

Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Демин Георгий Александрович

Методы построения рекомендательных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

д.ф-м.н., профессор

Дьяконов А. Г.

Москва, 2020

Содержание

1 Введение	3
2 Рекомендательных системы	3
2.1 Общая постановка задачи	3
2.2 Специальные постановки задач	4
2.3 Некоторые подходы к построению моделей	5
2.3.1 Memory-based model	5
2.3.2 Model-based method	5
2.3.3 Нейросетевые методы	7
3 Постановка задачи	8
4 Рекомендательные системы на сетях	8
4.1 GraphRec-WWW19	9
5 Рекомендательные системы на группах	10
5.1 AGREE	10
6 Предлагаемый подход	11
6.1 Моделирование графовой структуры	11
6.2 Обогащение обучающей выборки	12
6.3 Ансамблирование	12
7 Эксперименты	12
8 Результаты	13
9 Литература	14

Аннотация

В работе рассмотрены различные модели рекомендательных систем, решается задаче улучшения работы рекомендательной системы на группах с помощью моделирования графовой структуры и применения рекомендательной системы на сетях. Был предложен и реализован метод применения рекомендательных систем на сетях для данных, в которых пользователи объединены в группы, но не представляют собой граф (такие данные изначально рассчитаны на групповые рекомендательные системы). Проведены эксперименты, показывающие, что данный метод улучшает качество работы рекомендательной системы.

1 Введение

Рекомендательные системы - это класс моделей машинного обучения, предназначенных для решения задачи подбора товара (услуги) для пользователей. Постановки задач в конкретных случаях могут сильно отличаться, зависит это от предметной области и имеющихся данных. В работе будут рассмотрены различные примеры постановок задач и разработанные алгоритмы для их решения. Более подробно будут рассмотрены рекомендательные системы на сетях и группах пользователей и показано, как переходить от одной постановки задачи к другой.

2 Рекомендательных системы

2.1 Общая постановка задачи

Наиболее обще задачу можно поставить так: дано множество пользователей U , множество товаров I , а также R - множество взаимодействий пользователей и товаров, то есть существует не всюду определенное отображение $f : (U \times I) \rightarrow R$. Введём стандартные обозначения:

$r_{ui} := f(u, i)$ - известное взаимодействие пользователя u и товара i

\hat{r}_{ui} - предсказанное взаимодействие

U_i - множество пользователей, оценивших товар i

I_u - множество товаров, оценённых пользователем u

Задача состоит в том, чтобы восстановить значения f на множестве, на котором отображение не определено.

Обычно решается задача обучения с учителем: $\{(X, y)\} = \{(u, i), r_{ui}\}$. Множество $R \in \mathbb{Z}_+^{n \times m}$, представляет собой матрицу (ее называют матрицей интеракций), где на пересечении i столбца и u строки стоит число - оценка пользователя u товару i (здесь и далее под товарами будут пониматься любые сущности (посещенные рестораны, просмотренные фильмы, оказанные услуги), оценки которым пользователь может поставить. Также рассматриваются задачи, в которых матрица интеракции содержит не оценки товаров, а факт взаимодействия (1 - если пользователь что-либо сделал с

товаром, -1 - иначе). Такие модели называются неявными. [Подробнее про неявные модели](#)

	Товар 1	Товар 2	Товар 3	Товар 4	Товар 5
Клиент 1		3		5	
Клиент 2	1		1	1	
Клиент 3	2			3	2
Клиент 4		4			5
Клиент 5	5		2	3	4

Рис. 1: Пример матрицы интеракций

2.2 Специальные постановки задач

Далее будут рассмотрены различные постановки задач (в зависимости от наличия и характера дополнительной информации) и методы их решений (далее будет подразумеваться, что матрица интеракций известна всегда, задачи, в которых она не задана рассматриваться не будут)

1. Известна только матрица интеракций. Такие задачи называются collaborative filtering.
2. Информация о пользователях (например, социально-демографическая) или о товарах (категория товара, его цена, статистики продаж), задачи с такой информацией называются content-based filtering.
3. Информация о времени или месте покупки товара — context-aware collaborative filtering
4. Известна не только оценка пользователя, но и текстовый отзыв — Review-based recommender systems
5. Между пользователями существуют дружеские связи (Social networks recommender systems) или они объединены в группы (Group recommender systems)

2.3 Некоторые подходы к построению моделей

2.3.1 Memory-based model

Это самый наивный метод рекомендации, опишем его подробно, чтобы дать представление как проектируются алгоритмы рекомендаций и от каких характерных проблем страдают.

Этот подход основан на нахождении похожих пользователей (или товаров) и усреднении их оценок. Интуиция метода такова: пользователю понравится товар, если этот товар нравится похожим пользователям (которые оценивают товары примерно также как данный пользователь). Будем находить

$$\hat{r}_{ui} = \frac{1}{|U_i|} \sum_{u' \in U_i} \text{sim}(u, u') r_{u'i} \quad (1)$$

Для товара i находятся пользователи, которые его оценили u' , для каждого ищется его мера схожести с u (это может быть косинусное расстояние, коэффициент Жакарра, посчитанные по матрице инеракций) и рейтинги этих пользователей складываются с весами, соответствующими похожести пользователя на u . Таким образом мы получаем искомые рейтинг как некоторую линейную комбинацию всех рейтингов этого товара. Этот подход очень прост: он не требует больших вычислений, хранения дополнительных данных. В силу своей примитивности метод выдает плохие результаты для реальных датасетов. Он также подвержен недостаткам, общих для многих моделей колаборативной фильтрации: рекомендация самых популярных товаров и проблема холодного старта (для нового пользователя не можем посчитать его схожесть с другими). Последняя проблема характерна для многих моделей рекомендательных систем.

2.3.2 Model-based method

Основная идея данного метода - разложение разреженной матрицы интеракции на произведение матриц меньшего ранга (matrix factorization - MF).

$$\min_{P, Q} \|R - PQ^T\|_F^2 \quad (2)$$

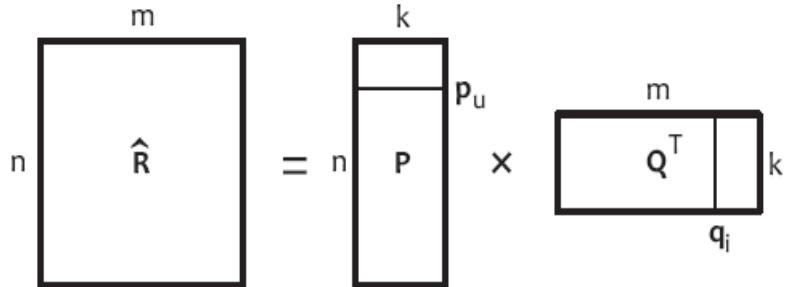


Рис. 2: Матричное разложение

В формуле (2) $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{m \times k}$ минимизируется норма Фробениуса. По сути для каждого товара и пользователя выучивается эмбеддинг размерности k . На практике вместо (2) используется (3): в этой оптимизационной задаче мы учим P и Q только по известным рейтингам в R , также она учитывает регуляризацию. Функционал в (3) соответствует функционалу Ridge-регрессии, для которого известно аналитическое решение, однако обычно эту задачу решают булочно-координатным спуском.

$$\sum_{u,i} (r_{u,i} - p_u q_i^T) + \lambda(\|p_u\|^2 + \|q_i\|^2) \rightarrow \min_{P,Q} \quad (3)$$

Эта модель требует дополнительного хранения P и Q и времени на обучение, однако как преимущество мы получаем эмбеддинги (эмбеддингом называется представление объекта вектором определенного размера, формально эмбеддинг может являться простым признаком описанием объекта, однако обычно используют термин «эмбеддинг», чтобы подчеркнуть, что такое представление объекта обучаемо, то есть в процессе настройки модели оно изменяется [2]), пользователей и товаров, которые можно использовать далее.

Рассмотренный метод построения эмбеддингов (как с помощью классических матричных разложений [1], так и с помощью нейросетей [3]) является одним из самых востребованных методов решения задачи колаборативной фильтрации при отсутствии дополнительной информации.

2.3.3 Нейросетевые методы

Данная категория объединяет большое разнообразие моделей рекомендательных систем, которые содержат в себе нейронные сети (или их элементы). В общем случае модель строится следующим образом.

1. Уровень представлений: каждый пользователь, товар (а также, возможно, другая информация) представляются в виде вектора. Он обычно является либо вектором признаков, либо эмбеддингом (то есть обучаемым вектором).
2. Различные модели нейросетей: полносвязные (полносвязные свои есть практически во всех моделях), сети с вниманием (механизм внимания применяется, чтобы определить, как сильно каждая входная переменная влияет на каждую выходную), рекуррентные нейронные сети (традиционно они используются при обработки последовательностей, а в рекомендательных системах применяются в задачах, в которых известен текстовый отзыв клиента), генеративно-состязательные сети (модель состоит из 2 блоков: первый генеративный блок пытается создать правдоподобный объект из обучающей выборки, второй предназначен для того, чтобы отличать сгенерированный объект от настоящего — в рекомендательных системах первый блок предсказывает рейтинг, а второй определяет, насколько он правдоподобен)
3. Целевая функция определяет какую задачу мы решаем. Для явного отклика обычно используется функции потерь из задачи регрессии (RMSE, MAE, Hinge-loss и другие). Для неявного отклика целевой функцией является сигмоида, которая определяет вероятность, с которой пользователю будет интересен товар.

Сегодня большинство новых предлагаемых и разрабатываемых моделей являются именно нейросетевыми. Основные достоинства и недостатки таких моделей совпадают с достоинствами и недостатками нейронных сетей для других задач машинного обучения. Главные плюсы: гибкость и сложность модели — потенциально из различных нейросетевых блоков можно построить сколь угодно сложную модель, учитывающую всю возможную информацию. Главной же проблемой является сложность в обучении модели. Поэтому зачастую конкретная архитектура нейронной сети выбирается исходя из структуры имеющихся данных для определенной задачи.

3 Постановка задачи

Выше приведено много различных постановок задачи рекомендательных систем и методов их решений. В связи с этим становится актуальной задача преобразования входных данных для обработки их различными моделями. Предлагается вместо разработки новой модели для входных данных, преобразовать их неким образом, чтобы иметь возможность сделать предсказание с помощью модели, которая работает с другими структурами данных и засчет ансамблирования 2 принципиально различных моделей получить лучший результат.

В частности в данной работе требуется подход, с помощью которого в данных, содержащих информацию о группах пользователей моделируется графовая структура. Таким образом, эти данные становятся доступны для рекомендательных систем на сетях.

[Улучшить стиль](#)

Рассмотрим подробнее задачи рекомендации на сетях и групповые рекомендации.

4 Рекомендательные системы на сетях

Пусть, кроме матрицы интеракции, известны связи между пользователями (дружеские, родственные и т.д.). Формально это означает, что задан граф $\Gamma = \langle U, E \rangle$ и $e_{uv} \in E \Leftrightarrow$ пользователи u и v связаны. Предполагается, что пользователи связаны симметрично, то есть граф ненаправленный, также предполагается, что граф невзвешенный (то есть все связи между пользователями равнозначны). Задача рекомендательной системы не изменяется: необходимо найти \hat{r}_{ui} .

На сегодняшний день для построения рекомендательных систем на сетях, как и других моделей, используют нейросетевые модели (обзор графовых рекомендательных систем можно найти в [4]). Для представления пользователя используются эмбеддинги, включающие в себя информацию о друзьях пользователя и оценках товаров. Иногда используется механизм случайного блуждания см [5]. Далее подробнее рассматривается конкретная модель графовых рекомендаций - Graph Neural Networks for Social Recommendation [6].

4.1 GraphRec-WWW19

Это модель колаборативной фильтрации с вниманием, ее архитектура показана на рис. 3. Важной особенностью данной модели является то, что эмбеддинги строятся практически для каждой сущности:

1. эмбеддинг каждого уникального рейтинга $r \in \{1, 2, 3, 4, 5\}$
2. эмбеддинг пользователя состоит из эмбеддинга относительно других пользователей (его соседей в графе) и эмбеддинга относительно товаров (которые он оценил)
3. эмбеддинг товара (линейная комбинация по все пользователям, оценивших товар, эмбеддинга пользователя сконкатенированного с эмбеддингом рейтинга)

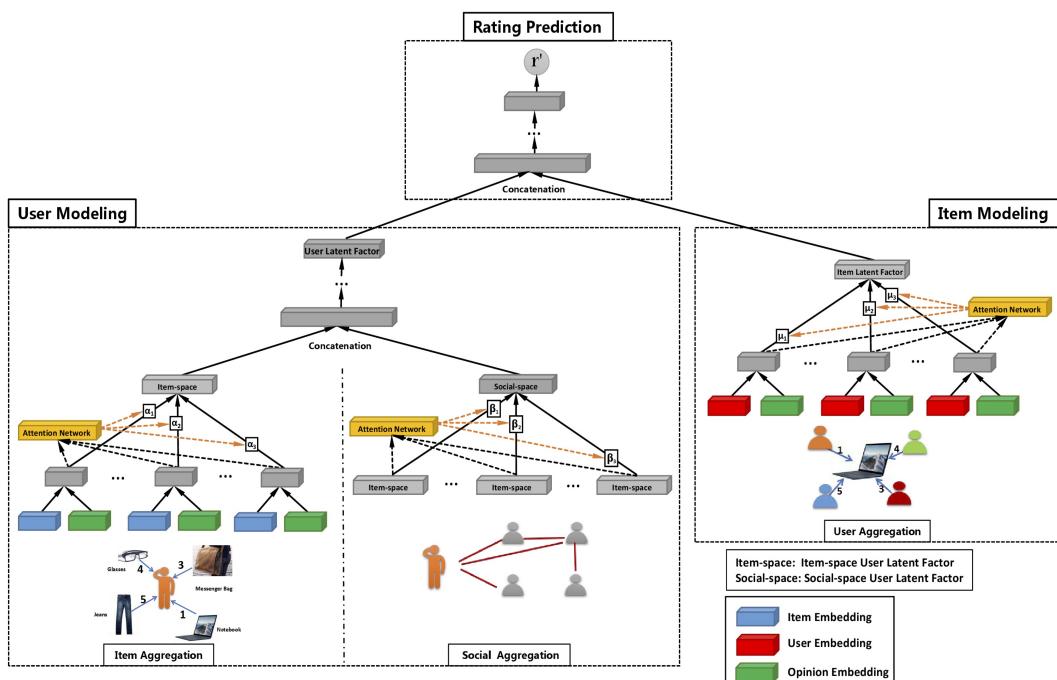


Рис. 3: Архитектура модели GraphRec

Таким образом модель может обучаться и предсказывать результат даже в случае, когда никакой дополнительной информации о пользователях и товарах не задано. Механизм внимания используется, чтобы определить с каким весом

1. каждый пользователь влияет на товар
2. каждый товар влияет на пользователя

3. каждый пользователь влияет на пользователя, который является его соседом

Эта модель используется для проведения экспериментов. Она была выбрана, так как является достаточно новой моделью (препринт опубликован летом 2019 года), а также код ее реализации выложен в открытый доступ.

5 Рекомендательные системы на группах

Пусть, кроме матрицы интеракций, задано разбиение пользователей по группам (возможно, пересекающимся) $U \rightarrow G = \{g_1, g_2, \dots, g_d\}$, группа также может поставить оценку товару $f_g : (G \times I) \rightarrow R$. По аналогии с обозначениями из 2.1 $r_{gi} := f_g(g, i)$ - известный рейтинг, \hat{r}_{gi} - предсказанный рейтинг для группы g и товара i , кроме того обозначим $U(g)$ - множество пользователей, входящих в группу g .

Требуется восстановить неизвестные рейтинги для пользователей \hat{r}_{ui} и (или) групп \hat{r}_{gi} . Для решения этой задачи в основном используются нейросети с механизмом внимания, который позволяет учитывать влияние каждого пользователя в группе на оценку всей группы (см, например, [7]). В качестве примера рекомендательной системы на группах рассматривается модель AGREE, описанная в [8]. Она выбрана по тем же причинам, что и GraphRec-WWW-19: статья с описанием модели вышла сравнительно недавно (2018 год) и код выложен в открытый доступ.

5.1 AGREE

Архитектура данной модели включает эмбеддинги для групп, пользователей и товаров, механизм влияния для того, чтобы определить степень влияния каждого пользователя на решение группы. Ее архитектура показана на рис. 4. Важно отметить, что изначально данная модель предложена авторами для задачи с неявным откликом, то есть для каждого пользователя и группы находится вероятность заинтересованности в товаре, который они еще не оценили. Однако в архитектуре нет существенных элементов, предназначенных именно для задач с неявным откликом, поэтому заменой итоговой функции сигмоиды

$$\sigma(F(x)) = \frac{1}{1 + \exp(-y_{im}F(x))}$$

на квадратичную функцию

$$er(x) = -(F(x) - y_{ex})^2$$

Мы будем обучать сеть не на предсказание вероятности, а на предсказание рейтинга. Здесь $F(x)$ обозначает выход нейросети, y_{im}, y_{ex} - обозначают неявный (+1 — если оценил и -1 — если не оценил) и явный (выставленный рейтинг) отклик соответственно.

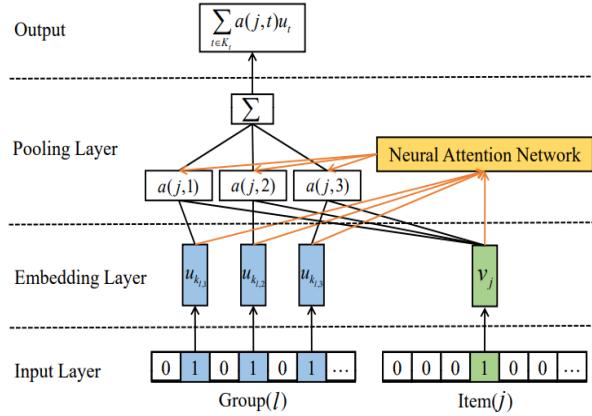


Рис. 4: Архитектура модели Agree

6 Предлагаемый подход

6.1 Моделирование графовой структуры

Для использования рекомендательной системы на графах, нам необходимо смоделировать графовую структуру в наших данных, для этого рассмотрим 3 способа:

1. Все пользователи, входящие в одну группу связаны ребрами друг с другом.
Пусть число связей равно M
2. Оставим каждое ребро, полученное первым способом, с вероятностью 0.5. Тогда мы получаем примерно $\frac{1}{2}M$ связей. Оставшиеся $\frac{1}{2}M$ заполним случайно (то есть случайно сэмплируем без возвращения пару пользователей, которые не связаны ребром).
3. Полностью случайно сэмплируем M связей

6.2 Обогащение обучающей выборки

Сформируем валидационную выборку (она потребуется нам при ансамблировании): отберем небольшое число (порядка 5%) пар рейтингов пользователь-товар и исключим их из обучающей. Так как в данных имеется информация о рейтингах группы-товар используем ее следующим образом: рассмотрим всех пользователей, входящих в группу $U(g)$ и рейтинги, которыми группа оценила товары $\{r_{gv_1}, \dots, r_{gv_k}\}$, и для каждого пользователя из группы $u \in U(g)$ добавим в матрицу интеракций рейтинг $r_{uv_j} = r_{gv_j}$. То есть, будем считать, что, если человек состоит в группе, то он оценил соответствующий товар той же оценкой, что и группа. Таким образом, мы обогатим матрицу интеракций большим количеством рейтингов и практически исключим негативный эффект от исключения данных в валидационную выборку.

6.3 Ансамблирование

После применения групповой и сетевой рекомендательной системы, мы получим предсказанные рейтинги двух моделей. Обучим простейшую модель регрессии вида

$$\sum_{(u,i) \in val_set} (r_{ui} - (\lambda r_{ui}^{group} + (1 - \lambda)r_{ui}^{graph}))^2 \rightarrow \min_{\lambda},$$

где r_{ui}^{group} — предсказание обученной групповой модели на паре валидационной выборке, r_{ui}^{graph} — предсказание обученной графовой модели на паре валидационной выборке. Таким образом будет подобран оптимальный вес, показывающий какой вклад вносит рейтинг каждой модели в ответ.

7 Эксперименты

Для тестирования предлагаемого подхода применяется датасет CAMRa2011 (он был использован в оригинальной статье [8]). Он представляет собой данные о рейтингах фильмов, просмотренных в семье. Содержит

1. 600 пользователей
2. 200 групп (в каждой группе в среднем 2.2 пользователя - это семья)
3. 7000 товаров

4. 140000 рейтинги группа-товар
5. 110000 рейтинги пользователь-товар
6. 600 рейтингов пользователь-товар для теста

В соответствии с предлагаемым подходом добавим к рейтингам пользователь-товар новые рейтинги пользователь-товар, полученные из рейтингов группы-товар.

В таблице 1 представлено сравнение метрик качества для каждой модели:

- in — модель GraphRec, все пользователи в одной группе друзья
 - half — модель GraphRec, половина связей случайные
 - random —модель GraphRec все связи случайные
- . _ens - означает ансамблированные результаты модели GraphRec (с соответствующей графовой структурой и AGREE). Мы видим, что качество при агрегировании моделей намного выше (метрики RMSE и MAE соответственно меньше), чем качество каждой модели в отдельности. Это позволяет говорить о том, что предлагаемый метод действительно улучшает качество моделей. Самый лучший результат достигнут при моделирование графовой структуры таким образом, чтобы все пользователи в одной группе становились друзьями.

	AGREE	in	half	rand	in_ens	half_ens	rand_ens
RMSE	0.9860	0.9740	0.9841	0.9833	0.9668	0.9698	0.9737
MAE	0.7542	0.7345	0.7432	0.7460	0.7291	0.7309	0.7420

Таблица 1: Сравнение метрик качества моделей

8 Результаты

Предложен подход, позволяющий использовать датасеты, содержащие группы пользователей для моделей рекомендаций на сетях. Данный подход включает:

1. моделирование графовой структуры

2. обогащение обучающей выборки
3. метод ансамблирования групповой и графовой моделей.

Список литературы

- [1] Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. 2014. Y Shi, M Larson, A Hanjalic - ACM Computing Surveys (CSUR), 2014 - dl.acm.org
- [2] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781[cs.AI]
- [3] S. Zhang, L. Yao, A. Sun, Y. Tay. 2019. Deep learning based recommender system: A survey and new perspectives - ACM Computing Surveys (CSUR), - dl.acm.org
- [4] Guandong Xu, Zhiang Wu, Zhiang Wu, Yanchun Zhang,Yanchun Zhang, Jie Cao. 2013. Social networking meets recommender systems: survey. Int. J. Social Network Mining, Vol. 2, No. 1, 2015 p. 64-96
- [5] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, Qing Li. 2019. Deep Social Collaborative Filtering arXiv:1907.06853[cs.AI]
- [6] Wenqi Fan, Yao Ma , Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In Proceedings of the 28th International Conference on World Wide Web (WWW).
- [7] Lucas Vinh Tran, Tuan-Anh Nguyen Pham, Yi Tay, Yiding Liu, Gao Cong, Xiaoli Li Interact and Decide: Medley of Sub-Attention Networks for Effective Group Recommendation. 2019. arXiv:1804.04327 [cs.AI]
- [8] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive Group Recommendation. In The 41st International ACM SIGIR Conference on Research Development in Information Retrieval (SIGIR '18). ACM, New York, NY, USA, 645-654.