. “KATRec: Knowledge Aware aTtentive Sequential Recommendations”. *DS*. 2021.

[37] Haiyan Wang, Kai-Lang Yao, Jian Luo, and Yi Lin. “An Implicit Preference-Aware Sequential Recommendation Method Based on Knowledge Graph”. *Wirel. Commun. Mob. Comput.* 2021 (2021), 5206228:1–5206228:12.

[38] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. “Embedding Entities and Relations for Learning and Inference in Knowledge Bases”. *CoRR* 1412.6575 (2015).

[39] Sergey Kolesnikov, Oleg Lashinin, Michail Pechatov, and Alexander Kosov. “TTRS: Tinkoff Transactions Recommender System benchmark”. *ArXiv* 2110.05589 (2021).

[40] Mathias Kraus and Stefan Feuerriegel. “Personalized Purchase Prediction of Market Baskets with Wasserstein-Based Sequence Matching”. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

[41] Shoujin Wang, Liang Hu, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Longbing Cao. “Modeling Multi-Purpose Sessions for Next-Item Recommendations via Mixture-Channel Purpose Routing Networks”. *IJCAI*. 2019.

[42] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. “Evaluating collaborative filtering recommender systems”. *ACM Trans. Inf. Syst.* 22 (2004), pp. 5–53.

[43] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. “How good your recommender system is? A survey on evaluations in recommendation”. *International Journal of Machine Learning and Cybernetics* 10 (2019), pp. 813–831.

[44] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. “Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems”. *The World Wide Web Conference* (2019).

[45] Jiacheng Li, Yujie Wang, and Julian McAuley. “Time Interval Aware Self-Attention for Sequential Recommendation”. *Proceedings of the 13th International Conference on Web Search and Data Mining* (2020).

[46] Jiaxi Tang and Ke Wang. “Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding”. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018).

[47] Maurizio Ferrari Dacrema, Paolo Cremonesi, and D. Jannach. “Are we really making much progress? A worrying analysis of recent neural recommendation approaches”. *Proceedings of the 13th ACM Conference on Recommender Systems* (2019).

[48] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. “A Re-visit of the Popularity Baseline in Recommender Systems”. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).

[49] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. “Modeling Personalized Item Frequency Information for Next-basket Recommendation”. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).