

GNN and Graph Features in OTTO – Multi-Objective Recommender System Kaggle Competition

Зырянов Илья, M05-116

30 декабря 2022 г.

Аннотация

Последнее время графовые нейронные сети вызывают все больший интерес в различных задачах построения рекомендательных систем. Это видно и по увеличивающемуся количеству связанных с ними публикаций и по новостям об их непосредственном применении некоторыми крупными компаниями в конкретных задачах. Но, при этом, они остаются далеко не самым популярным и основным подходом при построении рекомендаций на практике. Цель данной работы - попробовать на практическом примере проверить применимость GNN в конкретной задаче, что и было сделано в рамках соревнования "OTTO – Multi-Objective Recommender System Kaggle Competition" на платформе kaggle.

1 Вступление

Для начала расскажем немного о соревновании. Соревнование проводится компанией OTTO - это крупнейший в Германии онлайн магазин, в котором представлено более 10 миллионов товаров от более чем 16 тысяч брендов. Цель соревнования - научиться предсказывать клики, добавления в корзину и покупки пользователей, основываясь на предыдущих событиях в сессии, построив при этом единую многоцелевую рекомендательную систему. В подобных задачах часто на первый план выходят не какие-то хитрые и сложные архитектуры, а грамотный feature engineering и хорошо выбранные подходы для генерации фичей, что было в частности показано в рамках соревнования HM Personalized Fashion Recommendations. Поэтому мы попробуем применить в данном соревновании GNN для генерации фичей.

2 Данные в соревновании

Обучающая выборка представлена в формате: id сессии и последовательность событий, где каждое событие включает в себя id товара, время события и типа события (клик, добавление или покупка). Всего в обучающей выборке более 12 миллионов сессий, около 220 миллионов событий и почти 2 миллиона уникальных товаров. Тестовая выборка представляет из себя набор обрезанных сессий, отличных от сессий в обучении, по которым модель должна предсказать последующие действия в рамках сессии, вплоть до 20 объектов на каждый тип. Всего в тестовой выборке около 1.5 миллиона сессий.

Качество решения оценивается через Recall@20 для каждого типа событий, образуя финальную формулу:

$$score = 0.1 * R_{clicks} + 0.3 * R_{carts} + 0.6 * R_{orders}$$

3 Подходы участников

В данной секции мы рассмотрим различные основные подходы участников соревнования.

Наиболее популярным решением на текущий момент является дефолтный для рекомендательных систем подход, разбивающий пайплайн на 2 части: генерация кандидатов и их ранжирование.

Подход	Score
Отбор кандидатов + ранжирование без GNN	0.575
GRU4Rec	0.544
GCN pytorch geometric	0.143
RGCN dgl	0.108
Отбор кандидатов + ранжирование + GCN	0.463

Таблица 1: An example table.

Для генерации кандидатов в большинстве используется матричная факторизация. Иногда используются и другие подходы, так, например, одно из решений с высоким скором использует сразу 5 подходов для генерации кандидатов:

- Исторические данные
- По 20 наиболее популярных объектов для каждого из типов взаимодействия
- Различные матрицы совстречаемости товаров при разных типа взаимодействия

Такие эмпирические подходы дают неплохой прирост метрик.

Для ранжирования же в основном используются градиентные бустинги или ранкеры, основанные на них, обученные на различных сгенерированных фичах, таких как длина сессии и т.д. Также на удивление хорошо себя показывают подходы к ранжированию просто на основе каких-то эмпирических правил, например выдача большего приоритета в соответствии с совстречаемостью, тем, которые недавно были в корзине, просто популярным товарам. Это во многом демонстрирует сложность данной задачи, так как мы видим, что интересные подходы далеко не всегда могут превзойти обычные эмпирические правила.

Также особое внимание хочется обратить на несколько другой подход к решению данной задачи. Данная задача является по факту задачей последовательных рекомендаций. Часть решений была основана на попытке использовать это и попробовать крайне интересные архитектуры рекомендаций, такие как Transformers4Rec и GRU4Rec. Это популярные в данный момент подходы с точки зрения написания статей, но главная проблема того же BERT4Rec и других сходных моделей, что, несмотря на ресурсоемкость, они далеко не всегда могут дать хороший результат на практике. В данном же соревновании у таких подходов наблюдался довольно неплохой скор, отстающий от сора медалей всего на 5%.

4 Проведенные эксперименты

В наших экспериментах мы будем придерживаться схожих подходов, а именно делать ранжирование с помощью бустинга на основе обычных сгенерированных фичей, графовых фичей и этих фичей вместе и сравним результаты.

Для непосредственной генерации графовых фичей было попробовано 2 подхода:

- Graph Convolutional network на основе pytorch geometric. Для обучения модели данные были сконвертированы в формат HeteroData этой же библиотеки, который позволяет иметь несколько разных типов сущностей в графе, что нам и нужно для этой задачи. Модель обучалась на 50 эпохах, размер эмбедингов - 16
- RGCN на основе Deep Graph Library. Также HeteroGraphConv для графа, как и выше. Модель обучалась также на 50 эпохах и размер эмбедингов - 16

В Таблице 1 представлены попробованные подходы и их скоры в соревновании.

Как мы видим в рамках данных экспериментов не удалось достичь прироста метрик за счет использования GNN и такие подходы уступают подходам без них. Дело скорее всего в специфике соревнования, т.к. данные не имеют явную графовую структуру и очень важна последовательная их структура, поэтому и графовые признаки не позволяют извлечь много полезной для обучения и предсказаний информации.

5 Заключение

В данном соревновании остается широкий спектр способов применить графовые фичи и попробовать тем самым улучшить результат. Так, например, из можно попробовать поиспользовать в моделях Transformers4Rec, а так же можно попробовать их генерировать посредством других архитектур. К сожалению, все это попробовать не хватило времени, но до конца соревнования еще месяц, так что может что-нибудь попробую. Говоря об проведенных экспериментах, видно, что, к сожалению, в рамках них, графовые фичи не дали прироста метрик, а использование только их давало достаточно слабый результат. Это, как мне кажется, можно объяснить спецификой задачи, так как данные здесь довольно далеки от удобной графовой структуры. Даже сравнения с соревнованием "HM Personalized Fashion Recommendations где графовые фичи себя не плохо показали, в нем надо было предсказывать просто релевантные товары, а в этой задаче очень важен порядок, нужно предсказать какое именно действие над каким товаром будет и это представление крайне важно. Подтверждает эту важность неплохой результат моделей для последовательных рекомендаций. Но даже в такой сложной для графового представления задаче фичи показали себя не совсем плохо, позволяя делать валидные рекомендации, хоть и с ощутимо меньшим качеством, что говорит нам о больших перспективах области GNN в рекомендациях, но предостерегает от бездумного использования этого подхода.