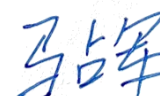


Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет науки и
технологий имени академика М.Ф. Решетнева»

На правах рукописи



Ма Чжаньцзюнь

**МЕТОДЫ ОБУЧЕНИЯ ГРАФА ЗНАНИЙ НА ОСНОВЕ
ОПТИМИЗАЦИИ СТРУКТУРЫ ГРАФА И ХЭШИРОВАНИЯ**

2.3.1 – Системный анализ, управление и обработка информации, статистика

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель
доктор технических наук,
профессор
Казаковцев Л.А.

Красноярск – 2026

Оглавление

ВВЕДЕНИЕ	4
1 ГРАФЫ ЗНАНИЙ И ИХ ОБУЧЕНИЕ	12
1.1 Базы и графы знаний	12
1.2 Текущее состояние исследований в области алгоритмов обучения представлению графа знаний	21
1.3 Базы данных графов знаний	34
1.4 Комбинаторные рекомендации: постановка задачи и примеры	37
1.5 Основные методы подбора комбинаций	41
1.6 Области применения комбинаторных рекомендаций.....	47
1.7 Выводы по главе 1	48
2 ОБУЧЕНИЕ ПРЕДСТАВЛЕНИЮ ГРАФОВ ЗНАНИЙ ДЛЯ НЕСВЯЗАННЫХ ГРАФОВ.....	54
2.1 Проблема несвязности в графах знаний.....	54
2.2 Графовые нейронные сети.....	59
2.3 Многореляционная графовая нейронная сеть на основе максимизации взаимной информации: общая структура Dual-FusionKG.....	66
2.4 Сеть внимания на основе графов слияния сущностей и связей и оценки максимальной взаимной информации.....	68
2.5 Экспериментальная оценка алгоритма Dual-FusionKG	80
2.6 Результаты главы 2.....	95
3 ОБУЧЕНИЕ ПРЕДСТАВЛЕНИЮ МУЛЬТИМОДАЛЬНЫХ ГРАФОВ ЗНАНИЙ.....	97
3.1 Мультимодальность в графах знаний и обзор методов.....	97
3.2 Алгоритм Hyperfusion-Net: общая структура и модуль слияния информации.....	101
3.3 Модуль агрегирования и проверки прогнозов в HyperFusion-Net.....	109
3.4 Экспериментальная оценка и анализ гиперпараметров	113
3.5 Результаты главы 3	127
4 АЛГОРИТМ КОМБИНАТОРНЫХ РЕКОМЕНДАЦИЙ НА ОСНОВЕ ХЭШИРОВАНИЯ	129
4.1 Задачи комбинаторных рекомендаций и методы хэширования	129

4.2	Технология хэширования и персонализированное сопоставление комбинаций на основе хэширования	139
4.3	Совместимость комбинаций и хэш-обучение.....	141
4.4	Целевая функция и мультимодальность	149
4.5	Экспериментальная оценка и анализ модели	152
4.6	Результаты главы 4	169
ЗАКЛЮЧЕНИЕ		171
ПРИЛОЖЕНИЕ А		187

ВВЕДЕНИЕ

Актуальность. Современные системы искусственного интеллекта, такие как поисковые системы, рекомендательные сервисы и вопросно-ответные комплексы, опираются на структурированное представление знаний о реальном мире. Графы знаний, также известные как семантические сети или онтологии, решают эту задачу, описывая связи между сущностями с помощью графовых структур, что позволяет моделировать сложные семантические отношения. Графовые нейронные сети (GNN) эффективно обучают представления сущностей, распространяя информацию по структуре графа через механизмы передачи сообщений. Однако существующие методы имеют ограничения: Во-первых, автоматически создаваемые графы знаний содержат множество несвязанных подграфов и изолированных узлов, что приводит к искаженным представлениям сущностей с «длинным хвостом» и ограничивает возможности извлечения глобальной структурной информации. Во-вторых, современные графы знаний содержат мультимодальные данные, но существующие методы агрегируют только отдельные триплетные признаки, игнорируя взаимодействия сущностей внутри структуры графа. Поэтому разработка алгоритмов обучения представлений, которые балансируют между характеристиками структуры графа и мультимодальным слиянием, имеет важное исследовательское значение. В-третьих, персонализированные рекомендации должны обрабатывать комбинации продуктов, а не отдельные продукты, что приводит к экспоненциальному росту пространства комбинаций и требует разработки более эффективных алгоритмов для вычисления комбинаций. Технологические прорывы в области графов знаний могут улучшить интерпретируемость рассуждений в таких областях, как медицинская диагностика; технологические прорывы в области рекомендательных систем могут вырабатывать предложения сопутствующих услуг и товаров в реальном

времени для платформ электронной коммерции с сотнями миллионов пользователей. Вместе они способствуют совместной эволюции когнитивного интеллекта и интеллекта принятия решений.

Степень разработанности тематики. В области обучения представлению графов знаний исследователи предложили различные методы и алгоритмы для решения таких проблем, как несвязность и многомодальное слияние. Среди методов, основанных на трансляции, в методе TransE Бордеса и др. (2013) [1] впервые применена парадигма моделирования векторной трансляции. Метод TransH Ванга и др. (2014) [2] обрабатывает сложные отношения посредством проекции гиперплоскости, TransR Лина и др. (2015) [3] вводит пространство, специфичное для отношений, а RotatE [4] Суна и др. (2019) моделирует отношения как сложные операции вращения в пространстве. Методы, основанные на тензорной декомпозиции, включают RESCAL Никеля и др. (2011) [5], декомпозицию Tucker [6], упомянутое в работе Колды и др. (2009) и ComplEx Труйона и др. (2016) [7], использующий сложные вложения. Что касается методов нейронных сетей, Деттмерс и др. В работе ConvE (2018) используются сверточные сети [8], а в работе НурЕР (2019) [9] Балазевича и др. используются гиперсети для генерации сверточных ядер, специфичных для отношений. В области графовых нейронных сетей Шлихткрулл и др. (2018) [10] впервые реализовали реляционную свертку графов в своей работе RGCN, а Вашишт и др. (2020) [11] улучшили производительность за счет совместного встраивания сущностей и отношений. Методы на основе трансформеров, такие как KG-BERT (2019) [12] Яо и др., CoKE (2019) [13] Ванга и др. и MKGformer (2021) Лю и др [14], используют предварительно обученные языковые модели для многомодального рассуждения. Однако существующие графы знаний часто неполны из-за проблем с шумом, создаваемых алгоритмами ручного или автоматического построения, что приводит к несвязным структурам графов. Эта проблема препятствует эффективному использованию существующих графовых нейронных сетей, основанных на принципах распространения

информации, в представлении графов знаний, то есть они не могут эффективно улавливать полную структурную информацию графов знаний. Современное обучение представлению графов знаний на основе графовых нейронных сетей в основном ориентировано на одномодальные графы знаний. Многомодальные взаимодействия в графах фрагментированы, и сущности в графах знаний не ограничиваются текстовыми описаниями, но могут также включать информацию в нескольких модальностях, таких как изображения и речь. В области рекомендаций пакетов услуг Сар Шалом и др. (2016) [15] улучшили показатели кликабельности пользователей с помощью коллаборативной фильтрации. Лю Г. и др. (2017) [16] оптимизировали производительность рекомендаций пакетов услуг с помощью своей модели VRM. Вэй П. и др. (2022) [17] улучшили полноту рекомендаций с помощью своего алгоритма персонализированного объединения. Вейт [18] А. и др. (2015) использовали сиамские сети для повышения точности рекомендаций пакетов услуг. Тансенг и др. В работе (2020) [19] значительно улучшен эффект рекомендаций по комплектации с помощью сквозного метода. Однако в реальных системах огромное количество элементов обычно увеличивает пространство комбинаций, что делает вычислительную эффективность критически важной проблемой. Поэтому обучение несвязанных графов, слияние мультимодальных графов и обеспечение высокой производительности рекомендательных систем остаются актуальными задачами.

Объектом исследования являются задачи обучения графов знаний, **предметом исследования** – являются алгоритмы для решения данных задач.

Цель исследования – повышение точности и вычислительной эффективности решения задач дополнения графов знаний и комбинаторных рекомендаций за счет разработки новых методов обучения представлений, учитывающих структурную несвязность, мультимодальность информации и необходимость работы в реальном времени.

Задачи исследования:

1. Разработать адаптивный алгоритм внимания для графа знаний (Dual-FusionKG), основанный на максимизации взаимной информации, улучшающий способность к обучению представлений несвязных графов посредством структурно-семантического двухканального слияния и исследовать его эффективность по показателям MRR и Hits@10 на наборах данных FB15k, WN18, FB15k237 и WN18RR.

2. Разработать алгоритм, основанный на слиянии тензоров низкого ранга (HyperFusion-Net) с использованием сложных пространственных операций взаимной корреляции для моделирования направленности мультимодальных связей, повышающий эффективность кросс-модального взаимодействия по показателям показателя Hits@10 и MRR на наборах данных FB15k-237, WN18RR, DB15K и YAGO15K.

3. Разработать алгоритм, совмещающий использование взвешенного хэша и алгоритм вероятностного динамического кодирования (MCHM-Net), сокращающий время ответа рекомендательной системы в режиме реального времени с помощью механизма поиска по двоичным векторам, исследовать эффективность на наборе данных Polyvore-U по показателям AUC (площадь под кривой) и NDCG (нормализованный дисконтированный кумулятивный прирост).

Новые научные результаты:

1. Разработан новый алгоритм Dual-FusionKG для обучения представлений в несвязных статических графах знаний, который, в отличие от известных методов, основан на использовании двухканальной архитектуры сети внимания (ERGAT) с максимизацией взаимной информации между локальными и глобальными представлениями графа, что позволяет эффективно извлекать структурную информацию даже при отсутствии прямых связей между сущностями.

2. Предложен новый алгоритм HyperFusion-Net для обучения представлений мультимодальных графов знаний, основанный на низкоранговом тензорном слиянии текстовых, визуальных и числовых признаков с последующей агрегацией структурной информации через сеть внимания ERGAT, что обеспечивает учет как межмодальных взаимодействий, так и направленности связей в графе.

3. Разработан новый алгоритм персонализированных комбинаторных рекомендаций MCHM-Net, использующий взвешенное хэширование и вероятностное кодирование на основе распределения Бернулли для представления пользователей и элементов в виде бинарных векторов, что позволяет достигать микросекундного времени отклика за счет табличного поиска при сохранении высокой точности рекомендаций.

Теоретическая значимость. Результаты обогащают методы структурно-семантического слияния, моделирования сложного пространства, мультимодального взаимодействия и эффективных рекомендаций хэша. В Dual-FusionKG впервые вводится механизм максимальной взаимной информации на графах для несвязных графов знаний, расширяя границы применения методов, основанных на взаимной информации в обучении графовых структур; HyperFusion-Net конструирует реляционную структуру моделирования, основанную на комплексном пространстве, эффективно объединяя технологию разложения тензоров низкого ранга и предоставляет новую теоретическую перспективу для многомодального слияния; а MCHM-Net вводит механизм совместимости взвешенного хэша и вероятностного динамического кодирования в комбинаторных рекомендациях, что решает ключевую проблему недостаточной гибкости традиционных методов хэширования. Вместе эти подходы способствуют развитию теории обучения представлениям в области мультимодального слияния информации, моделирования структурной неоднородности и эффективных рассуждений.

Практическая значимость. Предложенные алгоритмы имеют ценность в практических задачах, таких как завершение графа знаний и комбинаторные рекомендации. Так, алгоритм DualFusion-KG применим для завершения неполной информации в областях здравоохранения, финансов и т. д.; алгоритм HyperFusion-Net, способный точно моделировать семантические различия в мультимодальных средах данных, применим для совместного вывода текстовых, графических и структурных данных в социальных сетях, в системах интеллектуальных вопросов-ответов и в других областях; алгоритм MSHM-Net за счет способности точно моделировать семантические различия, подходит для совместного рассуждения о текстовых, графических и структурных данных в социальных сетях, интеллектуальных системах вопрос-ответ и др., повышая производительность и масштабируемость системы комбинаторных рекомендаций за счет эффективного и динамического механизма хеширования, который широко применим в электронной коммерции, коротких видео, новостных рассылках и других сценариях приложений с высокими требованиями к реальному времени и персонализации.

Алгоритм MSHM-NET внедрен в эксплуатацию составе рекомендательной системы технологической компанией "Бэйцзин Циху Кэцзи".

Методы исследования: В диссертации применяются методы из следующих научных направлений: теория графов и машинное обучение на графах (в частности, графовые нейронные сети); многомодальное обучение и слияние данных (использование предобученных моделей VGG-16, BERT, а также предложенный метод низкорангового тензорного слияния HyperFusion-Net); приближенный поиск и хэширование (методы обучения хэшированию, включая детерминированное и вероятностное на основе распределения Бернулли); анализ и обработка данных (статистические методы, метрики оценки AUC, MRR, Hits@K и др.). Конкретные технологии,

реализованные в рамках этих направлений, включают алгоритмы Dual-FusionKG, HyperFusion-Net и MCHM-Net, которые легли в основу экспериментальной части работы.

Положения, выносимые на защиту:

1. Алгоритм Dual-FusionKG позволяет улучшить структурную осведомлённость при сохранении возможностей семантического моделирования за счет двухканального механизма взаимной информации. По сравнению с традиционными методами графовых нейросетей (такими как RGCN и SACN), он демонстрирует преимущества в задачах завершения (устранения несвязности) графа знаний (предсказание недостающих триплетов и определение подлинности заданных триплетов) по показателям $MRR=0,362$ и $Hits@10$.

2. Алгоритм HyperFusion-Net демонстрирует преимущества в мультимодальном динамическом слиянии и моделировании направления связей, достигая максимального значения $Hits@10=0,694$ в задаче завершения графа знаний в эталонных наборах данных, что представляет собой улучшение в среднем на 9,6% по сравнению с лучшими известными моделями, при этом значение MRR улучшается на 7,2%.

3. Алгоритм MCHM-Net позволяет достигать времени отклика рекомендательной системы на уровне микросекунд, представляя рекомендуемые элементы и пользователей рекомендательной системы в виде двоичных векторов, сохраняя при этом точность персонализированных рекомендаций. На наборе данных Polyvore-U значения AUC (площадь под ROC-кривой) и NDCG достигают 0,9237 и 0,8674 соответственно, что значительно превосходит базовые алгоритмы (такие как FPITF и Bi-LSTM); при использовании алгоритма быстрой таблицы поиска время вычисления одной рекомендации составило всего 1,66 микросекунды.

Соответствие паспорту специальности. Работа соответствует п.4 (разработка методов и алгоритмов решения задач системного анализа,

оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта).

Апробация результатов. Результаты обсуждались на международных научно-практических конференциях «Инновационные исследования: опыт, проблемы внедрения результатов и пути решения» (г.Уфа, 2025), Database and Big Data Workshop (Москва, 2024), «Актуальные проблемы авиации и космонавтики» (г.Красноярск, 2021 и 2023 гг.), Прикладная физика и математика (ААРМ-IV 2025, г.Бухара, Узбекистан), ВНК «Цифровое общество: научные инициативы и новые вызовы» (2025, н.Москва), V Всероссийская (национальная) научная конференция «Достижения науки и технологий» (ДНиТ-V-2026, Красноярск).

Публикации. Результаты опубликованы в 16 работах, среди которых 7 статей в журналах, рекомендованных ВАК.

Структура и объем диссертации. Диссертация состоит из введения, 4 глав, заключения и приложения. Диссертационная работа представлена на 188 страницах, содержит 28 таблиц, 28 рисунков и приложение. Список литературных источников содержит 173 наименования.

1 ГРАФЫ ЗНАНИЙ И ИХ ОБУЧЕНИЕ

1.1 Базы и графы знаний

Техническое развитие картирования (систематизации) знаний можно проследить с середины XX века. В 1955 году Гарфилд и др. [20] предложили использовать сетевую цитируемость как инструмент для корреляционного анализа академической литературы, и этот ранний прототип «картирования знаний» по сути является визуализацией области знаний, хотя и существенно отличается от современного картирования знаний. В 1977 году, на Пятой международной конференции по искусственному интеллекту, была официально сформулирована технология баз знаний на основе экспертных систем, что представляло новый этап структурированной обработки знаний. Ключевым поворотным моментом в развитии технологии стал 2012 год, когда команда Google [21] впервые явно предложила концепцию Knowledge Graph (граф знаний) и реализовала прорыв в коммерческом применении сервисов знаний с помощью технологии семантического поиска. С точки зрения эволюции технологий, Knowledge Graph возник из системы исследований семантической паутины: архитектура Linked Data, предложенная Бернерсом-Ли в 2016 году [22], устанавливает ключевые технические спецификации, а её основными элементами являются три компонента: унифицированные идентификаторы ресурсов (URI), RDF (Resource Description Framework) и OWL (Web Ontology Language).

Графы знаний (Knowledge Graphs), представляющие собой структурированные формы семантических описаний сущностей и их взаимосвязей, в русскоязычной научной традиции также известны под терминами семантические сети и онтологии [23,24].

Кораблиновым В. И. Браславским П. в 2020 году был собран один из первых российских больших наборов данных для систем вопрос-ответ

RuBQ, предназначенный для тестирования систем вопрос-ответ, основанных на графах знаний Wikidata [25]. Ярошко Т., Коса В. и Игнатенко О. в 2024 г. предложили метод построения для построения графов открытых исследовательских знаний на основе академических публикаций и применения их в области исследований по борьбе с коррупцией [26]. Атаева О.М., Массель Л.В. и Серебряков В.А. в 2024 году создали междисциплинарный граф знаний научных журналов, используя методы интеллектуального анализа данных для достижения интеллектуальной навигации по контенту [27].

Кроме того, Сальников М., Ле Х., Раджпут П. и Никишина И. в 2023 г. исследовали механизм слияния больших языковых моделей и графов знаний, и предложили метод ответа на фактические вопросы на основе архитектуры Transformer и провели специальную его настройку с учетом особенностей русского языка [28]. В совокупности эти исследования способствовали эволюции парадигмы от традиционной инженерии знаний к современным технологиям графов знаний, что в конечном итоге привело к появлению новой системы представления знаний, интегрирующей граф знаний (семантическую сеть) и машинное обучение.

Граф знаний, как система структурированного представления знаний, по сути является представлением базы знаний, построенным на основе графовой структуры [29]. Система опирается на два ключевых элемента – сущность и отношение – для моделирования реального мира: Сущность используется для обозначения конкретных объектов или абстрактных понятий, а отношение – для описания семантических связей между сущностями [30]. С разных точек зрения графы знаний обладают множеством свойств: с семантической точки зрения это собрание знаний, состоящее из фактических троек; с точки зрения теории графов они представлены в виде семантической сети, состоящей из узлов (сущностей) и ребер (отношений). Как показано на рисунке 1.1, факт (Китай, столица, Пекин) может быть записан в виде тройки (триплета) в базе

знаний или преобразован в путь «Китай→Столица→Пекин» в ориентированном графе, и обе формы выражения полностью эквивалентны на семантическом уровне.

База знаний
(триплеты фактов)

(Китай, Столица, Пекин)
 (Китай, Население, 1,412 миллиона)
 (Китай, Территория, 9,600,000 кв.км)
 (США, Столица, Вашингтон)
 (США, Население, 333 миллиона)
 (США, Территория, 9,370,000 кв. км)
 (Великобритания, Столица, Лондон)
 (Великобритания, Население, 67 миллионов)
 (Великобритания, Территория, 244,100 кв.км)
 (Россия, Столица, Москва)
 (Россия, Население, 146 миллионов)
 (Россия, Территория, 17,098,242 кв.км)

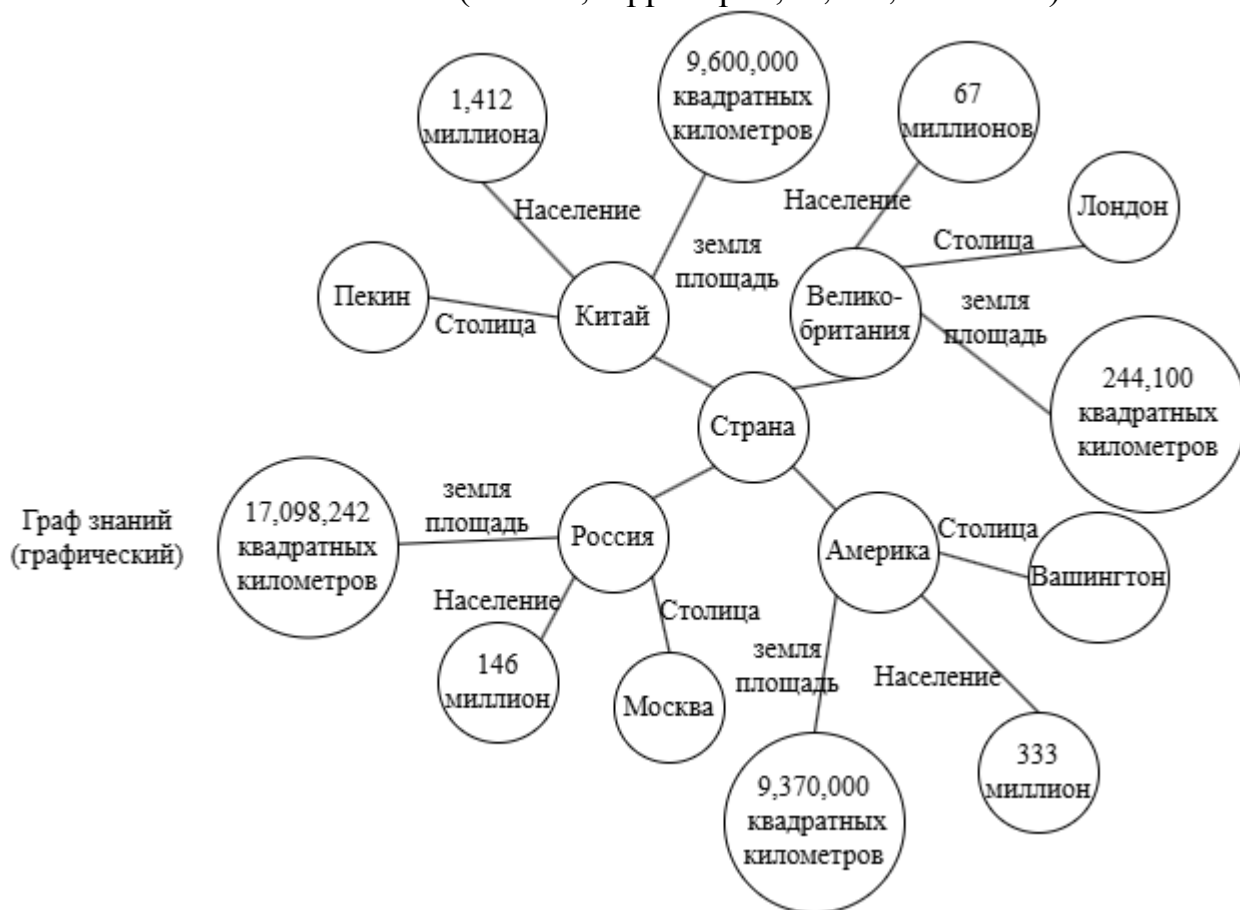


Рисунок 1.1 – База знаний и диаграмма графа знаний (в качестве примера – данные National Geographic)

Основное преимущество графа знаний заключается в реализации логики рассуждений на основе знаний через обучение представлением, что наделяет ИИ-системы (системы искусственного интеллекта) способностями к обработке знаний, подобными человеческим. Процесс его создания в основном состоит в извлечении связей между сущностями из неструктурированного текста и их интеграции для формирования масштабной сети знаний (рисунок 1.1).

После того, как Google в 2012 году запустил свой первый коммерческий продукт на основе графов знаний Knowledge Vault [31], технология стремительно развивалась по всему миру: системы Microsoft Bing Satori, платформы Baidu Knowing Heart и Knowledge Cube от Sogou [32] выходили одна за другой, демонстрируя, что граф знаний стал ключевым элементом инфраструктуры в эпоху искусственного интеллекта. Эти системы существенно повышают точность и интерпретируемость информационных сервисов за счет структурированного представления знаний.

Практика инженерного применения коммерциализированных графов знаний стимулировала академические исследования в этой области. Как показано на рисунке 1.2, текущие исследования графов знаний в основном сосредоточены на двух основных направлениях: методах построения графов знаний и методах обучения представлений графов знаний.

В процессе фактического построения графа знаний обычно используются следующие три типа источников данных для извлечения и интеграции знаний: во-первых, структурированные текстовые данные, имеющие чёткую спецификацию формата, типичными представителями которых являются различные виды баз данных. Во-вторых, массивные неструктурированные текстовые данные, включающие новостные отчёты, контент социальных сетей (например, микроблоги) и различные документы. Наконец, существуют полуструктурированные текстовые данные, которые объединяют как структурированные, так и неструктурированные

характеристики, включая расширяемый язык разметки (XML), облегченный формат обмена данными (JSON) и открытые источники знаний, такие как Википедия.



Рисунок 1.2 – Схематическая диаграмма содержания исследований по картированию знаний

После применения ключевых методов, таких как распознавание именованных сущностей [33], извлечение информации [34] и извлечение атрибутов [35], для извлечения базовых элементов, таких как сущности, взаимосвязи и атрибуты, из необработанных данных, построенный граф знаний можно рассматривать лишь как ряд дискретных коллекций триплетов фактов. Как показано на рисунке 1.2, эти базовые единицы фактов недостаточны для формирования полноценной системы знаний. Чтобы создать по-настоящему пригодный к использованию граф знаний, необходимо провести обучение представлений знаний на полученных триплетах, а также оценить и оптимизировать результаты построения с помощью таких последующих этапов обработки, как логика знаний, выравнивание сущностей, извлечение понятий и других, прежде чем они могут быть внедрены в

практические сценарии применения.

Основная цель изучения представления графов знаний [36] заключается в том, чтобы сущности и отношения, которые в исходном символьном виде непригодны для эффективных численных операций и не отражают семантической близости между объектами, преобразовать в числовые представления в векторном пространстве низкой размерности с помощью определённой модели обучения представлению. Это преобразование не только должно сохранять семантические особенности исходных данных, но и обеспечивать возможность применения к ним операций в векторном пространстве, что необходимо для последующего автоматизированного анализа и вывода. После получения векторных представлений сущностей и отношений обычно необходимо разработать функцию оценки, чтобы оценить достоверность построенных триплетов фактов. Наиболее представительной является функция оценки на основе расстояния, предложенная моделью TransE [1]: $\|h + r - t\|$. Процесс реализации этого метода таков: сначала вектор головного объекта h , вектор хвостового объекта t и вектор отношения r генерируются энкодером, а затем применяется вышеуказанная функция оценки для определения истинности троичной структуры – если оценка равна 0, то она определяется как ложная, а если оценка равна 1, то как истинная. Таким образом, исследование обучения представления знаний в графах сосредоточено на том, как разработать эффективные энкодеры для достижения точного векторного представления объектов и отношений.

В настоящее время основные энкодеры для обучения представлений графов знаний включают в себя следующие архитектуры: многослойный перцептрон (MLP) [31], сверточную нейронную сеть (CNN) [8], графовую нейронную сеть (GNN) [10] и модели с механизмом самовнимания [12]. Поскольку графы знаний по своей сути относятся к данным с графовой структурой, использование технологий графовых нейронных сетей для моделирования ассоциативных характеристик между сущностями и связями

позволяет полностью реализовать синергетические преимущества обоих подходов и достичь лучшего представления знаний. Преимущества этой комбинации в основном проявляются в двух аспектах: во-первых, графовые нейронные сети естественным образом подходят для работы с данными, структурированными в виде графов, и могут эффективно обучать векторное представление узлов (сущностей) и связей (отношений); во-вторых, графовые нейронные сети обладают мощными возможностями распространения информации и рассуждения, что позволяет им лучше поддерживать различные сценарии применения знаний, такие как классификация новостных текстов, рекомендательные системы с использованием знаний и т.д. Однако следует отметить, что текущая модель обучения представления знаний на основе графовых нейронных сетей все еще имеет ряд проблем и ограничений, которые необходимо решить.

Во-первых, необходимо решить проблему присущей неполноты графов знаний. Существующие графы знаний в основном создаются с помощью ручной аннотации или автоматизированных алгоритмов, которые страдают либо от недостаточности извлечения знаний, либо от внесения ошибок, что приводит к разреженным или разъединённым структурам графов. Хотя в существующих исследованиях обычно используются графовые нейронные сети для моделирования графов знаний, большинство из них просто преобразуют базу знаний в структуры графа для обработки, что неэффективно решает проблему смещения обучения представлений, вызванного разрывами в графе.

Во-вторых, возможности моделирования для специальных типов графов знаний недостаточны. С бурным ростом мультимодальных данных мультимодальные графы знаний, объединяющие текстовую, визуальную и другую многоканальную информацию, становятся все более важными. Существующие методы обычно используют стратегию простого объединения после извлечения субмодальных признаков, что позволяет работать с

мультимодальной информацией внутри одного триплета, но игнорирует общие графо-структурные свойства графов знаний. Кроме того, реальные знания часто имеют вероятностный характер, и традиционные методы приводят к трудностям эффективного моделирования таких неопределённых знаний.

Наконец, в области графов знаний все еще есть возможности для оптимизации на уровне приложений. Например, в системе рекомендаций современные основные методы по-прежнему ограничиваются введением предварительно обученного представления графа знаний в качестве внешней функции, не достигая глубокой интеграции графа знаний и системы рекомендаций. Хотя такие модели, как KGAT, пытаются строить гибридные графы «знания-пользователь-товар», стратегия отрицательной выборки, которую они используют, приводит к неэффективному обучению и затрудняет соответствие требованиям практического применения. Такой формальный подход к применению ограничивает полное раскрытие ценности графа знаний.

В качестве ответа на вышеуказанные проблемы имеет большое теоретическое и практическое значение проведение исследований по обучению представлений на основе структуры графа. В этом исследовании уделено внимание разработке более продвинутых архитектур графовых нейронных сетей для улучшения обучения представлений графов знаний.

Традиционные рекомендательные системы в основном рассчитаны на то, чтобы рекомендовать отдельные элементы, такие как фильмы и телепередачи, публикации или новостную информацию. На раннем этапе развития технологий алгоритмы рекомендаций, основанные на релевантности предметов [37], широко применялись в практических бизнес-сценариях благодаря своей отличной масштабируемости и вычислительной эффективности. Однако эта модель рекомендаций для отдельных элементов фокусируется только на индивидуальных предпочтениях пользователя и часто

игнорирует синергетические соответствия между элементами. Например, при подборе одежды пользователь может предпочитать каждый отдельный элемент набора, но стильное сочетание этих элементов может не быть согласованным. Аналогично, в сценарии рекомендаций для дома необходимо учитывать стили дизайна и функциональное взаимодействие различных мебельных изделий комплексно, а не рекомендовать отдельные продукты по отдельности. Эта задача рекомендации, которая подчеркивает синергетические отношения между несколькими предметами, в академическом сообществе определяется как комбинаторная рекомендация [38].

Ключевая особенность комбинаторных рекомендаций заключается в том, что несколько элементов в результате рекомендации образуют органическое целое. Однако число возможных комбинаций растет экспоненциально с увеличением размера комбинации, что не только усугубляет проблему информационной перегрузки, но и усложняет задачу оптимизации рекомендательной системы. В частности, комбинаторные рекомендации сталкиваются со следующими основными проблемами: во-первых, необходимо создать эффективную модель оценки совместимости элементов; во-вторых, требуется решить проблему вычислительной сложности, вызванную комбинаторным взрывом; и, наконец, необходимо балансировать между индивидуальными предпочтениями пользователей и общей координацией комбинаций. Эти особенности делают комбинаторные рекомендации сложным направлением исследований в области рекомендательных систем. Ранее нами были исследованы методы кластеризации текстовых данных, которые могут быть использованы при построении графов знаний из неструктурированных текстов [39].

1.2 Текущее состояние исследований в области алгоритмов обучения представлению графа знаний

Обучение представлению графа знаний в основном отображает сущности и отношения в графе знаний в определенное математическое пространство с помощью конкретной модели кодирования, а после получения его низкоразмерного векторного представления, его способность к представлению затем оценивается через задачи нижнего уровня, такие как логические рассуждения о взаимосвязях и дополнение графа знаний. В зависимости от различных подходов к кодированию, существующие методы обучения для представления графа знаний можно разделить на пять групп: модели представления знаний, основанные на моделях трансляции [40,41], модели представления знаний, основанные на разложении матриц [5,42], модели представления знаний, основанные на нейронных сетях [43-45], модели представления знаний, основанные на графовых нейронных сетях [46,47], и модели представления знаний [12], основанные на модели самовнимания для представления знаний. Для последующего обсуждения основные математические обозначения, используемые в этом разделе, сведены в таблицу 1.1.

Таблица 1.1 – Представление карты знаний с описанием символов, связанных с обучением

Обозначение	Описание
G	Граф знаний
\mathcal{E}	Набор сущностей
\mathcal{R}	Набор отношений
\mathcal{F}	Набор триплетов фактов
h, t, r	Головная сущность, хвостовая сущность и отношение
$f_r(h, t)$	Функция оценки (для триплета фактов)
w	Весовая матрица

В следующих подразделах описаны основные подходы, применяемые в данной области.

Модель представления знаний на основе модели трансляции

Предложенная Бордесом и др. модель TransE [1] использует инновационный подход к представлению знаний, проецируя сущности и отношения в графе знаний в векторное пространство низкой размерности. Основной механизм модели, показанный на рисунке 1.3, рассматривает отношения в каждом триплете (головная сущность, отношение, хвостовая сущность) как векторы переноса, соединяющие головную и хвостовую сущности. В ходе итеративного процесса оптимизации сумма вектора головной сущности и вектора отношения делается максимально близкой к вектору хвостовой сущности.

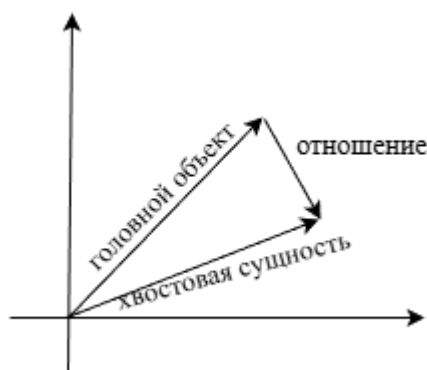


Рисунок 1.3 – Схема TransE на основе моделирования перевода

Этот подход к моделированию позволяет совместно использовать эмбединги (векторные образы) сущностей и отношений в одном и том же векторном пространстве. Математическое выражение модели можно записать следующим образом:

$$f_r(h, t) = \|h + r - t\|_{L_1/L_2}. \quad (1.1)$$

Здесь L_1/L_2 означает, что может использоваться норма L_1 или L_2 .

На основе модели TransE в последующих исследованиях были разработаны различные улучшенные методы обучения представлений. Модель TransH [2] оптимизирует работу TransE с учетом ограничений при работе с отношениями типа один-ко-многим, многие-к-одному и

многие-ко-многим. Модель предполагает, что хотя сущности и отношения по-прежнему разделяют одно и то же линейное пространство представлений, сущности в рамках разных отношений будут отображаться на разные гиперплоскости с помощью специальной матрицы весов отношения w_r .

$$f_r(h, t) = -\| (h - w_r^T h w_r) + r - (t - w_r^T t w_r) \|_2^2. \quad (1.2)$$

С другой стороны, TransR [3] исследует представление сущностей в графе знаний, классифицируя их в соответствии с их отношениями, и осуществляет преобразование в соответствующем реляционном пространстве через матрицу реляционного отображения M_r :

$$f_r(h, t) = -\| M_r h + r - M_r t \|_2^2. \quad (1.3)$$

В противоположность TransR (который считается слишком сложным), TransD [48] упрощает матрицу отображения отношений в пространство отношений, используя произведение двух векторов.

$$f_r(h, t) = -\| (rh^T + I)h + r - (rt^T + I)t \|_2^2. \quad (1.4)$$

Он разлагает M_r в (1.3) на $M_{rh} = rh^T + I$ и $M_{rt} = rt^T + I$. TranSparse [49] использует матрицы с разным числом параметров для обучения представлений, чтобы отражать различные сложные взаимосвязи. В то же время, TranSparse использует разные матрицы отображения для представления головной и хвостовой сущностей, чтобы решить проблему их неравного числа.

Существуют также работы, опирающиеся на идею моделирования трансляции, и утверждающие, что линейное пространство слишком простое для того, чтобы представить сложность графов знаний. В этих работах достигается лучшая способность к представлению знаний путем отображения сущностей и отношений в нелинейное пространство. ComplEx [7], использующий метод гамильтонова произведения, является первым подходом к реализации обучения представления графов знаний в комплексном пространстве, с преимуществом моделирования симметричной и

антисимметричной природы отношений:

$$f_r(h, t) = \text{Re}(h^T \text{diag}(M_r) t'). \quad (1.5)$$

Здесь $\text{Re}(\cdot)$ обозначает получение действительной части комплексного числа, а векторы в Уравнении (1.5) являются комплексными числами, t' обозначает комплексно-сопряженное число к t , а diag – диагональный вектор матрицы отношений.

Модель RotatE [4] создает эмбединги сущностей в пространстве комплексных чисел и отображает отношения между сущностями через операции вращения в пространстве комплексных чисел, что может использоваться для моделирования и вывода различных шаблонов отношений. Функция модели RotatE представлена как:

$$\begin{aligned} f_r(h, t) &= \|h \circ r - t\|, \\ \{h, r, t\} &= e^{i\theta} = \cos \theta + i \sin \theta. \end{aligned} \quad (1.6)$$

Сущности в RotatE представлены комплексными числами, а Гамильтоново произведение \circ представляет операцию вращения. QuatE [50] далее реализует обучение представлений графов знаний в пространстве кватернионов, и его модель представляет каждую сущность или связь с тремя мнимыми частями и одной действительной частью ($Q = a + b_i + c_j + d_k$), а также объединяет главные сущности и связи с помощью умножения Гамильтона.

Модель KG2E [51] реализует обучение представлений графа знаний в пространстве многомерных гауссовских распределений, где векторные средние многомерных гауссовских распределений используются для представления позиций сущностей и связей, а матрицы дисперсий используются для представления неопределенностей связей между сущностями. Модель оптимизируется с использованием расстояния Кульбака – Лейблера:

$$f_r(h, t) = \int_{x \in \mathcal{R}^{k_e}} N(x; \mu_r, \Sigma_r) \log \frac{N(x; \mu_e, \Sigma_e)}{N(x; \mu_r, \Sigma_r)} dx. \quad (1.7)$$

Здесь $N(x; \mu_r, \Sigma_r)$ – эмбеддинг отношений многомерного гауссовского распределения, представление сущности аналогично. TransG[40] преобразует сущности в смесь гауссовских распределений, связанных с отношениями, сэмплирует несколько представлений отношения с использованием байесовского непараметрического моделирования смеси гауссовских распределений, а затем выбирает лучшее представление отношения на основе контрольной информации триплета фактов.

Авторы ManifoldE [52] утверждают, что пространство представления вышеупомянутого подхода моделирования слишком велико по сравнению с семантической информацией графа знаний, что приводит к разреженным векторам представления графа знаний. ManifoldE расширяет моделирование сущностей (т.е. сущности моделируются как точки в векторном пространстве) традиционного подхода к обучению представления графа знаний до эмбеддингов на основе многообразий, что требует отображения сущностей и отношений из исходного линейного пространства в Гильбертово пространство:

$$f_r(h, t) = \| M(h, r, t) - D_r \|^2. \quad (1.8)$$

$M(\cdot)$ является функцией многообразия, а D_r – параметром многообразия, связанным с отношением.

TorusE [41] заменяет пространство представлений на n -мерное тороидальное пространство компактных групп Ли и реализует обучение представлений графа знаний на основе идеи TransE (головная сущность + хвостовая сущность = отношение).

Модели представления знаний на основе разложения матриц

Модель представления знаний на основе разложения матрицы обычно рассматривает обучение представлений графа знаний как процесс разложения трёхмерного тензора. Разложение тензоров, как правило, осуществляется

следующими способами:

$$X_{\text{hrt}} = h^T M_r t. \quad (1.9)$$

Здесь X_{hrt} – тензор представления фактических знаний предопределенного графа знаний, который может быть составлен из векторов сущностей h , t и матрицы отношений M_r . Обучение представлений графа знаний может быть реализовано через разложение тензора представления на основе данных для получения представления сущностей и представления отношений. Таким образом, как показано на рисунке 1.4, обучение представлений графа знаний может быть реализовано с помощью разложения матрицы на основе данных для разложения тензора представления фактических знаний с целью получения представления сущностей и представления отношений.

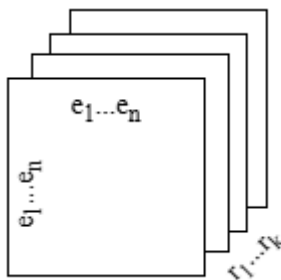


Рисунок 1.4 – Схема модели RESCAL на основе разложения матриц

RESCAL [5] является первым методом факторизации в этом направлении:

$$X_{\text{hrt}} = ERE^T. \quad (1.10)$$

Здесь E – матрица $\|\varepsilon\| \times \|\mathcal{R}\|$ информации о потенциальном представлении сущностей, а R – матрица отношений $\|\mathcal{R}\| \times \|\mathcal{R}\|$.

В методе на основе разложения Таккера [9] на основе RESCAL матричная факторизация заменена на тензорное разложение Таккера, которое представляет собой разложение тензора на ядро тензора, умноженное на векторы по каждой размерности, то есть:

$$f_r(h, t) = W \times_1 h \times_2 r \times_3 t. \quad (1.11)$$

Здесь $W \in \mathcal{R}^{d_e \times d_r \times d_e}$ – основная матрица разложения Таккера, а d_e и d_r представляют собой размерность вектора представления сущности и размерность вектора представления отношения, соответственно. Операция \times_n представляет собой тензорное произведение вдоль модальности n -го порядка.

Модели представления знаний на основе традиционных нейросетей

Традиционная модель представления знаний на основе нейронных сетей заключается в извлечении признаков сущностей и отношений для получения представления знаний графа знаний с использованием мощных возможностей извлечения признаков нейронных сетей. Существуют три подхода: многоуровневый перцептрон (MLP, Multilayer Perceptron), сверточная нейронная сеть (CNN, Convolutional Neural Network) и рекуррентная нейронная сеть (RNN, Recurrent Neural Network). Модель представления знаний на основе графовых нейронных сетей представлена в отдельном подразделе.

Многоуровневый перцептрон

С развитием линейных нейронных сетей некоторые исследователи решили использовать многослойный перцептрон для реализации обучения представления спектра знаний. Как показано на рисунке 1.5, он вводит сущности и отношения в многослойный перцептрон и определяет, правильны ли фактические знания, путем отображения матрицы весов и вычисления семантического коэффициента соответствия.

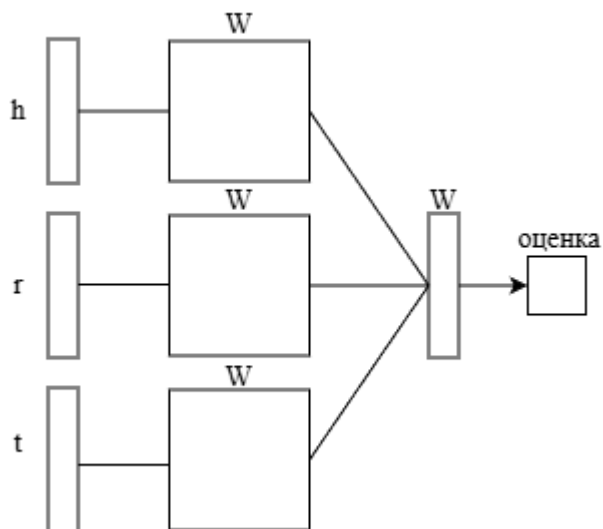


Рисунок 1.5 – Схема модели представления знаний на основе многослойного перцептрона

SME [53] кодирует сущности и отношения вместе в матрице весов (многослойный перцептрон) и использует перцептрон с сигмоидной функцией активации σ [54] для получения оценки знаний:

$$f_r(h, t) = \sigma(w^T \sigma(W[h, r, t])). \quad (1.12)$$

Здесь W – матрица весов, а $[h, r, t]$ получен путем соединения векторов сущностей и отношений

NTN [55] принимает в качестве входных данных векторы сущностей и дополнительную матрицу весов ассоциативных отношений, а его функция оценки знаний выглядит следующим образом:

$$f_r(h, t) = r^T \sigma(h^T R t + M_{r,1} h + M_{r,2} t + b_r). \quad (1.13)$$

Здесь $M_{r,1}$ и $M_{r,2}$ – матрицы весов отношений, связанные с векторами сущностей, а b_r – переменная смещения.

Сверточная нейросеть

Модель представления знаний, основанная на сверточной нейронной сети, использует уникальную способность сверточного ядра к извлечению нелинейных признаков для восстановления векторов головных сущностей и связей в двумерную матрицу, для извлечения многослойных нелинейных

признаков графа знаний. Как показано на рисунке 1.6, ConvE [8] сначала восстанавливает головные сущности и векторы связей в «изображение», а затем реализует извлечение признаков с помощью сверточного ядра и получает приблизительные головные сущности и векторы связей. Наконец, мы находим правильную хвостовую сущность путем сопоставления сходства с набором сущностей. Функция оценки определена в (1.14):

$$f_r(h, t) = \sigma(\text{vec}(\sigma([h, r] * \omega)))W)t . \quad (1.14)$$

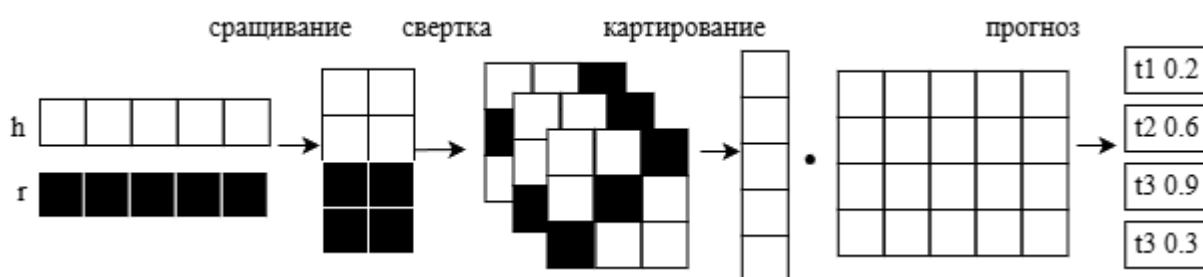


Рисунок 1.6 – Модель представления знаний на основе сверточной нейронной сети

Здесь ω – ядро свертки, а vec обозначает операцию преобразования двумерной матрицы в векторы.

ConvKB [56], с другой стороны, использует сверточную нейронную сеть для непосредственной обработки двумерных матриц, объединенных с векторами сущностей и отношений:

$$f_r(h, t) = \sigma(\text{concat}[h, r, t] * \omega)w . \quad (1.15)$$

ConvKB связывает извлеченные матрицы признаков и повышает обучаемость потенциальных признаков. Нурег [9], с другой стороны, отказывается от традиционной сверточной нейронной сети и использует гиперсеть для генерации специфического для отношений сверточного ядра, ориентированного на векторы отношений, и использует это сверточное ядро вместо операции свертки, что упрощает двумерную сверточную операцию на основе ConvE.

Рекуррентная нейросеть

Представленные выше нейросетевые модели представления знаний, основанные на многослойном перцептроне и сверточной нейронной сети, работают с объектами данных, представляющими собой триплеты фактов, а функция потерь для обучения представлению знаний строится путем оценки этих триплетов. Однако в графе знаний часто встречаются многоступенчатые реляционные зависимости. Например, знания в биологической области обычно имеют семь уровней классификации, таких как вид, род, семейство, отряд, тип и царство. Взяв в качестве примера кошку породы рэгдолл, ее видовая связь в карте знаний выглядит следующим образом: кошка рэгдолл, вид кошки, род кошки, семейство кошек, хищники, млекопитающие, хордовые, животные. Из рода «кошки» также можно найти других родственных животных, таких как сиамские кошки и циветты. Из этого рода можно найти и другие роды со схожими повадками, например, леопарды. Чтобы справиться с подобной длинной последовательной зависимостью отношений, некоторые исследователи предложили модель представления знаний на основе рекуррентных нейронных сетей, используя преимущества рекуррентных нейронных сетей в обработке последовательных задач. Как показано на рисунке 1.7, модели представления знаний на основе рекуррентных нейронных сетей [57-59] соединяют различные тройки фактов посредством отношений, образуя пути, и моделируют граф знаний через эти пути. Затем эти модели подают различные пути в рекуррентную нейронную сеть для завершения извлечения признаков и обучения представлению графа знаний с помощью предыдущей базовой модели. Модели генерируют каждый путь отношений (x_1, x_2, \dots, x_t) , случайным образом перемещаясь по графу знаний, где x_t может быть сущностью или отношением. Затем семантические признаки скрытого слоя каждого шага извлекаются блоками рекуррентной нейронной сети, и информация об сущностях и отношениях объединяется.

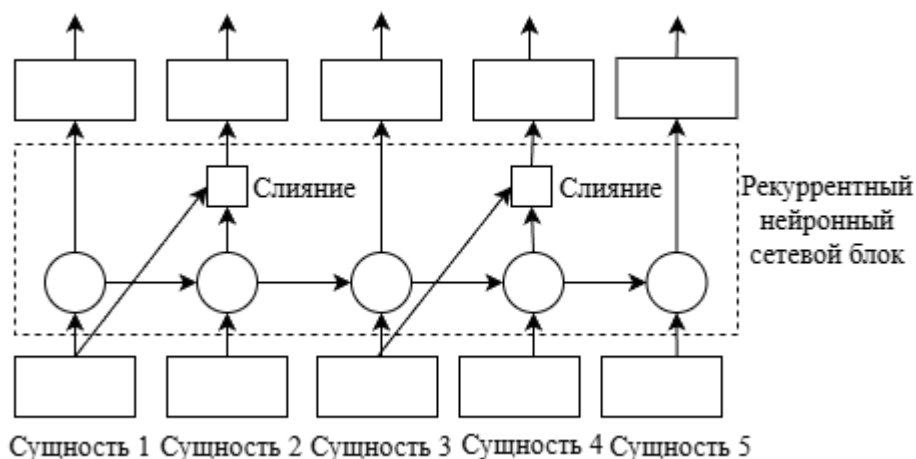


Рисунок 1.7 – Модель представления знаний на основе рекуррентной нейросети

$$h_t = \begin{cases} h_t & , x_t \in \mathcal{E}, \\ W_1 h_t + W_2 x_{t-1} & , x_t \in \mathcal{R}. \end{cases} \quad (1.16)$$

Здесь W_1 и W_2 – матрицы весов, h_t – признак сущности или связи скрытого слоя на шаге t , а x_t – это признак входной сущности или связи на шаге t .

Модель представления знаний на основе графовой нейросети

Преимущество графов знаний перед традиционными схемами описания ресурсов заключается в том, что структура графа, сформированная взаимосвязями между сущностями, может содержать дополнительную информацию. Факты представляют собой своего рода описание, тогда как знания – это вывод фактов, и без структуры графа знаний реальный вывод невозможен. С развитием графовых нейронных сетей в последние годы исследователи уделяют особое внимание извлечению структурной информации из графа знаний с помощью графовых нейронных сетей. Модель представления знаний, основанная на графовой нейронной сети, в основном опирается на операцию свертки графа для распространения информации об узлах сущностей и их соседних узлах, а также на совместное моделирование для изучения их структурной информации. На рисунке 1.8 показан процесс обучения представлению сущностей и связей в графе знаний с помощью

графовых нейросетей. Каждый раз, когда обновляется представление узла сущности, изучается информация о его соседних узлах сущностей и их связях. После обучения представлению «сущность-связь» обычно используется простая модель перевода, представленная ранее, в качестве декодера для преобразования представления «сущность-связь» в последующие подзадачи, такие как тройная классификация, прогнозирование связей и так далее. Реляционная графовая нейронная сеть [10] (RGCN) – первая модель графовой нейронной сети, предложенная для многореляционных графов (в которых узлы связаны между собой несколькими различными типами связей) и применяемая для обучения представлению графа знаний. Она обновляет информацию об узлах в графе знаний, чтобы получить информацию о соседних узлах, как показано на рисунке 1.8, а ее функция обновления определена в (1.17).

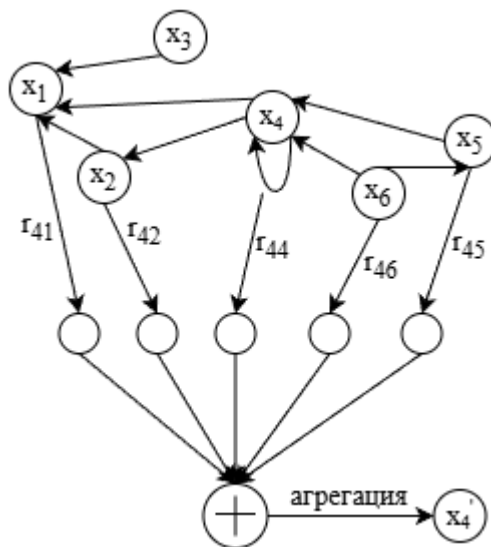


Рисунок 1.8 – Модель представления знаний на основе графовой нейросети

$$x_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} x_j^{(l)} + W_0^{(l)} x_i^{(l)} \right). \quad (1.17)$$

Здесь $x_i^{(l)} \in \mathcal{R}^{d^{(l)}}$ обозначает скрытую характеристику i -го узла сущности в слое l , N_i^r – соседний узел, с которым данный узел связан через отношение

r , $W_r^{(1)}$ и $W_0^{(1)}$ – соответствующие матрицы весов. $c_{i,r}$ – параметры, обычно количество соседних узлов. В RGCN каждая сущность имеет одинаковую степень важности (вес), и по этой причине в была предложена взвешенная графовая сверточная сеть (WGCN) [46]. WGCN вводит дополнительный вектор весов $\alpha_j^{(1)}$ для каждого соседнего узла во время обновления узла с помощью механизма внимания классов, так что разные соседние узлы генерируют разные веса для обновления целевого узла.

$$x_i^{(1+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(1)} \alpha_j^{(1)} x_j^{(1)} + W_0^{(1)} x_i^{(1)}\right). \quad (1.18)$$

Авторы модели ComrGCN [11] утверждают, что реляционные графовые нейронные сети эффективны при моделировании неориентированных графов или сетей с одной связью, но когда количество связей увеличивается, размер обучающих параметров растет настолько быстро, что их невозможно обучить, и они не могут достичь хороших результатов на графах с несколькими связями. Действительно, предыдущие модели не уделяют достаточного внимания обучению представлений связей в графе знаний и слишком много внимания уделяют обучению представлений узлов. Comrgcn предлагает подход, основанный на комбинации узлов и связей, для завершения обучения графовых нейронных сетей:

$$\begin{aligned} x_i^{(1+1)} &= \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(1)} \phi(h_r^{(1)}, x_j^{(1)})\right), \\ h_r^{(1+1)} &= W_{rel}^{(1)} h_r^{(1)}. \end{aligned} \quad (1.19)$$

Здесь $\phi(h_r^{(1)}, x_j^{(1)})$ обозначает операцию объединения связей между узлами, которая обычно осуществляется путем сложения, умножения и свертки векторов.

Модель представления знаний, основанная на моделировании самовнимания

Модели представления знаний, основанные на моделях самовнимания, учитывают согласованность знаний на основе единого триплета фактов и улучшают обучение представлению графа знаний за счет введения контекстной информации. Для использования контекстной информации модель CoKE [13] сначала выбирает пути графа знаний с помощью корреляционной модели, подобной рекуррентной нейронной сети, а затем кодирует эти пути с помощью модели самовнимания для обучения представлению. KG-BERT [12] использует предварительно обученный двунаправленный кодировщик-преобразователь представления (BERT) для извлечения признаков из предложений, состоящих из сущностей и отношений, с дополнительной информацией из предварительно обученных моделей. Кроме того, в области обучения представлению мультимодального графа знаний подход MKGformer [14] реализует моделирование многомодальных признаков сущностей на основе унифицированной модели самовнимания в многомодальном обучении представлений.

1.3 Базы данных графов знаний

Приведем распространенные и легкодоступные базы данных графов знаний. Эти данные могут как собираться вручную, так и создаваться автоматически, многие из них являются открытыми. Рассмотрим графы знаний в двух широких категориях: общие графы знаний и специализированные графы знаний.

Общие графы знаний

Под графами общих знаний, как правило, подразумевают четыре крупных баз данных общих знаний: Wikidata [60], Freebase [61], DBpedia [62] и YAGO [63], которые содержат наибольший объем информации среди баз

данных знаний, охватывая широкий спектр областей и широкий диапазон знаний. Эти четыре набора данных будут кратко представлены далее.

1) Wikidata

Хранилище базовой информации, созданное Фондом Wikimedia для обеспечения поддержки знаний и поиска данных в рамках его энциклопедической базы знаний и других медиапроектов. Википедия – это многоязычный справочный инструмент в глобальной сети Интернет, и по состоянию на сентябрь 2021 года Wikidata содержит более 80 миллионов элементов данных и более 2 миллиардов семантических связей. Данные охватывают множество предметных областей, включая историю, искусство, науку, культуру, географию и политику. Wikidata имеет автоматизированную систему хранения данных, современную технологическую архитектуру, которая позволяет эффективно извлекать и использовать постоянно растущий объем знаний, тем самым способствуя обмену знаниями в глобальном масштабе.

2) Freebase

Freebase – граф знаний, поддерживаемый компанией Metaweb Technologies, созданный в 2007 году для сбора и структурирования информации в мировом масштабе. В 2010 году Google приобрела Metaweb Technologies и включила Freebase в свой граф знаний. Freebase содержит более 250 миллионов сущностей и почти 2 миллиарда связей. Сущности и связи охватывают знания обо всем: от людей, мест и организаций до книг, фильмов и музыки. В отличие от Wikidata, граф знаний Freebase в основном редактируется и организуется вручную экспертами, а также собирает информацию из других источников, таких как Википедия, IMDb [64], Flickr [65] и другие веб-сайты или корпуса.

3) DBpedia

Открытый граф знаний, основанный на Википедии, который извлекает структурированную информацию из Википедии и хранит её в формате RDF.

DBpedia была впервые запущена в 2007 году и охватывает большую часть содержимого Википедии, включая статьи, классификации, изображения, географические местоположения, людей, события, книги и т. д., и стала одним из важных источников данных для связанных данных (Linked Data). По состоянию на сентябрь 2021 года DBpedia содержала более 650 миллионов сущностей и более 2 миллиардов триплетов (т. е. отношений субъект-предикат-объект), из которых более 16 миллионов сущностей были отнесены к более чем 500 категориям. Размер DBpedia продолжает расти, каждый месяц в граф знаний добавляются новые данные, чтобы лучше удовлетворять потребности различных областей. В то же время DBpedia предоставляет ряд инструментов и сервисов для запросов к графу знаний, анализа данных и разработки приложений.

4) YAGO

На основе иерархического анализа YAGO интегрирует источники данных, такие как wordnet [66] и geonames [67], с системой классификации Википедии и создает базу знаний с другой иерархической структурой. По состоянию на 2021 год YAGO4 содержит более 300 миллионов триплетов и охватывает более 100000 страниц Википедии.

Специализированные графы знаний

Графы знаний в отраслевых областях обычно строятся на основе данных, специфичных для данной отрасли, и часто называются отраслевыми графами знаний или графами знаний предметной области. Ниже приведены примеры отраслевых графов знаний, таких как IMDB [68], Musicbrainz [69] и Douban [70].

1) IMDB

База данных видеоработ, содержащая информацию об актерах, фильмах, телешоу, телезвездах и кинопроизводствах. По состоянию на февраль 2012 года IMDB собрала информацию о более чем 2 миллионах работ и 4,5 миллионах деятелей кино и телевидения.

2) Musicbrainz

Структурированная база данных музыки, цель которой – предоставить метаданные о музыке широкой публике и позволить зарегистрированным пользователям добавлять информацию или вносить свой вклад в пополнение данных. Musicbrainz обычно предоставляет данные пользователям двумя способами: в виде базы данных или веб-сервиса, а его веб-API имеет формат XML.

3) Douban

Социальная сеть, специализирующаяся на обзорах книг, фильмов и музыки, а также уникальный сайт в сфере аудиовизуальных материалов в китайском интернете. Позволяет пользователям самостоятельно оценивать различные фильмы и сериалы, став местом встречи любителей кино, телевидения, книг и кинокритиков в Китае. Рейтинги Douban стали важным показателем для оценки фильмов в Китае и привлекают все больше внимания. Благодаря информации об обзорах в режиме реального времени, предоставляемой Douban, его влияние на кино- и телеиндустрию становится еще более значительным, и он лучше отражает оценку фильмов и сериалов пользователями. Douban накопил большой объем данных о фильмах, предоставив тем самым важный ресурс для анализа киноиндустрии. Douban также предоставляет ценную основу для экономического моделирования и разработки маркетинговых кампаний для киноиндустрии. По состоянию на 2019 год Douban собрал пользовательские оценки почти 100 000 фильмов, создав всеобъемлющую базу данных кинорейтингов.

1.4 Комбинаторные рекомендации: постановка задачи и примеры

В области комбинаторных рекомендаций фокус исследований сместился от рекомендаций отдельных элементов к ансамблевым рекомендациям, состоящим из множества элементов, таких как наборы товаров, музыкальные

альбомы и комбинации книг. Это сделало анализ ассоциаций (AA) [71] между элементами ключевой темой исследований в комбинаторных рекомендациях. С.Шалом и другие ученые [15] оптимизировали показатель кликабельности (CTR) списков рекомендаций, выявляя потенциальные корреляции между элементами для повышения эффективности рекомендаций. Алгоритм AGREE [72] использует нейронную коллаборативную фильтрацию (NCF) [73] для удовлетворения потребностей в рекомендациях групп пользователей. Исследовательская группа Лю [16] в 2017 г. разработала модель BPM (Bundle Purchases with Motives – покупки комплектов с учетом мотивации) для решения проблемы персонализированных рекомендаций наборов продуктов. BPM – вероятностная модель, которая не только оптимизирует стратегию продуктового портфеля, но и частично раскрывает мотивы покупок пользователей путем анализа их совместных покупок. Аналогичный подход успешно применялся в исследовании рекомендаций портфеля для игр и их реквизита [74]. Вэй и др. в 2022 г. [17] пошли еще дальше, предложив метод динамической генерации персонализированных решений продуктового портфеля на основе предпочтений пользователя.

Музыкальные плейлисты и наборы книг привлекли большое внимание как типичные сценарии рекомендаций портфеля. Исследования в этих двух областях обычно взаимно дополняют друг друга, формируя синергетическое развитие. В ранних исследованиях команда Цао [75] в 2017 г. впервые предложила задачу рекомендации музыкальных плейлистов на основе данных облачной музыкальной платформы NetEase и разработала модель факторизации встраивания (EFM) для реализации совместной рекомендации отдельных песен и списков. Чен и др. [76] предложили модель DAM (Deep Attentive Multi-Task), использующую сеть внимания декомпозиции для интеграции характеристик элементов, и выпустили набор данных с наборами книг Yousu, что заложило важную основу для последующих исследований. Модель инновационно моделирует как взаимодействие пользователя с

элементом, так и взаимодействие пользователя с комбинацией элементов посредством многозадачной структуры обучения.

С появлением данных с таких платформ, как Spotify и Goodreads [77], исследователи начали фокусироваться на оптимизации согласованности между комбинациями. Команда Чанга [78] предложила модель BGCN (Bundle Graph Convolutional Network – сверточная сеть с пакетным графом) для достижения персонализации путем построения структуры графа комбинаций пользователей для персонализированных рекомендаций. Ван и др. [79] представили механизм взаимодействия соседей на этой основе для дальнейшей оптимизации архитектуры графовой нейронной сети. Алгоритм Attlist [80] фиксирует отношения включения комбинаций и предпочтения пользователей посредством иерархической структуры. Команда Чжана, вдохновленная моделью IGMC [81] использовала нейронные сети подграфов для решения проблемы междоменных рекомендаций.

Исследования показывают, что комбинаторные методы рекомендаций демонстрируют широкую применимость в реальных сценариях, однако процесс их разработки, очевидно, ограничен размером и качеством доступных эталонных данных. Стоит отметить, что в большинстве сценариев применения корреляция между элементами, как правило, подчиняется более явным комбинаторным правилам – например, треки в музыкальных плейлистах обычно имеют схожие жанровые или исполнительские характеристики, а функциональная взаимодополняемость должна поддерживаться между объединенными в пакет товарами. Эти присущие правилам правила в некоторой степени снижают гибкость комбинаций, тем самым упрощая сложность взаимосвязей между элементами.

С развитием социальных сетей контент, создаваемый пользователями и посвященный модной одежде, демонстрирует взрывной рост. По сравнению с традиционными сценариями рекомендаций по сочетаниям, рекомендации по подбору одежды обладают тремя важными особенностями: во-первых,

каждое сочетание одежды имеет более высокую степень свободы подбора; во-вторых, персональные предпочтения пользователя более выражены; и, наконец, значительно возрастает сложность взаимосвязей между предметами одежды. Кроме того, предметы одежды обычно обладают богатыми мультимодальными данными (включая высококачественные изображения и подробные текстовые описания), что делает рекомендации по подбору одежды идеальной экспериментальной платформой для проверки эффективности алгоритмов рекомендаций по сочетаниям.

Практический пример: рекомендации по сочетанию предметов одежды

Как неотъемлемая часть повседневной жизни, подбор одежды не только удовлетворяет базовые потребности людей, но и является важным носителем личного стиля и культурного самовыражения. Эта характеристика привлекает большое внимание в последние годы в исследованиях рекомендаций одежды [82] постепенно превращается в развивающуюся область исследований. Традиционные рекомендации одежды в основном сосредоточены на проблеме рекомендации отдельных товаров [83] их основная цель – помочь пользователям найти свои любимые предметы одежды, в частности, включая следующие направления исследований:

1. Поиск одежды одного типа [84]: сопоставление соответствующих товаров электронной коммерции с фотографиями повседневной одежды для удовлетворения потребности пользователей в покупке по принципу «что видишь, то и получаешь». Исследовательская команда Киапура [84] в 2015 г. впервые применила методы глубокого обучения для изучения междоменных ассоциаций признаков между фотографиями улицы и витрины. Чен и др. [85] в 2017 г. дополнительно расширили эту технику на область подбора видеоматериалов.

2. Рекомендация похожих товаров [86]: в основном предоставляет платформам электронной коммерции услуги по рекомендации

взаимозаменяемых товаров, фокусируясь на объективном сходстве товаров.

3. Рекомендация дополнительных товаров [87]: рекомендация сочетаний на основе субъективных ощущений пользователей. Макколи и др. [88] изучают взаимодополняющие отношения между товарами на основе общих данных о покупках и характеристик изображений, и это исследование тесно связано с рекомендациями наборов.

Для повышения эффективности рекомендаций существующие исследования также распространяются на распознавание атрибутов одежды [89], анализ стиля [90] и прогнозирование модных тенденций [91], что обеспечивает многомерное решение для рекомендаций одежды в сложных сценариях.

По сравнению с рекомендацией единственного товара, рекомендация полного набора одежды лучше отвечает реальным потребностям пользователей. Основная задача рекомендаций наборов заключается в поиске баланса между сложной взаимосвязью элементов и персонализированными предпочтениями пользователя. Существующие методы исследований в этой области можно разделить на три основные категории: методы сопоставления на основе элементов, методы целостного моделирования и методы моделирования на основе графов.

1.5 Основные методы подбора комбинаций

Алгоритмы сопоставления элементов в комбинаторных рекомендациях

Алгоритмы, основанные на сопоставлении элементов [92], преобразуют общую координацию сопоставления наборов в задачу комбинаторной оптимизации отношений пара-к-паре между элементами. Суть таких подходов заключается в установлении эффективной метрики релевантности элементов. Команда Маколея [88] впервые применила линейные

преобразования для обработки признаков изображений (извлеченных с помощью предварительного обучения CNN), но смогла обучить только присвоение весов измерениям признаков. Вейт и др. [93] представили архитектуру двойной сети (сиамскую сеть), которая используется в двухканальной сети с общими параметрами для достижения сквозного обучения встраиванию признаков, что значительно улучшает производительность модели. Учитывая многомерный характер фактических решений о парном сопоставлении, команда Вейта [18] предложила сети условного сходства (CSN – Conditional Similarity Networks) для установления метрик дифференциации в подпространстве различных ограничений атрибутов. Этот метод сталкивается с проблемой затрудненного доступа к атрибутивным данным, хотя он инновационно вводит условные факторы, такие как цвет и стиль и т. д. Василева и др. [94] улучшают этот подход, используя более доступную информацию о категориях в качестве условий для установления метрик в зависимом от категории подпространстве с помощью нелинейной функции расстояния. Однако метод имеет ограниченный эффект при обработке комбинаций категорий с разреженными данными. Чтобы преодолеть это ограничение, команда Янга (2019 г.) [92] использует подход моделирования «относительных отношений» [1] для преобразования информации о категориях в структурированные ограничения между элементами, что позволяет более адаптивно вычислять гетерогенное сходство. Этот сдвиг парадигмы эффективно смягчает проблему разреженности данных и предоставляет новые идеи для рекомендаций коллокации в сложных сценариях.

Поскольку явное введение условной информации требует достаточного количества данных для изучения метрических связей в различных условиях, производительность модели часто ограничена, когда условная информация отсутствует. Для решения этой проблемы Тан и др. [95] предложили неявный метод обучения в нескольких подпространствах. Метод имеет двухветвевую

структуру: ветвь признаков проецирует признаки элементов на несколько потенциальных подпространств, а ветвь весов динамически вычисляет комбинированные веса различных подпространств. Благодаря этому механизму модель способна достичь многомерного обучения сходству элементов без явной условной маркировки. Однако тот факт, что вычисление весов зависит от конкретных пар элементов, приводит к необходимости пересчитывать представление признаков для каждой оценки сходства, что увеличивает вычислительную сложность.

В области персонализированных рекомендаций существующие исследования в основном характеризуют персонализированные предпочтения путем введения параметров пользователя. Алгоритм *deepstyle*, предложенный командой Лю [96], использует двунаправленные встроенные представления пользователя и элемента, вычисляет оценки персонализации с помощью скалярного произведения векторов и улучшает представления признаков, используя информацию о категориях одежды. Инь и др. [97] дополнительно расширили подход к мультимодальному слиянию, интегрировав семантические и визуальные сигналы для изучения предпочтений пользователя. Хотя эти методы эффективны, они применимы только к сценариям рекомендаций отдельных товаров. Для задачи рекомендации наборов алгоритм функциональной попарной тензорной факторизации взаимодействия (FPITF) [98], использует метод функционализированного попарного тензорного разложения. Алгоритм не только разлагает совместимость наборов на попарные отношения между товарами, но и моделирует взаимодействие «пользователь-товар» посредством отображения в нескольких подпространствах для достижения персонализированной рекомендации наборов. Эти методы разложения широко используются в исследованиях по рекомендациям подбора одежды благодаря простоте их модели и интерпретируемости.

Целостный подход к моделированию в комбинаторных рекомендациях

Хотя разложение задачи рекомендации комбинации элементов (например, костюма) на задачи сопоставления элементов имеет определенную рациональность, такое явное разложение затрудняет выявление сложных ассоциаций более высокого порядка внутри комбинации. По этой причине исследователи в области рекомендаций комбинаций одежды начали изучать целостные подходы к моделированию костюмов, и к значительным достижениям относится работа команды Тансенга [99], которая впервые применила прямую целостную характеристику костюмов, объединив все характеристики элементов для формирования представления костюма. Однако этот метод имеет два очевидных ограничения: во-первых, он требует фиксированного количества элементов в наборе, и во-вторых, результаты характеристики чувствительны к порядку расположения элементов. Ли и др. [100] попытались закодировать наборы переменной длины в векторы фиксированной длины с помощью рекуррентных нейронных сетей (РНН), но все еще столкнулись с проблемой зависимости от порядка, и сеть имела тенденцию игнорировать информацию об элементах из предыдущих входных данных, что привело к чрезмерной зависимости представления от конечных элементов. Для решения этих проблем команда Хана представила двунаправленную сеть долговременной и кратковременной памяти (Bi-LSTM) [101] для оценки общей координации набора путем вычисления интеграции оценок каждого элемента. Этот подход не только снижает чувствительность к порядку элементов, но и значительно улучшает производительность модели, более полно улавливая межэлементные ассоциации посредством двунаправленного потока информации.

Для более эффективного устранения влияния порядка элементов на прогнозирование совместимости набора команда Янга [102] использовала механизм внимания вместо традиционной рекуррентной нейронной сети.

Метод динамически улучшает представление признаков каждого элемента путем вычисления интерактивных весов внимания между элементами в наборе. Этот подход к слиянию признаков на основе весов внимания характеризуется инвариантностью к перестановкам, что делает окончательное представление признаков полностью независимым от порядка ввода элементов. Хотя метод успешно устраняет зависимость от порядка на уровне представления элементов, в расчете общего показателя совместимости все еще присутствует некоторая чувствительность к порядку, что в некоторой степени ограничивает конечную производительность алгоритма.

Для сценария персонализированных рекомендаций наборов модель outfitnet, предложенная Лином и др. [103], использует двухэтапную структуру обучения: на первом этапе изучается связь коллокации между элементами с помощью сети связей между элементами моды (Fashion Item Relation Network, FIR), так что элементы одного набора могут быть близки друг к другу в пространстве вложений; а на втором этапе моделируются персонализированные предпочтения пользователей с помощью сети предпочтений наборов (Set Preference Network, OP). Модель представляет оценки наборов как взвешенные комбинации составляющих элементов, но поскольку сеть OP фокусируется только на взаимодействиях пользователя и элемента, процесс тонкой настройки может привести к постепенному затуханию многомерной информации об ассоциациях элементов, полученной сетью FIR, что влияет на общую производительность модели. Это явление подчеркивает важность сквозных структур обучения для поддержания многоуровневой семантической информации.

Графовые модели и алгоритмы в комбинаторных рекомендациях

В задачах рекомендаций наборов данных часто проявляются сложные свойства графовой структуры, и такую структурную информацию трудно эффективно получить с помощью традиционных методов метрического обучения. Графовые нейронные сети (GNN – Graph Neural Networks) [104],

как алгоритмическая структура, специализирующаяся на обработке данных графовой структуры, широко используются в различных смежных задачах [105]. Стандартная структура графа может быть формально представлена как $G=(V,\varepsilon)$, где $V = \{v_1, \dots, v_n\}$ обозначает множество из n узлов, а $\varepsilon \subseteq V \times V$ определяет связь ребер между узлами. Эта структура может быть математически представлена матрицей смежности $A \in \mathcal{R}^{n \times n}$, где $a_{ij}=1$ указывает на наличие ребра (v_i, v_j) , а $a_{ij}=0$ указывает на отсутствие связей. В практических приложениях узлы обычно имеют представления признаков, а ребра могут также нести информацию о весе.

Алгоритмы рекомендаций, основанные на моделях графов [106], повышают точность прогнозирования за счет извлечения информации о структуре графа из данных. Кукурулл и др. [107] преобразовали задачу сопоставления элементов в задачу прогнозирования связей (Link Prediction), построив структуру графа, где элементы являются узлами, а отношения совместного появления ребрами. Команда Сингхала [108] улучшила представление признаков узлов, введя семантическую информацию о категориях. Для решения проблемы ненадежной структуры графа из-за разреженности данных Лю и др. [109] предложили построить граф категорий G_c , где узлы представляют категории товаров, а веса ребер отражают вероятности совместного появления категорий, и получить стабильное представление структуры графа посредством нормализации. Метод также моделирует каждый набор как полный граф и инициализирует веса ребер, используя информацию графа категорий. Ли и др. [110] дополнительно применяют граф категорий к сценариям персонализированных рекомендаций и разрабатывает иерархическую графовую модель. Команда Лю [111] аналогичным образом использует полный граф для характеристики наборов, исследуя при этом проблему разнообразия рекомендаций. Чжан и др. [112] показали, что значительные результаты рекомендаций также могут быть

достигнуты с использованием только цветовых признаков в качестве атрибутов узлов. Основное преимущество такого подхода заключается в возможности одновременного использования структуры графа для улучшения представления элементов и изучения сложных ассоциативных моделей элементов с помощью графовых нейронных сетей. Однако необходимо отметить две ключевые проблемы: во-первых, случайно сгенерированные отрицательные образцы могут привести к переобучению модели локальным графовым шаблонам, что влияет на способность к обобщению; во-вторых, построение и обработка крупномасштабных графовых структур может значительно увеличить вычислительную сложность. Эти факторы необходимо тщательно учитывать в практических приложениях.

1.6 Области применения комбинаторных рекомендаций

Изучение релевантности предметов одежды формирует важную основу для рекомендательных систем и связанных с ними производных задач. В ответ на разнообразные потребности в практических приложениях, существующие исследования углубленно изучают проблему рекомендаций одежды в различных сложных сценариях. Гуань и др. [113] исследовали прогнозирование совместимости предметов с ограниченной информацией об управлении. Аль-Халах и его команда [114] разработали метод, основанный на разложении неотрицательной матрицы, для изучения особенностей стиля одежды и прогнозирования будущих модных тенденций. С другой стороны, Сяо и др. сосредоточились на конкретном сценарии персонализированных рекомендаций гардероба [115]. В области рекомендаций по подбору костюмов исследования должны учитывать не только личные предпочтения пользователя и общую координацию подбора, но и множество других аспектов, таких как характеристики телосложения пользователя, интерпретируемость алгоритмов [116], а также методы дополнения и

генерации костюмов [117]. Между тем, методы генерации изображений одежды, ориентированные на совместимость, также привлекли широкое внимание, включая синтез одежды [118] и виртуальную примерку [119], а также другие направления исследований.

1.7 Выводы по главе 1

Содержание настоящего исследования последовательно охватывает алгоритмы обучения представлению в классических графах знаний и мультимодальных графах знаний, а также разработку новых алгоритмов на основе графовых нейронных сетей для полного изучения структурной информации в графах знаний. В задаче рекомендаций по сопоставлению комплектов одежды переход от рекомендательного объекта к комбинации нескольких элементов, в сочетании с огромным количеством размеров и категорий элементов, приводит к взрывному росту возможных комбинаций пользователь-комплект, что требует разработки более эффективных вычислительных методов.

В частности, вклад данной диссертации отражен в следующих трех аспектах:

1) Обучение представлению графов знаний для несвязных графов

В исследовании информации о структуре графов классических графов знаний в данной работе обнаружено, что создаваемые графы часто являются несвязными из-за проблемы дефицита знаний в графах знаний. Существующие модели графовых нейронных сетей, такие как RGCN, WGCN и CompGCN, осуществляют обновление информации о сущностях и отношениях в графе знаний путем распространения информации между узлами для реализации обучения представлению графа знаний. Эти модели графовых нейронных сетей, основанные на распространении информации, часто не способны уловить полную информацию о структуре графа в случае

таких несвязанных графов. В данной диссертации используется идея Deep Graph Infomax (DGI) для улавливания полной информации о структуре графа, а также семантической информации графа знаний с помощью контрастного обучения. В частности, предлагается новый алгоритм Dual-FusionKG для решения задачи обучения представлению несвязанного графа знаний, который состоит из двух идентичных сетей внимания графа слияния сущностей и отношений (ERGAT). Алгоритм состоит из двух идентичных сетей внимания графа слияния сущностей и отношений (ERGAT) для извлечения структурной и семантической информации соответственно. В сети внимания графа слияния сущностей и отношений модель использует взаимные корреляции для обмена информацией между сущностями и отношениями, а также включает механизм внимания к отношениям для обучения векторных представлений сущностей. Кроме того, модель устанавливает отдельные матрицы весов для отношений, чтобы динамически обновлять векторные представления отношений. Первый модуль сети внимания графа слияния сущностей и отношений изучает полную структурную информацию графа знаний без учителя, максимизируя взаимную информацию между глобальным и локальным представлениями графа знаний. В теории вероятностей и теории информации взаимная информация (MI) двух случайных переменных является мерой взаимозависимости между переменными, и таким образом этот модуль побуждает кодировщик передавать полную структурную информацию графа также и в локальном представлении. Второй модуль сети внимания графа слияния сущностей и отношений, подобно традиционному подходу к моделированию на основе графовых нейронных сетей, получает семантическую информацию посредством обучения с учителем на основе известной информации о триаде фактов из графа знаний. Наконец, алгоритм Dual-FusionKG объединяет структурную и семантическую информацию для завершения обучения представлению графа знаний.

2) Обучение представлению графа знаний для мультимодальных графов

С ростом объемов мультимодальных данных обучение представлению мультимодальных графов знаний также стало актуальной темой исследований в области обучения представлению графов знаний и их применения. Традиционные методы обучения представлению графов знаний не могут охватить мультимодальную информацию узлов в мультимодальном графе знаний, межмодальные связи между одними и теми же сущностями и мультимодальные семантические связи между различными сущностями, а также не могут обеспечить эффективное и точное представление узлов и связей в мультимодальном графе знаний. В то же время мультимодальный граф знаний представляет собой гетерогенный граф с множеством связей, и традиционный подход на основе графовых нейронных сетей не может эффективно обучать представлению таких графов. Для обучения представлению мультимодальных графов знаний в данной работе предложен алгоритм HyperFusion-Net. Учитывая мультимодальную природу графов знаний, алгоритм HyperFusion-Net предлагает модуль слияния информации. В этом модуле алгоритм HyperFusion-Net сначала получает предварительно обученные векторы встраивания для каждой модальности сущности, а затем использует слияние тензоров низкого ранга (LTF) для получения мультимодального представления каждой сущности, чтобы смоделировать процесс информационного взаимодействия внутри одной модальности и между различными модальностями в разных сущностях с мультимодальным представлением.

В данной диссертации используются возможности обработки графовых данных в графовых нейронных сетях, в качестве объектов исследования рассматриваются традиционные графы знаний и мультимодальные графы знаний, проводится исследование разрывности графов знаний и сложности работы с мультимодальной информацией, а также предлагаются целенаправленные методы решения ключевых проблем. Одновременно в

диссертации разработаны алгоритм совместимости взвешенного хеширования и алгоритм вероятностного динамического кодирования для повышения вычислительной эффективности и гибкости, а также сокращения времени отклика рекомендаций в реальном времени для огромного количества возможных комбинаций пользователей и костюмов в задаче подбора костюмов.

Структура глав организована следующим образом:

В главе 2 исследуется обучение представлению традиционных графов знаний. Исходя из проблемы непересекаемости в графах знаний, предлагается алгоритм многореляционной графовой нейронной сети Dual-FusionKG, основанный на максимальной взаимной информации в графах. Одновременно в главе подробно представлена базовая модель кодирования данной диссертации – разновидность реляционной графовой нейронной сети, также известная как сеть внимания графа слияния сущностей и связей. Алгоритм Dual-FusionKG извлекает информацию о структуре графа, а также семантическую информацию графа знаний с помощью двух сетей внимания, основанных на слиянии сущностей и отношений, соответственно. Эксперименты по оценке производительности показывают, что предложенный в этой главе алгоритм превосходит другие алгоритмы не только по эффективности, но и по скорости сходимости в процессе обучения.

В главе 3 исследуется обучение представлению многомодального графа знаний, и на основе представления многомодальной информации предлагается алгоритм HyperFusion-Net, который эффективно реализует извлечение мультимодальной информации и представление информации о структуре графа. В этой главе также кратко представлены исследовательский контекст и прогресс в области обучения представлению многомодального графа знаний. Алгоритм HyperFusion-Net в основном состоит из модуля слияния информации и модуля агрегирования информации. Первый модуль слияния информации реализует извлечение многомодальной информации, а

второй модуль агрегирования информации реализует представление информации о структуре графа. В ходе сравнительных экспериментов показано, что предложенный в этой главе алгоритм HyperFusion-Net обеспечивает лучшее представление на многомодальных графах знаний, а также проведены эксперименты для подтверждения вклада индивидуальной модальной информации в обучение представлению многомодального графа знаний.

В главе 4 исследуется проблема эффективности в комбинаторных рекомендациях. Предложен новый алгоритм MCHM-Net для персонализированных комбинаторных рекомендаций на основе хеширования, позволяющий решить проблему сложного баланса между эффективностью и персонализацией в задачах комбинаторных рекомендаций из-за взрыва многомерного комбинаторного пространства. Алгоритм повышает вычислительную эффективность за счет представления элементов и пользователей в виде бинарных векторов и поддерживает точность персонализированных рекомендаций, одновременно повышая вычислительную эффективность. Предварительные результаты по оптимизации рекомендательных алгоритмов были представлены в наших ранних работах [120-123], где рассматривались различные подходы к коллаборативной фильтрации и гибридным моделям рекомендаций. Кроме того, в ранних исследованиях рассматривались методы оптимизации, основанные на линейном программировании [124].

Взаимосвязь исследуемых моделей и методов представлена на рисунке 1.9.



Рисунок 1.9 – Организационная структура содержания настоящего исследования

2 ОБУЧЕНИЕ ПРЕДСТАВЛЕНИЮ ГРАФОВ ЗНАНИЙ ДЛЯ НЕСВЯЗАННЫХ ГРАФОВ

2.1 Проблема несвязности в графах знаний

Обучение представлению знаний в виде графов знаний является важным направлением исследований в области машинного обучения. Граф знаний по сути представляет собой семантическую сеть, которая раскрывает ассоциативные связи между сущностями и способна структурно представлять сложные связи различных вещей в реальном мире. В настоящее время эта техника широко используется в нескольких известных крупных базах знаний, включая Wikidata [60], Freebase [61] и Dbpedia [62]. Графы знаний обычно используют форму триплета {голова, отношение, хвост} для представления знаний, где голова и хвост обозначают соответственно головную и хвостовую сущности, а отношение характеризует семантическую связь между ними. Например, («Альберт Эйнштейн», «Профессия», «Ученый») и («Кристофер Нолан», «Профессия», «Режиссер») являются типичными примерами таких троек. Однако такие дискретные символические представления имеют присущие им ограничения: даже когда сущности и отношения кодируются идентификаторами, они по-прежнему сталкиваются с двойной проблемой разреженности данных и высокой вычислительной сложности. В последние годы технология глубоких нейронных сетей, основанная на обучении представлений, совершила значительные прорывы и способна преобразовывать мультимодальные данные, такие как изображения, текст и идентификаторы продуктов, в плотные низкоразмерные векторные представления, тем самым поддерживая эффективные вычисления в векторном пространстве. На основе этого обучение представлению графов знаний позволяет преобразовывать вывод, слияние и применение знаний в

задачи векторной арифметики путем моделирования встраивания сущностей и отношений и отображения их в непрерывное семантическое пространство.

В настоящее время обучение представлению графов знаний широко использует различные типы архитектур нейронных сетей для извлечения семантической информации, включая традиционные методы, такие как линейные нейронные сети и сверточные нейронные сети. Однако эти методы, как правило, фокусируются только на представлении семантической информации в виде триплетов фактов, пренебрегая топологическими свойствами графов знаний как данных, структурированных по графам. Для эффективного захвата структурных особенностей исследователи последовательно предлагали методы обучения представлений на основе графовых нейронных сетей (GNN), среди которых можно выделить RGCN [10], SACN [46] и CompGCN [11]. Эти методы в основном используют механизм агрегирующей функции для достижения динамического обновления представления узлов графа путем интеграции собственных характеристик узла с информацией о его окрестностях.

Несмотря на значительный прогресс, существующие графы знаний по-прежнему в целом сталкиваются с проблемой недостаточной целостности, основные причины которой можно суммировать в следующих четырех аспектах:

1. Ограничения источников данных: построение графов знаний обычно опирается на определенные источники данных (например, веб-сайты энциклопедий, научная литература и специализированные базы данных), которые могут иметь присущий им недостаток неполного охвата;

2. «Узкое место» в извлечении информации: извлечение структурированных знаний из необработанных данных часто требует больших затрат на ручную разметку, и точность и полнота процесса извлечения трудно гарантировать;

3. Эффект задержки обновления: карту знаний необходимо постоянно и

итеративно обновлять для поддержания актуальности, но из-за ограниченности ресурсов фактическая частота обновления часто не соответствует требованиям;

4. Пробел в охвате предметной области: существующая карта знаний с трудом охватывает все профессиональные области, и неизбежно возникают «слепые пятна» знаний, специфичные для отдельных областей.



Рисунок 2.1 – Примеры фрагментов графов знаний об Эйнштейне и Нолане во Freebase

Неполнота графа знаний приводит к тому, что соответствующие графы обычно несвязны (имеют несвязанные между собой части). Например, как показано на рисунке 2.1, Эйнштейн и Нолан достигли больших успехов в своих областях, но между ними нет прямой связи в графе знаний, несмотря на то, что у них богатый набор соседних сущностей и схожие структуры графов. Такая структура графа подразумевает, что Эйнштейн и Нолан фактически разделяют потенциально идентичную семантическую информацию – известность и даже научно-фантастические фильмы Нолана могут ссылаться на достижения Эйнштейна в физике. Однако существующим моделям нейронных сетей на основе графов, использующим агрегацию, сложно

работать с такими разъединенными структурами графов, поскольку как распространение информации, так и агрегация в нейронных сетях на основе графов реализуются на основе связей между узлами графа. Между тем, в данной диссертации установлено, что большинство графов знаний имеют подобную разъединенность. Для проверки этой гипотезы в данной диссертации обучающие наборы FB15k-237 [125] и WN18RR [8], которые обычно используются в обучении представлению графов знаний, преобразуются в неориентированные графы, и для этих неориентированных графов осуществлен обход для вычисления количества их связных компонент. Связная компонента – подграф графа, в котором любые два узла могут быть соединены друг с другом путями, в то время как узлы с разными связными компонентами не могут быть соединены друг с другом путями. В результате обхода алгоритма в данной работе установлено, что FB15k-237 содержит 41 независимую связную компоненту, а WN18RR – 430 независимых связных компонент. Анализ разреженности данных и методов их обработки был предварительно исследован нами в работе [126] на примере анализа лесных пожаров, где применялись методы прогнозирования на основе неполных данных. Для решения проблемы несвязности графа знаний в настоящем исследовании утверждается, что новая модель графовой нейронной сети должна не только извлекать локальную структурную информацию узлов, но и учитывать глобальную структурную информацию всего графа.

В последние годы метод обучения представлению графа, основанный на максимизации взаимной информации (Deep Graph Infomax, DGI), предлагает новые идеи для обучения графов без учителя. В отличие от традиционных методов, основанных на функции потерь при реконструкции графа, DGI способен более полно улавливать структурные особенности графа, максимизируя взаимную информацию между локальными представлениями узлов и глобальным представлением графа. Однако у стандартного подхода DGI есть два очевидных ограничения: во-первых, он применим только к

неориентированным графам; и во-вторых, он не способен обучаться реляционным представлениям, что ограничивает его применение в графах знаний. Для решения этих проблем в данной главе предлагается улучшенный алгоритм максимизации взаимной информации в глубоких реляционных графах, Dual-FusionKG. Ключевым нововведением этого алгоритма является использование двухканальной архитектуры Entity-Relation Fusion Graph Attention Network (ERGAT), которая обрабатывает структурную и лингвистическую информацию графов знаний отдельно.

Принцип работы предлагаемого в настоящей диссертации алгоритма Dual-FusionKG заключается в следующем: первый модуль ERGAT фокусируется на структурном обучении, получая локальные и глобальные структурные характеристики неконтролируемым образом путем максимизации взаимной информации; второй модуль ERGAT захватывает семантическую информацию на основе фактических триплетов с использованием контролируемого обучения; и, наконец, генерирует полное вложенное представление графа знаний путем объединения структурной и семантической информации, а ConvE используется в качестве декодера для прогнозирования тройной оценки.

В частности, следует отметить, что предложенный в этой главе модуль ERGAT вносит инновационные улучшения в следующие аспекты: введение операций слияния сущностей и связей для эффективной интеграции информации о соседних сущностях и связях; и разработка механизма внимания к связям для динамического управления процессом обновления представления узлов.

Основные результаты этой главы можно суммировать в три пункта:

1. Разработка алгоритма Dual-FusionKG, который впервые применяет максимизацию взаимной информации к обучению представлению многореляционного графа и обеспечивает сквозное встраивание графа знаний.

2. Разработка кодировщика ERGAT, реализующего адаптивное обучение представлений посредством механизмов взаимной корреляции и реляционного внимания.

3. Проверка эффективности модели с помощью экспериментов, особенно демонстрация преимуществ в скорости сходимости.

2.2 Графовые нейронные сети

Сверточные нейронные сети (CNN), как важный представитель методов глубокого обучения, совершили прорыв в компьютерном зрении, обработке естественного языка и других областях благодаря своей мощной способности извлекать признаки. Этот тип сети особенно подходит для обработки евклидовых пространственных данных с регулярной сеткой, таких как изображения, речь и текст. Ключ к их успеху заключается в использовании свойства трансляционной инвариантности данных: путем скольжения общего сверточного ядра по регулярной сетке можно извлекать локальные признаки единообразным способом, тем самым создавая иерархические представления признаков.

Однако в реальном мире существует большое количество неевклидовых пространственных данных, таких как данные с графовой структурой, например, социальные сети, сети белковых взаимодействий и транспортные сети. В отличие от данных с регулярной сеткой, каждый узел графовых данных имеет уникальную локальную схему связей с различным количеством соседних узлов и методами связей, и эта структурная неоднородность полностью нарушает предположение о трансляционной инвариантности, на котором основаны традиционные CNN. В частности, в данных графов невозможно определить ни регулярные отношения соседства, ни равномерные сверточные операции, что делает принципиально сложным прямое применение CNN к данным графов.

Именно это существенное различие привело к исследованиям и разработке графовых нейронных сетей (GNN). В отличие от CNN, GNN специально разработаны для работы с нерегулярными данными графовой структуры и способны эффективно улавливать сложные топологические связи между узлами за счет разработки новых механизмов передачи и агрегирования информации. Такая целенаправленная разработка делает GNN идеальными для обработки данных графовой структуры и открывает новый технический путь для анализа сложных системных взаимосвязей.

В последние годы, с быстрым развитием методов глубокого обучения, графовые нейронные сети (GNN) стали важным методом обработки данных графовой структуры. Такие сети заимствуют идею локального извлечения признаков из сверточных нейронных сетей путем моделирования зависимостей между узлами на графе, и поэтому в некоторых источниках их также называют графовыми сверточными нейронными сетями. Однако построение графовых нейронных сетей сталкивается с рядом ключевых проблем.

Во-первых, присущие графовым данным неевклидовы свойства затрудняют прямое применение традиционных сверточных операций. Тот факт, что каждый узел имеет уникальную локально связанную структуру, нарушает предположение о трансляционной инвариантности, на котором основаны традиционные сверточные нейронные сети, что требует от исследователей разработки специализированных операторов свертки графов. Во-вторых, графовые данные в реальном мире демонстрируют значительное разнообразие. Взяв в качестве примера социальные сети, можно увидеть, что различные взаимодействия, такие как подписка, лайк и ретвит, имеют различное семантическое значение и силу связей; в энергетических сетях важность узлов динамически меняется в зависимости от их топологического расположения и функций; а узлы в сети Интернет могут даже содержать гетерогенные типы данных. Это разнообразие предъявляет более высокие

требования к способности графовых нейронных сетей к выражению признаков. Наконец, обработка больших объемов графовых данных также представляет серьезные проблемы. Реалистичные графы, содержащие миллионы или даже десятки миллионов узлов и ребер, требуют от алгоритмов достижения баланса между вычислительной эффективностью, объемом памяти и производительностью модели.

В ответ на эти вызовы современные исследования в области графовых нейронных сетей ведутся по трем основным направлениям: проектирование архитектуры сети, моделирование сложной информации и оптимизация обучения. В данной статье рассматривается прогресс в исследованиях по построению архитектуры сети. В настоящее время основные методы построения графовых нейронных сетей можно разделить на две категории: методы, основанные на теореме свертки, и методы, основанные на функции агрегирования.

Графовая нейронная сеть, основанная на теореме о свертке, происходит из спектрального метода в области обработки сигналов. Согласно теореме о свертке, операция свертки во временной области эквивалентна операции умножения в частотной области:

$$F(f * g) = F(f) \cdot F(g). \quad (2.1)$$

Здесь F обозначает преобразование Фурье. Обратное преобразование Фурье F^{-1} получается следующим образом:

$$f * g = F^{-1}(F(f) \cdot F(g)). \quad (2.2)$$

Эта теорема обеспечивает математическую основу для свертки графов. Определение преобразования Фурье на графе требует помощи разложения матрицы Лапласа L графа на собственные значения. Пусть U – матрица собственных векторов матрицы L . Преобразование Фурье сигнала X графа имеет вид:

$$\hat{X} = U^T X. \quad (2.3)$$

Исходя из этого, операция свертки графа для двух сигналов графа X и Y может быть выражена следующим образом:

$$X_G * Y = U((U^T X) \odot (U^T Y)). \quad (2.4)$$

Здесь \odot обозначает произведение Адамара. Пусть g_θ – обучаемое ядро свертки, тогда свертку графа можно упростить следующим образом:

$$\hat{X}_G = U g_\theta U^T X. \quad (2.5)$$

Самые ранние спектральные сверточные нейронные сети (Spectral CNN) были реализованы путем наложения таких графовых сверточных слоев:

$$X^{(1+1)} = \sigma(U F^{(1)} U^T X^{(1)} W^{(1)}). \quad (2.6)$$

Здесь $F^{(1)}$ – диагональное ядро свертки 1-го слоя, $W^{(1)}$ – матрица весов, а σ – функция активации. Для снижения вычислительной сложности было предложено усовершенствование, упрощающее вычисления за счет использования свойства .

$$X^{(1+1)} = \sigma(L X^{(1)} W^{(1)}). \quad (2.7)$$

Используется нормализованная матрица Лапласа

$$L^{\text{sym}} = D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} = D^{-\frac{1}{2}} (D - A) D^{-\frac{1}{2}}. \quad (2.8)$$

Полученная в итоге эффективная формула спектральной свертки выглядит следующим образом:

$$X^{(1+1)} = \sigma(D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} X^{(1)}). \quad (2.9)$$

Кроме того, вейвлет-преобразование может также заменить преобразование Фурье в качестве основы для свертки графов [127]. Поскольку данная работа посвящена нейронным сетям на основе графов, основанным на агрегированных функциях, для получения более подробной информации о спектральных методах (например, Чебинет [128] и др.) читателю рекомендуется обратиться к соответствующим монографиям.

Графовые нейронные сети, основанные на теореме о свертке, строятся путем определения графовой свертки в пространстве спектральной области.

Однако этот подход имеет существенный недостаток: как решение обратной матрицы Лапласа, так и умножение матриц для обновления параметров узлов в графовой нейронной сети сталкиваются с большими трудностями при работе с крупномасштабными графовыми данными. Это связано с тем, что крупномасштабные графовые данные могут содержать миллионы или даже десятки миллионов узлов, а операции обратного преобразования или умножения матриц на такой многомерной матрице являются непомерно затратными как по времени вычислений, так и по объему памяти. Поскольку суть свертки заключается в извлечении локальной информации о признаках данных, эффект графовой нейронной сети также может быть достигнут путем аппроксимации графовой свертки рекурсивным агрегированием информации о соседях вокруг каждого узла непосредственно из области пространства.

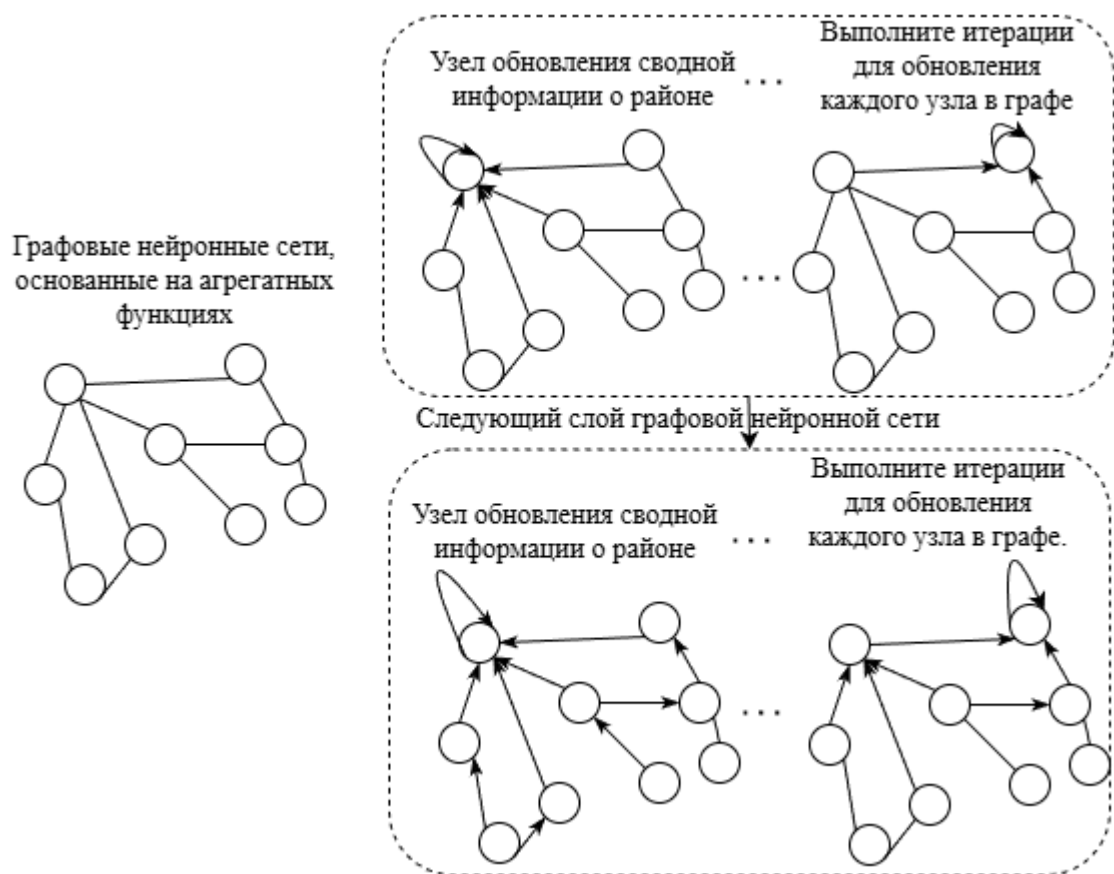


Рисунок 2.2 – Графовая нейронная сеть на основе функции агрегации

Как показано на рисунке 2.2, графовая нейронная сеть, основанная на функции агрегирования, фокусируется на информации об узлах и окружающих их соседних узлах. При обновлении представления каждого центрального узла информация о соседних узлах агрегируется с помощью функции агрегирования $f_{\text{aggregate}}$, после чего обновляется представление самого узла. Затем графовая нейронная сеть проходит по графу, выполняя функцию агрегирования для обновления каждого узла. Путем рекурсивного наложения вышеуказанных операций графовая нейронная сеть может получать информацию о соседях, находящихся дальше в более глубоких слоях сети. Функция агрегирования узла i в $(l+1)$ -м слое обычно определяется следующим образом:

$$f_{\text{aggregate}}(x_i^{(l+1)}) = \sigma\left(\sum_{j \in N_i} \frac{1}{c_i} W^{(l)} x_j^{(l)} + W_0^{(l)} x_i^{(l)}\right). \quad (2.10)$$

Здесь N_i – множество соседних узлов узла i , $x_i^{(l+1)}$ – вектор представления узла i после $(l+1)$ -го слоя графовой нейронной сети, а W – соответствующая матрица весов. В следующем разделе мы представим графовую нейронную сеть, основанную на функции агрегирования, представленной GraphSage [129].

На основе графовых нейронных сетей, использующих функции агрегации, метод `graphsage` вводит операцию сэмплирования (выборки), то есть каждая агрегация не объединяет информацию от всех соседних узлов, а учитывает только некоторые из них. Процесс агрегации в `graphsage` осуществляется на графе циклическим образом, при этом представление каждого узла обновляется в соответствии с представлениями его соседних узлов.

Этот подход можно рассматривать как метод обучения представлений без учителя, поскольку он обучается представлениям узлов без использования размеченных данных. Впоследствии эти обученные представления узлов могут быть применены к различным задачам, таким как классификация

графов, прогнозирование связей и кластеризация узлов и т. д. Масштабируемость и эффективность graphsage делают его хорошо подходящим для обучения представлений графов в больших масштабах. Было показано, что Graphsage превосходит традиционные методы обучения представлений графов в нескольких тестах производительности. Методы обучения широко используются в различных областях, таких как веб-аналитика, рекомендательные системы и биоинформатика.

Графовые нейронные сети на основе агрегатных функций также вводят механизмы, такие как механизм внимания, для определения уровня участия агрегированных соседних узлов [130], вместо предыдущего подхода, где вся информация о соседних узлах имеет одинаковый вес. Для каждого соседнего узла определяется вес внимания:

$$a_{i,j} = \frac{\exp(\text{Leakly ReLU}(W[x_i \parallel x_j]))}{\sum_{k \in N_i} \exp(\text{Leakly ReLU}(W[x_i \parallel x_k]))}. \quad (2.11)$$

Здесь $x_i \parallel x_j$ обозначает сшивание векторов узлов x_i и x_j . LeaklyReLU – функция активации, вариант ReLU.

Результаты обзора литературы по графовым нейросетям двух типов сведены в табл.2.1.

Таблица 2.1 – Репрезентативные работы по графовым нейронным сетям

Алгоритмы	Соответствующие работы
Графовые нейронные сети, основанные на теореме свертки.	Spectral CNN[131], Chebynet[128],GCN[132], GWNN[127],GraphHeat [133],PPNP [134],SPC[135].
Графовые нейронные сети, основанные на агрегатных функциях.	MoNet[136],MPNNs[137],GNNs[138],GAT[130], GraphSAGE[129],DCNN[139],ConfGCN[140],HGNNL[141].

В то же время, исследователи в данной области работают и над изучением более сложных функций агрегирования. Традиционные функции агрегирования обычно включают только суммирование векторов представления соседних узлов или вычисление векторного произведения. В отличие от этого, GraphSage вводит функцию агрегирования на основе среднего значения, а также функцию агрегирования на основе сети долговременной кратковременной памяти (LSTM [142]). В частности, в процессе агрегирования информации об узлах graphsage заменяет традиционную операцию суммирования операцией вычисления среднего значения или использует LSTM для извлечения признаков.

2.3 Многореляционная графовая нейронная сеть на основе максимизации взаимной информации: общая структура Dual-FusionKG

В данном исследовании, в его части, направленной на решение проблемы одновременного учета структурной и семантической информации при обучении представлению графа знаний, предлагается новый алгоритм многореляционной графовой нейронной сети Dual-FusionKG. Ключевое нововведение алгоритма заключается в разработке двухканальной архитектуры сети внимания графа слияния сущностей и связей (ERGAT – Entity-Relationship Fusion Graph Attention Network), которая используется для извлечения различных измерений признаков графа знаний. С точки зрения технической реализации, алгоритм Dual-FusionKG обеспечивает многостороннее обучение признакам с помощью двух параллельных алгоритмов ERGAT: с одной стороны, он использует сигналы триплетов фактов для точного моделирования семантических связей между сущностями; с другой стороны, он эффективно захватывает топологические структурные особенности графа знаний, максимизируя взаимную информацию между локальными представлениями узлов и глобальными представлениями графа.

Эта двухканальная конструкция преодолевает зависимость традиционных методов от явных связей и позволяет обнаруживать потенциальные семантические связи. В частности, алгоритм ERGAT внедряет механизм внимания, учитывающий взаимосвязи, который обеспечивает дифференцированную обработку разнородных связей путем динамического вычисления весовых коэффициентов важности соседних узлов. Модуль использует стратегию слияния признаков сущностей и связей для объединения структурной и семантической информации в общем пространстве представлений, что обеспечивает более богатые представления признаков для последующих задач вывода знаний.

Алгоритм Dual-FusionKG представляет собой алгоритм кодирования-декодирования для обучения представлению графа знаний, как показано на рисунке 2.3. Алгоритм состоит из двух независимых алгоритмов ERGAT, образующих кодировщик для извлечения векторов представления сущностей и отношений, и использует ConvE [8] в качестве декодера для оценки оценок троек фактов. Для заданного графа знаний Dual-FusionKG преобразует его входные данные в две формы данных для получения множественной информации о графе знаний. Сначала Dual-FusionKG передает матрицу смежности входного графа первому алгоритму ERGAT для захвата полной структурной информации графа знаний путем максимизации взаимной информации. Затем второй алгоритм ERGAT извлекает семантическую информацию на основе существующих троек фактов, используя традиционные методы обучения с учителем. На рисунке 2.4 показано, как ERGAT осуществляет агрегацию информации. После завершения извлечения информации Dual-FusionKG объединяет семантическую информацию графа знаний со структурной информацией для генерации окончательных векторов встраивания сущностей и отношений. Наконец, используя convе в качестве декодера, вычисляются оценки троек фактов, подлежащих оценке, что, в свою очередь, позволяет предсказать

недостающие сущности.

Функцию, выполняемую алгоритмом Dual-FusionKG, можно определить следующим образом:

$$f_r(h, t) = f(\text{vec}(f(\text{ER} - \text{GAT}[X_h; X_r] * \omega))W)X_t. \quad (2.12)$$

Здесь $\text{ER} - \text{GAT}[X_h; X_r]$ – представление встраивания (эмбединга) с семантической информацией и полной структурной информацией, полученной из ER-GAT, f – операция на полносвязном слое. В контексте ConvE вектор встраивания будет адаптирован как двумерная матрица в качестве входных данных для ConvE-декодера, vec – преобразование (растяжение) карты признаков, сгенерированной обычной свёрткой, в вектор, $*$ – операция свёртки, ω – ядро свёртки в ConvE, а W – матрица весов признаков, извлечённых ConvE. Далее подробно рассмотрены алгоритм кодировщика ERGAT, способы извлечения семантической информации с помощью ERGAT и способы максимизации взаимной информации для получения полной структурной информации графа знаний.



Рисунок 2.3 – Схема работы алгоритма Dual-FusionKG

2.4 Сеть внимания на основе графов слияния сущностей и связей и оценки максимальной взаимной информации

При обучении представлений многосвязных графов (например, графов знаний) большинство существующих алгоритмов графовых нейронных сетей просто используют информацию о связях для оптимизации только представлений узлов, игнорируя взаимоотношения между связями и узлами.

Например, в шести триплетах фактов обучающих данных FB15k-237, перечисленных в таблице 2.2, можно наблюдать примечательное явление:

В триплетах фактов связь может быть сильно коррелирована с хвостовой сущностью или же с головной сущностью. Для учета направленности таких связей и корреляций сущностей предлагается сеть внимания на основе слияния графов сущностей и связей (ER-GAT).

Таблица 2.2 – некоторые триплеты фактов в FB15k-237

Головная сущность	Отношения (связи)	Хвостовая сущность
Billy Ray Cyrus	/common/topic/webpage	Official Website
Technical University of Berlin	/common/topic/webpage	Official Website
Pennsylvania Academy of the Fine Arts	/common/topic/webpage	Official Website
10th Screen Actors Guild Awards	/award/award_honor/award_winner	Tony Shalhoub
10th Screen Actors Guild Awards	/award/award_honor/award_winner	Charlize Theron
10th Screen Actors Guild Awards	/award/award_honor/award_winner	Elijah Wood

В первых трех триплетах наблюдается сильная корреляция между связями и хвостовыми сущностями, в то время как в последних трех триплетах корреляция между связями и головными сущностями более значительна. Чтобы уловить эту направленность корреляции связей и сущностей, в этой данном исследовании предлагается сеть внимания на основе графа слияния сущностей и связей (ERGAT – Entity-Relation Fusion Graph Attention Network) для точного обучения представлений многосвязных графов путем параметрического представления связей и сущностей. Как показано на рисунке 2.4, ERGAT опирается на идею функции агрегации в графовых нейронных сетях и использует представление, объединяющее узлы и отношения для обновления скрытой информации узлов, вместо того, чтобы

полагаться только на представление соседних узлов для участия в процессе агрегации. Одновременно ERGAT внедряет уникальный механизм внимания к отношениям, способный изучать важность соседних узлов и отношений для целевого узла, что позволяет осуществлять адаптивное обновление скрытой информации узлов. Кроме того, ERGAT создает независимую матрицу параметров для динамического обновления представления встраивания отношений, обеспечивая адаптивное обучение представлению отношений в графах с множеством отношений.

Функция агрегирования ER-GAT математически определяется следующим образом:

$$x_i^{(l+1)} = \sigma\left(\sum_{r \in R} \sum_{j \in N} \frac{1}{c_{i,r}} \alpha_{i,j}^{(l)} \varphi(x_i^{(l)}, r_{i,j}^{(l)})\right). \quad (2.13)$$

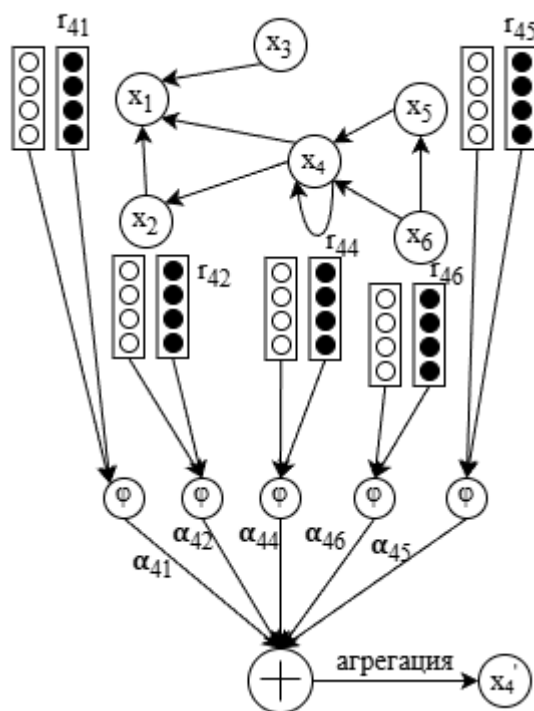


Рисунок 2.4 – Объединение сущностей и связей в сети внимания графа

На рисунке 2.4 представлена визуализация (пример работы) функции агрегирования ERGAT. Граф содержит шесть узлов, с акцентом на узел 4. При обновлении скрытого представления узла 4 ERGAT сначала объединяет

информацию о соседних узлах и их соответствующих отношениях посредством операции взаимного отношения. Затем ERGAT вычисляет оценку внимания к отношениям для каждого соседнего узла. После этого информация об узлах 1, 2, 5 и 6 и их соответствующих отношениях агрегируется в узел 4 в соответствии с уравнением функции агрегирования. В последующем процессе обратного распространения ошибки представления узла и отношений будут обновляться автоматически.

Здесь l представляет собой l -й слой ER-GAT, $\alpha_{i,j}^{(l)}$ – показатель внимания узла x_i к связям (отношениям) $r_{i,j}$ с соседним узлом x_j . Процесс вычислений показан на рисунке 2.5. $c_{i,r} = |N_i^r|$ – коэффициент нормализации, который обозначает соседние узлы, с которыми узел i связан через отношение r . $\varphi(x_i^{(l)}, r_{i,j}^{(l)})$ – операция объединения информации о взаимосвязях между узлами. Головная сущность и связи объединяются посредством операции кросс-корреляции $\varphi(h, r) = h \otimes r$ для создания промежуточного представления, чувствительного к направлению. Здесь \otimes обозначает операцию свертки.

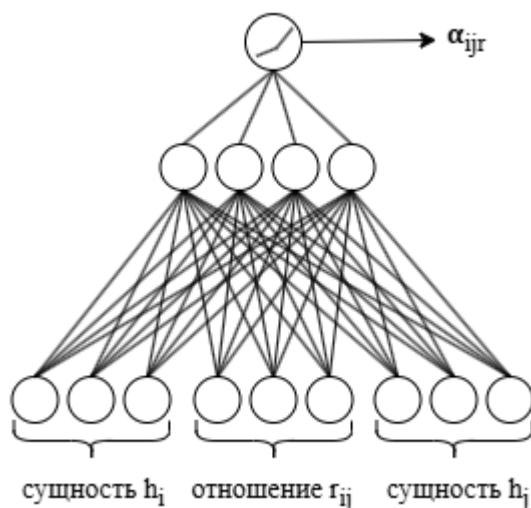


Рисунок 2.5 – Механизм реляционного внимания (внимания к отношениям)

С помощью ER-GAT модель может получать локальную информацию графа знаний, и когда для обучения с учителем вводится известный триплет фактов, она может получать семантическую информацию графа знаний. Однако существуют также случаи, когда узлы не связаны напрямую с целевым узлом. Например, предположим, что триплет фактов (x_7, r_{74}, x_4) не встречается в графе знаний. Поскольку в этой главе рассматривается статический граф знаний, где все сущности и отношения фиксированы на момент обучения, для x_7 возможны два сценария:

1) Узел x_7 может достичь узла x_4 по пути, например, $(x_7, r_{75}, x_5, r_{54}, x_4)$. ERGAT, как алгоритм графовой нейронной сети, позволяет узлу x_7 получать информацию о своем соседнем узле x_4 через два слоя ERGAT. В частности, первый слой ERGAT суммирует информацию от x_7 до x_5 как $x_5^1 = \alpha_{75} \varphi(x_7, r_{75})$, где x_5^1 – это представление эмбединга x_5 , α_{75} – это оценка внимания к отношениям между x_7 и x_5 , и $\varphi(x_7, r_{75})$ – это информация о слиянии сущностей и отношений. Затем второй слой ERGAT агрегирует информацию от обновленного x_5 до x_4 , т.е. $x_4^2 = \alpha_{54} \varphi(x_5^1, r_{54})$. Таким образом, хотя x_4 и x_7 не связаны напрямую, x_4 может получить доступ к информации (x_7, r_{75}) таким образом. Из этого примера видно, что ERGAT способен получать информацию о соседних узлах более высокого порядка. При обучении многослойных графовых нейронных сетей с увеличением числа слоев векторные представления узлов в графе стремятся к равенству: явление, известное как «чрезмерное сглаживание», поэтому число слоев графовой нейронной сети обычно устанавливается равным двум.

2) x_7 не может достичь x_4 никаким путем, что означает, что x_7 и x_4 являются разными связными компонентами. Как упоминалось во введении к этой главе, несвязность графа знаний приводит к неспособности графовой нейронной сети справиться с этой ситуацией. Поэтому в данной главе эта проблема решается путем максимизации взаимной информации между

локальными и глобальными представлениями графа знаний, чтобы алгоритм мог улавливать глобальную структурную информацию.

Оценка максимальной взаимной информации на основе сетей внимания с графом слияния сущностей и связей

Далее мы разъясняем, как использовать сеть внимания на основе графа слияния сущностей и связей (ERGAT) для захвата полной структурной информации графа знаний. Сначала мы подробно рассмотрим концепцию оценки максимальной взаимной информации на графах, а затем изучим, как реализовать оценку максимальной взаимной информации на графе с множественными связями.

Оценка максимальной взаимной информации

Предположим, что имеются два сигнала – X и Z . Их взаимная информация определяется следующим образом:

$$\begin{aligned} I(X; Z) &= H(X) - H(X | Z) \\ &= \int_{xz} \log \frac{dP_{XZ}}{dP_X \times dP_Z} dP_{XZ} \quad . \quad (2.14) \\ &= D_{KL}(P_{XZ} \parallel P_X \times P_Z) \end{aligned}$$

Здесь P_{XZ} – совместное распределение X и Z , P_X и P_Z – их маргинальные распределения. Видно, что взаимная информация двух сигналов означает нахождение разброса Кульбака-Лейблера их совместного и маргинального распределений. Предположим, что X – входные данные до кодирования, обычно случайное представление вложений, а Z – представление эмбединга, полученное в результате обучения представлений. Произведение совместного и маргинального распределений подразумевает два разных случая:

1. P_{XZ} подразумевает рассмотрение входной переменной X и выходного представления Z совместно.

2. $P_X \times P_Z$ подразумевает рассмотрение входной переменной X и выходного представления Z по отдельности.

Если распределения в двух вышеуказанных случаях сильно различаются, то X и Z можно рассматривать как сильно коррелированные. С точки зрения теории информации, обучение представлению можно рассматривать как канал, соединяющий входное и выходное представления, поэтому максимизация взаимной информации между входом и выходом также может означать, что при передаче информации по этому каналу среднее количество информации, передаваемой на символ, максимизируется. Это соответствует передаче большего количества информации меньшим количеством символов, что означает, что обучение представлению в этом случае более эффективно. Однако вычисление взаимной информации между двумя неизвестными переменными является сложной задачей, если использовать методы, основанные на дискриминаторе нейронной оценки взаимной информации (MINE) и глубокой оценке максимизации взаимной информации (DIM). Максимизация взаимной информации может обеспечить нижнюю границу оценки дисперсии Кульбака-Лейблера с помощью представления Донскера-Варадхана:

$$D_{\text{KL}}(P \parallel Q) \geq \sup_{T: \Omega \rightarrow \mathbb{R}} E_P[T] - \log(E_Q[e^T]). \quad (2.15)$$

Таким образом, (2.14) можно заменить на:

$$\begin{aligned} I(X; Z) &= D_{\text{KL}}(P_{XZ} \parallel P_X \times P_Z) \\ &\geq \sup_{T: \Omega \rightarrow \mathbb{R}} E_{P_{XZ}}[T] - \log(E_{P_X \times P_Z}[e^T]). \end{aligned} \quad (2.16)$$

В приведенном выше уравнении T принадлежит множеству функций, областью определения которых является P_{XZ} или $P_X \times P_Z$, а областью значений – \mathbb{R} . Задача состоит в том, чтобы найти функцию T из множества функций \mathbb{R} , которая может максимизировать взаимную информацию между X и Z . Однако пространство поиска множества функций настолько велико, что найти правильную T с помощью традиционных методов оптимизации сложно. Поэтому здесь T обычно параметризуется как нейронная сеть, а аппроксимация T находится путем подгонки функции с помощью нейронной

сети. Естественно, максимизацию взаимной информации в приведенном выше уравнении можно также рассматривать как оценку отрицательных примеров, которая вводит дискриминаторную функцию $T=D(X,Z)$. Входной сигнал X и соответствующий ему выходной сигнал Z рассматриваются как пары положительных примеров, а случайный входной сигнал и соответствующий ему выходной сигнал – как пары отрицательных примеров. Затем максимизация функции правдоподобия (2.19) приводит к максимизации оценки взаимной информации (поскольку дивергенция Кульбака-Лейблера теоретически неограничена, здесь используется дисперсия JS-дивергенция (Дженсена-Шеннона) для повышения эффективности оценки). Дивергенция Кульбака-Лейблера измеряет степень различия между двумя вероятностными распределениями, тогда как JS-дивергенция является ее улучшенной версией, решающей проблему асимметрии дивергенции Кульбака-Лейблера посредством обработки симметрии.

$$D_{JS}(P_X \parallel P_Z) = \frac{D_{KL}(P_X \parallel \frac{P_X + P_Z}{2})}{2} + \frac{D_{KL}(P_Z \parallel \frac{P_X + P_Z}{2})}{2}, \quad (2.17)$$

$$\begin{aligned} I(X; Z) &= D_{JS}(P_{XZ} \parallel P_X \times P_Z) \\ &\geq \max_D (E_{P_{XZ}} [\log \sigma(D(X, Z))] \\ &\quad + E_{P_X \times P_Z} [\log(1 - \sigma(D(\tilde{X}, \tilde{Z})))]). \end{aligned} \quad (2.18)$$

На этом этапе оценивается максимизация взаимной информации сигналов X и Z .

Оценка максимальной взаимной информации на графах знаний.

Граф знаний – это классический многосвязный граф G , содержащий $|N|$ типов сущностей и $|R|$ типов связей. A – матрица смежности графа знаний, элементы которой равны типам связей сущностей.

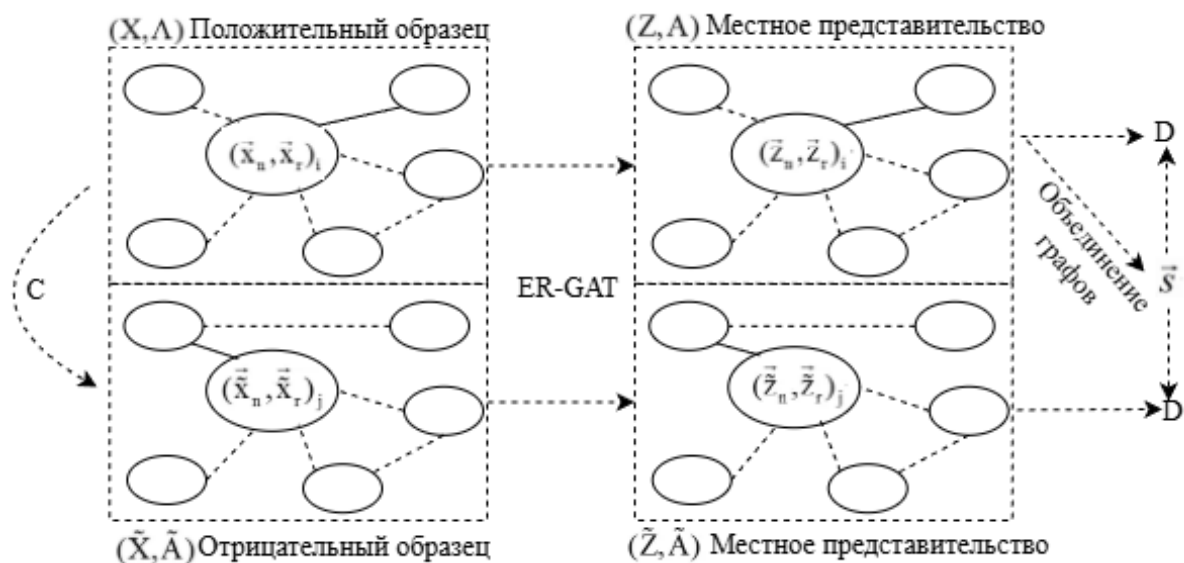


Рисунок 2.6 – Максимальная взаимная информация на графах на основе ER-GAT

На рисунке 2.6 представлено графическое изображение оценки максимальной взаимной информации с помощью ER-GAT. $X = (\bar{x}_n, \bar{x}_r)$ – начальный случайный эмбединге графа знаний, где \bar{x}_n и \bar{x}_r – начальные эмбединги сущностей и связей. $Z = (\bar{z}_n, \bar{z}_r)$ – представление эмбедингов, полученных с помощью ER-GAT, основанное на X , где распределения \bar{z}_n и \bar{z}_r обозначают полученные встраивания сущностей и связей. $\tilde{X} = (\tilde{x}_n, \tilde{x}_r)$ – начальный случайный эмбединг отрицательных примеров, а $\tilde{Z} = (\tilde{z}_n, \tilde{z}_r)$ – представление эмбединга, полученного с помощью ER-GAT, основанное на \tilde{X} . Благодаря операции слияния сущностей и связей и механизму внимания к связям в ER-GAT, каждая сущность в графе может получить информацию об окружающих соседях через функцию агрегирования, поэтому Z можно рассматривать как локальное представление графа знаний. Для получения структурной информации на глобальном уровне Dual-FusionKG преобразует полученное локальное представление Z в глобальное представление, используя функцию объединения графов R:

$$\vec{S} = \frac{1}{|N|} \sum_i^{|N|} (z_n)_i. \quad (2.19)$$

Здесь $|N|$ – количество сущностей, а $(z_n)_i$ – эмбединг i -й сущности, изученное на основе представления ER-GAT. Путем агрегирования всех локальных представлений можно эффективно получить глобальную информацию графа знаний.

Согласно уравнению (2.18), Dual-FusionKG использует дискриминатор D , где $D(Z, \vec{S})$ выполняется оценка отрицательной выборки. Z и связанный с ним \vec{S} рассматриваются как положительная локально-глобальная структурная информация, подразумевающая выборку из совместного распределения локальных и глобальных представлений. Затем отрицательные выборки получаются из исходного графа знаний с помощью механизма нарушения C :

$$(\tilde{X}, \tilde{A}) = C(X, A). \quad (2.20)$$

Отрицательные примеры \tilde{Z} с правильными значениями \tilde{S} будут рассматриваться как отрицательная локально-глобальная структурная информация, то есть они выбираются из произведения распределений ребер локального и глобального представлений. Граф с отрицательными примерами имеет точно такой же состав узлов, как и исходный граф, но положения узлов изменяются по сравнению с исходным графом, что приводит к другому локальному представлению. Когда Dual-FusionKG оценивает максимальную взаимную информацию локальных и глобальных структурных представлений для объединения графов, узлы с различными компонентами связности будут обмениваться информацией для лучшего представления. В итоге, алгоритм получает локальное представление Z исходного графа с помощью прохода (X, A) алгоритма ER-GAT и локальное представление \tilde{Z} графа с отрицательными примерами с помощью прохода (\tilde{X}, \tilde{A}) алгоритма ER-GAT. Следуя идее оценки максимальной взаимной информации, дискриминатор D устанавливается как билинейная функция:

$$D(\bar{z}_i, \bar{s}) = \sigma(\bar{z}_i W \bar{s}). \quad (2.21)$$

Здесь $\bar{z}_i = (\bar{z}_n, \bar{z}_r)_i$. Максимизация оценки взаимной информации на графе знаний представляет собой процесс слияния локального представления Z с глобальным представлением \bar{S} . Полная структурная информация может быть получена как с локальной, так и с глобальной точек зрения, а итоговая функция потерь для максимизации оценки взаимной информации на графе знаний выглядит следующим образом:

$$\begin{aligned} L_{MI} = & \frac{1}{N+M} \left(\sum_{i=1}^N E_{(X,A)} [\log(D((\bar{z}_n, \bar{z}_r)_i, \bar{s}))] \right. \\ & \left. + \left(\sum_{j=1}^M E_{(\tilde{x}, \tilde{A})} [\log(1 - D((\tilde{z}_n, \tilde{z}_r)_j, \bar{s}))] \right) \right). \end{aligned} \quad (2.22)$$

Здесь N – количество положительных примеров, а M – количество отрицательных примеров, обычно $N=M$.

Полный псевдокод процесса обучения для алгоритма Dual-FusionKG выглядит следующим образом (Алгоритм 2.1):

Алгоритм 2.1 Алгоритм обучения Dual-FusionKG:

Дано: Граф $G=(N,R,F)$, матрица смежности A , признаки X

Параметр: $\theta = \{W_{Structure}, W_{r Structure}, W_{Semantic}, W_{r Semantic}, W_D, \theta_{ConvE}\}$

для epoch из $[1, T]$:

#Циклы обучения

#Прямое распространение

1. Структурное кодирование (обучении без учителя)

1) Случайным образом перемешать и сгенерировать отрицательные образцы $\tilde{A} = \text{shuffle}(A)$.

2) Вычислить локальные эмбединги Z и эмбединги отрицательных образцов \tilde{Z} .

$$Z = \text{ERGAT}(X, A; W_{\text{Structure}}, W_{\text{r Structure}})$$

$$\tilde{Z} = \text{ERGAT}(X, \tilde{A}; W_{\text{Structure}}, W_{\text{r Structure}})$$

- 3) Вычислить глобальные эмбединги S положительных образцов: $S = \text{mean}(Z)$.
 - 4) Вычислить степень соответствия локальной и глобальной оценок d (оценки для положительного образца) и \tilde{d} (оценки для отрицательного образца) для каждого узла.
 - 5) Рассчитать структурные потери L_{MI} .
2. Семантическое кодирование (обучение с учителем)

- 1) Обучение троек путем выборки из набора фактов: $\text{Batch} = \text{sample}(F)$
- 2) Вычислить семантические векторные представления с помощью семантического кодирования ERGAT (используя ту же архитектуру кодировщика, но другие параметры):
- 3) $H_{\text{Semantic}} = \text{ERGAT}(X, \text{Batch}; W_{\text{Semantic}}, W_{\text{r Semantic}})$.
- 4) Функция потерь при обучении с учителем $L_{\text{sup}} = \text{RankingLoss}(H_{\text{Semantic}}, \text{Batch})$.

3. Слияние и декодирование

- 1) $H_{\text{final}} = \alpha \cdot Z + \beta \cdot H_{\text{Semantic}}$. Взвешенная сумма, α, β являются обучаемыми параметрами.
- 2) Прогнозирование связей, $\text{score} = \text{ConvE}(H_{\text{final}}; \theta_{\text{ConvE}})$
- 3) $L_{\text{task}} = \text{CrossEntropy}(\text{score})$
- 4) Общее значение функции ошибки $L = L_{\text{MI}} + \lambda * L_{\text{sup}} + \gamma * L_{\text{task}}$

#Обратное распространение

1. Дискриминатор, $\nabla W_{\text{D}} = \partial L / \partial W_{\text{D}}$

2. Структурный кодировщик

$$\nabla Z = \partial L_{\text{MI}} / \partial Z + \gamma * \alpha * \partial L_{\text{task}} / \partial H$$

$$\nabla W_{\text{Structure}}, \nabla W_{\text{r Structure}} = \text{ERGAT}_{\text{Backward}}(\nabla Z, \text{cache})$$

3. Семантический кодировщик

$$\nabla H_{\text{Semantic}} = \partial L_{\text{sup}} / \partial H_{\text{Semantic}} + \gamma * \beta * \partial L_{\text{task}} / \partial H$$

$$\nabla W_{\text{Semantic}}, \nabla W_{\text{r Semantic}} = \text{ERGAT_Backward}(\nabla H_{\text{Semantic}}, \text{cache})$$

4. Decoder

$$\nabla \theta_{\text{ConvE}} = \partial L_{\text{task}} / \partial \theta_{\text{ConvE}}$$

#Обновление параметров (алгоритм ADAM)

AdamUpdate({ $W_{\text{Structure}}, W_{\text{r Structure}}, W_{\text{Semantic}}, W_{\text{r Semantic}}, W_{\text{D}}, \theta_{\text{ConvE}}$ }, скорость обучения η)

Возвратить θ

На данном этапе модель завершает оценку максимальной взаимной информации на графе на основе слияния сущностей и связей в сети внимания графа на графе знаний и реализует извлечение полной структурной информации из графа знаний.

2.5 Экспериментальная оценка алгоритма Dual-FusionKG

Для оценки производительности предложенного алгоритма Dual-FusionKG мы опирались на методологию сравнительного анализа, разработанную в [143], где проводилось тестирование различных алгоритмов глобального поиска. Предложенный алгоритм Dual-FusionKG был протестирован на четырех больших эталонных наборах данных, все из которых используются для задачи дополнения графа знаний. Цель задачи дополнения графа знаний (рисунок 2.7) – восстановить недостающие связи из известных троек фактов.

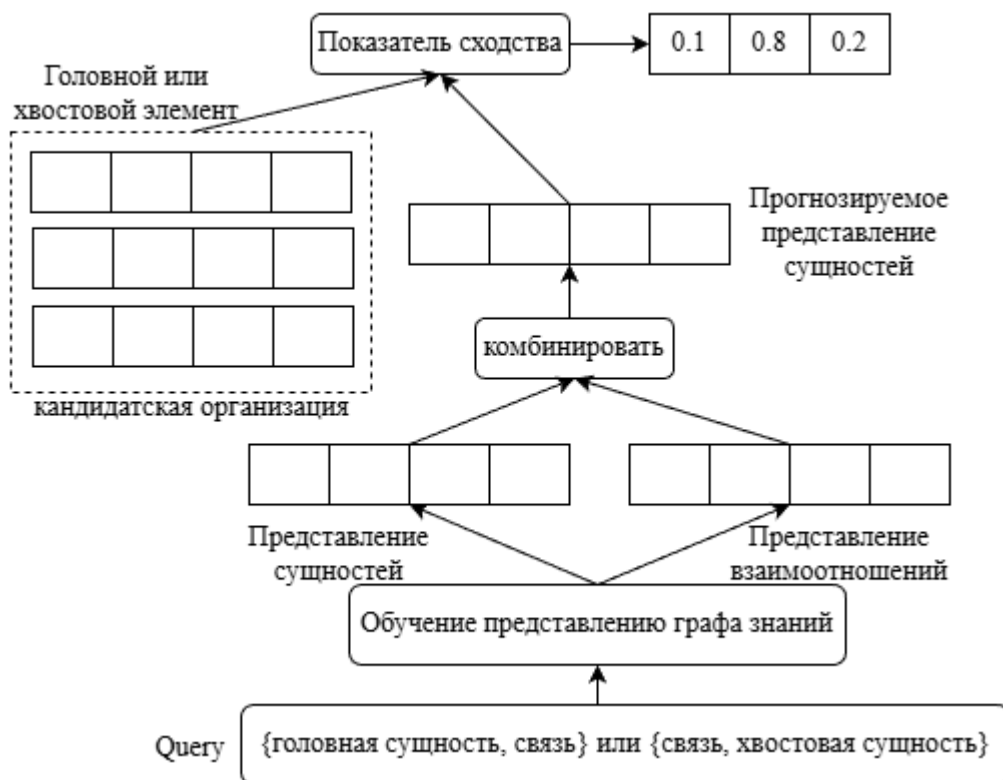


Рисунок 2.7 – Задача по дополнению графа знаний на основе обучения представлению графа знаний

Дана неполная графовая структура знаний $G = (N, R, F)$, где N обозначает множество сущностей, R – множество отношений, а F – множество известных триплетов фактов. Задача дополнения графа знаний заключается в предсказании недостающих триплетов фактов $F^* = (h, r, t) | (h, r, t) \notin F$. Как правило, дополнение графа знаний осуществляется с помощью функции оценки $f(h, r, t)$, основанной на обучении представлений графа знаний, которая используется для оценки достоверности троек фактов. Когда достоверность триады фактов высока, оценка функции f соответственно увеличивается. Как показано на рисунке 2.7, задача дополнения графа знаний основана на обучении представлений графа знаний. В частности, в задаче дополнения графа знаний сначала используется кодировщик для обучения встроенным представлениям сущностей и отношений, а затем вычисляются оценки сходства между всеми

сущностями-кандидатами и неполными тройками фактов, эти сущности-кандидаты ранжируются, и выводятся к сущностям с наивысшим рейтингом. Оценка ранжирования правильной сущности должна быть выше. Следовательно, оценочные метрики для этого эксперимента также связаны с рейтингом.

Таблица 2.3 – Информация об экспериментальном наборе данных

Набор данных	FB15K	FB15k-237	WN18	WN18RR
Число сущностей	14981	14541	40943	40943
Число связей (отношений)	1345	237	18	11
Число триплетов фактов (обучающее множество)	483142	272115	141442	86835
Число триплетов фактов (множество для проверки)	50000	17535	5000	3034
Число триплетов фактов (тестовое множество)	59071	20466	5000	3134
Количество компонентов связи	9	41	13	430

Эксперименты включают четыре набора данных: FB15K [1], FB15k-237 [125], WN18 [1] и WN18RR [8]. В таблице 2.3 представлена сводная статистическая информация по этим наборам данных, а обучающие данные из больших наборов данных преобразуются в неориентированные графы, и вычисляется количество их связных компонентов.

FB15K – граф знаний, основанный на реальных фактах, созданный Freebase, охватывающий сущности в областях кино, актеров, наград, спорта и физических упражнений. Он содержит 14981 сущность и 1345 связей, а его обучающие данные могут быть преобразованы в 9 связных компонентов.

FB15k-237 – подмножество FB15K, из которого удалены обратные ребра из обучающего. Он содержит 14541 сущность и 237 связей, а его обучающие данные могут быть преобразованы в 41 связный компонент.

WN18 – граф знаний, построенный на основе WordNet, описывающий отношения между словами со строгой иерархической структурой. Он

содержит 40943 сущности и 18 типов отношений, а его обучающие данные могут быть преобразованы в 13 компонентов связности.

WN18RR – подмножество WN18, в котором удалены дублирующиеся данные, сгенерированные путем инвертирования триплетов фактов в WN18. Он содержит 40943 сущности и 11 отношений, а его обучающие данные могут быть преобразованы в 430 компонентов связности.

Для сравнения эффективности использованы семь базовых алгоритмов, основанных на различных декодерах. Среди них TransE, Rotate и TorusE – методы представления знаний на основе модели перевода; ConvE и HyperE – это методы представления знаний на основе сверточных нейронных сетей; RGCN и SACN – это методы представления знаний на основе графовых нейронных сетей. Ниже приведены описания этих алгоритмов:

TransE [1] – первый метод представления знаний на основе модели трансляции, реализующее представление графов знаний путем встраивания сущностей и их взаимосвязей в низкоразмерное линейное пространство.

RotatE [4] – представление знаний на основе модели трансляции, которое встраивает сущности в комплексные пространства и использует операции вращения над комплексными векторами для описания взаимосвязей между сущностями.

TorusE [41] – представление знаний на основе модели трансляции, которое расширяет пространство отображения от традиционного линейного пространства до групп Ли.

ConvE [8] – представление знаний на основе сверточных нейронных сетей, которое извлекает признаки сущностей и взаимосвязей с помощью сверточных нейронных сетей.

Hyper [9] – это представление знаний на основе сверточной нейронной сети, которое сводит сверточную нейронную сеть в convE к набору фильтрующих представлений через гиперсеть.

RGCN [10] – представление знаний, основанное на графовых

нейронных сетях, которое реализует представление реляционного встраивания в графовых сверточных нейронных сетях для извлечения структурной информации из графов знаний.

SACN [46] – метод представления знаний, основанный на графовых нейронных сетях, и он предлагает взвешенную графовую нейронную сеть. В отличие от RGCN, который по умолчанию присваивает равные веса всем соседям в функции агрегирования, SACN присваивает обучаемые веса различным типам отношений во время агрегирования.

Эти алгоритмы охватывают основные направления исследований в области обучения представлению графов знаний и могут быть использованы для проверки эффективности алгоритма Dual-FusionKG. Кроме того, добавлены эксперименты с использованием только сети внимания графа слияния сущностей и отношений (ERGAT) для проверки эффективности алгоритма максимизации взаимной информации в графах. В последующих экспериментах ERGAT обозначает алгоритм без максимизации взаимной информации на графе, а Dual-FusionKG – полный алгоритм, включающий сеть внимания графа слияния сущностей и связей и максимизацию взаимной информации на графе.

В качестве метрик оценки используются классические метрики для задачи завершения графа знаний, а именно следующие:

1. Hits@1, Hits@3 и Hits@10: доля правильных сущностей, занимающих первые 1-е, 3-е и 10-е места среди всех сущностей-кандидатов. В уравнении (2.23) $\Pi(\cdot)$ – индикаторная функция (значение функции равно 1, если условие истинно, и 0 в противном случае), а Q – количество всех тестовых образцов.

$$\text{Hits @ } k = \sum_i \frac{\Pi(\text{rank}_i \leq k)}{Q}. \quad (2.23)$$

2. Средний рейтинг (MR): средняя доля правильно ранжированных объектов среди всех объектов-кандидатов.

$$MR = \frac{1}{Q} \sum_i \text{rank}_i. \quad (2.24)$$

3. Средний обратный рейтинг (MRR): средняя сумма обратных рейтингов всех объектов.

$$MRR = \frac{1}{Q} \sum_i \frac{1}{\text{rank}_i}. \quad (2.25)$$

Проведено сравнение скоростей сходимости алгоритмов, при этом в качестве базовых алгоритмов используются convE (алгоритм обучения представлению графа знаний на основе сверточных нейронных сетей) и SACN (алгоритм обучения представлению графа знаний на основе графовых нейронных сетей), а также проверяется влияние обучения представлению на сущности с различной степенью разреженности. Наконец, проводится серия экспериментов по изменению размерности вектора эмбединга, кодировщика ERGAT и способов объединения структурной и семантической информации.

Гиперпараметры алгоритма в этом эксперименте следующие (где выделены жирным шрифтом наилучшие значения гиперпараметров): отсечение кодировщика {0,0, **0,1**, 0,2, 0,3}, размерность вектора встраивания {50, 100, 150, **200**, 500}, отсечение декодера {0,0, 0,1, 0,2, **0,3**}, а размер сверточного ядра convE составляет 10 * 20. Количество сверточных ядер равно 200 (см. оптимальные настройки convE). Каждый набор данных разделен на три части: обучающий набор, проверочный набор и тестовый набор, как показано в таблице 2.3. Алгоритм обновляется с использованием алгоритма Адама [144], а скорость обучения устанавливается равной {1E-2, **1E-3**, 1E-4, 1E-5} в процессе обучения.

Результаты эксперимента

Результаты решения задач по дополнению графа знаний, представлены в таблицах 2.4 и 2.5. Выделенные жирным шрифтом результаты в каждом столбце соответствуют алгоритмам с наилучшими результатами на четырех больших эталонных наборах данных.

Таблица 2.4 – Экспериментальные результаты для задач на дополнение графа знаний (наборы данных FB15k-237 и WN18RR)

Алгоритм	FB15k-237					WN18RR				
	Hits					Hits				
	@10	@3	@1	MR	MRR	@10	@3	@1	MR	MRR
Transe	0.420	0.284	0.174	-	0.257	0.444	0.295	0.027	-	0.182
Rotate	0.533	0.375	0.241	177	0.338	0.571	0.492	0.428	3340	0.476
Toruse	0.485	0.337	0.219	-	0.307	0.534	0.480	0.429	-	0.464
ConvE	0.501	0.356	0.237	244	0.325	0.520	0.440	0.400	4187	0.430
Hyper	0.520	0.376	0.252	250	0.341	0.522	0.477	0.436	5798	0.465
RGCN	0.420	0.230	0.150	-	0.250	-	-	-	-	-
SACN	0.540	0.390	0.260	-	0.350	0.540	0.480	0.430	-	0.470
ERGAT	0.501	0.356	0.237	199	0.325	0.522	0.465	0.421	5131	0.454
Dual-FusionKG	0.549	0.399	0.270	131	0.362	0.543	0.496	0.445	3223	0.491

Таблица 2.5 – Экспериментальные результаты для задач на дополнение графа знаний (наборы данных FB15K and WN18)

Алгоритм	FB15K					WN18				
	Hits					Hits				
	@10	@3	@1	MR	MRR	@10	@3	@1	MR	MRR
Transe	0.749	0.578	0.297	-	0.463	0.943	0.888	0.113	-	0.495
Rotate	0.884	0.830	0.746	40	0.797	0.955	0.952	0.944	309	0.949
Toruse	0.832	0.771	0.674	-	0.733	0.954	0.950	0.943	-	0.947
Conve	0.831	0.723	0.558	51	0.657	0.956	0.946	0.935	374	0.943
Hyper	0.885	0.829	0.734	44	0.790	0.958	0.955	0.947	431	0.951
RGCN	0.825	0.736	0.541	-	0.651	0.955	0.928	0.686	-	0.814
ERGAT	0.885	0.822	0.715	71	0.776	0.956	0.929	0.891	478	0.898
Dual-FusionKG	0.896	0.854	0.746	38	0.798	0.957	0.950	0.943	436	0.948

Как видно из таблиц, алгоритм Dual-FusionKG превосходит другие методы по приведенным показателям. В частности, Dual-FusionKG демонстрирует значительное улучшение по сравнению с алгоритмом обучения для представления графов знаний, который также основан на графовой нейронной сети. На наборе данных FB15k-237 показатель MRR алгоритма Dual-FusionKG на 844,8% выше, чем у RGCN, и на 3,4% выше, чем у SACN. Кроме того, Dual-FusionKG превосходит по этому показателю сеть внимания на основе графа слияния сущностей и связей (ERGAT) на 11,2%. На наборе данных WN18RR показатель MRR алгоритма Dual-FusionKG на 2,0%

лучше, чем у SACN. На наборе данных WN18RR показатель MRR алгоритма Dual-FusionKG улучшился на 2,0% по сравнению с SACN и на 5,5% по сравнению с сетью внимания на основе графа слияния сущностей и связей (ERGAT). На наборах данных FB15K и WN18 результаты MRR алгоритма Dual-FusionKG на 22,6% и 16,5% лучше, чем результаты MRR алгоритма RGCN, соответственно.

Из экспериментальных результатов видно, что экспериментальная производительность Dual-FusionKG на WN18 не очень выдающаяся. WN18 – это граф знаний, содержащий отношения «слово к слову», построенный на основе WordNet, содержащий всего 18 отношений и 141442 обучающих триплета фактов. Кроме того, WN18 содержит множество дублирующихся триплетов фактов, полученных путем инвертирования объектов и субъектов. Небольшое количество типов отношений и дублирующихся кортежей фактов в WN18 могло привести к переобучению алгоритма Dual-FusionKG на этом наборе данных, и этот недостаток необходимо учитывать в будущих исследованиях.

В целом, из этих результатов можно заключить, что алгоритм Dual-FusionKG эффективен и обеспечивает улучшение производительности в обучении представлению графа знаний.

На рисунке 2.8 приведены скорости сходимости трех алгоритмов – Dual-FusionKG, SACN и ConvE – в процессе обучения на наборах данных FB15k-237 и WN18RR. Из рисунка видно, что после 50 итераций обучения на FB15k-237 алгоритм Dual-FusionKG демонстрирует лучшие экспериментальные результаты, чем SACN и convE. После 20 итераций обучения на WN18RR алгоритм Dual-FusionKG показывает лучшие экспериментальные результаты, чем SACN и ConvE. Эти экспериментальные результаты демонстрируют разницу в скорости сходимости между алгоритмами, доказывая, что дополнительная полная структурная информация помогает алгоритмам быстрее сходиться к лучшим результатам.

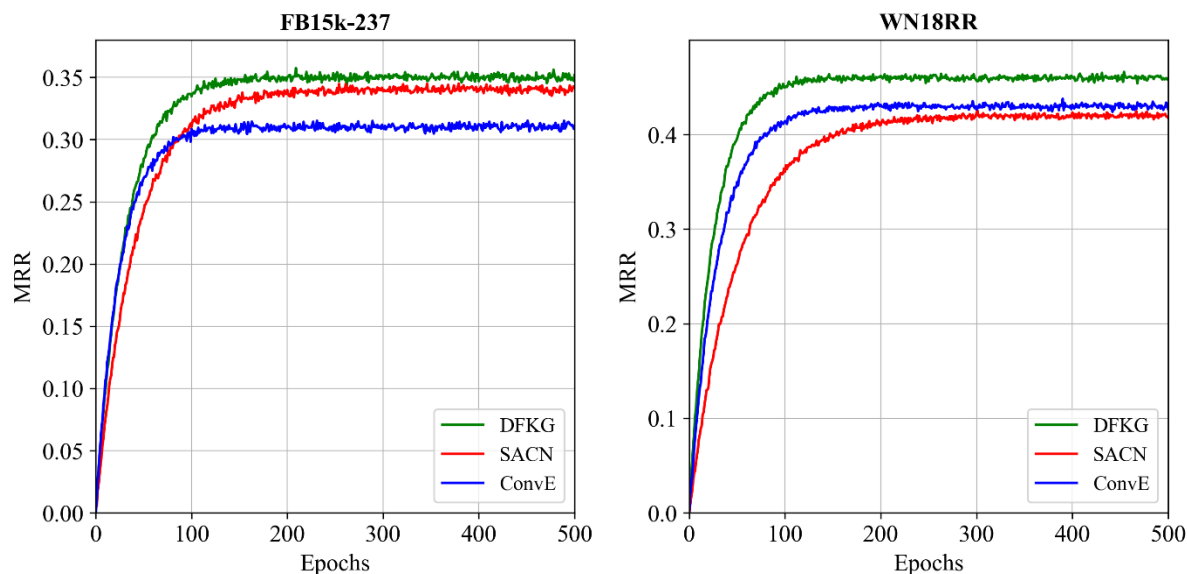


Рисунок 2.8 – Скорость сходимости алгоритмов Dual-FusionKG, SACN и ConvE во время обучения на наборах данных FB15k-237 (a) и WN18RR (b)

Ниже мы исследуем производительность алгоритма Dual-FusionKG на наборе данных FB15K для различных категорий отношений (связей).

Эти классы отношений классифицируются по среднему числу головных сущностей h_{p_r} , связанных с каждой хвостовой сущностью, и числу хвостовых сущностей tp_{h_r} , связанных с каждой головной сущностью. Согласно, если $h_{p_r}, tp_{h_r} < 1.5$, то отношение r считается отношением «один к одному». Если $h_{p_r} < 1.5$ и $tp_{h_r} \geq 1.5$, то отношение r считается отношением «один к N».

Отношение r считается отношением «N к 1», если $h_{p_r} \geq 1.5$ и $tp_{h_r} < 1.5$. Если $h_{p_r} \geq 1.5$ и $tp_{h_r} \geq 1.5$, отношение r считается отношением «N к N».

Таблица 2.6 – Экспериментальные результаты работы алгоритмов с различными типами отношений на FB15K (Hits@10)

Задача	Предсказание головной сущности				Предсказание хвостовой сущности			
	1 к 1	1 к N	N к 1	N к N	1 к 1	1 к N	N к 1	N к N
TransE	0.437	0.657	0.182	0.472	0.437	0.197	0.667	0.500
TransH	0.667	0.817	0.302	0.574	0.637	0.301	0.832	0.608
TorusE	0.775	0.906	0.479	0.859	0.773	0.592	0.893	0.890
ConvE	0.885	0.947	0.404	0.683	0.888	0.511	0.939	0.750
Dual-FusionKG	0.916	0.958	0.639	0.861	0.931	0.742	0.969	0.921

В таблице 2.6 представлены результаты Hits@10 для Dual-FusionKG на данных различных классов отношений в FB15K. Поскольку некоторые методы из таблицы 2.4 не участвовали в этом эксперименте, в качестве базовых алгоритмов для этой части эксперимента были выбраны TransE, ConvE и TorusE. Из таблицы 2.6 видно, что Dual-FusionKG превосходит другие методы во всех категориях отношений. Результаты по категории отношений в FB15K доказывают эффективность Dual-FusionKG в выявлении корреляции между сущностями и отношениями.

Проведен экспериментальный анализ производительности сущностей с различными уровнями разреженности. Было обнаружено, что, поскольку количество связей каждой сущности в графе знаний сильно варьируется, необходимо анализировать производительность сущностей с различными уровнями разреженности. В этом эксперименте сущности в FB15k-237 разделены на семь групп на основе общего количества связей в триплетах фактов: количество связей сущностей в разных группах составляет [0, 100], [100, 200], [200, 300], [300, 400], [400, 500], [500, 1000] и более 1000 соответственно.

Таблица 2.7 – Экспериментальные результаты работы алгоритма для объектов с разным уровнем разреженности на наборе данных FB15k-237

Диапазон чисел связей	SACN		Conve		Dual-FusionKG	
	Hits		Hits		Hits	
	@10	@3	@10	@3	@10	@3
[0,100]	0.195	0.134	0.192	0.125	0.532	0.401
[100,200]	0.441	0.253	0.441	0.245	0.484	0.323
[200,300]	0.705	0.429	0.696	0.446	0.401	0.220
[300,400]	0.806	0.577	0.829	0.558	0.370	0.225
[400,500]	0.868	0.663	0.894	0.661	0.707	0.411
[500,1000]	0.891	0.695	0.918	0.767	0.908	0.545
[1000,max]	0.981	0.922	0.992	0.941	0.919	0.452

Количество сущностей составляет 14272, 151, 43, 29, 24, 20 и 15 соответственно. Видно, что более 98,1% сущностей имеют менее 100 связей, в то время как только 0,1% сущностей имеют более 1000 связей, что также называется феноменом длинного хвоста, то есть большинство узлов являются разреженными, а лишь немногие – плотными.

В таблице 2.7 приведен анализ производительности для сущностей с различными уровнями разреженности и сравниваются метрики Hits@10 и Hits@3 предложенного алгоритма Dual-FusionKG с алгоритмами SACN и ConvE, которые представляют собой алгоритм графовой нейронной сети, ориентированный на структурную информацию, и алгоритм сверточной нейронной сети, ориентированный на семантическую информацию. Видно, что, во-первых, SACN работает лучше, чем ConvE, для сущности с небольшим количеством связей, поскольку его функция агрегации помогает сущности получить дополнительную структурную информацию из агрегации соседних сущностей. Однако агрегация структурной информации в SACN блокируется из-за несвязности FB15k-237, в то время как Dual-FusionKG с максимизированной взаимной информацией захватывает полную структурную информацию и достигает 53,2% попаданий при 10 совпадениях для сущностей с количеством связей [0, 100]. Во-вторых, для сущностей с

большим количеством связей (что означает наличие большего количества соседних сущностей) алгоритмы представления знаний на основе графовых нейронных сетей SACN и Dual-FusionKG работают хуже, чем ConvE, поскольку их представления встраивания существенно сглаживаются избыточной информацией о соседних сущностях внутри функции агрегации. Однако доля сущностей в этой части общего набора данных очень мала. Этот эксперимент демонстрирует эффективность использования полной структурной информации из Dual-FusionKG для максимизации взаимной информации в графе знаний, а также ограничения алгоритмов представления знаний на основе графовых нейронных сетей.

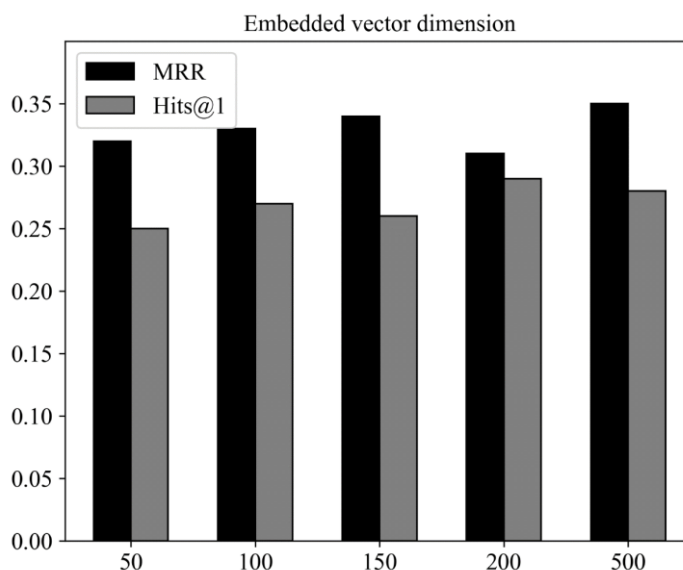


Рисунок 2.9 – Влияние различных размерностей векторов эмбедингов на производительность алгоритма (FB15k-237)

Далее обсуждаются три ключевых фактора, влияющих на производительность предлагаемого алгоритма Dual-FusionKG, а именно размерность векторов эмбедингов, декодер Dual-FusionKG и операция слияния семантической и структурной информации. Влияние этих факторов оценено на примере набора данных FB15k-237 (табл. 2.8-2.10).

Размерность вектора эмбединга очень важна для обучения представлений. Слишком малая размерность вектора приводит к ограниченной емкости представления для алгоритма, в то время как слишком большая размерность вектора приводит к слишком разреженным векторам представления, нерациональному использованию места для хранения и иногда даже к худшим результатам представления. В этом разделе для проверки эффективности алгоритма Dual-FusionKG используются пять различных размеров размерности вектора: 50, 100, 150, 200 и 500. Как показано на рисунке 2.9, в этом эксперименте, когда размерность вектора встраивания меньше 200, большая размерность вектора обеспечивает лучшую производительность; когда размерность вектора больше 200, размерность вектора встраивания отрицательно коррелирует с производительностью.

Как упоминалось выше, декодер по умолчанию в Dual-FusionKG – ConvE, а в данном эксперименте используются различные декодеры, которые применялись в RGCN и SACN, а именно TransE, dismult и простой многослойный перцептрон (MLP). Результаты экспериментов с различными декодерами на FB15k-237 приведены в таблице 2.9, и было установлено, что Dual-FusionKG+ConvE является наилучшей комбинацией кодировщика и декодера.

Как показано на рисунке 2.3, в Dual-FusionKG имеются два кодировщика, извлекающие семантическую и структурную информацию соответственно. Поэтому влияние объединения двух типов информации на производительность эксперимента очень важно. В этом эксперименте используются три различных операции объединения: сложение, умножение и взвешенное суммирование (веса являются динамически обучаемыми параметрами). Результаты различных операций слияния между кодировщиками для FB15k-237 приведены в таблице 2.9. Установлено, что наилучшее представление графа знаний получается путем взвешенного суммирования семантической и структурной информации.

Таблица 2.8 – Влияние типа декодера на производительность алгоритма (FB15k-237)

Dual-FusionKG	TransE	Distmult	MLP	ConvE
Hits@10	0.518	0.531	0.521	0.549
Hits@3	0.375	0.378	0.376	0.399
Hits@1	0.241	0.251	0.248	0.270
MR	214	192	201	131
MRR	0.336	0.348	0.345	0.362

Таблица 2.9 – Влияние используемых операций слияния на производительность алгоритма (FB15k-237)

Merge operation	Addition	Multiplication	Weighted sum
Hits@10	0.536	0.529	0.549
Hits@3	0.388	0.368	0.399
Hits@1	0.265	0.242	0.270
MR	146	186	131
MRR	0.354	0.338	0.362

В сети внимания графа слияния сущностей и связей (ERGAT) функция агрегирования использует взаимное отношение $\varphi(x, r) = x \otimes r$ для объединения информации между сущностями и связями в графе знаний, а также учитывает свойство асимметрии направления связи. Был проведен эксперимент для проверки влияния различных операций слияния, включая результаты без таковой операции, с операциями умножения и взаимной корреляции соответственно. Как показано в таблице 2.10, наилучшие результаты были достигнуты при слиянии сущностей и связей с взаимной корреляцией. Кроме того, экспериментальные результаты также показали, что слияние сущностей и связей на основе умножения хуже, чем без операции. Предварительный анализ показывает, что причиной плохих результатов является то, что метод умножения игнорирует свойство асимметрии направления связи.

Таблица 2.10 – Влияние различных операций слияния сущностей и связей на производительность алгоритма (FB15k-237)

Операции	Без операции	Умножение	Взаимная корреляция
Hits@10	0.353	0.534	0.549
Hits@3	0.385	0.380	0.399
Hits@1	0.260	0.252	0.270
MR	221	289	131
MRR	0.351	0.345	0.362

Как показано на рисунке 2.3, Dual-FusionKG имеет два алгоритма ERGAT, и был проведен дополнительный эксперимент для проверки времени обучения алгоритмов для этих двух отдельных алгоритмов ERGAT и Dual-FusionKG. Первый алгоритм ERGAT направлен на получение глобальной структурной информации путем максимизации взаимной информации в графе знаний, в то время как второй алгоритм ERGAT направлен на получение семантической информации из известных фактических троек. В таблице 2.11 проверяется время, затрачиваемое двумя алгоритмами ERGAT на четырех больших эталонных наборах данных, а также время обучения Dual-FusionKG от начала до конца до сходимости алгоритма. Единицей измерения времени обучения является количество секунд, используемых в каждом раунде обучения, а единицей измерения времени сходимости – часы.

В экспериментах алгоритму ERGAT, максимизирующему взаимную информацию в графе знаний, потребовалось 10000 раундов обучения для сходимости, и алгоритму ERGAT, максимизирующему взаимную информацию в графе знаний, также потребовалось 10000 раундов обучения для сходимости.

Таблица 2.11 – Время обучения алгоритма на различных наборах данных

	FB15k-237	FB15K	WN18RR	WN18
Число ребер	272115	483142	86835	141442
Число узлов	14541	14951	40943	40943
Структурная информация (с.)	0.031	0.061	0.029	0.041
Семантическая информация (с.)	128.12	311.36	93.61	151.78
Время сходимости алгоритма (ч.)	17.82	43.36	13.07	21.19

ERGAT, получающий семантическую информацию на основе известных троек фактов, требует 500 итераций обучения для обеспечения сходимости. Из этих экспериментальных результатов можно сделать вывод, что количество ребер (т.е. количество троек фактов) в графе знаний оказывает большое влияние на временную производительность.

2.6 Результаты главы 2

В данной главе предложен алгоритм Dual-FusionKG для решения проблемы несвязности в графе знаний, состоящий из двух кодировщиков и декодера. Кодировщики представляют собой две независимые сети внимания для графов слияния сущностей и отношений (ERGAT). Входом первого кодировщика является матрица смежности графа знаний, которая обучается полной структурной информации графа знаний путем максимизации оценки взаимной информации в графе. Входом второго кодировщика являются триплеты фактов в графе знаний, которые используются для обучения семантической информации в графе знаний с помощью контролируемого обучения; кроме того, ERGAT использует механизмы взаимного и реляционного внимания для самоадаптивного обновления представлений встраивания сущностей и отношений. В этой главе проведены всесторонние эксперименты на четырех эталонных наборах данных, и результаты показывают, что предложенный алгоритм Dual-FusionKG не только демонстрирует лучшие результаты на этих стандартных наборах данных, но и

имеет более быструю сходимость, а также лучшее представление разреженных сущностей по сравнению с существующими методами благодаря своей способности улавливать полную структурную информацию.

В будущих исследованиях будут рассмотрены такие методы, как дистилляция знаний, для снижения сложности алгоритма. Поскольку алгоритм Dual-FusionKG использует две сети внимания графа слияния сущностей и связей в качестве декодеров, пространственная сложность алгоритма все еще вдвое выше, чем у обычного алгоритма графовой нейронной сети, несмотря на то, что для достижения того же времени обучения можно использовать распределенное обучение. Статья опубликована [145].

Алгоритм Dual-FusionKG, предложенный в этой главе, эффективно решает задачу обучения представлениям для несвязанных графов знаний, но он предполагает, что сущности представляются только идентификаторами и отношениями, игнорируя богатые мультимодальные представления сущностей в реальном мире (такие как изображения, текст и числовые значения). Это ограничивает полноту векторов представления. Развитие предложенного метода с использованием геометрических представлений (box embeddings) было продолжено в нашей работе [146], где исследуются графовые нейронные сети с учетом геометрии пространства, но комплексное решение этой проблемы представлено в главе 3 настоящей диссертации, где модель расширена для работы с мультимодальными графами знаний, интегрируя визуальную, текстовую и числовую информацию во время обучения, сохраняя при этом возможность захвата сложных графовых структур.

3 ОБУЧЕНИЕ ПРЕДСТАВЛЕНИЮ МУЛЬТИМОДАЛЬНЫХ ГРАФОВ ЗНАНИЙ

3.1 Мультимодальность в графах знаний и обзор методов

Классические графы знаний – это представления знаний, характеризующие сущности реального мира и их ассоциации в виде семантических сетей, основная структура которых состоит из триады фактов $\{h,r,t\}$, где головная сущность (h) и хвостовая сущность (t) образуют направленную ассоциацию посредством определенного отношения (r). Построение таких графов обычно опирается на структурированные базы знаний, включая Wikidata [60], открытую базу знаний, поддерживаемую Фондом Викимедиа, Freebase [61], типичный ранний крупномасштабный граф знаний, и авторитетные источники данных, такие как Dbpedia [62], которая формируется путем автоматического извлечения структурированных данных из статей Википедии.

В области обучения представлению графов знаний разработано множество методов исследования, основанных на различных кодировщиках. В частности, существующие методы можно разделить на следующие основные категории: модели, основанные на трансляции, моделируют отношения между сущностями путем отображения сущностей в пространство представления и использования математических операций в этом пространстве; и модели, основанные на разложении матриц, моделируют тройные отношения графа знаний как задачу разложения на 3D-тензоры; модели, основанные на нейронных сетях, используют линейные или билинейные нейронные сети для кодирования вложений сущностей и отношений; Модели на основе сверточных нейронных сетей используют сверточные операции для извлечения глубоких семантических признаков тройственной структуры путем объединения векторных представлений

сущностей и отношений, в то время как модели на основе графовых нейронных сетей в полной мере используют структурированную природу графов знаний для динамического обновления неявных представлений сущностей и отношений посредством механизма передачи сообщений. Каждый из этих подходов имеет свои преимущества, предоставляя разнообразные технические пути для обучения представлению графов знаний.

Современные основные методы обучения представлению графов знаний имеют очевидные ограничения. Эти методы обычно фокусируются только на информации о триаде отношений в графе знаний и сводят сущности и отношения к дискретным идентификаторам меток для моделирования, игнорируя при этом богатые мультимодальные признаки, заложенные в сущностях в реальном мире. Такие мультимодальные свойства неэффективно используются в традиционных методах обучения представлений на основе идентификаторов, в результате чего понимание семантики сущностей моделью остается на поверхностном уровне, что затрудняет получение полного семантического представления знаний реального мира.

По сравнению с традиционными графами знаний, мультимодальные графы знаний значительно повышают богатство и точность описания сущностей за счет интеграции нескольких типов данных. Современные исследования в этой области сосредоточены на интеграции различной модальной информации в традиционную структуру графа знаний: в текстовой модальности An et al [147] и Yao et al инновационно добавляют описания на уровне предложений к сущностям и используют модель BERT [148] для глубокого семантического кодирования; в визуальной модальности Xie и др. [149] первыми ввели информацию об изображениях в графы знаний, а Liu и др. [150] систематически дополнили визуальные признаки и числовые атрибуты для сущностей в графе знаний FB15k, построив первый полный мультимодальный граф знаний (ММКГ). На этих основах предложенная в

этой главе сеть внимания на основе реляционных графов гиперточек реализует глубокое слияние трех типов модальной информации, а именно текстовой (на уровне предложений), визуальной и числовой, и предоставляет более точный метод моделирования для обучения представлению многомодальных графов знаний.

В последние годы в области обучения представлению мультимодальных графов знаний появился ряд инновационных подходов. KBLRN [151], как новаторская работа в этой области, впервые предложил структуру обучения многомодальному представлению, которая интегрирует потенциальные реляционные признаки с числовыми признаками. Впоследствии МКВЕ [152] инновационно разработал многонейронную архитектуру кодировщика для достижения эффективного обучения многомодальным векторным представлениям сущностей путем оснащения различной модальной информации специальными кодировщиками и органичного объединения их с традиционными реляционными моделями. В медицинской области МКНАН [153] разработал иерархическую сеть внимания, которая не только кодирует мультимодальные медицинские знания, но и поддерживает интерпретируемые задачи медицинских вопросов и ответов, а MMRFAN [154] дополнительно предложил модель представления медицинских знаний на основе графических нейронных сетей, которая в полной мере использует топологическую информацию графов знаний. Эти методы ориентированы на различные сценарии применения, каждый со своими особенностями в механизме слияния модальностей и проектировании архитектуры сети, что в совокупности способствует развитию обучения представлению мультимодальных графов знаний.

Методы обучения мультимодальных графов знаний

В последние годы в области мультимодальных графов знаний появился ряд важных результатов исследований: Ан и др. впервые ввели текстовые описания на уровне предложений в представления графов знаний; в методе

KG-BERT но-новому была использована модель BERT для кодирования троичных кортежей сериализованным образом; в IKLR впервые получили кросс-модальное представление знаний на основе визуальных вложений; в методе KBLRN создали новый набор данных, содержащий числовые признаки; в МКВЕ разработали многокодировочную архитектуру для объединения многомодальных данных; в МКНАН разработали иерархическую сеть внимания для медицинских вопросов и ответов; в MMRFAN предложили обучение признакам с помощью состязательного подхода для достижения кросс-модального выравнивания; Сун и др. [155] применили многомодальное отображение для оптимизации рекомендательных систем; Чан и др. [156] создали структуру обнаружения фейковых новостей на основе семантики знаний; а Чаудхари и др. [157] исследовали применение графов знаний в поиске изображений.

Однако у существующих подходов есть очевидные ограничения: большинство исследований сосредоточены только на обработке мультимодальной информации, игнорируя присущие графам знаний свойства графовой структуры. Хотя MMRFAN пытается внедрить графовые нейронные сети, его несквозная конструкция требует наличия нескольких независимых сетей для обработки различных модальностей, что приводит к высокой сложности модели и неэффективности. Для решения этих проблем алгоритм HyperFusion-Net, предложенный в этой главе, может одновременно захватывать мультимодальные признаки и информацию о структуре графа с помощью одной графовой нейронной сети, реализуя сквозное обучение представлению мультимодального графа знаний в истинном смысле этого слова. Методологическая разработка алгоритма подробно описана в следующем разделе.

3.2 Алгоритм Hyperfusion-Net: общая структура и модуль слияния информации

Hyperfusion-Net реализует обучение с использованием многомодального представления графа знаний посредством трех основных модулей: модуля слияния информации, модуля агрегирования информации и модуля проверки прогнозов. Ниже описаны конкретные функции каждого модуля (рисунок 3.1).

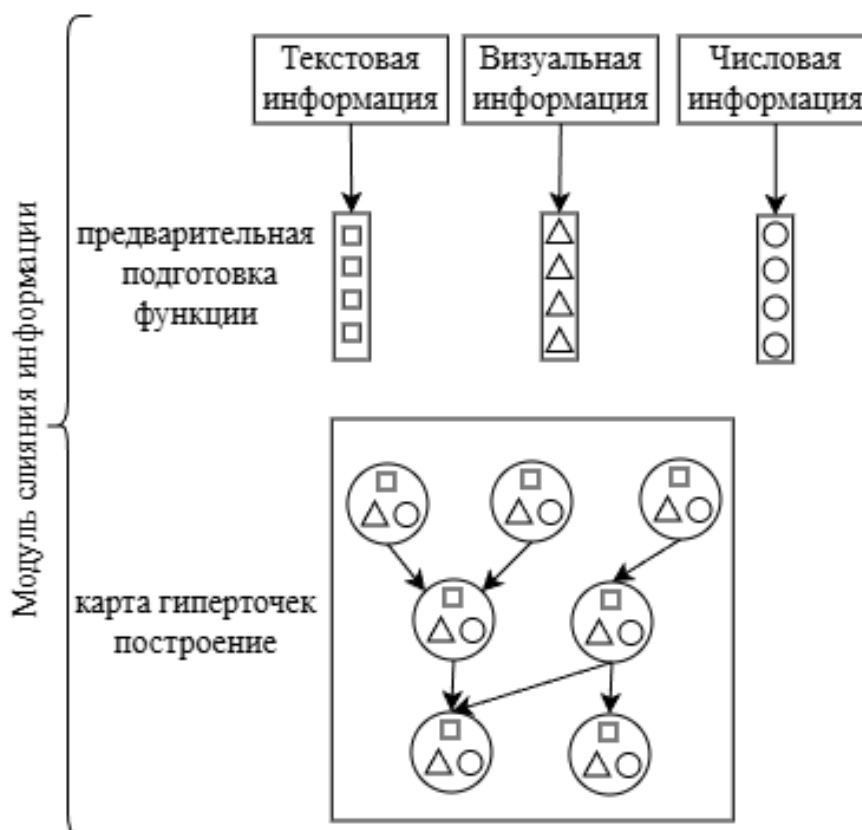


Рисунок 3.1 – Модуль слияния информации

Модуль слияния информации: этот модуль сначала получает предварительно обученное встраивание каждой модальной информации с помощью предварительно обученной модели для генерации исходного унимодального представления. Затем модуль слияния информации объединяет каждое унимодальное представление с помощью операции слияния тензоров низкого ранга для захвата внутри- и межмодальных взаимодействий признаков. Все вычислительные операции в слиянии

тензоров низкого ранга выполняются на микроуровне (отдельных узлов), что гарантирует возможность обучения нейронной сети от начала до конца. Для решения проблемы дисбаланса мультимодальной информации сущностей модуль слияния информации вводит фактор декомпозиции в слияние тензоров низкого ранга, что позволяет ему гибко обрабатывать различные объемы модальной информации, эффективно решая таким образом проблему отсутствия или дисбаланса мультимодальной информации.

Модуль агрегирования информации: в отличие от традиционных графовых нейронных сетей, этот модуль использует в качестве базовой модели сеть внимания на основе графа слияния сущностей и связей, которая способна более эффективно извлекать информацию о структуре графа в мультимодальном графе знаний.

Модуль проверки прогнозов: В модуле проверки прогнозов Hyperfusion-Net использует ConvE в качестве декодера для вычисления достоверности триплетов фактов на основе векторных представлений сущностей и отношений многомодального графа знаний, чтобы выполнить задачу обучения представлению графа знаний.

Модуль слияния информации

Основная задача обучения представлению мультимодального графа знаний заключается в том, как эффективно захватывать дополнительную модальную информацию, учитывая при этом информацию о структуре графа знаний. Предложенный в данной статье подход работает в тандеме с помощью модуля слияния информации и модуля агрегирования информации: модуль слияния информации сначала использует предварительно обученную модель для получения предварительно обученных встроенных представлений каждой модальности и преобразует традиционный граф знаний в граф гиперточек посредством слияния тензоров низкого ранга; впоследствии модуль агрегирования информации дополнительно захватывает информацию о структуре графа в графе гиперточек.

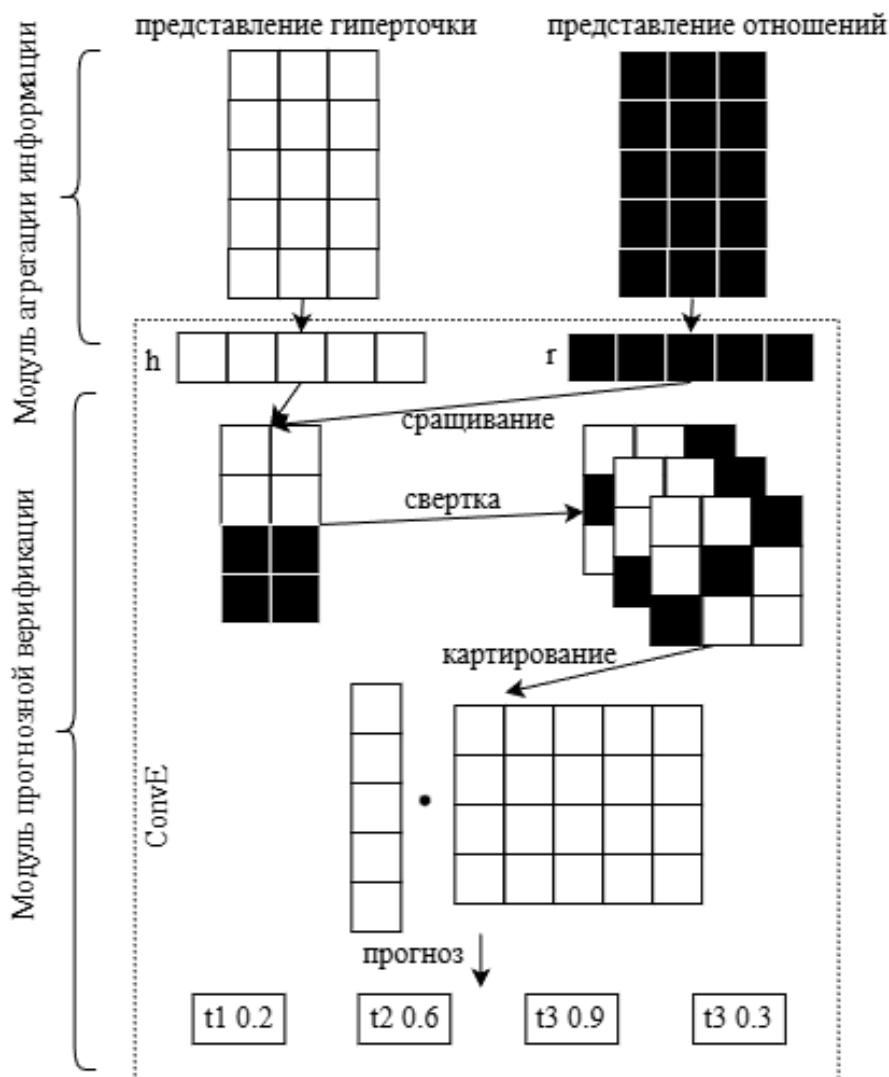


Рисунок 3.2 – Модуль агрегирования информации и проверки прогнозов

Предварительно обученные признаки

Во-первых, каждая модальность извлекает свой предварительно обученный вектор признаков с помощью соответствующего предварительно обученного кодировщика, специфичного для данной модальности:

Текстовая информация: в области обучения представлению графов знаний сущности и связи (отношения) обычно сводятся к дискретным идентификаторам меток для моделирования, и это представление является вычислительно эффективным, но теряет богатую семантическую описательную информацию сущностей. Для преодоления этого ограничения в настоящем исследовании используется усовершенствованная предварительно

обученная языковая модель для захвата тонких текстовых характеристик сущностей. В частности, модуль слияния информации выбирает Sentence-BERT [158] в качестве кодировщика текста, основанного на архитектуре Transformer, и глубоко кодирует описания сущностей на уровне предложений с помощью 12-слойной сети (содержащей 768-мерные скрытые состояния и 12 заголовков внимания), и в конечном итоге выдает 768-мерные семантические векторы et. Эта предварительно обученная модель, основанная на контекстной осведомленности, не только захватывает сложные взаимодействия между словами, но и поддерживает семантическую целостность на уровне предложений, обеспечивая более богатые текстовые представления для мультимодальных графов знаний. Общее количество параметров модели составляет 110 МБ, что обеспечивает хороший баланс между возможностями семантического представления и вычислительной эффективностью.

Визуальная информация: Сущности в графе знаний обычно представляют собой конкретные объекты в реальном мире, поэтому их можно описать с помощью какого-либо изображения. Предварительно обученный кодировщик визуальной информации декодирует визуальную информацию с помощью VGG-16 [159], которая представляет собой модель сверточной нейронной сети, состоящую из модуля признаков и модуля классификации. Модуль признаков представляет собой стек слоев сверточной нейронной сети, состоящий из сверточных ядер 3×3 с общим количеством 13 сверточных слоев. Модуль классификации представляет собой стек слоев сверточной нейронной сети размером 3×3 с общим количеством 13 сверточных слоев. Модуль классификации представляет собой стек из 3 полносвязных слоев, и кодировщик визуальной информации удаляет последний полносвязный слой и слой softmax для получения предварительно обученного визуального представления. Входными данными для VGG-16 является изображение формаиа RGB фиксированного размера 224×224 , а результирующее

встраивание визуальной информации объекта представляется в виде 4096-мерного вектора ev .

Числовая информация: с точки зрения обработки числовой информации для графов знаний, данное исследование фокусируется на совместном представлении атрибутов объекта и соответствующих числовых характеристик. Как показано на рисунке 3.1, объекты обычно содержат важные числовые атрибуты (например, «рост Кайли Миноуг составляет 1,52 метра», «дата рождения – 1968, апрель» и т. д.), и эта информация существует в виде пар (ключ, значение). Для обработки 116 различных атрибутов в наборе данных FB15k мы разработали двухканальную архитектуру кодирования: сначала ключ атрибута кодируется как 768-мерный семантический вектор с помощью модели BERT, а одновременно числовое значение отображается во вложенное представление той же размерности с использованием слоя полной связности (FC); затем два 768-мерных вектора объединяются, и затем используется еще один 768-мерный слой полной связности для слияния признаков. В результате слияния признаков генерируются 768-мерные векторы эмбедингов, содержащие полную числовую семантическую информацию. Этот подход преодолевает ограничения традиционного обучения представлению графов знаний, которое игнорирует числовые признаки, и реализует органическое единство семантики атрибутов и числовых признаков с помощью глубоких нейронных сетей, которые обеспечивают структурированные числовые представления для последующего мультимодального слияния.

При разработке модуля слияния информации мы используем предварительно обученные модели VGG-16 и BERT в качестве базовых экстракторов признаков. Для обеспечения совместимости мультимодальных признаков модуль сначала равномерно проецирует векторы встраивания различных модальностей (визуальной, текстовой и числовой) в одномерное пространство признаков. В частности, 4096-мерные признаки VGG для

визуальной модальности, 768-мерные признаки BERT для текстовой модальности и 768-мерные признаки для числовой модальности будут нормализованы к представлениям встраивания одинаковой размерности с помощью слоя линейного преобразования. Затем эти выровненные мультимодальные признаки глубоко взаимодействуют и интегрируются с помощью инновационной техники слияния тензоров низкого ранга, и исходный граф знаний в конечном итоге преобразуется в представление графа гиперточек, содержащее богатую семантическую информацию. Эта обработка не только сохраняет уникальные признаки каждой модальности, но и обеспечивает кросс-модальное выравнивание и слияние признаков, закладывая прочную основу для последующего обучения структуры графа. Все предварительно обученные модели на этом этапе сохраняют свои параметры неизменными, и оптимизируются только параметры части проекции и слияния признаков.

Слияние тензоров низкого ранга

Слияние мультимодальной информации позволяет сущностям использовать дополнительную информацию, существующую в разных модальностях в мультимодальном графе знаний. Ранние подходы к обучению представлений мультимодального графа знаний использовали механизмы слияния или внимания для реализации слияния мультимодальной информации с целью захвата мультимодальной информации, однако эти подходы обычно игнорировали динамическое взаимодействие признаков внутри и между модальностями. Слияние тензоров [160] – это успешный метод слияния мультимодальной информации, который преобразует входные векторы представления каждой модальности в высокоразмерные тензоры, а затем отображает их в низкоразмерные векторы представления с помощью слияния тензоров. Мультимодальное представление e , сформированное с помощью слияния тензоров, вычисляется следующим образом:

$$e = g(Z; W, b) = W \cdot Z + b, (e, b \in \mathcal{R}^{d_e}) \quad . \quad (3.1)$$

Здесь W – матрица весов, b – вектор смещения. $Z = \otimes_{m=1}^M e_m$ – многомерный тензор, образованный путем выполнения внешнего тензорного произведения \otimes над набором векторов унимодального представления $e_m = (e^t, 1), (e^v, 1), (e^n, 1)$ с добавлением 1, m обозначает m -ю Модальность. Z можно определить как

$$\begin{aligned} Z &= \begin{bmatrix} e^t \\ 1 \end{bmatrix} \otimes \begin{bmatrix} e^v \\ 1 \end{bmatrix} \otimes \begin{bmatrix} e^n \\ 1 \end{bmatrix} \\ &= (e^t, e^v, e^n) + (e^t \otimes e^v, e^t \otimes e^n, e^v \otimes e^n) + e^t \otimes e^v \otimes e^n. \end{aligned} \quad (3.2)$$

Первые три элемента представляют собой унимодальные представления встраивания, которые отражают признаки в рамках одной модальной информационной модальности. Затем последние три элемента $e^t \otimes e^v, e^t \otimes e^n, e^v \otimes e^n$ представляют собой бимодальные межмодальные взаимодействия признаков, а последний подтерм $e^t \otimes e^v \otimes e^n$ представляет собой тримодальные межмодальные взаимодействия признаков, которые могут отражать межмодальные взаимодействия признаков при тензорном слиянии мультимодальной информации.

Тем не менее, этот метод требует предварительного явного создания многомерного тензора, содержащего мультимодальную информацию, причём размерность этого многомерного тензора будет экспоненциально расти с увеличением числа модальностей, а его пространственная сложность также будет расти экспоненциально. Используя идею низкорангового разложения векторных пространств, слияние тензоров можно заменить слиянием низкоранговых тензоров. Во-первых, слияние низкоранговых тензоров использует низкоранговое разложение весов вместо весовой матрицы W :

$$W = \sum_{i=1}^k \otimes_{m=1}^M w_m^{(i)}. \quad (3.3)$$

Минимальное значение k , удовлетворяющее вышеуказанному разложению матрицы, называется рангом тензора. Множество векторов

называется коэффициентом разложения исходной весовой матрицы, где m представляет m -ю модальность данных, а i – номер коэффициента разложения. Следовательно, уравнение (3.1) можно преобразовать в следующую форму (вектор смещения b здесь опущен):

$$\begin{aligned} e &= \left(\sum_{i=1}^k \otimes_{m=1}^M w_m^{(i)} \right) \cdot Z = \sum_{i=1}^k \left(\otimes_{m=1}^M w_m^{(i)} \cdot Z \right) \\ &= \sum_{i=1}^k \left(\otimes_{m=1}^M W_m^{(i)} \cdot \otimes_{m=1}^M e_m \right) = \Lambda_{m=1}^M \left[\sum_{i=1}^k w_m^{(i)} \cdot e_m \right]. \end{aligned} \quad (3.4)$$

Здесь $\Lambda_{m=1}^M$ определяется как $\Lambda_{m=1}^3 e_t \circ e_v \circ e_n$, а \circ – произведение Адамара. Стоит отметить, что e_m – унимодальный вектор представления с дополнительной 1, то есть $e_m = (e^m, 1)$. Низкоранговое тензорное представление e трехмодальной информации может быть выражено следующим образом:

$$e = \left(\sum_{i=1}^k w_t^{(i)} \cdot e_t \right) \circ \left(\sum_{i=1}^k w_v^{(i)} \cdot e_v \right) \circ \left(\sum_{i=1}^k w_n^{(i)} \cdot e_n \right). \quad (3.5)$$

Используя (3.5), можно предварительно обучить представление эмбединга непосредственно из индивидуального унимодального предварительного обучения входных данных, используя коэффициент разложения конкретной модальности для вычисления ее низкорангового тензорного представления e , избегая необходимости вычисления произведения большого входного модального тензора Z и его весовой матрицы W . В то же время, можно комбинировать различное количество модальностей, расширяя уравнение (3.5) (две модальности могут быть представлены двумя произведениями в уравнении (3.5)). Наконец, и что наиболее важно, все операции низкорангового тензорного представления являются полностью дифференцируемыми, что позволяет обучать параметр от начала до конца с помощью обратного распространения ошибки.

3.3 Модуль агрегирования и проверки прогнозов в HyperFusion-Net

В модуле агрегирования информации структурная информация графа собирается путем агрегирования информации о гиперточках соседей; сначала для каждой гиперточки измеряется важность ее соседей с использованием механизма внимания, специфичного для отношений. Затем для каждой пары «сущность-связь» информация о соседе и информация о связи объединяются с помощью операции слияния «сущность-связь». Наконец, информация о соседях агрегируется с использованием функции агрегации на основе внимания, специфичной для связей, и обновляется представление суперузла. Кроме того, модуль агрегации информации обучает эмбединги связей (реляционные эмбединги) с помощью независимой матрицы реляционных параметров. Эта матрица обеспечивает адаптивное обучение реляционных представлений посредством обратного распространения ошибки во время обучения, что дополнительно повышает способность модели моделировать сложные структуры графов. Такая конструкция позволяет модулю агрегации информации эффективно собирать информацию о структуре графа в многомодальных графах знаний, одновременно улучшая представление сущностей и отношений. Для измерения важности соседа для целевой гиперточки используется показатель внимания, специфичный для отношений, в адаптивной сети внимания графа, который вычисляется следующим образом:

$$a_{ij} = \text{soft max}(b_{ij}) = \frac{\exp(b_{ij})}{\sum_{n \in N_i^r} \exp(b_{in})} , \quad (3.6)$$

$$b_{ij} = W_i e_i + W_r r_{ij} + W_j e_j . \quad (3.7)$$

Здесь N_i^r – сосед i -й гиперточки, связанной отношением r , e_i и e_j – скрытые представления гиперточек в виде вложений, а r_{ij} – скрытое реляционное представление в виде вложений.

В многомодальных графах знаний сущности и связи тесно коррелированы, и для решения этой проблемы операция слияния сущностей и связей $\varphi(e,r)$ в адаптивных сетях внимания к графам может объединять информацию между гиперузлами и связанными с ними отношениями (связями). Операция $\varphi(e,r)$ вычисляется следующим образом:

$$\varphi(e,r) = e * r. \quad (3.8)$$

Здесь $*$ обозначает вычисление свертки.

Вдохновленные графовыми нейронными сетями, основанными на функциях агрегирования, обучение представлений гиперточек и представлений отношений может быть достигнуто путем агрегирования соседей. То есть каждое представление гиперточки более высокого порядка обновляется путем агрегирования представлений соседних гиперточек более низкого порядка. Процесс агрегирования реализуется следующим образом:

$$e_i^{(l+1)} = \sigma\left(\sum_{j \in N_i^r} \frac{1}{c_{i,r}} a_{ij} W_{i,j}^{(l)} \varphi(e_i^{(l)}, r_{i,j}^1)\right). \quad (3.9)$$

Здесь $e_i^{(l)}$ обозначает представление встраивания i -й гиперточки в l -м слое адаптивной сети внимания к отношениям, $c_{i,r} = |N_i^r|$ – коэффициент нормализации, N_i^r – сосед i -й гиперточки со связью r , а $W_{i,j}^{(l)}$ – матрица весов гиперточки e_i и ребра $r_{i,j}$ с соседней гиперточкой e_j . Также определим матрицу W_r для обновления представления связи, при этом i -я связь будет обновлено следующим образом:

$$r_i^{(l+1)} = W_r^{(l)} r_i^{(l)}. \quad (3.10)$$

Модуль проверки прогнозов

Для реализации многомодального обучения представлению графа знаний, следуя традиционной модели обучения представлению графа знаний, представления встраивания сущностей и встраивания отношений вычисляются на основе кодировщика графовой нейронной сети, а оценки

троек фактов с полученными эмбедингами вычисляются декодером. Функция потерь модели реализуется путем вычисления разницы между прогнозируемой оценкой и истинной оценкой. В данной работе в качестве декодера выбран ConvE, а функция оценки для каждой тройки фактов имеет следующий вид:

$$f(h, r, t) = \sigma(\text{vec}([e_h; r] * W) e_t). \quad (3.11)$$

Здесь e_h и e_t представляют собой встраивания головной и хвостовой сущностей соответственно, а r – эмбединг связи между ними, $*$ и ω представляют собой операцию свертки и ядро свертки соответственно, а vec – операцию преобразования карты признаков в векторы, W – матрица отображения, а σ – сигмоидная функция активации. С помощью описанного выше подхода модель может вычислить, является ли истинным или ложным утверждение, составленное между сущностями на основе некоторой связи. В данной работе в качестве функции потерь выбрана бинарная кросс-энтропия:

$$\mathcal{L}(f, t) = -\frac{1}{|F|} \sum_i (t_i \cdot \log(f_i) + (1 - t_i) \cdot \log(1 - f_i)). \quad (3.12)$$

Здесь $|F|$ – общее количество обученных троек фактов, t_i – метка триплета фактов, а f_i – оценка предсказания i -го триплета фактов на основе многомодальной сети внимания графа знаний. $T=1$, когда триплет фактов истинен, и 0 , когда он ложен.

Псевдокод алгоритма HyperFusion-Net можно представить следующим образом:

Алгоритм 3-1: Процесс обучения HyperFusion-Net:

Дано: Мультимодальный граф знаний $G=(V, R, T)$, где V – множество сущностей, R – множество связей (отношений), T – множество триплетов, Мультимодальные признаки сущностей: X_t (текстовые), X_v (визуальные), X_n (числовые),

Гиперпараметры: размерность эмбедингов d , количество факторов низкого ранга k , количество слоев GNN L , размер пакета B .

Вывод алгоритма: матрица эмбедингов сущностей $H \in \mathbb{R}^{|V| \times d}$, матрица эмбедингов связей $R_{\text{emb}} \in \mathbb{R}^{|R| \times d}$.

Инициализация:

Загрузить предобученный текстовые кодировщики `Encoder_BERT`, `Encoder_VGG`, `Encoder_BERT_FC`.

Инициализировать параметры тензорного слияния низкого ранга W_t .

Инициализация параметров внимания графа: W_a (матрица внимания), W_r (веса связей), ω (ядро свертки)..

Инициализация параметров декодера: Ω (ядро ConvE), W_{mlp} (матрица отображения).

Инициализировать алгоритм оптимизации Adam, установить скорость обучения η .

Основной цикл обучения:

Для $\text{epoch} = 1$ до MaxEpoch :

1. //Модуль слияния информации (построение суперточек).

Для каждого $v_i \in V$:

$z_t \leftarrow \text{Encoder_BERT}(X_t)$, //Вектор текстовых признаков.

$z_v \leftarrow \text{Encoder_VGG}(X_v)$, //Вектор визуальных признаков.

$z_n \leftarrow \text{Encoder_BERT_FC}(X_n)$, //Вектор числовых признаков.

`LowRankFusion`($z_t, z_v, z_n, W_t, W_v, W_n$) // генерирует представление суперточек.

2. //Модуль агрегирования информации (ERGAT).

Для $l = 1$ до L :

For каждого $v_i \in V$:

Множество соседей $N_i \leftarrow \{(v_j, r) \mid (v_i, r, v_j) \in T\}$ //

Для каждого $(v_j, r) \in N_i$:

Вычислить коэффициенты внимания, специфичные для

связей. a_{ij} .

Операция слияния сущность-связь φ_{ij} .

//Сводная информация о соседях и ее обновление.

$h_{vi}^{(l)} \leftarrow \text{Aggregate}(N_i, \{a_{ij}\}, \{\varphi_{ij}\}, W_r)$

$r_{emb_r}^{(l)} \leftarrow \text{UpdateRelation}(r_{emb_r}^{(l-1)}, W_r)$

Получить окончательный эмбединг сущности $H \leftarrow \{h_{v_i}^{(L)}\}$.

Получить окончательный эмбединг связи $R_emb \leftarrow \{r_{emb_r}^{(L)}\}$.

3.// Модуль прогнозирования и проверки (ConvE)

Для batch $B \subseteq T$:

Множество положительных образцов $T_{pos} \leftarrow B$

Множество отрицательных образцов set $T_{neg} \leftarrow$

NegativeSample(B)

Для each $(h, r, t) \in T_{pos} \cup T_{neg}$:

$score_{\{h,r,t\}} \leftarrow \text{ConvEDecoder}(H[h], R_{emb[r]}, H[t], \Omega, W_{mlp})$

// Рассчитать потери бинарной кросс-энтропии.:

$mathcal{L} \leftarrow \text{BinaryCrossEntropy}(\{score_{\{h,r,t\}}\}, T_{pos}, T_{neg})$

// Обратное распространение ошибки и обновление параметров

$mathcal{L}.backward()$

UpdateParameters(Adam, η)

//Проверка достоверности и досрочная остановка

$MRR_{val} \leftarrow \text{EvaluateValidationSet}(H, R_{emb})$

Если EarlyStop(MRR_{val} , patience= p):

Прервать цикл

Возвратить H, R_{emb}

3.4 Экспериментальная оценка и анализ гиперпараметров

Эксперименты в данной главе сосредоточены на проверке эффективности алгоритма HyperFusion-Net по четырем ключевым аспектам:

во-первых, комплексная эффективность алгоритма в задаче дополнения графа знаний по сравнению с другими методами обучения представлению графа знаний; во-вторых, исследуется степень влияния различной модальной информации в мультимодальном графе знаний на обучение представлению; затем, вклад основных компонентов, таких как слияние мультимодальной информации, слияние низкоранговых тензоров, механизм реляционного внимания и операция слияния сущностей и связей, как основных компонентов, влияющих на эффективность модели; наконец, исследуется влияние гиперпараметров, таких как размер размерности встроеного представления и количество низкоранговых факторов, на эффективность модели, что позволяет всесторонне оценить технологические преимущества алгоритма и его чувствительность к параметрам.

Предложенный алгоритм HyperFusion-Net оценивается на четырех эталонных наборах данных, а именно FB15k-237, WN18RR, DB15K и YAGO15K. В набор данных для этого эксперимента собрана текстовая информация об сущностях FB15k-237, WN18RR, DB15K и YAGO15K из KG-BERT. В то же время визуальная и числовая информация о сущностях FB15k-237, DB15K и YAGO15K собрана из MMKG для изучения мультимодальной информации графа знаний. В таблице 3.1 приведены сводные статистические данные по набору данных. Количество триплетов фактов – общее количество триплетов фактов в графе знаний, который в этом эксперименте разделен на обучающую, тестовую и валидационную выборки. Соотношение объемов выборок в методе RGCN применяется к наборам данных FB15k-237 и WN18RR, а соотношение в DB15K и YAGO15K установлено таким же, как и для FB15k-237. Количество текстовых информационных сущностей – количество сущностей с текстовыми описаниями на уровне предложений. Количество визуальных информационных сущностей – количество сущностей с графическими описаниями. Количество числовой информации – это количество элементов

числовой информации, содержащейся в триплете фактов, а количество атрибутов – это количество типов атрибутов, соответствующих числовой информации, например, широта и дата.

Таблица 3.1 – Статистическая информация о наборах данных для экспериментов

Набор данных	FB15k-237	WN18RR	DB15K	YAGO15K
Сущности	14541	40943	12842	15404
Связи	237	11	279	32
Триплеты фактов	310116	93012	89197	122886
Сущности текстовой информации	14541	40943	12842	11199
Сущности визуальной информации	13444	-	12837	11194
Элементы числовой информации	29395	-	48080	23532
Атрибуты	116	-	225	7

FB15k-237 [1] – набор данных из проекта Freebase, содержит 14541 сущность и 237 связей. В FB15k-237 удалены инвертированные связи из обучающего набора FB15k. Что касается мультимодальной информации, каждая сущность в FB15k-237 имеет текстовое описание в виде предложения, 13444 сущности снабжены визуальной информацией, а 29395 (т.е. большинство сущностей) – числовую информацию, охватывающую большинство сущностей.

WN18RR [1] – набор данных английского словаря, содержит 4094 сущности и 11 связей. WN18RR получен путем удаления инвертированных связей из обучающего набора WN18. Поскольку WN18RR – словарь, он содержит только текстовую информацию (предложения).

DB15K [62] является производной от DBPedia, которая построена путем выравнивания сущностей с FB15K. Цель DBPedia – извлечение структурированной информации и контента из Википедии.

YAGO15K [161] – подмножество YAGO (Yet Another Great Ontology),

которое также построено путем выравнивания сущностей с FB15K. База знаний YAGO уже была представлена в главе 1, посвященной текущему состоянию исследований.

В эксперименте сравнивается алгоритм HyperFusion-Net с девятью базовыми алгоритмами, основанными на различных кодировщиках, и ниже приведено краткое описание каждого алгоритма.

TransE [1] – это классическая модель обучения представлению графа знаний, основанная на модели трансляции, которая кодирует сущности и отношения в линейное пространство.

RotatE [4] – это модель обучения представлению графа знаний, основанная на модели трансляции, которая отображает векторы представления в комплексное пространство и использует операции вращения в комплексном пространстве для описания реляционной информации.

JointE [162] – совместно использует одномерные и двумерные сверточные операции для эффективной реализации обучения представлению графов знаний.

ConvE [8] – модель сверточной нейронной сети, которая использует свертку для захвата корреляции между сущностями и отношениями.

КМАЕ [163] – модель сверточной нейронной сети, которая расширяет сверточные ядра в атрибутах сущностей и отношений соответственно для выполнения обучения представлению графа знаний.

RGCN [10] – нейронная сеть реляционных графов, которая применяет свертку реляционных графов для обучения представлению многореляционных данных, таких как графы знаний.

WGNCN [46] – реляционная графовая нейронная сеть, которая присваивает разные веса разным типам отношений во время агрегации, и веса адаптивно обновляются во время обучения сети.

KBLRN [151] – исторически первый подход к моделированию, работающий с многомодальными графами знаний, и осуществляет обучение

представлениям многомодальных графов знаний, содержащих реляционную информацию и числовые признаки.

MMRFAN [154] – метод обучения представлению многомодальных графов знаний на основе графовой нейронной сети, который изучает многомодальную информацию об сущностях с помощью модуля обучения признакам с помощью состязательного подхода, а затем завершает встраивание представления многомодальных графов знаний с помощью операции свертки реляционных графов.

Следует отметить, что хотя существуют и другие методы обучения многомодальному представлению графов знаний, такие как IKLR[149] и МКНАН[153], они являются методами моделирования данных медицинской области, а эталонные наборы данных, используемые в этой работе, взяты из общедоступных графов знаний, поэтому две вышеупомянутые модели не рассматриваются в этом эксперименте.

Предварительные признаки для обучения задаются вручную со следующими гиперпараметрами: начальная размерность текстового эмбединга составляет 768, начальная размерность визуального эмбединга – 4068, а начальная размерность числового встраивания – 768. Гиперпараметры модели следующие (где выделены жирным шрифтом оптимальные гиперпараметры): размерность графа знаний составляет {50, 100, 150, **200**, 500}, а коэффициент отсечения кодировщика – {0,0, **0,1**, 0,2, 0,3}. Гиперпараметры декодера ConvE установлены такими же, как в WGCN. Алгоритм оптимизации модели – ADAM, а скорость обучения установлена равной {1e-2, **1e-3**, 1e-4, 1e-5}. При этом метрики оценки в этом эксперименте такие же, как и в экспериментах с многореляционными графовыми нейронными сетями на основе максимальной взаимной информации на графах в главе 2.

Результаты экспериментов и их анализ

В этом разделе алгоритм HyperFusion-Net, предложенный в этой главе, сравнивается с различными базовыми алгоритмами на эталонных наборах данных FB15k-237, WN18RR, DB15K и YAGO15K, главным образом для проверки того, насколько хорошо алгоритм HyperFusion-Net справляется с задачей дополнения графа знаний по сравнению с другими методами обучения представлению графа знаний. В частности, FB15k-237, DB15K и YAGO15K содержат информацию всех трех модальностей, в то время как WN18RR содержит только текстовую информацию.

Таблицы 3.2 и 3.3 показывают результаты задачи дополнения графа знаний, при этом лучшие результаты в каждом столбце метрик выделены жирным шрифтом. На популярных наборах данных FB15k-237 и WN18RR предложенный в этой главе алгоритм HyperFusion-Net демонстрирует наилучшие результаты по большинству оцениваемых метрик по сравнению с другими базовыми алгоритмами. Как видно из таблицы 3.2, алгоритм HyperFusion-Net значительно улучшает производительность дополнения графа знаний. Из результатов следует, что алгоритм HyperFusion-Net превосходит все традиционные методы обучения представлению графа знаний.

Это подтверждает значимость обучения представлению многомодального графа знаний. Кроме того, сравнение с KBLRN и MMRFAN, классом методов обучения представлению многомодального графа знаний, показывает, что алгоритм HyperFusion-Net использует предложенный модуль слияния информации и сеть внимания графа слияния сущностей и связей для достижения значительного улучшения.

Таблица 3.2 – Результаты экспериментов по дополнению графов знаний (FB15k-237 и WN18RR)

Алгоритм	FB15k-237					WN18RR				
	Hits					Hits				
	MR	MRR	@1	@3	@10	MR	MRR	@1	@3	@10
Transe	357	0.257	0.174	0.284	0.420	-	0.182	0.027	0.295	0.444
Rotate	177	0.338	0.241	0.375	0.533	3340	0.476	0.428	0.492	0.571
Jointe	177	0.356	0.262	0.393	0.543	4655	0.537	0.438	0.483	0.537
Conve	244	0.325	0.237	0.356	0.501	4187	0.430	0.400	0.440	0.520
KMAE	235	0.326	0.240	0.358	0.502	4441	0.448	0.415	0.465	0.524
R-GCN	-	0.25	0.15	0.23	0.42	-	-	-	-	-
WGCN	-	0.350	0.260	0.390	0.540	-	0.470	0.430	0.480	0.540
KBLRN	209	0.309	0.219	0.342	0.493	3871	0.471	0.435	0.479	0.538
MMRFAN	256	0.297	0.205	0.326	0.485	-	-	-	-	-
Hyperfusion-Net	156	0.366	0.271	0.404	0.542	2685	0.491	0.454	0.503	0.567

Предыдущие исследования не сообщали о результатах производительности алгоритмов дополнения графов знаний на наборах данных на наборах данных DB15K и YAGO15K. Поэтому были реализованы некоторые классические алгоритмы для проверки эффективности предлагаемого метода, с использованием DGL-KE [151], а также классический метод для дополнения мультимодального графа знаний KBLRN в качестве базовых моделей. DGL-KE – программный пакет для обучения представлению графа знаний, который включает в себя широкий спектр популярных моделей, таких как rotate, TransE и TransR. В этом разделе показано, что алгоритм HyperFusion-Net, предложенный в этой главе, превосходит все базовые алгоритмы в таблице 3.3. В частности, KBLRN достигает конкурентоспособной производительности, используя только механизм внимания, по сравнению с традиционными методами обучения представлением графа знаний. KBLRN хуже только метода на основе графовой нейронной сети WGCN в базовом варианте, что еще раз подтверждает эффективность использования мультимодальной информации. На основании полученных результатов можно сделать вывод, что алгоритм HyperFusion-Net эффективен для обучения многомодальным представлениям графов знаний.

Таблица 3.3 – Результаты эксперимента по дополнению графа знаний (DB15K и YAGO15K)

Алгоритм	DB15K					YAGO15K				
	Hits					Hits				
	MR	MRR	@1	@3	@10	MR	MRR	@1	@3	@10
Transe	3	0.257	0.174	0.284	0.420	-	0.182	0.027	0.295	0.444
Rotate	1805	0.52	0.5	0.531	0.551	1732	0.356	0.289	0.391	0.479
R-GCN	2101	0.531	0.479	0.58	0.574	1540	0.304	0.258	0.308	0.461
WGCN	1243	0.601	0.569	0.640	0.664	740	0.454	0.358	0.508	0.611
KBLRN	1357	0.578	0.531	0.623	0.657	1182	0.414	0.335	0.492	0.578
Hyperfusion-Net	917	0.630	0.597	0.647	0.694	474	0.480	0.406	0.524	0.637

Кроме того, в этом эксперименте MRR также был выбран в качестве показателя для оценки скорости сходимости модели на наборе данных FB15k-237. На рисунке 3.3 показаны результаты скорости сходимости пяти моделей в процессе обучения: HyperFusion-Net, compgcn, WGCN, Можно заметить, что HyperFusion-Net показывает лучшие результаты после 50 итераций обучения по сравнению с четырьмя другими алгоритмами. В частности, показатель MRR для HyperFusion-Net достигает 0,118 после одного раунда обучения, в то время как MRR для WGCN и convе составляет всего 0,091 и 0,062 соответственно. Разница между этими алгоритмами показывает, что мультимодальная информация может способствовать быстрой сходимости обучения представлений графа знаний.

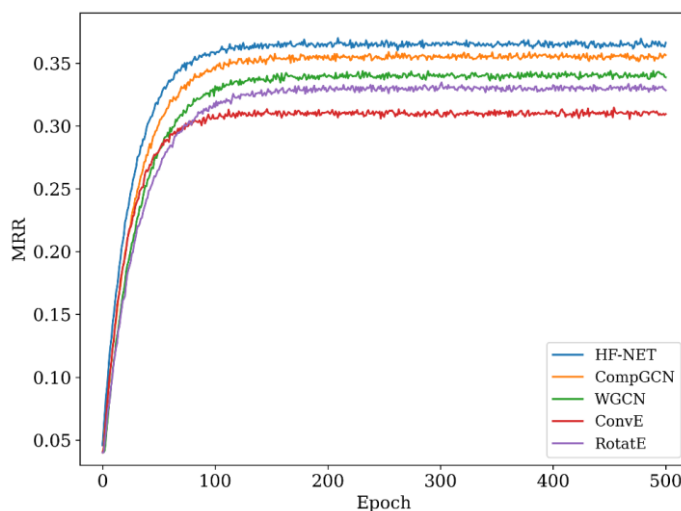


Рисунок 3.3 – Скорость сходимости HyperFusion-Net, compgcn, WGCN, RotatE и ConvE на наборе данных FB15k-237.

Для проверки влияния информации различных модальностей на обучение многомодальным представлениям графов знаний в этом разделе проводится эксперимент по изучению влияния каждой из модальностей информации на решение задачи дополнения многомодального графа знаний. В таблице 3.4 показаны результаты дополнения графа знаний с использованием информации различных модальностей на наборе данных FB15k-237.

Результаты показывают, что наиболее важной информацией в многомодальном графе знаний является текстовая информация, за которой следует числовая информация и, наконец, визуальная информация. Причины таких результатов следующие. Во-первых, визуальная и числовая информация неполны, но каждая сущность имеет подробное текстовое описание в FB15k-237. Во-вторых, сложно описать сложные сущности единственным изображением. В-третьих, предварительно обученная модель может быть неспособна учитывать полную иультимодальную информацию, в то время как BERT, используемый для текстовой информации, предварительно обучен на крупномасштабном корпусе данных с высокой способностью к извлечению признаков.

Таблица 3.4 – Результаты дополнения графа знаний для информации различных модальностей (FB15k-237)

Модальности	Hits@10	Hits@3	Hits@1	MRR	MR
Числовая	0.516	0.366	0.242	0.344	211
Визуальная	0.495	0.355	0.231	0.332	253
Текстовая	0.527	0.381	0.256	0.355	190
Все модальности (полная информация)	0.542	0.404	0.271	0.366	156

Общее сравнение производительности показывает, что именно HyperFusion-Net демонстрирует наилучшие результаты в обучении представлений многомодальных графов знаний. Был также проведен ряд экспериментов по удалению компонентов для проверки эффективности

каждого из основных компонентов алгоритма HyperFusion-Net, в основном путем сравнения HyperFusion-Net с упрощенными («ослабленными») версиями HyperFusion-Net на наборе данных FB15k-237. Перечислим эти версии.

Hyperfusion-Net-IF: в этой версии удален модуль слияния информации на рисунке 3.2. Эта версия модели основана на графе знаний без мультимодальной информации и проверяет влияние мультимодальной информации на результаты эксперимента. Исходный граф знаний подразумевает только один слой эмбеддингов для идентификации сущностей и отношений и преобразования их в соответствующие скрытые представления эмбеддингов.

Hyperfusion-Net-LMF: в этой версии операция слияния низкоранговых тензоров в модуле слияния информации на рисунке 3.2 заменена простым соединением. Эта версия игнорирует динамику взаимодействия внутри- и межмодальных признаков. Данная версия предназначена для проверки эффективности слияния тензоров низкого ранга.

Hyperfusion-Net-RA: в этой версии удален механизм внимания, специфичный для отношений, из модуля слияния информации, что призвано проверить эффективность механизма внимания к связям.

Hyperfusion-Net-ERF: в этой версии удалена операция слияния сущностей и отношений из модуля агрегирования информации, что призвано проверить эффект сильной корреляции между сущностями и связями.

В таблице 3.5 показано сравнение производительности предложенной HyperFusion-Net и четырех его «слабых» версий в задаче дополнения графа знаний на FB15k-237. Наблюдаются следующие экспериментальные результаты.

Влияние мультимодальной информации: для проверки эффекта введения мультимодальной информации в граф знаний в данном эксперименте экспериментально сравниваются HyperFusion-Net и

HyperFusion-Net-IF. Как показано в таблице 3.5, можно заметить, что HyperFusion-Net значительно превосходит HyperFusion-Net-IF по всем оценочным показателям, что указывает на эффективность модуля мультимодальной информации и слияния информации. Кроме того, экспериментальные результаты на HyperFusion-Net и HyperFusion-Net-LMF показывают, что слияние тензоров низкого ранга превосходит простые методы слияния мультимодальной информации при объединении мультимодальной информации из мультимодальных графов знаний.

Влияние реляционной информации: для проверки влияния реляционной информации (информации о связях) на обучение представлению мультимодального графа знаний в этом эксперименте сравниваются HyperFusion-Net, HyperFusion-Net-RA и HyperFusion-Net-ERF, результаты представлены в таблице 3.5.

Таблица 3.5 – Сравнительный анализ производительности различных версий HyperFusion-Net в задаче дополнения графа знаний (FB15K-237)

Версии алгоритма	Hits@10	Hits@3	Hits@1	MRR	MR
Hyperfusion-Net	0.542	0.404	0.271	0.366	156
Hyperfusion-Net-IF	0.536	0.382	0.251	0.345	211
Hyperfusion-Net-LMF	0.535	0.385	0.261	0.355	201
Hyperfusion-Net-RA	0.537	0.391	0.262	0.357	192
Hyperfusion-Net-ERF	0.528	0.369	0.241	0.338	232

Сравнение HyperFusion-Net с HyperFusion-Net-RA показывает, что механизмы внимания, специфичные для связей, оказывают слабое положительное влияние на обучение представлению мультимодального графа знаний. Однако этот эксперимент выявил значительное улучшение результатов HyperFusion-Net по сравнению с HyperFusion-Net-ERF, что указывает на важность информации о связях для графов знаний. Поскольку эффект механизма внимания, специфичного для связей, не особенно очевиден, первоначально предполагается, что это связано с тем, что механизм внимания, специфичный для связей, не учитывает направление связи, в то время как

кросс-корреляция, используемая в операции слияния сущностей и отношений, учитывает направленность связи. Следовательно, для обучения представлению графа знаний особенно важна не только информация о связях, но и об их направленности.

Анализ влияния гиперпараметров алгоритма

В этом разделе в ходе экспериментов по определению гиперпараметров оценивается влияние различных гиперпараметров на производительность предложенного алгоритма HyperFusion-Net на наборе данных FB15k-237. Результаты представлены на рисунке 3.4, 3.5 и 3.6.

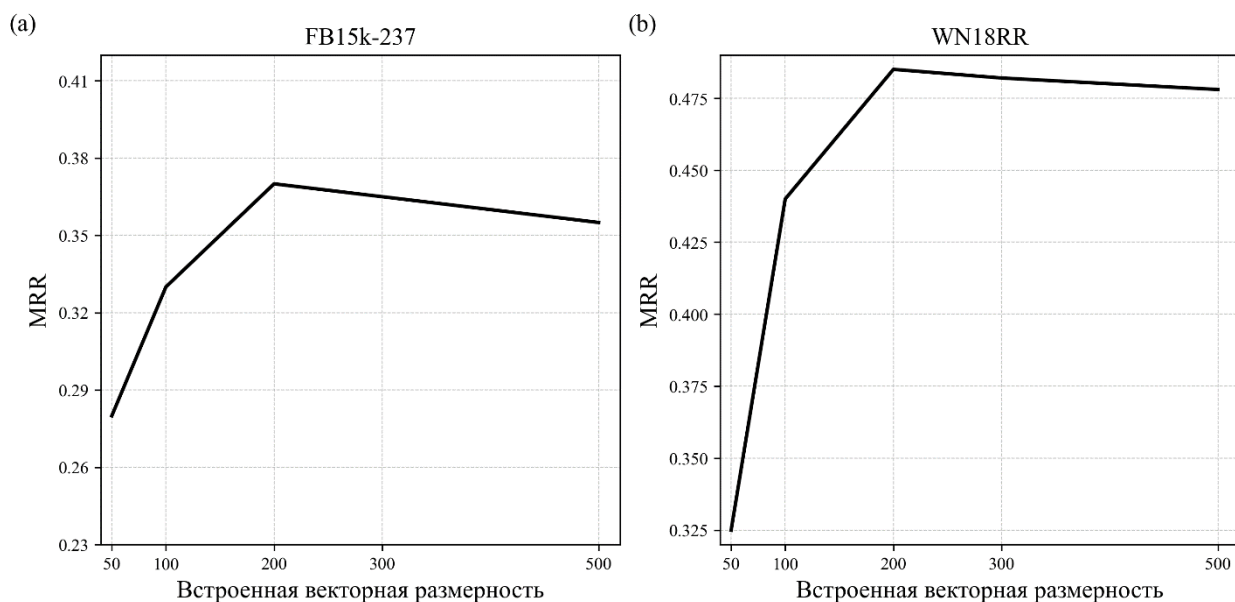


Рисунок 3.4 – Влияние размерности вектора эмбединга на производительность алгоритма для FB15k-237 (a) и WN18RR (b)

Для оценки влияния размерности представления графа знаний (т. е. размера эмбедингов суперузлов и связей в мультимодальном графе знаний) на модель, были использованы значения размерности от 50 до 500, а в качестве метрики оценки был выбран MRR (Multimodal Knowledge Rank). Как показано на рисунке 3.4, производительность алгоритма сначала увеличивается, а затем уменьшается с увеличением размерности эмбединга. Когда размерность вектора представления встраивания установлена равной 50, модель не обладает достаточной емкостью для кодирования информации

многомодального графа знаний. Когда размерность встраивания превышает 200, эффективность модели не увеличивается, что доказывает, что модель может достигать лучших результатов даже при меньших размерностях эмбеддингов.

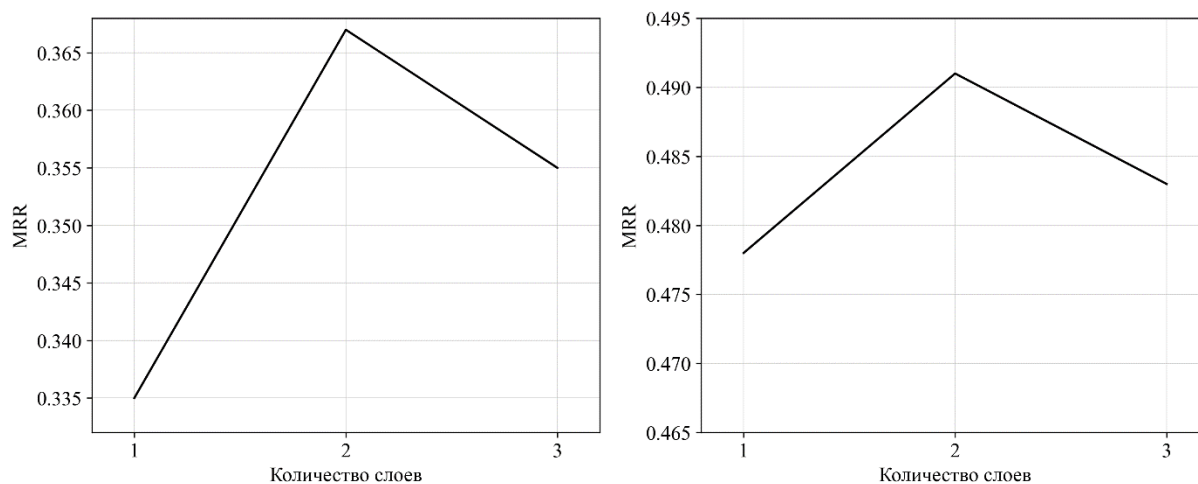


Рисунок 3.5 – Влияние количества слоев на производительность модели на наборах данных FB15k-237 (a) и WN18RR (b)

Оценим влияние количества слоев модуля агрегации информации на производительность модели. Модуль агрегации информации – модель графовой нейронной сети, которая агрегирует информацию о соседних узлах целевого узла, а затем обновляет представление встраивания целевого узла путем интеграции информации о соседних узлах. Этот метод агрегации может привести к чрезмерному сглаживанию представления каждого узла, то есть по мере увеличения количества слоев графовой нейронной сети в процессе обучения представления каждого узла в графе становятся похожими. Для исследования влияния количества слоев модуля агрегации информации на производительность модели, ее производительность на наборе данных FB15k-237 была исследована путем изменения количества слоев модуля агрегации информации от 1 до 3.

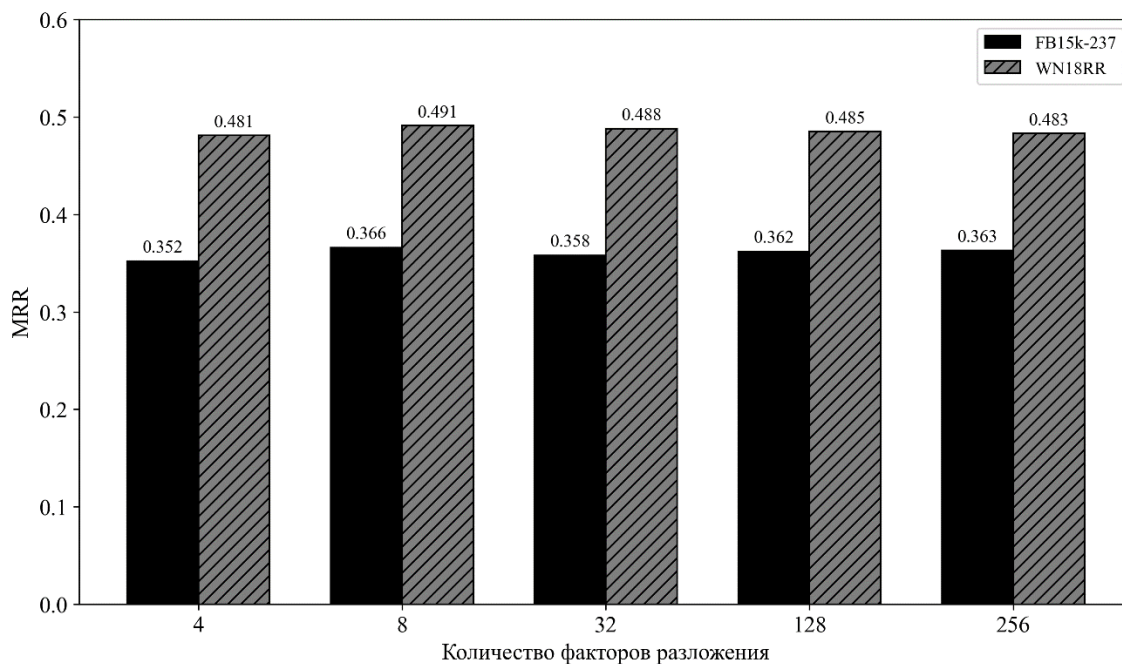


Рисунок 3.6 – Влияние числа факторов декомпозиции на производительность модели

По мере увеличения количества слоев модуля агрегации информации требования к памяти экспоненциально возрастают, а память графического процессора ограничена, поэтому максимальное количество слоев в этом эксперименте установлено равным 3, и проверить большее количество слоев модуля агрегации информации оказалось невозможно. Как показано на рисунке 3.5, при количестве слоев модуля агрегации информации, равном 2, модель демонстрирует наилучшую производительность. Из рисунка можно сделать следующий вывод: когда количество слоев модуля агрегации информации слишком мало, информация в графе знаний не может быть достаточно агрегирована для изучения структурной информации графа знаний, и некоторые полезные соседи более высокого порядка не могут быть захвачены, и производительность модели в этом случае очень низкая. Когда количество слоев модуля агрегации информации слишком велико, представление узлов будет чрезмерно сглажено, и производительность модели ухудшится.

Наконец, были проведены эксперименты для исследования влияния

количества факторов разложения при слиянии тензоров низкого ранга на производительность модели. В экспериментах измерялось изменение производительности модели при различном количестве факторов разложения на наборах данных FB15k-237 и WN18RR. Результаты представлены на рисунке 3.6, из которого видно, что результаты практически не изменяются с увеличением количества факторов разложения. Небольшого количества факторов разложения достаточно для достижения хорошей производительности при слиянии мультимодальной информации.

3.5 Результаты главы 3

В настоящей главе предложен алгоритм HyperFusion-Net для реализации обучения представлений для мультимодальных графов знаний. Алгоритм сначала использует три различных предварительно обученных кодировщика для получения предварительно обученных вложений мультимодальной информации, а именно текстовой, визуальной и числовой. Затем традиционный граф знаний преобразуется в граф гиперточек, и выполняется слияние тензоров низкого ранга на основе предварительно обученных вложений сущностей, узлы которых содержат объединенную мультимодальную информацию. Кроме того, HyperFusion-Net обновляет представление гиперточек с помощью модуля агрегации информации для захвата структурной информации в графе знаний. Модуль агрегации информации повторно использует сеть внимания графа слияния сущностей и связей (ERGAT), упомянутую в главе 2, где механизм внимания, специфичный для связей, и операция слияния сущностей и связей позволяют модели получать более качественные представления связей. После получения окончательного встраивания мультимодального графа знаний модель использует ConvE для задачи дополнения графа знаний с целью завершения обучения с учителем. В этой главе проведены всесторонние эксперименты по

обучению представлению мультимодального графа знаний на четырех эталонных наборах данных. Результаты показывают, что предложенная HyperFusion-Net не только превосходит существующие методы обучения представлению графа знаний, но и обладает более высокой скоростью сходимости. Для проверки влияния различной одномодальной информации и различных компонентов в HyperFusion-Net на обучение представлению мультимодального графа знаний в этой главе также проведен ряд экспериментов по исключению компонентов. Результаты главы опубликованы в статье [165].

Алгоритм HyperFusion-Net, представленный в настоящей главе, позволяет получать более полные и точные векторные представления для многомодальных графов знаний. Полученные векторы эмбедингов сущностей и отношений имеют значение для решения многих прикладных задач, включая рекомендательные системы. Однако в сценариях, требующих перечисления огромного количества комбинаций (например, персонализированное сопоставление одежды), прямое использование этих представлений сталкивается с проблемами вычислительной эффективности. Комбинаторный взрыв делает невозможным применение стандартных методов попарного сравнения в режиме реального времени. Для преодоления этого ограничения в главе 4 предлагается принципиально иной подход, основанный на хешировании. Алгоритм MCHM-Net может преобразовывать плотные векторные представления в компактные бинарные коды, открывая путь к созданию эффективных комбинаторных рекомендательных систем на основе графовых знаний, которые могут работать в режиме реального времени.

4 АЛГОРИТМ КОМБИНАТОРНЫХ РЕКОМЕНДАЦИЙ НА ОСНОВЕ ХЭШИРОВАНИЯ

Вопросы построения рекомендательных систем на основе коллаборативной фильтрации подробно рассматривались в наших ранних работах [166-168], где исследовались различные стратегии снижения ложных оценок и повышения точности рекомендаций. В реальных задачах составления комбинаторных рекомендаций огромное количество элементов приводит к экспоненциальному расширению комбинаторного пространства, в результате чего вычислительная эффективность становится ключевой проблемой, которую нельзя игнорировать. По этой причине в данной главе предлагается алгоритм MCHM-Net, который повышает производительность персонализированной комбинаторной системы рекомендаций в аспектах вычислительной эффективности и точности рекомендаций за счет совместного обучения двоично-кодированным представлениям пользователей и элементов.

4.1 Задачи комбинаторных рекомендаций и методы хэширования

Задача составления наборов/комплектов – задача комбинированных рекомендаций – широко изучается в последнее время. В этой задаче совместимость, например, одежды отличается от простого визуального сходства (например, по цвету, атрибутам и т. д.), и соответствие между предметами одежды носит субъективный характер. Разные пользователи могут иметь диаметрально противоположные предпочтения, что затрудняет описание соответствия наборов с помощью объективных правил. В персонализированном сопоставлении наборов важно не только измерять корреляцию между несколькими предметами, но и моделировать различные предпочтения пользователей. Разные пользователи имеют существенно

различающиеся предпочтения в соответствии с выбранным стилем. Предположим, что набор состоит из нескольких предметов x_1, \dots, x_n , содержащих предметы из разных категорий, а пользователи представлены вектором признаков u . Задача персонализированной рекомендации по сопоставлению наборов может быть кратко сведена к обучению функции оценки $f_\theta(x_1, \dots, x_n; u)$ для рекомендации наборов, подходящих для разных пользователей, где θ – обучаемый параметр. Предположим, что n предметов в наборе относятся к n различным категориям, при этом каждая категория содержит L_n предметов. Всего существует возможное количество сопоставлений (комплектов) $\prod_{i=1}^n L_i$ при рассмотрении наборов фиксированной длины. На практике количество элементов L_i в каждой категории огромно, что приводит к взрывному росту общего числа наборов с увеличением категорий элементов, затрудняя алгоритму вычисление всех возможных вариантов. Поэтому на практике эффективность поиска данных является важным вопросом, который необходимо учитывать и исследовать. Для повышения эффективности вычислений в этой главе сначала моделируется функция оценки f_θ на основе явной формы, и используется метод хеширования для эффективного вычисления попарных связей.

На основе различных способов моделирования функции оценки f_θ предлагаются различные алгоритмы. Один из подходов на основе явного моделирования заключается в разложении функции оценки на сумму корреляций два-к-двум между элементами и элементами, а также между элементами и пользователями. Многомерная взаимосвязь сопоставления наборов ограничивается низкоразмерной попарной моделью, в то время как задача прогнозирования совместимости наборов сводится к задаче сопоставления элементов. Например, Veit et al. [18] использовали сети-близнецы для изучения сходства между различными элементами. Hu et al. [98] дополнительно рассмотрели сходство между пользователями и

элементами на основе модели тензорного разложения. Поскольку персонализированные предпочтения между разными пользователями часто имеют схожие закономерности, а высокоразмерные отношения имеют низкоранговый характер, подход, основанный на явном разложении, позволяет получить более высокую точность рекомендаций, используя модель низкорангового разложения. В дополнение к алгоритмам явного моделирования на основе совместимости, некоторые опубликованные работы рассматривают набор как единое целое для изучения потенциальных корреляций более высокого ранга между элементами. Однако в этом случае функция прогнозирования f_{θ} часто не имеет явной структуры, и вычислительную сложность оценки совместимости трудно уменьшить с помощью существующих алгоритмов поиска ближайших соседей.

Целью хеширования является представление данных в виде набора компактных бинарных кодов (БК) для эффективного вычисления сходства. Поэтому методы хеширования естественным образом и широко используются в таких областях, как поиск изображений. В этой главе также предлагается новый подход к бинарному кодированию на основе выборки с помощью вероятностной интерпретации на основе хеш-функции (ХФ). Этот подход позволяет алгоритму находить компромисс между точностью рекомендаций и их эффективностью.

Алгоритм хеширования в глубоком обучении

Алгоритм хэширования основан на приблизительном поиске ближайших соседей (ANN – Approximate Nearest Neighbors). Благодаря своей эффективности в хранении и поиске данных, хэширование широко используется в таких задачах, как поиск изображений (IR – Image Retrieval). В зависимости от данных, методы хэширования в основном делятся на два типа: независимые от данных и зависимые от данных. Среди независимых от данных алгоритмов хэширования ранним представителем является локально-чувствительное хэширование (LSH – Locally Sensitive Hashing).

Основной принцип алгоритмов локально-чувствительного хэширования заключается в отображении точек, расположенных ближе друг к другу, в одно и то же кодирование с большей вероятностью. Поскольку такие алгоритмы не учитывают распределение данных, трудно добиться хорошей производительности в сценариях применения с большими объемами и многомерными данными. Поэтому алгоритмы хэширования на основе машинного обучения получили широкое внимание и стали предметом исследований. Алгоритмы хэширования, основанные на машинном обучении, используют различные распределения данных и учитывают потенциальные особенности данных, чтобы лучше обучаться бинарному кодированию образцов. Благодаря недавнему развитию технологий глубокого обучения, использование глубокого обучения для улучшения алгоритмов хэширования стало нормой, и одновременно был достигнут значительный прогресс. Поэтому в этом разделе мы сосредоточимся на алгоритмах хэширования с учителем, основанных на глубоком обучении.

Основная проблема алгоритмов хэширования

В хеш-кодировании двоичный код длины D представляет собой вектор из D элементов. Каждый элемент вектора называется битом, и для каждого бита существует только два возможных значения. Пусть это будет $\{H, T\}$. Пусть b_i и b_j – два различных двоично-кодированных вектора, а расстояние Хэмминга между векторами определяется как количество различных элементов в двух векторах, т.е.:

$$\text{dist}(b_i, b_j) = \sum_{k=1}^D \delta(b_{ik} \neq b_{jk}). \quad (4.1)$$

Здесь $\delta(\cdot)$ – индикаторная функция, которая определяется как $\delta(\cdot) = 1$, когда условие в скобках выполняется, $\delta(\cdot) = 0$ в противном случае. Аналогично, сходство двух двоичных кодов определяется как количество одинаковых элементов двух векторов, т.е.:

$$\text{sim}(\mathbf{b}_i, \mathbf{b}_j) = \sum_{k=1}^D \delta(\mathbf{b}_{ik} = \mathbf{b}_{jk}). \quad (4.2)$$

Описанные выше формулы расстояний и соотношения подобия могут быть эффективно вычислены с помощью побитовых операций XOR. На практике в качестве возможных значений бинарного вектора используется $\{-1,1\}$. На основе этого бинарного значения количество одинаковых элементов связывается со скалярным произведением бинарного вектора следующим образом:

$$\sum_{k=1}^D \delta(\mathbf{b}_{ik} = \mathbf{b}_{jk}) = (\mathbf{b}_i^T \mathbf{b}_j + D) / 2. \quad (4.3)$$

Поскольку скалярное произведение $\mathbf{b}_i^T \mathbf{b}_j$ линейно связано с показателем сходства, определенным в приведенном выше уравнении, то без потери общности скалярное произведение бинарных векторов, закодированных как $\{-1,1\}$, можно использовать непосредственно в качестве показателя сходства для упрощения обсуждения. Аналогично, расстояние Хэмминга в уравнении можно упростить, используя отрицательное скалярное произведение.

В процессе обучения алгоритма каждый бит в двоичном коде не обязательно имеет одинаковую важность. Если каждый бит рассматривать как различный атрибут образца (например, цвет, категория и т. д.), то становится ясно, что каждый атрибут по-разному влияет на вычисление сходства. Поэтому в существующих работах для дальнейшего повышения производительности алгоритмов хеширования распространенной практикой является расширение скалярного произведения до битово-взвешенной формы, т. е.

$$\mathbf{b}_i^T \Lambda \mathbf{b}_j = \sum_{k=1}^D \lambda_k \mathbf{b}_{ik} \mathbf{b}_{jk}. \quad (4.4)$$

Здесь диагональная матрица $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_D)$ обычно представляет

собой набор обучаемых параметров для весов различных битов в двоичном кодировании.

Пусть N – набор данных выборки, а матрица $Y \in \{-1, 1\}^{N \times N}$ – информация о метках между парами образцов. Цель алгоритма хеширования – изучить θ -параметризованную хеш-функцию $h_\theta(\cdot)$ на основе набора данных, полученных с помощью обучения с учителем. Хеш-функция отображает образцы x_i в компактное D -мерное бинарное представление $b_i \in \{-1, 1\}^D$, позволяя при этом образцам сохранять как можно большее сходство в бинарном пространстве с исходным пространством. В данных с метками Y , $y_{ij} = 1$ означает, что (x_i, x_j) коррелированы, а $y_{ij} = -1$ означает, что (x_i, x_j) некоррелированы. Следовательно, основная математическая форма алгоритма хеширования может быть выражена следующим образом:

$$\begin{aligned} \theta^* &= \arg \min_{\theta} \mathcal{L} = \sum_{i,j} \left(y_{ij} - \frac{1}{D} b_i^T b_j \right)^2, \\ \text{s.t. } b_i &= h_\theta(x_i) \in \{-1, 1\}^D. \end{aligned} \quad (4.5)$$

В этом случае определение функции потерь L зависит от сходства y_{ij} в исходном пространстве и сходства в бинарном пространстве, и ее цель состоит в том, чтобы сохранить неизменной связь сходства образцов в бинарном пространстве. В алгоритме хеширования на основе глубокого обучения используются глубокие нейронные сети, которые применяют мощные возможности нейронных сетей по извлечению признаков для обучения выборочных данных. Поскольку хеш-функция $b = h_\theta(x) \in \{-1, 1\}^D$ дискретна, градиент функции потерь L относительно параметра θ в уравнении (4.5) равен нулю. Это делает невозможным прямую оптимизацию параметра θ методом градиентного спуска.

Методы оптимизации для обучения хешированию

Для задачи оптимизации в (4.5) простой подход заключается в пошаговой оптимизации. В частности, сначала из матрицы подобия $Y \in \{-1, 1\}^{N \times N}$ получают двоичное кодовое представление B , соответствующее N образцам. Затем хеш-функция $h_\theta(\cdot)$ непосредственно обучается на основе полученного двоичного кодирования. Например, Ся и др. используют алгоритм координатного спуска для получения двоичного кодирования образцов:

$$B^* = \arg \min_B \left\| Y - \frac{1}{D} B B^T \right\|_2. \quad (4.6)$$

Соответствующий двоичный код для каждого образца затем обучается непосредственно по данным с помощью нейронной сети:

$$\theta^* = \arg \min_\theta \sum_i \| h_\theta(x_i) - b_i^* \|_2. \quad (4.7)$$

Двоичное кодирование нового образца можно получить, взяв знак хеш-функции, т.е. $b = \text{sign}(h_\theta^*(x))$, где функция знака определяется следующим образом:

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ -1, & x < 0. \end{cases} \quad (4.8)$$

Как видно, этот тип алгоритма разделяет основную задачу оптимизации в уравнении (4.5) на два отдельных шага и поэтому также известен как двухэтапный алгоритм. Как и следовало ожидать, производительность двухэтапного алгоритма в значительной степени ограничена качеством результата разложения матрицы подобия в уравнении (4.6). Поскольку на этом шаге образцы не характеризуются, а используется только корреляционная матрица, общая производительность будет значительно снижена.

Для сквозного обучения задачи оптимизации в уравнении (4.5) необходимо решить проблему нулевого градиента относительно функции знака. Одна из распространенных стратегий, используемых исследователями,

заключается в введении вектора релаксации $h \in [-1, 1]^D$ в качестве аппроксимации соответствующего бинарного вектора $b \in \{-1, 1\}^D$ и оптимизации параметра θ на основе непрерывно дифференцируемого отображения $h = h_\theta(x)$. Соответствующий бинарный вектор b затем получается путем квантования h . Сквозной алгоритм позволяет модели обрабатывать данные вне выборки путем введения ошибки квантования $Q(h, b)$, которая заставляет непрерывный вектор h аппроксимировать бинарный вектор b . Цель оптимизации можно сформулировать следующим образом:

$$\mathcal{L}(h) + Q(h, b). \quad (4.9)$$

Например, Лиу и др. предлагают алгоритм хеширования, основанный на векторе релаксации h , для прямой оптимизации. Алгоритм основан на следующей функции потерь:

$$\begin{aligned} \mathcal{L} = & \frac{1}{2}(1 - y_{ij}) \|h_i - h_j\|_2^2 + \frac{1}{2} y_{ij} \max(m - \|h_i - h_j\|_2^2, 0) \\ & + \alpha(\| |h_i| - 1 \|_1 + \| |h_j| - 1 \|_1). \end{aligned} \quad (4.10)$$

Здесь $h_i = h_\theta(x_i)$ содержит нейросеть глубокого обучения, а m и α – различные коэффициенты в целевой функции. Первые два члена функции потерь используются для сближения похожих пар образцов в низкоразмерном пространстве и для отдаления несхожих пар в низкоразмерном пространстве. Последний член функции потерь уменьшает ошибку квантования, аппроксимируя значение переменной ослабления h к бинарному кодированию $\{-1, 1\}^D$ с помощью однопараметрического ограничения $\| |h| - 1 \|_1$. Соответствующий бинарный вектор b образца квантуется путем взятия знака выходных данных $h_\theta(\cdot)$, т.е. $b = \text{sign}(h)$.

Описанная выше сквозная пара данных использует одну и ту же хеш-функцию для бинарного кодирования всех данных. Для получения лучшего бинарного кодирования существующих данных в выборке исследователи предложили алгоритм, основанный на чередующейся

оптимизации. Одним из преимуществ алгоритма, основанного на алгоритме чередующейся оптимизации, является то, что алгоритм может обучать хеш-функцию $h_\theta(\cdot)$, используя только часть информации о признаках данных. Например, Цзян и др. предложили следующий алгоритм:

$$\begin{aligned} \min_{h,b} \mathcal{L} &= \sum_{i \in \Omega} \sum_{j \in \Gamma} L(b_i, b_j; y_{ij}) + \sum_{i,j \in \Omega} L(b_i, b_j; y_{ij}), \\ \text{s.t. } b_i &\in \{-1, 1\}^D, \forall i \in \Omega. \end{aligned} \quad (4.11)$$

Здесь Γ обозначает множество всех образцов, а Ω – подмножество образцов. Получив двоичное кодирование образцов B , сеть обучает хеш-функцию для оптимизации H . Аналогично, авторы оптимизируют двоичное кодирование образцов в подмножестве B с помощью бинарного квадратичного программирования (BQP), получив на вход выходные данные сети H . Авторы также оптимизируют двоичное кодирование образцов в подмножестве B с помощью BQP.

Далее рассматривается текущее состояние исследований алгоритмов хеширования с учителем, основанных на глубоком обучении. Существующие работы подробно исследуют различные функции потерь, а также различные ограничения, что значительно улучшает производительность алгоритмов, способствует развитию обобщенных методов хеширования и обеспечивает теоретическую поддержку для других применений методов хеширования. Для повышения точности персонализированных рекомендаций мы также исследовали методы взвешенной коллаборативной фильтрации [169], которые легли в основу разработки весового хеширования в MCHM-Net.

Алгоритм хеширования и коллаборативная фильтрация

Коллаборативная фильтрация – классический алгоритм в рекомендательных системах. Предположим, есть M пользователей и N товаров, оценка товаров пользователями представлена матрицей Y . Цель алгоритма коллаборативной фильтрации – изучить представление пользователя U и представление товара V таким образом, чтобы

минимизировать следующие потери:

$$\mathcal{L} = \|Y - UV^T\|_2^2. \quad (4.12)$$

Сравнивая с целевой функцией оптимизации алгоритма хеширования в уравнении (4.5), можно обнаружить, что алгоритм коллаборативной фильтрации может реализовать бинарное представление пользователей и элементов с помощью существующей техники хеширования. Например, Чжан и др. предложили дискретную коллаборативную фильтрацию (DCF – Discrete Collaborative Filtering) для внедрения методов хеширования в алгоритмы коллаборативной фильтрации и предложили следующую целевую функцию оптимизации:

$$\begin{aligned} & \arg \min_{U, V} \sum_{i, j} (y_{ij} - u_i^T v_j)^2, \\ & \text{s.t. } U \in \{-1, 1\}^{M \times D}, V \in \{-1, 1\}^{N \times D}, \\ & V^T \mathbf{1} = 0, U^T \mathbf{1} = 0, \\ & \frac{1}{D} V^T V = I, \frac{1}{D} U^T U = I. \end{aligned} \quad (4.13)$$

Здесь y_{ij} – оценка товара j пользователем i , u_i – бинарный вектор пользователей, а v_j – бинарный вектор товаров. Последние два ограничения в члене ограничения обозначают соответственно ограничения баланса кодирования и декорреляции. Авторы считают, что -1 и 1 имеют одинаковую вероятность появления в каждом бите, в то время как разные биты некоррелированы. Поскольку алгоритм чистой коллаборативной фильтрации игнорирует информацию о содержании товара. Далее, в последующих работах рассматривался вопрос обучения бинарных представлений пользователей и товаров при наличии вспомогательной информации о пользователе или товаре. Кроме того, Лиу и др. применили хеш-алгоритм к задачам, связанным с рекомендациями предметов одежды. Однако алгоритм учитывает только отношение соответствия товаров и не рассматривает проблему рекомендации комбинаций товаров. Поэтому проблемы, связанные

с эффективностью в области персонализированных рекомендаций по подбору костюмов, до сих пор недостаточно изучены.

4.2 Технология хэширования и персонализированное сопоставление комбинаций на основе хэширования

Цель персонализированных рекомендаций – вырабатывать рекомендации товаров в соответствии с предпочтениями разных пользователей. В прошлых системах рекомендаций часто рекомендовались отдельные товары. Однако в задаче рекомендации на основе сопоставления наборов предметом рекомендации является уже не один товар, а комбинация нескольких товаров. В этом случае одна и та же комбинация часто оценивается только отдельными пользователями. Для прямого применения коллаборативной фильтрации слишком мало релевантных данных между разными пользователями. Поэтому в этой задаче алгоритм хэширования необходимо переосмыслить и пересмотреть. Пусть N – количество категорий одежды, L_n – общее количество товаров в категории, а количество пользователей равно U . Без потери общности предположим, что комплект одежды состоит из N товаров из разных категорий. Обозначим множество O_i :

$$O_i = \{x_{i_1}^{(1)}, x_{i_2}^{(2)}, \dots, x_{i_N}^{(N)}\}. \quad (4.14)$$

Здесь $i = (i_1, \dots, i_N)$ обозначает индекс различных элементов в наборе в соответствующих им категориях. Например, $x_{i_n}^{(n)}$ – это элемент в n -й категории. Задача персонализированной рекомендации одежды может быть кратко описана как обучение функции оценки Γ_{u, O_i} таким образом, чтобы разные пользователи U оценивали разные наборы O_i . Эта функция оценки измеряет не только совместимость набора O_i , но и персонализированные предпочтения пользователя. На практике количество пользователей, а также количество элементов в каждой категории часто огромно, что приводит к

возможности того, что все комбинации пользователь-набор будут огромными. Вероятность всех комбинаций (u, O_i) равна $U \times L_1 \times \dots \times L_N$, и это число будет расти полиномиально с количеством рассматриваемых категорий. Поэтому в практических системах необходим более эффективный способ вычисления попарной совместимости, чтобы алгоритмы рекомендаций могли быть эффективно развернуты. Хотя существующие работы были очень успешными с точки зрения точности рекомендаций, вычислительные и проблемы хранения часто игнорируются и недостаточно исследуются.

Алгоритм MCHM-Net: общая структура

В настоящей главе предлагается алгоритм MCHM-Net, основанный на хешировании, для прогнозирования предпочтений пользователя в отношении различных комбинаций одежды. Общая архитектура алгоритма MCHM-Net показана на рисунке 4.1.

Алгоритм состоит из трех модулей. Первый модуль – сеть признаков для извлечения признаков. Для визуальных данных используется сверточная нейронная сеть для извлечения представлений изображений, и выбор конкретной структуры сверточной нейронной сети не является предметом рассмотрения в этой главе.

Второй модуль – это хеш-модуль. Поскольку существуют разные категории предметов одежды, предметы и пользователи из разных категорий рассматриваются как разные типы. Поэтому для каждой категории в модели используются разные хеш-функции. Наконец, модуль сопоставления вычисляет функцию оценки пользователя для набора комбинаций на основе различных характеристик предметов и пользователей. Каждый пользователь представлен One-hot-кодированием, которое используется для индексации пользователя. В конечном итоге модель кодирует пользователей и предметы в виде бинарных векторов $\{\pm 1\}$ с D измерениями. Здесь может быть полезно использовать $b_u^{(u)}$ для обозначения u -го пользователя, а $b_{t_n}^{(n)}$ для

обозначения i_n -го элемента в n -й категории.

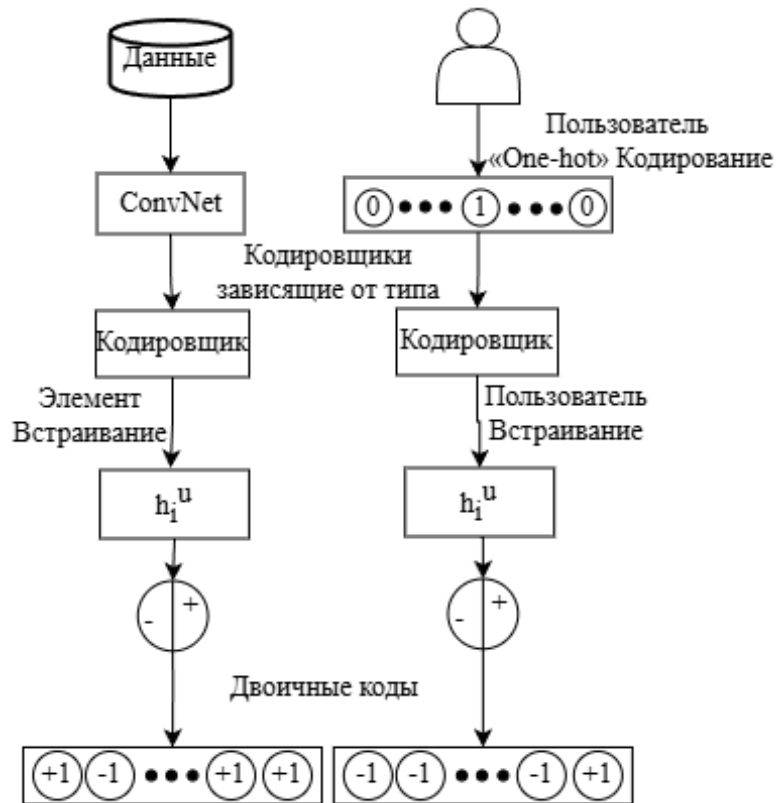


Рисунок 4.1 – Схема алгоритма MCHM-Net

4.3 Совместимость комбинаций и хэш-обучение

Тензорное разложение – способ моделирования отношений более высоких порядков. Существуют быстрые алгоритмы поиска максимального значения для повышения вычислительной эффективности на основе тензорного разложения. Формально, пусть $b_u^{(u)} \in \{-1,1\}^D$ – бинарный вектор u -го пользователя, а $b_{i_n}^{(u)} \in \{-1,1\}^D$ – i_n -й элемент в n -й категории. Широко используемым тензорным разложением является модель CANDECOMP/PARAFAC (CP), которая определяется следующим образом:

$$r_{u,O_i}^{CP} = \sum_{k=1}^D b_{u,k}^{(u)} b_{i_1,k}^{(1)} \cdots b_{i_N,k}^{(N)}. \quad (4.15)$$

Из-за проблемы разреженности данных, для отношений более высокого

порядка, содержащиеся в r_{u,O_i}^{CP} , могут возникать трудности их эффективного изучения. Более того, в реальных наборах данных каждый элемент может использоваться только один раз лишь в нескольких комбинациях, что еще больше усугубляет проблему разреженности данных. Между тем, алгоритмы декомпозиции на основе коллаборативной фильтрации требуют, чтобы каждая комбинация была видна существующим пользователям во время тестирования. Это требование трудно выполнить в рекомендациях комбинаций. Для решения вышеуказанных проблем в этом разделе предлагается модель попарной декомпозиции, основанная на контент-анализе. Соответствующая совместимость костюмов должна быть определена в следующей форме:

$$r_{u,O_i} = \frac{1}{z_1} \sum_n b_{i_n}^{(n)T} b_u^{(u)} + \frac{1}{z_2} \sum_{n \neq m} b_{i_n}^{(n)T} b_{i_m}^{(m)}. \quad (4.16)$$

Чтобы модель могла обрабатывать комбинации различной длины, общая оценка нормализуется. Здесь z_1 и z_2 обозначают количество пар векторов. В этой модели совместимости есть два слагаемых: первое указывает, насколько пользователь предпочитает комбинацию, а второе описывает, насколько хорошо сама внутренне соответствует. Для дальнейшего повышения производительности оценки здесь улучшаются с помощью алгоритма взвешенного хэширования, т.е.:

$$r_{u,O_i} = \frac{1}{z_1} \sum_n b_{i_n}^{(n)T} \Lambda_1 b_u^{(u)} + \frac{1}{z_2} \sum_{n \neq m} b_{i_n}^{(n)T} \Lambda_2 b_{i_m}^{(m)}. \quad (4.17)$$

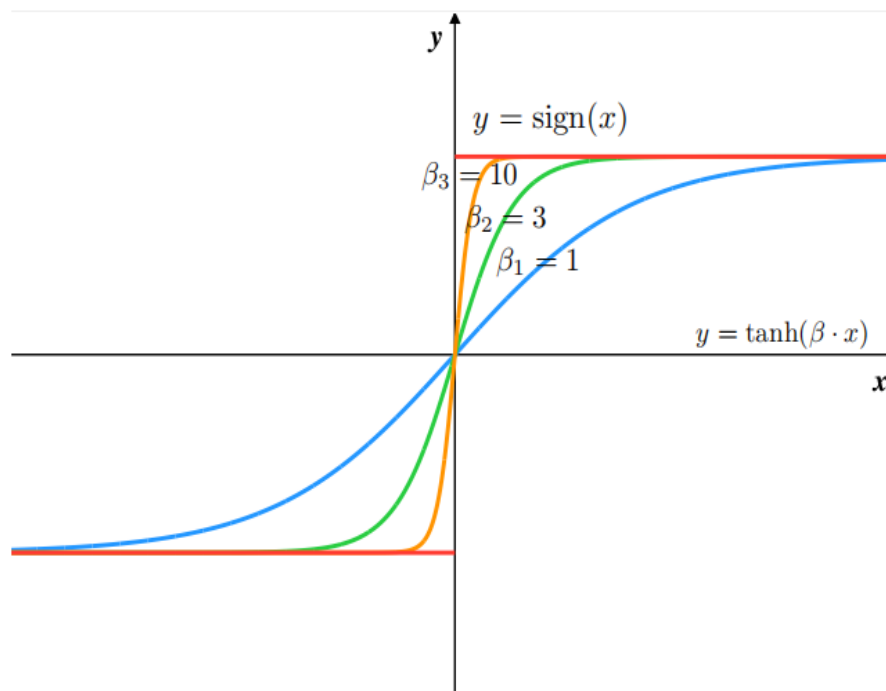


Рисунок 4.2 – Функция активации ScaletanH

Здесь $\Lambda^{(u)}$ и $\Lambda^{(i)}$ – две неотрицательные диагональные матрицы, используемые для определения важности различных битов в бинарном векторе.

В оценке совместимости есть два элемента, которые измеряют релевантность между элементами и персонализированными предпочтениями пользователя. Обозначим релевантность элемента через $r_{(i)}$, а релевантность пользователя через $r_{(u)}$. Вышеуказанную совместимость можно записать следующим образом:

$$r_{u,o_i} = \alpha \cdot r_{u,o_i}^{(u)} + r_{u,o_i}^{(i)}. \quad (4.18)$$

Здесь α используется для уравнивания слагаемых оценки совместимости.

Хэш-обучение

Без потери общности, \mathbf{b} используется для представления бинарного вектора элементов или пользователей. Поскольку сложно напрямую оптимизировать целевую функцию с бинарными ограничениями, необходимо

аппроксимировать b , используя вместо этого непрерывную переменную h . Следовательно, необходимо аппроксимировать исходную задачу, используя непрерывную переменную h вместо b , минимизируя при этом ошибку квантования, т.е. $Q(h,b) = \|h - b\|_1$. В этом разделе предлагаются два различных алгоритма обучения хешей для обучения персонализированной оценки костюма.

Ошибка квантования является одним из основных факторов, влияющих на производительность алгоритма, и в то же время из-за ошибки квантования возникает большая ошибка между двоичным кодированием и непрерывным вектором, что делает алгоритм неспособным точно получить двоичное кодирование. Поэтому Цао и др. предложили сквозной алгоритм без ошибки квантования. Алгоритм аппроксимирует функцию знака функцией активации ScaleTanH, а именно

$$\text{ScaleTanH}(x) = \text{TanH}(\beta_t \cdot x), \beta_t > 0. \quad (4.19)$$

При этом количество итераций β_t увеличивается. Как показано на рисунке 5.2, $\text{TanH}(\beta_t \cdot x)$ непрерывно приближается к значению функции знака $\text{sign}(x)$ по мере увеличения β_t . В частности, при $\beta_t \rightarrow \infty$,

$$\text{sign}(x) = \lim_{\beta_t \rightarrow \infty} \text{TanH}(\beta_t \cdot x). \quad (4.20)$$

Таким образом, алгоритму сквозного хэширования на основе функции активации scaletanh достаточно увеличивать значение β_t с каждой итерацией t в процессе оптимизации для получения качественного бинарного кодирования. Алгоритм хэширования на основе функции активации ScaleTanH прост и легко реализуем.

Подход, основанный на функции активации, реализуется путем добавления слоя ScaleTanH после выходных данных признаков пользователя и товара. Предположим, что на шаге t конечное представление товара (элемента комбинации) – это $h_{i,t}^{(n)}$, а конечное представление пользователя – это $h_{u,t}^{(u)}$.

Тогда взвешенная оценка на основе хеша равна

$$r_{u,0_i} = \frac{1}{Z_1} \sum_n h_{i_n,t}^{(n)T} \Lambda_1 h_{u,t}^u + \frac{1}{Z_2} \sum_{n \neq m} h_{i_n,t}^n \Lambda_2 h_{i_m,t}^m. \quad (4.21)$$

В существующих работах двоичное кодирование выборок в основном осуществляется с помощью знаковой функции, что эквивалентно использованию детерминированной хеш-функции $h_\theta(\cdot)$. Следовательно, параметры модели необходимо переобучать или корректировать при увеличении длины двоичного кодирования для достижения лучшей производительности. Вопросы оптимизации вероятностного кодирования рассматривались в контексте байесовских классификаторов в нашей работе [170], что повлияло на выбор распределения Бернулли для стохастической бинаризации. Тот факт, что алгоритму необходимо переобучать все данные, делает его недостаточно эффективным при изменении реальных требований. Между тем, из-за наличия знаковой функции существующие алгоритмы неизбежно вводят потери квантования. Чтобы избежать прямой оптимизации ошибки квантования, в этом разделе процесс бинаризации рассматривается с вероятностной точки зрения, и предлагается алгоритм хеширования, основанный на дискретизации распределения Бернулли.

Для заданного D -мерного бинарного вектора $b \in \{-1;1\}^D$ предполагается, что b представляет собой единственное наблюдение из D возможных распределений Бернулли $V = \{p_1, \dots, p_D\}$, то есть $b \in V$. Здесь значение b_k в k -м измерении принимает 1 с вероятностью p_k и -1 с вероятностью $1-p_k$. Алгоритм обучения выражает каждый образец как набор распределений Бернулли, как указано выше, путем обучения параметризованного отображения, соответствующее бинарное представление которого b может быть получено путем выборки из соответствующего распределения. На основе этого предположения можно получить вероятность того, что k -е бинарное

кодирование, соответствующее образцам x_i и x_j , принимает одно и то же значение:

$$\begin{aligned} P(b_{ik} = b_{jk}) &= P(b_{ik} = b_{jk} = 1) + P(b_{ik} = b_{jk} = -1) \\ &= p_{ik}p_{jk} + (1 - p_{ik})(1 - p_{jk}) = 2p_{ik}p_{jk} - p_{ik} - p_{jk} + 1. \end{aligned} \quad (4.22)$$

Исходя из этого, можно вычислить математическое ожидание b_{ik}

$$\begin{aligned} E[b_{ik} \cdot b_{jk}] &= (+1) \times P(b_{ik} = b_{jk}) + (-1) \times P(b_{ik} \neq b_{jk}) \\ &= 4p_{ik}p_{jk} - 2p_{ik} - 2p_{jk} + 1 = 2P(b_{ik} = b_{jk}) - 1. \end{aligned} \quad (4.23)$$

Значение $b_{ik} \cdot b_{jk}$ в приведенном выше уравнении отражает вероятность того, что b_{ik} и b_{jk} равны. Аналогично, математическое ожидание скалярного произведения бинарных векторов равно:

$$\begin{aligned} E[b_i \cdot b_j] &= E\left[\sum_{k=1}^D b_{ik} \cdot b_{jk}\right] \\ &= \sum_{k=1}^D E[b_{ik} \cdot b_{jk}] = \sum_{k=1}^D (4p_{ik}p_{jk} - 2p_{ik} - 2p_{jk}) + D. \end{aligned} \quad (4.24)$$

Это математическое ожидание соответствует вероятности того, что два бинарных вектора одинаковы. Для взвешенного случая существует следующая расширенная форма:

$$\begin{aligned} E[b_i^T \Lambda b_j] &= E\left[\sum_{k=1}^D \lambda_k b_{ik} b_{jk}\right] = \sum_{k=1}^D \lambda_k E[b_{ik} b_{jk}] \\ &= \sum_{k=1}^D \lambda_k P(b_{ik} = b_{jk}) - \lambda_k P(b_{ik} \neq b_{jk}) \\ &= \sum_{k=1}^D 2\lambda_k (2p_{ik}p_{jk} - p_{ik} - p_{jk}) + \text{tr}(\Lambda). \end{aligned} \quad (4.25)$$

Здесь $\text{tr}(\Lambda)$ – след диагональной матрицы.

Предположим, что выборка соответствует распределению Бернулли при $p \in \mathcal{R}^D$. Связь между уравнениями (4.23) и (4.25) указывает на то, что распределение вероятностей может быть непосредственно использовано в

качестве представления характеристик элемента. Пусть h – расширенный вектор распределения вероятностей:

$$h = [p, 1 - p] \in \mathcal{R}^{2D}. \quad (4.26)$$

Скалярное произведение выборок, основанных на h , равно:

$$h_i \cdot h_j = p_i^T p_j + (1 - p_i)^T (1 - p_j) = \frac{1}{2} (E[b_i + b_j] + D). \quad (4.27)$$

Аналогично, в случае с учетом весовых коэффициентов:

$$h_i^T \Lambda h_j = p_i^T \Lambda p_j + (1 - p_i)^T \Lambda (1 - p_j) = \frac{1}{2} (E[b_i^T \Lambda b_j] + \text{tr}(\Lambda)). \quad (4.28)$$

Приведенное выше уравнение показывает, что $(b_i \cdot b_j + D)/2$ и $(b_i^T \Lambda b_j + \text{tr}(\Lambda))/2$ являются несмещенной оценкой относительно $h_i \cdot h_j$, а также $h_i^T \Lambda h_j$, соответственно. Следовательно, можно оптимизировать целевую задачу непосредственно на основе представления признаков h , а затем сэмплировать образцы на основе h для получения соответствующего бинарного вектора b . В этом случае нет необходимости оптимизировать функцию потерь квантования. Существующие алгоритмы обучения представлению признаков могут быть непосредственно внедрены в структуру модели для повышения производительности алгоритма.

Кроме того, методы хеширования на основе выборки могут повысить производительность за счет многократного выбора каждого потенциального распределения Бернулли p_k для получения нескольких бинарных представлений. Такой подход устраняет необходимость переобучения модели для получения более длинных бинарных векторов. В частности, имея распределение Бернулли $p = \{p_1, \dots, p_D\}$, соответствующее выборке, отбор каждого распределения n_s раз дает следующий расширенный вектор:

$$\hat{b} = [b_1, \dots, b_{n_s}] \in \{-1, 1\}^{n_s D}. \quad (4.29)$$

Интуитивно понятно, что когда n_s достаточно велико, сходство, полученное на основе расширенного бинарного вектора \hat{b} , может быть восстановлено с небольшой погрешностью путем восстановления сходства, вычисленного на основе непрерывного вектора h .

Для получения соответствующего распределения Бернулли из выборки, определим для выборки x следующее отображение, позволяющее напрямую сопоставить выборку с потенциальным распределением Бернулли $V = p_1, \dots, p_n$:

$$h = [p_1(x), \dots, p_D(x), 1 - p_1(x), \dots, 1 - p_D(x)] \in \mathcal{R}^{2D}. \quad (4.30)$$

Здесь h_k и h_{k+D} обозначают вероятность того, что первые k бит двоичного вектора b примут значения -1 и 1 соответственно, и задается функцией Softmax для нормализации соответствующих признаков.

$$(h_k + h_{D+k}) = \text{Soft max}(v_k / \tau, v_{k+D} / \tau), k \leq D. \quad (4.31)$$

Здесь τ используется для управления степенью концентрации распределения вероятностей. По мере увеличения значения τ , потери квантования уменьшаются. Помимо стохастической бинаризации, аналогичные существующие алгоритмы могут использоваться для детерминированной бинаризации. Для выборки x_i , если $p_{ik} > 0.5$, то $b_{ik} = 1$, иначе $b_{ik} = -1$. Производительность бинаризации фиксирована, поскольку модель определена, в то время как производительность стохастической бинаризации может быть скорректирована путем регулирования количества n_s в соответствии с различными потребностями.

Алгоритм хеширования на основе функции активации является частным случаем алгоритма хеширования на основе распределения Бернулли. Поскольку выходные данные, полученные с помощью алгоритма ScaleTanH, находятся в диапазоне $[-1, 1]$, их можно легко преобразовать, чтобы рассматривать каждый бит как распределение Бернулли.

Таким образом, алгоритм хеширования на основе ScaleTanH обеспечивает сходимость распределения Бернулли к детерминированному вероятностному распределению путем управления значением β_t , и, следовательно, может рассматриваться как алгоритм распределения Бернулли в сочетании с алгоритмом затухающего огня. В то же время, метод детерминированной бинаризации, используемый ScaleTanH, при стремления β_t к бесконечности, становится эквивалентен детерминированной бинаризации, и ошибка квантования стремится к нулю. Однако не обязательно существует сильная корреляция между малой ошибкой квантования и хорошей производительностью. Увеличение ошибки квантования усложняет оптимизацию модели. При этом алгоритм на основе Бернулли обеспечивает более щадящий процесс оптимизации.

4.4 Целевая функция и мультимодальность

Имея пару положительных и отрицательных образцов, модель должна правильно предсказать положительные образцы, поставив их выше отрицательных. Обучающий набор для модели состоит из следующих пар положительных и отрицательных образцов:

$$p = \{(u, i, j) \mid r_{u, o_i} - r_{u, o_j}\}. \quad (4.32)$$

Здесь событие (u, i, j) обозначает, что пользователь u предпочитает комбинацию O_i комбинации O_j . В данном случае в качестве целевой функции используется байесовская функция потерь персонализированного ранжирования (BPR). Конкретная форма функции потерь BPR выглядит следующим образом:

$$L_{\text{rank}} = \sum_{(u, i, j) \in P} \log(1 + \exp(-(r_{u, o_i} - r_{u, o_j}))). \quad (4.33)$$

Оценка Γ_{u,o_i} вычисляется на основе непрерывного представления признаков h , что позволяет оптимизировать весь алгоритм от начала до конца. На основе получения представлений признаков элемента и пользователя, алгоритм выборки преобразует их в соответствующие бинарные векторы.

Мультимодальность при обучении

Помимо информации об изображении, предметы одежды зачастую сопровождаются текстовыми описаниями. Эти текстовые описания предоставляют семантическую информацию, которую можно использовать для обучения. Кроме того, предыдущие работы показали, что эта семантическая информация может значительно улучшить производительность сети. Предполагая, что v и f являются соответственно признаками на основе изображения и текста, сходство между двумя модальными данными может быть выражено как скалярное произведение, т.е.

$$s(v, f) = v \cdot f. \quad (4.34)$$

Выравнивание признаков представлений из разных модальностей осуществляется путем введения следующих визуально-семантических ограничений:

$$\mathcal{L} = \sum_{v,k} \max\{0, c - v \cdot f + v \cdot f_k\} + \sum_{f,k} \max\{0, c - f \cdot v + f \cdot v_k\}. \quad (4.35)$$

Здесь (v, f) – визуальная и семантическая информация одного и того же элемента, а (v, f_k) и (f, v_k) – мультимодальная информация разных элементов. Модель повышает производительность алгоритмов рекомендаций за счет выравнивания мультимодальных представлений одного и того же элемента. В отличие от предыдущего подхода, использующего только мультимодальность в качестве ограничения модели, алгоритм в этой главе дополнительно повышает производительность алгоритма за счет вычисления оценок на основе различных модальностей.

Псевдокод алгоритма MСНМ-Net выглядит следующим образом (Алгоритм 4.1):

Алгоритм 4.1. Процесс обучения MCHM-Net.

Инициализация

Загрузить кодировщики: извлечение признаков с помощью сверточной нейросети (CNN-энкодер), N-классовый кодировщик, кодировщик пользователей

Выберите стратегию хеширования: $method \in \{ScaleTanH, Bernoulli\}$

Инициализация обучаемых параметров: Λ_1 (веса пользователь-товар), Λ_2 (веса товар-товар)

Этап обучения (в офлайн-режиме)

для $(u, pos_outfit, neg_outfit)$ в Training data:

1. Извлечение признаков

$$V_pos = CNN(pos_outfit.images)$$

$$V_neg = CNN(neg_outfit.images)$$

2. HashEncode(Выбрать ScaleTanH или Bernoulli)

$$H_pos = HashEncode(V_pos, method, is_training=True)$$

$$H_neg = HashEncode(V_neg, method, is_training=True)$$

$$U = HashEncode(UserOneHot(u), method, is_training=True)$$

3. Рассчитать оценку совместимости

$$score_pos = \alpha \cdot Compat(H_pos, U, \Lambda_1) + Pairwise(H_pos, \Lambda_2)$$

$$score_neg = \alpha \cdot Compat(H_neg, U, \Lambda_1) + Pairwise(H_neg, \Lambda_2)$$

4. Оптимизация

$$loss = BPR(score_pos, score_neg)$$

Обновление обратного распространения (CNN, encoder, Λ_1 ,

Λ_2)

Этап вывода (онлайн-режим)

1. Предобработка: Бинаризация базы данных товаров

$$Database = \{item.id: HashEncode(CNN(item.image), method, False)$$

для item из candidate pool}

2. Рекомендация в реальном времени

функция `Recommend(user_id, candidates)`:

```
u_code = HashEncode(UserOneHot(user_id), method, False)
```

```
scores = []
```

для outfit из candidates:

```
codes = [Database[id] для id из outfit]
```

Быстрый поиск в таблице и расчет совместимости..

```
score = FastWeightedHamming(codes, u_code,  $\Lambda_1$ ,  $\Lambda_2$ )
```

```
scores.append(score)
```

```
возвратить TopK(candidates, scores)
```

4.5 Экспериментальная оценка и анализ модели

Набор данных Polyvore-U

Набор данных с веб-сайта Polyvore широко используется в исследованиях, связанных с рекомендациями одежды. Среди наборов данных, предоставленных в существующих работах, некоторые наборы данных не могут быть использованы для решения задачи персонализированных рекомендаций одежды, поскольку они не содержат информации о пользователе, а некоторые наборы данных содержат недостаточный объем данных, хотя и содержат информацию о пользователе. По этой причине в данной работе собраны новые наборы данных, содержащие информацию о пользователе, с веб-сайта Polyvore. После обработки здесь остаются только три наиболее важных типа предметов в наборе: верхняя одежда, нижняя одежда и обувь. Большинство комбинаций (костюмов) в собранном наборе данных содержат только один или два предмета верхней одежды, один предмет нижней одежды и одну пару обуви. Поэтому здесь рассматриваются два разных сценария: первый сценарий – комбинация содержит только три предмета: один предмет верхней одежды, один предмет нижней одежды и одну пару обуви, а второй сценарий – комбинация содержит один предмет

нижней одежды и одну пару обуви, но может содержать один или два предмета верхней одежды.

В первом случае путем фильтрации было получено 683 пользователя, и, разделив их на две части, мы получили Polyvore-630 и Polyvore-53, тогда как во втором случае был получен 551 пользователь, и, снова разделив их, мы получили Polyvore-519 и Polyvore-32. Для всех подмножеств данных было обеспечено отсутствие пересечения элементов в тестовом и обучающем наборах. Новый набор данных называется Polyvore-U, где U обозначает количество пользователей в соответствующем подмножестве данных, а соответствующие статистические данные представлены в таблице 4.1. Стоит отметить, что Polyvore-630/5 и Polyvore-519/32 представляют собой два разных разделения, поэтому между этими подмножествами данных существует перекрытие пользователей. В большинстве последующих экспериментов для проверки алгоритма используются два больших набора данных: Polyvore-630 и Polyvore-519, а наборы Polyvore-53 и Polyvore-32 предназначены для проверки способности модели обобщать результаты на новые сценарии использования.

Таблица 4.1 – Сведения о наборах данных Polyvore-U

	Число комбинаций		Число элементов	
	Обучающее множество	Тестовое множество	Обучающее множество	Тестовое множество
P-630	127326	23054	159729	45505
P-53	10712	1944	20230	4437
P-519	83416	14654	146475	39085
P-32	5133	898	14594	2797

Поскольку набор данных содержит только положительные примеры, то есть пользователи указывали только те комбинации наборов, которые соответствовали их задачам, а не те, которые, по их мнению, не соответствовали, необходимо случайным образом генерировать отрицательные примеры для тестирования из существующего набора данных.

Для рационализации генерации отрицательных примеров здесь используется статистическая информация из положительных примеров. В частности, при наличии положительного примера генерация соответствующих отрицательных примеров сохраняет согласованность по количеству и категории. В частности, это эквивалентно замене каждого элемента в положительном примере случайным выбором элемента из той же категории для получения отрицательного примера. В рекомендательных системах, содержащих только положительные примеры, распространенным предположением является то, что неизвестные наборы считаются менее подходящими, чем наборы, предоставленные пользователем.

Помимо случайно сгенерированных отрицательных примеров, возможен и другой тип отрицательных примеров, а именно пары наборов от других пользователей. Этот тип отрицательных примеров сложнее предсказать в процессе прогнозирования, поскольку методы рекомендаций одежды, не учитывающие персонализацию, не могут различать эти примеры. Для того, чтобы объективно сравнить различные алгоритмы, основным критерием является случайно сгенерированная простая отрицательная выборка, при этом соотношение количества сгенерированных отрицательных и положительных выборок составляет 10:1.

Условия эксперимента

Для экспериментов были рассмотрены следующие базовые алгоритмы: siamesenet, Bi-LSTM, SCE-Net, FPITF, NGNN и outfitnet. Среди них siamesenet использует сети-близнецы для изучения корреляции между парами элементов, где сеть изучает признаки, сближая представления похожих элементов и удаляя представления между непохожими элементами. На основе этой сети оценка набора определяется как среднее значение сходства всех пар элементов в наборе. Bi-LSTM рассматривает набор как последовательность, а элементы – как задачу прогнозирования. Совместимость всего набора

оценивается путем вычисления вероятности появления всех элементов. Type-aware разлагает совместимость всего набора на совместимость между парами элементов. Он также использует информацию о категориях для сопоставления разных элементов с разными подпространствами для обучения метрике.

Различные подпространства для обучения метрике. SCE-Net, с другой стороны, выполняет метрическое обучение путем изучения нескольких условных вложений элементов. Вес каждого условного вложения вычисляется с помощью механизма внимания. FITF использует вручную извлеченные признаки в качестве представления каждого элемента. Алгоритм использует модель попарного разложения на основе тензоров для разложения общего показателя совместимости на отношения два на два между пользователями и элементами, а также между элементами. NGNN строит реляционный граф на основе категорий элементов для повышения производительности алгоритма. Веса ребер между узлами в графе категорий указывают на величину вероятности того, что две категории встречаются вместе. Outfitnet использует механизм внимания для изучения различных весов, используемых для каждого элемента, чтобы получить общую совместимость.

Эксперименты в этом разделе сосредоточены на двух различных задачах рекомендаций. Первая задача – рекомендация одежды, которая включает сортировку комбинаций в соответствии с совместимостью для каждого пользователя. Качество ранжирования комбинаций (костюмов) оценивается по площади под кривой ROC (AUC) и нормализованному дисконтированному кумулятивному выигрышу (NDCG) характеристик работы субъекта.

Вторая задача – задача завершения комбинации (заполнение пропусков, FITB). Задача FITB заключается в выборе наиболее подходящего элемента из набора элементов-кандидатов заданной категории, при условии неполной комбинации (костюма). Здесь, ссылаясь на предыдущие работы, количество

элементов-кандидатов для задачи завершения набора установлено равным 4.

В экспериментах в этом разделе в качестве базовой сети для обучения и извлечения признаков на основе изображений используется alexnet. Чтобы иметь возможность обрабатывать изображения разных размеров, вместо всех полносвязных слоев в alexnet используется сверточный слой, а также добавлен слой усредняющего пулинга для получения 4096-мерного вектора признаков. Для текстовых данных текстовые признаки получаются из описаний элементов и меток с использованием существующей модели последовательности, где размерность текстовых признаков составляет 2400.

Результаты вычислительных экспериментов

Чтобы минимизировать ошибку, вызванную случайно сгенерированными отрицательными образцами в результатах, каждый алгоритм тестировался 10 раз, и в качестве окончательного результата использовалось среднее значение этих 10 тестов.

Каждый из этих тестов повторно генерировал набор отрицательных образцов, используемых для сравнения. Для экспериментов по точности рекомендаций алгоритмов использовались четыре набора данных: Polyvore-519, Polyvore-630, Polyvore-180 и IQON.

Наборы данных Polyvore-519 и Polyvore-630 представляют собой наборы данных, собранные в настоящем исследовании, содержащие информацию о пользователях, а Polyvore-180 и IQON взяты из разных существующих работ, соответственно.

Сначала были проведены эксперименты по тестированию различных алгоритмов на Polyvore-519 и Polyvore-630, а конкретная производительность различных алгоритмов приведена в таблице.

Таблица 4.2 – Производительность на данных Polyvore-U

	Polyvore-630			Polyvore-519		
	FITB	AUC	NDCG	FITB	AUC	NDCG
Siamesenet	0.5501	0.8123	0.6623	0.5621	0.8315	0.6736
Bi-LSTM	0.5610	0.8211	0.6943	0.5977	0.8569	0.7153
Type-Aware	0.5576	0.8166	0.6756	0.5600	0.8234	0.6801
SCE-Net	0.5858	0.8400	0.7161	0.5763	0.8348	0.6951
MCHM-Net -T(scaletanh)	0.5620	0.8686	0.7786	0.5450	0.8666	0.7711
MCHM-Net -T (Deterministic)	0.5718	0.8739	0.7903	0.5655	0.8773	0.7929
MCHM-Net -T ($n_s=3$)	0.5729	0.8747	0.7936	0.5674	0.8793	0.7969
MCHM-Net -V (scaletanh)	0.6145	0.8979	0.8178	0.6027	0.8974	0.8106
MCHM-NET-V (Deterministic)	0.6172	0.8992	0.8199	0.6145	0.9032	0.8205
MCHM-NET-V ($n_s=3$)	0.6263	0.9036	0.8294	0.6191	0.9068	0.8272
MCHM-NET-VSE	0.6628	0.9237	0.8674	0.6575	0.9282	0.8722

Таблица 4.3 – Производительность на данных IQON и Polyvore-180

Methods	IQON			Polyvore-180		
	FITB	AUC	NDCG	FITB	AUC	NDCG
Siamesenet	0.4778	0.8054	0.6532	0.5163	0.7921	0.6219
Bi-LSTM	0.5139	0.8127	0.6650	0.5269	0.7871	0.6267
Type-Aware	0.4581	0.7910	0.6301	0.5071	0.7776	0.6030
SCE-Net	0.4877	0.8232	0.6820	0.5148	0.7921	0.6328
NGNN	<u>0.4933</u>	<u>0.8636</u>	<u>0.7615</u>	0.4806	0.7753	0.5958
Outfitnet	0.4291	0.8308	0.6956	<u>0.5451</u>	<u>0.8841</u>	<u>0.7798</u>
MCHM-Net (scaletanh)	0.5404	0.8961	0.8018	0.5700	0.8907	0.7827
MCHM-Net (Deterministic)	0.5582	0.9115	0.8299	0.6187	0.9154	0.8407
MCHM-Net ($n_s=3$)	0.5637	0.9149	0.8374	0.6220	0.9156	0.8426

В таблице «Т» означает, что модель использует только текстовые признаки, «V» означает, что модель использует только признаки изображений, а «VSE» означает, что модель использует информацию из обеих модальностей. Сравнение производительности двух алгоритмов хеширования, предложенных в этой главе, а именно алгоритма хеширования на основе функции активации scaletanh и алгоритма хеширования на основе Бернулли, приведено отдельно для моделей, использующих данные одной модальности. Для алгоритма хеширования на основе Бернулли производительность снова

приведена отдельно для двух подходов к бинаризации: первый – детерминированная бинаризация, а второй – путем выборки каждого распределения Бернулли. Для мультимодальных данных показан только метод детерминированной бинаризации на основе Бернулли с лучшей производительностью, а также приведена производительность без алгоритма взвешенного хеширования. В качестве базового алгоритма в качестве сети извлечения признаков для данных изображений использовалась сеть resnet-18. По результатам сравнения можно выделить четыре основных вывода:

1) Предложенная в этой главе хеш-модель MCHM-Net-VSE значительно превосходит другие базовые алгоритмы по всем метрикам. Модель MCHM-Net-V по-прежнему превосходит другие методы, использующие мультимодальную информацию, даже при использовании только информации об изображениях. Это доказывает эффективность предложенного в этой главе алгоритма персонализированных рекомендаций на основе хеширования. Это показывает, что данный алгоритм достигает высокого уровня точности рекомендаций, одновременно повышая вычислительную эффективность за счет технологии хеширования;

2) Мультимодальная информация способствует повышению производительности алгоритма. Сравнивая производительность двух модальностей, можно обнаружить, что информация об изображениях более полезна, чем текстовая информация в этой задаче. Основная причина этого заключается в том, что сопоставление одежды больше ориентировано на визуальное единство, а текстовая информация о каждом предмете ограничена. Эти две причины одновременно позволяют модели, основанной на информации об изображениях, достигать лучших результатов. Объединение текстовой и графической информации еще больше повышает производительность алгоритма;

3) Взвешенное хеширование обеспечивает лучшую производительность. Сравнивая результаты вычисления сходства с использованием уравнений

(4.16) и (4.4) в качестве бинарных векторов соответственно, можно обнаружить, что результаты с использованием взвешенного хеширования значительно превосходят результаты без взвешенного хеширования;

4) Превосходство алгоритмов хеширования на основе вероятностной выборки. Хотя достаточно малая ошибка квантования может быть получена с помощью `scaletanh` для аппроксимации функции знака, градиент неэффективен для обновления весов модели, поскольку с увеличением p это приводит к тому, что выходные данные приближаются к области насыщения функции активации `scaletanh`. Кроме того, не обязательно существует связь между малой ошибкой квантования и хорошей производительностью. Экспериментальные результаты показывают, что предложенный в этой главе метод на основе выборки Бернулли может дополнительно улучшить производительность алгоритма.

Для дальнейшего сравнения производительности модель, представленная в этой главе, сравнивается с другими базовыми моделями с использованием предварительно обученных признаков изображений. Здесь используются наборы данных Polyvore-180 и IQON. Результаты экспериментов показаны в таблице 4-3. Для справедливого сравнения алгоритм MCHM-Net, предложенный в этой главе, также использует предварительно обученные признаки в качестве входных данных. Видно, что предложенный в этой главе алгоритм по-прежнему демонстрирует наилучшие результаты на этих данных. Кроме того, в экспериментах сравнивается производительность алгоритма MCHM-Net с производительностью алгоритма FPITF, не использующего глубокое обучение. Для справедливого сравнения используется $D = 30$, и используются исходные отрицательные образцы. NDCG для FPITF составляет 0,5576, тогда как NDCG для MCHM-Net составляет 0,6231, что представляет собой улучшение производительности примерно на 12% по сравнению с FPITF.

Анализ модели

Дополнительно проанализируем производительность алгоритма MСНМ-Net путем проведения подробных вспомогательных экспериментов на модели и визуализации результатов.

Таблица 4.4 – Доля выигрышных результатов (%) алгоритмов, описанных в этой главе, относительно других базовых алгоритмов

Даборы данных	Siamesenet	Bi-LSTM	Type-Aware	SCE-Net
Polyvore-630	99.2	98.6	98.9	97.3
Polyvore-519	96.7	97.1	97.3	97.3

Моделирование пользователей необходимо, поскольку сопоставление комбинаций субъективно, и разные пользователи могут давать диаметрально противоположные оценки одному и тому же сопоставлению. Чтобы показать важность информации о пользователях для модели, в эксперименте вычисляется AUC для каждого пользователя, а также подсчитывается доля пользователей, для которых алгоритм MСНМ-Net работает лучше, чем другие алгоритмы, как показано в таблице 4.4. Видно, что алгоритм MСНМ-Net работает лучше как минимум для 96,7% пользователей. Приведенные выше результаты показывают, что учет предпочтений пользователей важен для повышения точности рекомендаций. По сравнению с другими алгоритмами, алгоритм MСНМ-Net, предложенный в этой главе, значительно превосходит их.

В экспериментах в этой главе в качестве отрицательных примеров использовались случайно сгенерированные наборы, а сложной задачей было измерение производительности алгоритма с использованием комбинаций наборов других пользователей в качестве отрицательных примеров. Эта схема дополнительно иллюстрирует важность персонализированного моделирования, поскольку без моделирования пользователя было бы невозможно справиться с этой ситуацией. Результаты, полученные на основе сложных отрицательных примеров, представлены в таблице 4.5.

MCHM-Net-R обозначает модель, которая не включает сложные отрицательные примеры в процессе обучения, а MCHM-Net-H – модель, которая обучается с использованием сложных отрицательных примеров. Поскольку базовый метод не учитывает информацию о пользователе, его производительность эквивалентна случайному угадыванию. MCHM-Net, с другой стороны, позволяет распознавать такие сложные отрицательные примеры, поскольку учитывает информацию о пользователе. Кроме того, производительность алгоритма может быть дополнительно улучшена за счет включения сложных отрицательных примеров в процесс обучения.

Таблица 4.5 – Производительность алгоритма на сложных отрицательных образцах

Алгоритмы	Polyvore-630-H		Polyvore-519-H	
	AUC	NDCG	AUC	NDCG
Siamesenet	0.4993	0.2808	0.4997	0.2731
Bi-LSTM	0.4992	0.2817	0.4990	0.2739
Type-Aware	0.5000	0.2790	0.4995	0.2740
SCE-Net	0.4994	0.2810	0.4996	0.2738
MCHM-Net-R	0.7654	0.5552	0.7550	0.5369
MCHM-Net-H	0.8440	0.6869	0.8361	0.6685

Таблица 4.6 – Нормализованные положения исходных элементов

Категория элемента	Siamese	Bi-LSTM	Type-Aware	SCE-Net	MCHM-Net
Верх	26.91	22.38	25.49	23.28	17.88
Низ	26.81	23.71	26.76	24.16	17.83
Обувь	25.88	22.50	24.01	23.07	17.33

Рекомендация товаров является важным приложением. В задаче рекомендации товаров на основе множеств обычно задается несколько товаров для формирования множества, и для завершения множества (комбинации) выбирается наиболее подходящий товар из определенной категории. Из-за большого количества товаров сложно ранжировать исходные товары в топ-10, поэтому для количественного сравнения производительности алгоритмов в качестве индикатора сравнения между различными

алгоритмами используется нормализованное положение исходных товаров в сортировке. В таблице 4.6 показаны результаты сравнения, где меньшее значение (меньшая доля, в процентах) означает, что исходные товары занимают первые места в рейтинге, и, следовательно, производительность алгоритма лучше. Видно, что алгоритм MCHM-Net, предложенный в этой главе, значительно превосходит другие базовые алгоритмы.

Таблица 4.7 – Эффективность различных алгоритмов в задачах для новых пользователей.

Алгоритмы	Polyvore-53		Polyvore-32	
	AUC	NDCG	AUC	NDCG
Siamesenet	0.7802	0.5818	0.7891	0.5894
Bi-LSTM	0.8253	0.6740	0.8525	0.7235
Type-Aware	0.8015	0.6367	0.8030	0.6512
SCE-Net	0.8207	0.6570	0.8125	0.6692
MCHM-Net	0.9211	0.8688	0.9233	0.8608

Холодный старт – распространённая и важная проблема в рекомендательных системах. Необходимо анализировать новых пользователей, когда в систему постоянно добавляются новые пользователи. В реальной системе количество новых пользователей, добавляемых ежедневно, огромно, и переобучение всей сети чрезвычайно ресурсоёмко и нецелесообразно. Поскольку каждый пользователь используется только в качестве D-мерного параметра в алгоритме MCHM-Net, сеть признаков, связанная с элементом, уже хорошо обучена на основе существующих данных. Поэтому для новых пользователей можно обучить только бинарное представление D-мерного параметра пользователя, чтобы изучить новых пользователей. Для сравнения производительности алгоритмов в сценарии с новыми пользователями в качестве данных существующих пользователей используются данные Polyvore-630 и Polyvore-519, а в качестве данных новых пользователей – соответствующие им наборы данных Polyvore-53 и Polyvore-32. Алгоритм использует модели, обученные на данных Polyvore-630 и Polyvore-519, в

качестве основы, а Polyvore-53 и Polyvore-32 – для обучения только параметров, связанных с пользователем. Производительность алгоритма в задаче с новыми пользователями представлена в таблице 4.7. Поскольку предпочтения различаются от пользователя к пользователю, дообученная модель неизбежно покажет снижение производительности по сравнению с производительностью на Polyvore-630/519, но видно, что снижение производительности алгоритма MСНМ-Net является приемлемым. При увеличении размера набора данных примерно на 7% предложенный в этой главе метод может достичь производительности, сопоставимой с исходным алгоритмом, за счет дообучения только представления новых пользователей.

Таблица 4.8 – Производительность с различными слагаемыми оценки

Алгоритм	Polyvore-630			Polyvore-519		
	FITB	AUC	NDCG	FITB	AUC	NDCG
Training based on $r(i)$	0.62	0.86	0.75	0.61	0.86	0.75
Testing based on $r(i)$	0.60	0.85	0.73	0.60	0.85	0.74
Training based on $r(u)$	0.56	0.88	0.80	0.55	0.89	0.80
Testing based on $r(u)$	0.55	0.87	0.78	0.54	0.88	0.79
$R(u) + r(i)$	0.66	0.92	0.87	0.66	0.93	0.87

В уравнении (4.18) оценка персонализации пользователя для набора состоит из двух слагаемых, $r^{(u)}$ и $r^{(i)}$, где $r^{(u)}$ отражает предпочтения пользователя в отношении персонализации различных элементов, а $r^{(i)}$ отражает отношение совместимости между элементами. Для сравнения эффективности использования различных слагаемых рассматриваются два разных сценария. Первый сценарий заключается в тестировании $r^{(u)}$ и $r^{(i)}$ по отдельности на основе полностью обученной модели и измерении эффективности использования различных элементов в общей модели. Вторым сценарием является обучение модели на основе $r^{(u)}$ и $r^{(i)}$ по отдельности и исследовании верхнего предела производительности, который может быть достигнут различными оценочными элементами. Результаты производительности для различных настроек представлены в таблице 4.8.

Во-первых, для каждого элемента $r^{(u)}$ и $r^{(i)}$ производительность обучения непосредственно на основе соответствующего элемента лучше, чем производительность тестирования на основе каждого элемента, но улучшение ограничено. Это указывает на то, что полная модель, основанная на оценке, хорошо обучается на каждом элементе. В задаче FITB $r^{(i)}$ более полезен, чем $r^{(u)}$. Это связано с тем, что в задаче рекомендации товаров необходимо уделять больше внимания корреляции между различными товарами. В задаче рекомендации наборов $r^{(u)}$ важнее, чем $r^{(i)}$, поскольку персонализированные предпочтения играют более важную роль. Это показывает, что оба параметра $r^{(u)}$ и $r^{(i)}$ незаменимы, а их комбинация еще больше улучшает производительность.

Проведем сравнение различных методов хеширования: В этой главе предлагаются два различных метода обучения хешированию, а именно алгоритм хеширования, основанный на функции активации ScaleTanH, и алгоритм хеширования, основанный на распределении Бернулли. Алгоритм хеширования на основе распределения Бернулли позволяет получать бинарные векторные представления различной длины, многократно используя различные распределения Бернулли. В этом разделе сначала подробно анализируется взаимосвязь между количеством выборок и производительностью, а результаты показаны на рисунке 4.3. На рисунке отдельно показана производительность для различных типов бинаризации. Когда случайный подход к бинаризации недостаточен для малых значений n_s , алгоритм может быть дополнен гибридным подходом к бинаризации.

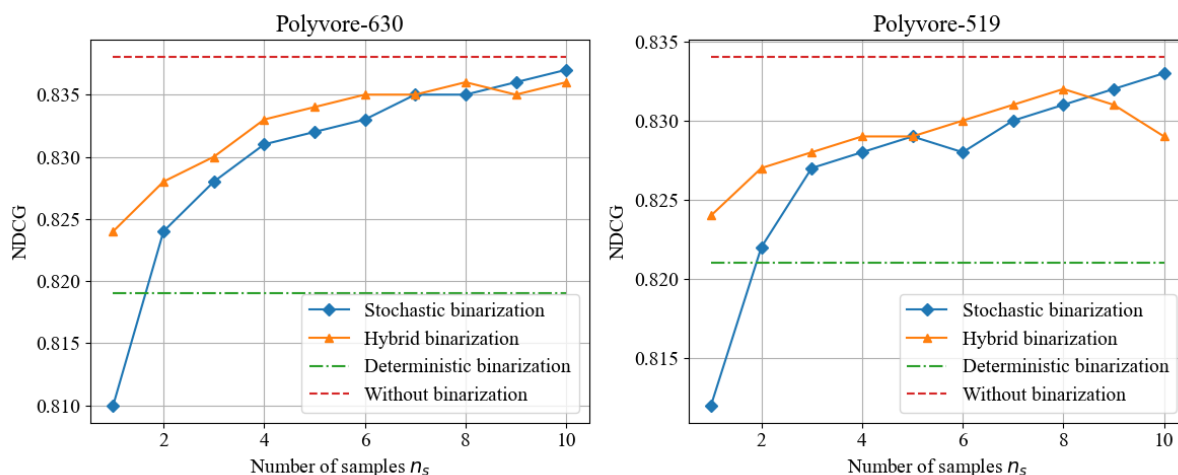


Рисунок 4.3 – Сравнение размера выборки (n_s) и показателей эффективности (NDCG), Stochastic binarization – стохастическая бинаризация, Hybrid binarization – гибридная бинаризация, Deterministic binarization – детерминированная бинаризация, Without binarization – без бинаризации.

Кроме того, алгоритм хеширования, основанный на функции активации ScaleTanH, можно рассматривать как частную форму алгоритма хеширования на основе распределения Бернулли в условиях имитации отжига, при этом алгоритм хеширования на основе распределения Бернулли регулирует плотность распределения вероятностей путем фиксации τ . Размер τ влияет на потери квантования, и чем меньше τ , тем ниже потери квантования, хотя не обязательно существует положительная корреляция между потерями квантования и τ . Это иллюстрируется здесь сравнением производительности при различных τ . Из таблицы 4.9 видно, что $\tau = 0,5$ обеспечивает оптимальную производительность. В таблице β_t обозначает алгоритм хеширования на основе функции активации, который, напротив, имеет небольшую ошибку квантования, но субоптимальную производительность. Основная причина этого явления заключается в том, что при малой ошибке квантования модель не может обеспечить эффективный градиент для обновления параметров модели, что приводит к преждевременной сходимости модели и, следовательно, к невозможности дальнейшего улучшения ее производительности.

Таблица 4.9 – Взаимосвязь между количественной ошибкой и производительностью

T	IQON			Polyvore-180		
	AUC	NDCG	$\ h - b\ _1$	AUC	NDCG	$\ h - b\ _1$
0.1	0.8627	0.7417	0.0101	0.9118	0.8324	0.1306
0.5	0.9115	0.8299	0.1690	0.9154	0.8407	0.2021
1.0	0.9107	0.8284	0.1915	0.9114	0.8310	0.2324
2.0	0.9083	0.8233	0.2087	0.9060	0.8219	0.2612
β_t	0.8961	0.8018	0.0139	0.8907	0.7827	0.0036

Изучим взаимосвязь между длиной кодирования и производительностью. В целом, производительность улучшается с увеличением длины кодирования. Чтобы проиллюстрировать влияние длины кодирования бинарного вектора на производительность, здесь производительность алгоритма оценивается путем обучения алгоритма на основе различных длин кодирования, а именно {16, 32, 64, 128, 256}, а производительность алгоритма с различными длинами кодирования показана на рисунке 4.4. Обратите внимание, что для алгоритма хеширования на основе распределения Бернулли длину кодирования можно увеличить, увеличив n_s . Здесь, чтобы лучше представить взаимосвязь между длинами кодирования, используется $n_s=1$, т.е. дополнительная выборка не выполняется. Как видно из рисунка, улучшение производительности алгоритма при увеличении длины кодирования примерно пропорционально логарифму длины кодирования. Однако слишком малая длина кодирования приводит к значительному снижению производительности. Это связано с тем, что D -битный бинарный вектор может кодировать до $2D$ объектов без перекрытия. Когда $D = 16$, максимальное количество объектов, которые могут быть представлены бинарным вектором, составляет $2^{16} = 65536$, что значительно меньше количества объектов, уже имеющихся в наборе данных. Следовательно, точность алгоритма значительно снижается при $D = 16$.

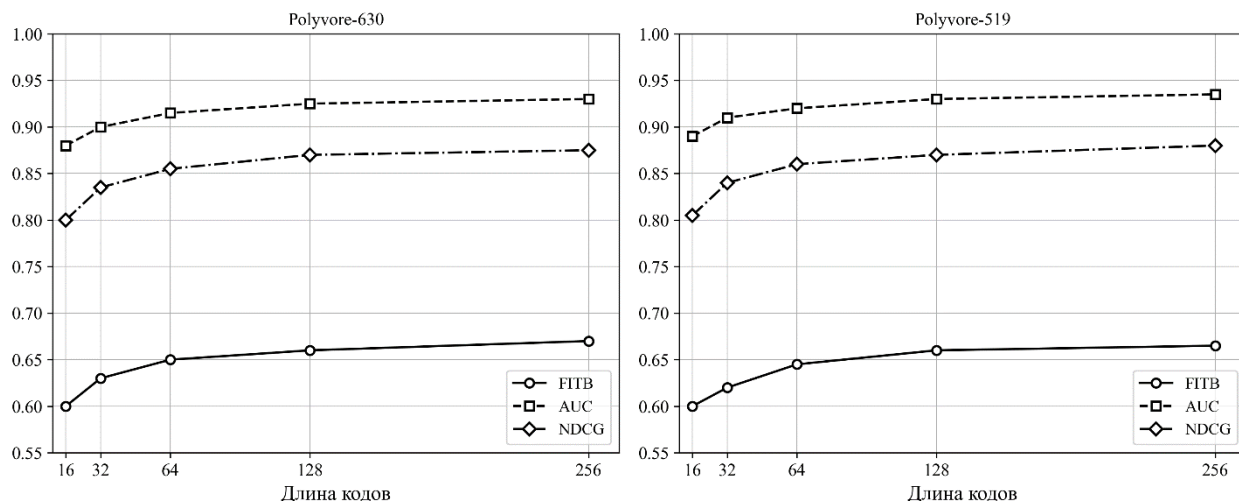


Рисунок 4.4 – взаимосвязь между длиной кодирования и производительностью

Хотя двоичное представление в некоторой степени снижает производительность алгоритма, оно обеспечивает значительное вычислительное удобство. Эффективность вычисления двоичных векторов демонстрируется здесь путем сравнения времени выполнения различных алгоритмов с точки зрения совместимости наборов. Прежде всего, расстояние Хэмминга между двоичными векторами может быть быстро вычислено с использованием лежащей в основе гетероскедастической операции и таблицы поиска. Составление таблицы поиска представляет собой постоянное слагаемое в сложности времени поиска. Поэтому здесь рассматривается только сравнение времени вычислений взвешенных двоичных векторов. Фан и др. в 2013 г. [171] предложили стратегию таблицы поиска для эффективного вычисления взвешенного расстояния Хэмминга. Имея два 8-битных двоичных кода $a, b \in \mathbb{B}^8$ и соответствующие веса $w \in \mathcal{R}^8$, мы сначала получаем все результаты $k = a \otimes b$ побитовых перестановок различных двоичных векторов, при этом очевидно, что k имеет 256 возможных значений. Как показано в таблице 4.10, взвешенное расстояние v предварительно устанавливается для каждого возможного k с помощью веса w . Таким образом, для вычисления сходства между любыми двумя бинарными кодами результат можно получить с помощью одной побитовой операции несходства и поиска в таблице, что

гораздо эффективнее, чем вычисление скалярного произведения двух векторов, поскольку сложность поиска в таблице является постоянной величиной. В D -мерном случае мы можем сначала разделить каждый бинарный код на $s = \lfloor D/8 \rfloor$ частей и построить s таблиц поиска. Итоговое взвешенное расстояние Хэмминга – это сумма всех частей, т.е. $\sum_i^s v_i$, где v_i – результат i -й таблицы поиска. Поскольку таблицу поиска нужно построить только один раз, это эффективно. Более того, увеличение количества бинарных кодов вносит лишь небольшие вычислительные затраты.

Таблица 4.10 – Таблица соответствия для взвешенного расстояния Хэмминга

К	V
0	0
1	W_1
2	W_2
3	$W_1 + W_2$
· · ·	· · ·
254	$W_2 + \dots + W_8$
255	$W_1 + \dots + W_8$

Таблица 4.11 – Сравнение продолжительности воспроизведения

	МСНМ -Fast	МСНМ	Siamesenet	Bi-LSTM	Type-Aware	SCE-Net
Время	1.66 μ s	27.2 μ s	38.9 μ s	1.23ms	59.1 μ s	192 μ s
Коэффициент улучшения	-	~16.4 \times	~23.4 \times	~741.0 \times	~35.6 \times	~115.7 \times

Вычисления с использованием алгоритма быстрого поиска обозначаются как МСНМ-Fast, а сравнение времени выполнения различных алгоритмов приведено в таблице 4.11, где указано время, затраченное с использованием и без использования алгоритма быстрого поиска. Прежде всего, видно, что при простой реализации производительность на основе алгоритма быстрого поиска как минимум в 16 раз выше, чем производительность без алгоритма быстрого поиска. Кроме того, видно, что Bi-LSTM является самым медленным методом, поскольку в Bi-LSTM

необходимо сравнивать вероятности всех возможных элементов при вычислении совместимости.

4.6 Результаты главы 4

В настоящей главе предложен алгоритм персонализированных комбинаторных рекомендаций на основе хэширования. Эффективность вычислений является существенной проблемой из-за возможности большого количества комбинаций элементов в комбинаторных рекомендациях. Алгоритмы хэширования повышают эффективность вычислений, представляя элементы и пользователей в виде бинарных векторов. Существующие алгоритмы хэширования в основном ориентированы на задачу поиска изображений и недостаточно изучены в задаче комбинаторных рекомендаций. В этой главе мы используем эффективность вычислений хэширования и применяем технологию хэширования к задаче комбинаторных рекомендаций, что значительно повышает вычислительную эффективность алгоритма и обеспечивает более высокую точность рекомендаций. Предложены два различных алгоритма хэширования для обучения бинарных представлений элементов и пользователей для задачи персонализированных рекомендаций наборов в онлайн-режиме. Кроме того, в данной работе используются мультимодальные данные элементов, из которых составляются комбинации, что позволяет согласовать представление изображений элементов с семантическим пространством и дополнительно повысить точность алгоритма. Для сравнения эффективности различных алгоритмов в этой главе представлен более масштабный набор данных о персонализированных рекомендациях костюмов, а также проверена эффективность алгоритмов на основе большого количества достаточных экспериментов.

Результаты главы опубликованы в [172], зарегистрирована [173] и внедрена в эксплуатацию (Приложение А) программа для ЭВМ.

Как показывают приведенные эксперименты, разработанный в данной главе алгоритм МСНМ-Net успешно решает ключевую проблему комбинаторных рекомендаций – проблемы вычислительной эффективности – за счет применения методов хеширования. Используя бинарное кодирование и предложенный вероятностный метод обучения, алгоритм достигает времени отклика на уровне микросекунд, сохраняя при этом высокую точность персонализации. Этот алгоритм завершает логическую цепочку данного исследования: от углубленного анализа структуры графа знаний (глава 2) и обогащения графа с использованием мультимодальных признаков (глава 3) до, наконец, построения сервиса, имеющего практическое значение и высокую производительность. Таким образом, результаты глав 2 и 3 обеспечивают необходимую семантическую и структурную глубину представления, а алгоритм в главе 4 гарантирует его практическую применимость в условиях реального времени.

ЗАКЛЮЧЕНИЕ

В настоящей работе предложен ряд новых алгоритмов, направленных на повышение эффективности решения задач по дополнению графа знаний за счет снижения вычислительной сложности в мультимодальных комбинаторных пространствах за счет решения проблемы несвязности структуры графа знаний и многомодальности при обучении представлению графа знаний на основе графовых нейронных сетей. Также предложены алгоритмы для реализации системы онлайн-рекомендаций по сочетанию предметов одежды на основе графов знаний. В ходе исследования решены следующие задачи.

1. Предложен алгоритм обучения представлению графа знаний Dual-FusionKG, который объединяет структурную и семантическую информацию. Алгоритм моделирует локальную структурную информацию и троичную семантическую информацию по отдельности, используя двухканальную адаптивную сеть внимания графа (ERGAT), и максимизирует локальное и глобальное представления за счет максимизации взаимной информации. Это эффективно устраняет проблему, связанную с трудностями распространения межсущностной информации в несвязных графах. Экспериментальные результаты показывают, что Dual-FusionKG значительно превосходит существующие алгоритмы на нескольких эталонных наборах данных, что подтверждает эффективность предлагаемого алгоритма в задаче завершения графа знаний.

2. Предложен алгоритм HyperFusion-Net для динамического моделирования взаимодействий с учетом направленности гетерогенных связей мультимодальных графов знаний посредством низкорангового тензорного слияния со сложными пространственно взаимосвязанными сетями слияния сущностей и связей, что значительно повышает производительность задач дополнения графов знаний. Эксперименты показывают, что алгоритм

HyperFusion-Net улучшает показатель Hits@10 до 9,6% на четырёх тестовых наборах данных, будучи и устойчив к пропускам модальности, обеспечивая эффективное решение для сложных задач анализа знаний.

3. Предложен алгоритм МСНМ-Net, который значительно повышает эффективность и точность комбинаторной рекомендательной системы благодаря методу хеширования и достигает AUC=0,9237 при времени отклика на уровне микросекунд, используя алгоритм взвешенного хеширования и вероятностного кодирования Бернулли.

Результаты исследования, представленные в данной работе, открывают несколько направлений для дальнейших исследований. Во-первых, изучение применения методов дистилляции знаний для сжатия модели Dual-FusionKG с целью уменьшения ее пространственной сложности без существенной потери точности является практически важной темой. Во-вторых, с точки зрения расширения мультимодальных методов (HyperFusion-Net), включение дополнительных модальностей (таких как аудиоданные или временные ряды) и изучение более сложных механизмов внимания для учета семантической направленности отношений демонстрируют многообещающие перспективы. Наконец, для алгоритма МСНМ-Net ценным направлением является разработка методов «динамического» расширения длины кодирования для адаптации к изменениям пользовательских предпочтений и достижения более высокого качества рекомендаций; одновременно, применение разработанного метода хеширования к другим типам задач рекомендаций, помимо подбора одежды, также имеет широкие перспективы применения.

СПИСОК ЛИТЕРАТУРЫ

1. Bordes A., Usunier N., García-Durán A., et al. Translating embeddings for modeling multi-relational data // *Advances in Neural Information Processing Systems*. 2013. №26. P.2787–2795.
2. Wang Z., Zhang J., Feng J., et al. Knowledge graph embedding by translating on hyperplanes // *Proc. of the 28th AAAI Conference on Artificial Intelligence*. Québec City, Canada. 2014. P.1112–1119.
3. Lin Y., Liu Z., Sun M., et al. Learning entity and relation embeddings for knowledge graph completion // *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Austin, Texas, USA. 2015. P.2181–2187.
4. Sun Z., Deng Z., Nie J., et al. RotatE: Knowledge graph embedding by relational rotation in complex space // *7th International Conference on Learning Representations (ICLR)*. 2019.P.1-18.
5. Nickel M., Tresp V., Kriegel H. P. A three-way model for collective learning on multi-relational data // *Proc. of the 28th International Conference on Machine Learning (ICML)*. Bellevue, Washington, USA. 2011. P.809–816.
6. Kolda T. G., Bader B. W. Tensor decompositions and applications // *SIAM Review*. 2009. №51(3). P.455–500.
7. Trouillon T., Welbl J., Riedel S., et al. Complex embeddings for simple link prediction // *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. 2016. P.2071–2080.
8. Dettmers T., Minervini P., Stenetorp P., et al. Convolutional 2D knowledge graph embeddings // *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, USA. 2018. P.1811–1818.
9. Balazevic I., Allen C., Hospedales T. M. Hypernetwork knowledge graph embeddings // *Proc. of the 28th International Conference on Artificial Neural Networks (ICANN)*. Munich, Germany. 2019. P.553–565.
10. Schlichtkrull M. S., Kipf T. N., Bloem P., et al. Modeling relational data with graph convolutional networks // *Proceedings of the 15th International Conference on The Semantic Web (ESWC 2018)*. Heraklion, Crete, Greece. 2018. P.593–607.
11. Vashishth S., Sanyal S., Nitin V., et al. Composition-based multi-relational graph convolutional networks // *8th International Conference on Learning Representations (ICLR)*. 2020.P.1-16.
12. Yao L., Mao C., Luo Y. KG-BERT: BERT for knowledge graph completion // *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. Macao, China. 2019. P.2901–2907.

13. Wang Q., Huang P., Wang H., et al. CoKE: Contextualized knowledge graph embedding [Электронный ресурс] // CoRR. 2019. arXiv:1911.02168. (дата обращения: 30.12.2025)
14. Chen X., Zhang N., Li L., et al. Hybrid transformer with multi-level fusion for multimodal knowledge graph completion // Proc. of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. Madrid, Spain. 2022. P.904–915.
15. Sar Shalom O., Koenigstein N., Paquet U., et al. Beyond collaborative filtering: The list recommendation problem // Proceedings of the 25th International Conference on World Wide Web. 2016. P.63–72.
16. Liu G., Fu Y., Chen G., et al. Modeling buying motives for personalized product bundle recommendation // ACM Transactions on Knowledge Discovery from Data. 2017. №11(3). P.1–26.
17. Wei P., Liu S., Yang X., et al. Towards personalized bundle creative generation with contrastive non-autoregressive decoding // Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022. P.2634–2638.
18. Veit A., Belongie S., Karaletsos T. Conditional similarity networks // 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P.830–838.
19. Tangseeng P., Okatani T. Toward explainable fashion recommendation // 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). 2020. P.2153–2162.
20. Garfield E. Citation indexes for science: A new dimension in documentation through association of ideas // Science. 1955. №122(3159). P.108–111.
21. Singhal A. Introducing the knowledge graph: things, not strings // Official Google Blog [Электронный ресурс]. 2012. URL <https://blog.google/products-and-platforms/products/search/introducing-knowledge-graph-things-not> (дата обращения: 30.12.2025).
22. Shadbolt N., Berners-Lee T., Hall W. The semantic web revisited // IEEE Intelligent Systems. 2006. №21(3). P.96–101.
23. Лозовский В.С. Семантические сети // Представление знаний в человеко-машинных и робототехнических системах. 1984. С.84–121.
24. Лозовский В.С. Экстенциональная база знаний на основе семантических сетей // Известия АН СССР. Техническая кибернетика. 1982. №5. С.23–42.
25. Кораблинов В., Браславский П. RuBQ: Российский набор данных для ответов на вопросы на основе Wikidata // Труды Международной конференции по семантической сети (ISWC). Афины, Греция (виртуальное мероприятие). Springer. 2020. С. 108–124.

26. Ярошко Т., Коса В., Игнатенко О. Создание научных графов знаний из публикаций: пример использования в борьбе с коррупцией // Коммуникации в области компьютерных и информационных наук (CCIS). Шам, Швейцария. Springer. 2024. № 2083. С. 225–241.
27. Атаева О. М., Массель Л. В., Серебряков В. А. Добыча данных при построении графа знаний междисциплинарного журнала // Труды Российской конференции по вычислительной математике и информационным технологиям. Владивосток, Россия. Springer. 2024. С. 1–12.
28. Сальников М., Ле Х., Раджпут П., Никишина И. Большие языковые модели и графы знаний для ответа на фактоидные вопросы // CoRR. 2023. arXiv:2310.02166. С. 1–12.
29. Cao Q., Zhao Y. The realization process and related applications of knowledge graph // Information Theory and Practice. 2015. №38(12). P.127–132.
30. Xu Z., Sheng Y., He L., et al. Review on knowledge graph techniques // Journal of University of Electronic Science and Technology of China. 2016. №45(4). P.589–606.
31. Dong X., Gabrilovich E., Heitz G., et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion // Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA. 2014. P.601–610.
32. Liu Q., Li Y., Duan H., et al. Knowledge graph construction techniques // Journal of Computer Research and Development. 2016. №53(3). P.582–600.
33. Sun Z., Wang H. Overview of the advance of the research on named entity recognition // New Technology of Library and Information Service. 2010. №(6). P.42–47.
34. Zhao J., Liu K., Zhou G., et al. Open information extraction // Journal of Chinese Information Processing. 2011. №25(6). P.98–110.
35. Wang Y., Tan S., Liao X., et al. Extended domain model based named attribute extraction // Journal of Computer Research and Development. 2010. №47(9). P.1567–1573.
36. Liu Z., Sun M., Lin Y., et al. Advances in Knowledge Representation Learning Research // Journal of Computer Research and Development. 2016. №53(2). P.247–261.
37. Linden G., Smith B., York J. Amazon.com recommendations: item-to-item collaborative filtering // IEEE Internet Computing. 2003. P: 76-80.
38. Chang J., Gao C., He X., et al. Bundle recommendation with graph convolutional networks // Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020. P.1673–1676.
39. Ма Ч., Гао М., Цепкова М.И., Резова Н.Л., Ступина А.А. Алгоритм группирования текстовых данных с рекомбинацией по принципу химической

реакции // Системы управления и информационные технологии. 2022. №3(89). С. 61-65.

40. Xiao H., Huang M., Zhu X. TransG: A generative model for knowledge graph embedding // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL). 2016. P.2316–2325.

41. Ebisu T., Ichise R. Generalized translation-based embedding of knowledge graph // IEEE Transactions on Knowledge and Data Engineering. 2020. №32(5). P.941–951.

42. Balazevic I., Allen C., Hospedales T. M. Tucker: Tensor factorization for knowledge graph completion // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019. P.5184–5193.

43. Bordes A., Weston J., Collobert R., et al. Learning structured embeddings of knowledge bases // Proc. of the 25th AAAI Conference on Artificial Intelligence. San Francisco, California, USA. 2011. P.301–306.

44. Yang B., Yih W., He X., et al. Embedding entities and relations for learning and inference in knowledge bases // Proceedings of the 3rd International Conference on Learning Representations (ICLR). 2015.

45. Liu H., Wu Y., Yang Y. Analogical inference for multi-relational embeddings // Proceedings of the 34th International Conference on Machine Learning (ICML). 2017. P.2168–2178.

46. Shang C., Tang Y., Huang J., et al. End-to-end structure-aware convolutional networks for knowledge base completion // Proc. of the 33rd AAAI Conference on Artificial Intelligence. Honolulu, Hawaii, USA. 2019. P.3060–3067.

47. Nathani D., Chauhan J., Sharma C., et al. Learning attention-based embeddings for relation prediction in knowledge graphs // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL). 2019. P.4710–4723.

48. Xiong S., Huang W., Duan P. Knowledge graph embedding via relation paths and dynamic mapping matrix // Proc. of the Joint International Conference on Conceptual Modeling (ER) Workshops. Xi'an, China. 2018. P.106–118.

49. Ji G., Liu K., He S., et al. Knowledge graph completion with adaptive sparse transfer matrix // Proc. of the 30th AAAI Conference on Artificial Intelligence. Phoenix, Arizona, USA. 2016. P.985–991.

50. Zhang S., Tay Y., Yao L., et al. Quaternion knowledge graph embeddings // Advances in Neural Information Processing Systems 32 (NeurIPS 2019). Vancouver, Canada. 2019. P.2731–2741.

51. He S., Liu K., Ji G., et al. Learning to represent knowledge graphs with gaussian embedding // Proc. of the 24th ACM International Conference on

Information and Knowledge Management (CIKM). Melbourne, Australia. 2015. P.623–632.

52. Xiao H., Huang M., Hao Y., et al. From one point to a manifold: Orbit models for knowledge graph embedding // CoRR. 2015. abs/1512.04792.

53. Bordes A., Glorot X., Weston J., et al. A semantic matching energy function for learning with multi-relational data: Application to word-sense disambiguation // Machine Learning. 2014. №94(2). P.233–259.

54. Lewicki G., Marino G. Approximation of functions of finite variation by superpositions of a sigmoidal function // Applied Mathematics Letters. 2004. №17(10). P.1147–1152.

55. Socher R., Chen D., Manning C. D., et al. Reasoning with neural tensor networks for knowledge base completion // Advances in Neural Information Processing Systems 26 (NIPS 2013). Lake Tahoe, Nevada, USA. 2013. P.926–934.

56. Nguyen D. Q., Nguyen T. D., Nguyen D. Q., et al. A novel embedding model for knowledge base completion based on convolutional neural network // Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). New Orleans, Louisiana, USA. 2018. P.327–333.

57. Gardner M., Talukdar P. P., Krishnamurthy J., et al. Incorporating vector space similarity in random walk inference over knowledge bases // Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar. 2014. P.397–406.

58. Neelakantan A., Roth B., McCallum A. Compositional vector space models for knowledge base completion // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP). 2015. P.156–166.

59. Guo L., Sun Z., Hu W. Learning to exploit long-term relational dependencies in knowledge graphs // Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. P.2505–2514.

60. Vrandečić D., Krötzsch M. Wikidata: A free collaborative knowledgebase // Communications of the ACM. 2014. №57(10). P.78–85.

61. Bollacker K., Evans C., Paritosh P., et al. Freebase: A collaboratively created graph database for structuring human knowledge // Proc. of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver, BC, Canada. 2008. P.1247–1250.

62. Auer S., Bizer C., Kobilarov G., et al. DBpedia: A nucleus for a web of open data // Proc. of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (ISWC/ASWC). Busan, Korea. 2007. P.722–735.

63. Suchanek F.M., Kasneci G., Weikum G. YAGO: A large ontology from Wikipedia and WordNet // Journal of Web Semantics. 2008. №6(3). P.203–217.

64. Hu Y., Wang Z., Wu W., et al. Recommendation for movies and stars using YAGO and IMDB // Proc. of the 12th International Asia-Pacific Web Conference (APWeb). Busan, Korea. 2010. P.123–129.
65. Lerman K., Jones L. Social browsing on Flickr [Электронный ресурс] // CoRR. 2006. abs/cs/0612047 (дата обращения: 30.12.2025).
66. Miller G.A. WordNet: A lexical database for English // Communications of the ACM. 1995. №38(11). P.39–41.
67. Maedche A., Staab S. The text-to-onto ontology learning environment // Proc. of the 8th International Conference on Conceptual Structures (ICCS) – Software Demonstration. Darmstadt, Germany. 2000.P.1-5.
68. IMDb official site [Электронный ресурс]. 2025. URL <http://www.imdb.com> (дата обращения: 30.12.2025).
69. MusicBrainz official site [Электронный ресурс]. 2025. URL <http://www.musicbrainz.org> (дата обращения: 30.12.2025).
70. Douban official site [Электронный ресурс]. 2025. URL <http://www.douban.com> (дата обращения: 30.12.2025).
71. Agrawal R., Srikant R. Fast algorithms for mining association rules // Proceedings of 20th International Conference on Very Large Data Bases (VLDB). 1994. P.487–499.
72. Cao D., He X., Miao L., et al. Attentive group recommendation // Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. 2018. P.645–654.
73. He X., Liao L., Zhang H., et al. Neural collaborative filtering // Proceedings of the 26th International Conference on World Wide Web. 2017. P.173–182.
74. Pathak A., Gupta K., McAuley J. Generating and personalizing bundle recommendations on Steam // Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017. P.1073–1076.
75. Cao D., Nie L., He X., et al. Embedding factorization models for jointly recommending items and user generated lists // Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017. P.585–594.
76. Chen L., Liu Y., He X., et al. Matching user with item set: Collaborative bundle recommendation with deep attention network // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. P.2095–2101.
77. He Y., Zhang Y., Liu W., et al. Consistency-aware recommendation for user-generated item list continuation // Proceedings of the 13th International Conference on Web Search and Data Mining. 2020. P.250–258.

78. Chang J., Gao C., He X., et al. Bundle recommendation and generation with graph neural networks // *IEEE Transactions on Knowledge and Data Engineering*. 2021. №35(3). P.2326–2340.

79. Wang X., Liu X., Liu J., et al. Relational graph neural network with neighbor interactions for bundle recommendation service // *2021 IEEE International Conference on Web Services*. 2021. P.167–172.

80. He Y., Wang J., Niu W., et al. A hierarchical self-attentive model for recommending user-generated item lists // *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2019. P.1481–1490.

81. Zhang M., Chen Y. Inductive matrix completion based on graph neural networks // *CoRR arXiv preprint*. 2019. arXiv:1904.12058.

82. Cheng W. H., Song S., Chen C. Y., et al. Fashion meets computer vision: A survey // *ACM Computing Surveys*. 2022. №54(4). P.1–41.

83. Wang X., Zhang T. Clothes search in consumer photos via color matching and attribute learning // *Proceedings of the 19th ACM International Conference on Multimedia*. 2011. P.1353–1356.

84. Kiapour M. H., Han X., Lazebnik S., et al. Where to buy it: Matching street clothing photos in online shops // *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015. P.3343–3351.

85. Cheng Z. Q., Wu X., Liu Y., et al. Video2shop: Exact matching clothes in videos to online shopping images // *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. P.4048–4056.

86. Jing Y., Liu D., Kislyuk D., et al. Visual search at pinterest // *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015. P.1889–1898.

87. Zhao K., Hu X., Bu J., et al. Deep style match for complementary recommendation // *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 2017. P.1–7.

88. McAuley J., Targett C., Shi Q., et al. Image-based recommendations on styles and substitutes // *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2015. P.43–52.

89. Yamaguchi K., Okatani T., Sudo K., et al. Mix and match: Joint model for clothing and attribute recognition // *Proceedings of the British Machine Vision Conference 2015*. 2015.

90. Kiapour M. H., Yamaguchi K., Berg A. C., et al. Hipster wars: Discovering elements of fashion styles // *The European Conference on Computer Vision (ECCV)*. 2014. P.472–488.

91. Nguyen T. V., Liu S., Ni B., et al. Sense beauty via face, dressing, and/or voice // Proceedings of the 20th ACM International Conference on Multimedia. 2012. P.239–248.
92. Vasileva M. I., Plummer B. A., Dusad K., et al. Learning type-aware embeddings for fashion compatibility // The European Conference on Computer Vision (ECCV). 2018. P.390–405.
93. Veit A., Kovacs B., Bell S., et al. Learning visual clothing style with heterogeneous dyadic co-occurrences // 2015 IEEE International Conference on Computer Vision (ICCV). 2015. P.4642–4650.
94. Yang X., Ma Y., Liao L., et al. TransNFCM: Translation-based neural fashion compatibility modeling // Proceedings of the AAAI Conference on Artificial Intelligence. 2019. №33(01). P.403–410.
95. Tan R., Vasileva M. I., Saenko K., et al. Learning similarity conditions without explicit supervision // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019. P.1–10.
96. Liu Q., Wu S., Wang L. DeepStyle: Learning user preferences for visual recommendation // Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017. P.841–844.
97. Yin R., Li K., Lu J., et al. Enhancing fashion recommendation with visual compatibility relationship // The World Wide Web Conference. 2019. P.3434–3440.
98. Hu Y., Yi X., Davis L. S. Collaborative fashion recommendation: A functional tensor factorization approach // Proceedings of the 23rd ACM International Conference on Multimedia. 2015. P.129–138.
99. Tangsang P., Yamaguchi K., Okatani T. Recommending outfits from personal closet // Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops. 2017. P.2275–2279.
100. Li Y., Cao L., Zhu J., et al. Mining fashion outfit composition using an end-to-end deep learning approach on set data // IEEE Transactions on Multimedia. 2017. №19(8). P.1946–1955.
101. Han X., Wu Z., Jiang Y. G., et al. Learning fashion compatibility with bidirectional LSTMs // Proceedings of the 25th ACM International Conference on Multimedia. 2017. P.1078–1086.
102. Yang X., Xie D., Wang X., et al. Learning tuple compatibility for conditional outfit recommendation // Proceedings of the 28th ACM International Conference on Multimedia. 2020. P.2636–2644.
103. Lin Y., Moosaei M., Yang H. OutfitNet: Fashion outfit recommendation with attention-based multiple instance learning // Proceedings of The Web Conference 2020. 2020. P.77–87.
104. Sperduti A., Starita A. Supervised neural networks for the classification of structures // IEEE Transactions on Neural Networks. 1997. №8(3). P.714–735.

105. Ying R., He R., Chen K., et al. Graph convolutional neural networks for web-scale recommender systems // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2018. P.974–983.

106. Zhan H., Lin J., Ak K. E., et al. A³-FKG: Attentive attribute-aware fashion knowledge graph for outfit preference prediction // IEEE Transactions on Multimedia. 2022. №24. P.819–831.

107. Cucurull G., Taslakian P., Vazquez D. Context-aware visual compatibility prediction // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019. P.12617–12626.

108. Singhal A., Chopra A., Ayush K., et al. Towards a unified framework for visual compatibility prediction // 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). 2020. P.3607–3616.

109. Cui Z., Li Z., Wu S., et al. Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks // The World Wide Web Conference. 2019. P.307–317.

110. Li X., Wang X., He X., et al. Hierarchical fashion graph network for personalized outfit recommendation // Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020. P.143–152.

111. Liu X., Sun Y., Liu Z., et al. Learning diverse fashion collocation by neural graph filtering // IEEE Transactions on Multimedia. 2021. №23. P.2894–2901.

112. Zhang H., Yang X., Tan J., et al. Learning color compatibility in fashion outfits // Proceedings of the 28th ACM International Conference on Multimedia. 2020. P.2680–2688.

113. Guan W., Wen H., Song X., et al. Partially supervised compatibility modeling // IEEE Transactions on Image Processing. 2022. №31. P.4733–4745.

114. Al-Halah Z., Stiefelhagen R., Grauman K. Fashion forward: Forecasting visual style in fashion // 2017 IEEE International Conference on Computer Vision (ICCV). 2017. P.388–397.

115. Hsiao W. L., Grauman K. Creating capsule wardrobes from fashion images // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2018. P.7161–7170.

116. Hidayati S. C., Goh T. W., Chan J. S. G., et al. Dress with style: Learning style from joint deep embedding of clothing styles and body shapes // IEEE Transactions on Multimedia. 2021. №23. P.365–377.

117. Jiang Y., Xu Q., Cao X., et al. Who to ask: An intelligent fashion consultant // Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (ICMR). 2018. P.525–528.

118. Yu C., Hu Y., Chen Y., et al. Personalized fashion design // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019. P.1–10.

119. Han X., Wu Z., Wu Z., et al. VITON: An image-based virtual try-on network // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2018. P.7543–7552.

120. Ma Z. Online ordering recommendation method based on bipartite graph projection and node2vec algorithm / Z.Ma, M.Gao, L.A.Kazakovtsev // Системы управления и информационные технологии. 2025. №1(99). С. 22-27.

121. Гао М. Построение динамической гибридной модели рекомендаций и исследования пользователей / М. Гао, Ч. Ма, Л.А. Казаковцев // Сибирский аэрокосмический журнал. 2025. Т. 26, №4. С. 466–477.

122. Ma Z. Design and implementation of news recommender system based on collaborative filtering and machine learning / Z.Ma, M.Gao // Innovation Science. 2025. No. 2-2. P. 53-58.

123. Ma Z. Design and implementation of author-based collaborative filtering book recommendation algorithm // Z.Ma, M.Gao // Инновационные исследования: опыт, проблемы внедрения результатов и пути решения: сборник статей Международной научно-практической конференции. Уфа, 2025. P. 31-38.

124. Ma Z. Research and analysis of purchasing investment products based on linear programming model // Актуальные проблемы авиации и космонавтики: Сборник материалов VII Международной научно-практической конференции. Красноярск, 2021. Vol. 2. С. 83-86.

125. Toutanova K., Chen D. Observed versus latent features for knowledge base and text inference // Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC). 2015. P.57–66.

126. Guo Z. Analysis and prediction of data on the basic situation of forest fires / Z. Ma, Z. Guo, E.A. Sopov // Science prospects. 2024. №5(176). P. 66-70.

127. Xu B., Shen H., Cao Q., et al. Graph wavelet neural network // Proc. of the 7th International Conference on Learning Representations (ICLR). New Orleans, Louisiana, USA. 2019.

128. Defferrard M., Bresson X., Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering // Advances in Neural Information Processing Systems (NeurIPS). Barcelona, Spain. 2016. P.3837–3845.

129. Hamilton W. L., Ying R., Leskovec J. Inductive representation learning on large graphs // Advances in Neural Information Processing Systems (NeurIPS). Long Beach, California, USA. 2017. P.1024–1034.

130. Velickovic P., Cucurull G., Casanova A., et al. Graph attention networks // International Conference on Learning Representations (ICLR). 2018.

131. Bruna J., Zaremba W., Szlam A., et al. Spectral networks and locally connected networks on graphs // 2nd International Conference on Learning Representations (ICLR). 2014.

132. Kipf T. N., Welling M. Semi-supervised classification with graph convolutional networks // Proc. of the 5th International Conference on Learning Representations (ICLR). Toulon, France. 2017.

133. Xu B., Shen H., Cao Q., et al. Graph convolutional networks using heat kernel for semi-supervised learning // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI). 2019. P.1928–1934.

134. Klicpera J., Bojchevski A., Günnemann S. Predict then propagate: Graph neural networks meet personalized pagerank // 7th International Conference on Learning Representations (ICLR). 2019.

135. Wu F., Souza A. H., Zhang T., et al. Simplifying graph convolutional networks // Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. P.6861–6871.

136. Monti F., Boscaini D., Masci J., et al. Geometric deep learning on graphs and manifolds using mixture model CNNs // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P.5425–5434.

137. Gilmer J., Schoenholz S. S., Riley P. F., et al. Neural message passing for quantum chemistry // Proceedings of the 34th International Conference on Machine Learning (ICML). 2017. P.1263–1272.

138. Scarselli F., Gori M., Tsoi A. C., et al. The graph neural network model // IEEE Transactions on Neural Networks. 2009. №20(1). P.61–80.

139. Atwood J., Towsley D. Diffusion-convolutional neural networks // Advances in Neural Information Processing Systems (NeurIPS). Barcelona, Spain. 2016. P.1993–2001.

140. Vashishth S., Yadav P., Bhandari M., et al. Confidence-based graph convolutional networks for semi-supervised learning // Proc. of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS). Naha, Okinawa, Japan. 2019. P.1792–1801.

141. Feng Y., You H., Zhang Z., et al. Hypergraph neural networks // Proc. of the 33rd AAAI Conference on Artificial Intelligence. Honolulu, Hawaii, USA. 2019. P.3558–3565.

142. Sak H., Senior A. W., Beaufays F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition // CoRR. 2014. arXiv:1402.1128.

143. Ма Ч. Сравнение производительности алгоритмов глобального поиска с использованием набора тестовых функций // Актуальные проблемы авиации и космонавтики: Сборник материалов VII Международной научно-практической конференции. Красноярск, 2021. Том 2. С. 79-82.

144. Kingma D. P., Ba J. Adam: A method for stochastic optimization // International Conference on Learning Representations (ICLR). 2015. arXiv:1412.6980.

145. Ma Z., Kazakovtsev L.A. Мультиреляционная графовая нейронная сеть на основе взаимной информации // Материалы V Всероссийской (национальной) научной конференции «Достижения науки и технологий».2039-ДНИТ-V-2026, С.73-79.

146. Ma Z., Kazakovtsev L.A. Geometry-Aware Graph Neural Network with Box Embeddings // Системы управления и информационные технологии. 2025. №4(102). С. 4-10.

147. An B., Chen B., Han X., et al. Accurate text-enhanced knowledge graph representation learning // Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). New Orleans, Louisiana, USA. 2018. P.745–755.

148. Devlin J., Chang M., Lee K., et al. BERT: Pre-training of deep bidirectional transformers for language understanding // Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Minneapolis, Minnesota, USA. 2019. P.4171–4186.

149. Xie R., Liu Z., Luan H., et al. Image-embodied knowledge representation learning // Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI). Melbourne, Australia. 2017. P.3140–3146.

150. Liu Y., Li H., García-Durán A., et al. MMKG: Multi-modal knowledge graphs // Proc. of the 16th Extended Semantic Web Conference (ESWC). Portorož, Slovenia. 2019. P.459–474.

151. García-Durán A., Niepert M. KBLRN: End-to-end learning of knowledge base representations with latent, relational, and numerical features // Proc. of the 34th Conference on Uncertainty in Artificial Intelligence (UAI). Monterey, California, USA. 2018. P.372–381.

152. Pezeshkpour P., Chen L., Singh S. Embedding multimodal relational data for knowledge base completion // Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). Brussels, Belgium. 2018. P.3208–3218.

153. Zhang Y., Qian S., Fang Q., et al. Multi-modal knowledge-aware hierarchical attention network for explainable medical question answering // Proc. of the 27th ACM International Conference on Multimedia. Nice, France. 2019. P.1089–1097.

154. Zhang Y., Fang Q., Qian S., et al. Multi-modal multi-relational feature aggregation network for medical knowledge representation learning // Proc. of the 28th ACM International Conference on Multimedia. Seattle, USA. 2020. P.3956–3965.

155. Sun R., Cao X., Zhao Y., et al. Multi-modal knowledge graphs for recommender systems // Proc. of the 29th ACM International Conference on Information and Knowledge Management (CIKM). Virtual Event, Ireland. 2020. P.1405–1414.

156. Qian S., Hu J., Fang Q., et al. Knowledge-aware multi-modal adaptive graph convolutional networks for fake news detection // ACM Transactions on Multimedia Computing, Communications, and Applications. 2021. №17(3). P.98:1–98:23.

157. Chaudhary C., Goyal P., Goyal N., et al. Image retrieval for complex queries using knowledge embedding // ACM Transactions on Multimedia Computing, Communications, and Applications. 2020. №16(1). P.13:1–13:23.

158. Reimers N., Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks // Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China. 2019. P.3980–3990.

159. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // Proc. of the 3rd International Conference on Learning Representations (ICLR). San Diego, California, USA. 2015.

160. Zadeh A., Chen M., Poria S., et al. Tensor fusion network for multimodal sentiment analysis // Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP). Copenhagen, Denmark. 2017. P.1103–1114.

161. Suchanek F.M., Kasneci G., Weikum G. YAGO: A core of semantic knowledge // Proc. of the 16th International World Wide Web Conference (WWW). Banff, Canada. 2007. P.697–706.

162. Zhou Z., Wang C., Feng Y., et al. JoinTE: Jointly utilizing 1D and 2D convolution for knowledge graph embedding // Knowledge-Based Systems. 2022. №240. P.108100.

163. Jiang D., Wang R., Yang J., et al. Kernel multi-attention neural network for knowledge graph embedding // Knowledge-Based Systems. 2021. №227. P.107188.

164. Zheng D., Song X., Ma C., et al. DGL-KE: Training knowledge graph embeddings at scale // Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Xi'an, China. 2020. P.739–748.

165. Ma Z., Kazakovtsev L.A. Multimodal knowledge graph attention networks // Системы управления и информационные технологии, №2(100), 2025. С.58-65.

166. Ma Z., Gao M. Design and implementation of tourism visualization attraction recommendation system based on collaborative filtering algorithm // Innovation Science. 2025. №2(2). P. 48-53.

167. Ma Z., Gao M., Go Z. Collaborative filtering-based recommendation strategy for reducing false ratings // Innovation Science. 2024. №1(2)1-2. P. 56-59.

168. Ма Ч., Гао М. Стратегия рекомендаций на основе коллаборативной фильтрации для снижения ложных оценок // Актуальные проблемы авиации и космонавтики: Сборник материалов IX Международной научно-практической конференции. Красноярск, 2023. С. 74-77.

169. Ma Z., Gao M. A study on weighted collaborative filtering recommendation algorithm based on multi-cluster hybridization // Innovation Science. 2025. №2(2). P. 58-63.

170. Ma Z., Gao M. Self-optimizing weighted Bayesian classifier based on heuristic algorithm // Innovation Science. 2023. №6(1). P. 41-46.

171. Fan B., Kong Q., Yuan X., et al. Learning weighted hamming distance for binary descriptors // International Conference on Acoustics, Speech and Signal Processing. IEEE. 2013. P.2395-2399.

172. Ma Z., Gao M., Guo Z. The design of clothing combination recommendation system based on hash technology// Системы управления и информационные технологии, №1(99), 2025.

173. Ма Ч. NCHM-NET. Свидетельство о государственной регистрации программы для ЭВМ / Ма Ч., Казаковцев Л.А. // Роспатент. рег. № 2026615791 от 03.03.2026. заявл. 23.02.2026

ПРИЛОЖЕНИЕ А
АКТ О ВНЕДРЕНИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИИ

学位论文成果实施法案

关于学位论文研究成果的实施

本文件特此确认，北京奇虎科技有限责任公司已成功部署由马占军研发的 MCHM-Net 算法，并将其应用于电商服装推荐系统。该算法采用二进制编码学习技术，显著提升了推荐效率。作为马占军博士论文研究阶段的核心成果，该算法的实际应用使推荐系统的准确度获得平均 1.5% 的优化提升，系统的响应时间缩短十分之一。

主任签字：



北京奇虎科技有限公司

Перевод с китайского языка на русский

**АКТ О ВНЕДРЕНИИ РЕЗУЛЬТАТОВ
ДИССЕРТАЦИОННОГО ИССЛЕДОВАНИЯ**

Настоящим подтверждаю, что ООО «Бэйцзин Циху Кэцзи» успешно внедрила алгоритм MCHM-Net, разработанный Ма Чжаньцзюнем, и применила его в своей системе рекомендаций одежды для электронной коммерции. Этот алгоритм использует технологию обучения с бинарным кодированием, что значительно повышает эффективность рекомендаций. Основным результатом диссертационного исследования аспиранта Ма Чжаньцзюня стало то, что практическое применение этого алгоритма повысило точность рекомендательных систем в среднем на 1,5% и сократило время отклика системы на одну десятую.

Подпись директора: /Ван Синтянь/

Печать ООО «Бэйцзин Циху Кэцзи»: [ООО «Бэйцзин Циху Кэцзи»; 11010510024446]

--конец перевода--

Перевод с китайского языка на русский язык верен.

Переводчик УМС

