Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №4
по курсу
”Проектирование Рекомендательных Систем”

Тема Матричная факторизация

Студент Якуба Д. В.

Группа ИУ7-33М

Оценка (баллы) _____

Преподаватели Быстрицкая А.Ю.

Москва — 2023 г.

Оглавление

ВВЕДЕНИЕ

Цель работы – изучить алгоритмы матричной факторизации на примере funk SVD и SVD++.

Для достижения поставленной цели потребуется:

- привести описание алгоритмов;
- привести описание используемых для исследования данных;
- привести зависимости скорости работы алгоритмов от заданных параметров.

1. Аналитический раздел

1.1 Матричная факторизация

Матричная факторизация – это класс алгоритмов коллаборативной фильтрации, используемых в рекомендательных системах. Данные алгоритмы работают путем разложения матрицы взаимодействия пользователя с объектами на произведение двух прямоугольных матриц меньшей размерности. Зачастую матричная факторизация используется для улучшения качества персонализированных рекомендаций, позволяя выявить скрытые паттерны и взаимосвязи между пользователями и товарами.[1]

Методы матричной факторизации в рекомендательных системах обладают следующими аспектами:

- Снижение размерности – уменьшение объема вычислений и улучшение эффективности;
- Скрытые факторы – данные методы предполагают, что в системе присутствуют некоторые латентные признаки, которые влияют на предпочтения пользователей и характеристики товаров;
- Эффективность работы с разреженными данными – матричная факторизация может эффективно работать с разреженными данными, заполняя недостающие значения.

1.2 Funk SVD

Funk SVD – это один из методов матричной факторизации, который был предложен Саймоном Функом и является одним из ранних подходов к коллаборативной фильтрации.

Целью обучения Funk SVD является минимизация разницы между фактическими оценками пользователей и предсказанными на основе разложения матрицы. Для оптимизации параметров разложения и нахождения оптимальных значений скрытых факторов используется градиентный спуск. [2]

Для оценки качества модели обычно используются среднеквадратичная ошибка (RMSE) и средняя абсолютная ошибка (MAE).

Funk SVD имеет также и свои ограничения – он не способен учитывать неявные обратные связи и отсутствие возможности холодного старта.

Прогнозируемую оценку можно рассчитать как:

$$\widetilde{R} = HW \tag{1}$$

где $\widetilde{R} \in \mathbb{R}^{users \times items}$ — матрица оценок пользователя;

$H \in \mathbb{R}^{userx \times latentfactors}$ — содержит латентные признаки пользователя;

$W \in \mathbb{R}^{latentfactors \times items}$ — скрытые признаки объекта.

В частности, прогнозируемая оценка пользователя u объекту i:

$$\widetilde{r}_{ui} = \sum_{f=0}^{nfactors} H_{u,f} W_{f,i} \tag{2}$$

1.3 SVD++

SVD++ – это усовершенствование SVD, которая было разработано для решения некоторых ограничений традиционных SVD-моделей. SVD++ учитывает не только явные оценки или взаимодействия пользователей с товарами, но и неявные взаимодействия. Кроме того, он также учитывает предвзятость пользователя к объекту.

Прогнозируемая оценка, которую пользователь u поставит объекту i, рассчитывается как:

$$\widetilde{r}_{ui} = \mu + b_i + b_u + \sum_{f=0}^{nfactors} H_{u,f} W_{f,i} \tag{3}$$

где μ — относится к общей средней оценке;

b_i, b_u — относятся к наблюдаемому отклонению объекта i и пользователя u от среднего.

Главным недостатком SVD++ является то, что при добавлении нового пользователя требуется переобучение модели.

2. Конструкторский раздел

2.1 MovieLens 100K Dataset

В качестве источника данных был взят датасет, располагающийся в свободном доступе на веб-сайте Grouplens [3]. Датасет включает в себя 100000 оценок от 1000 пользователей на 1700 фильмов.

3. Технологический раздел

3.1 Средства реализации

В качестве используемого был выбран язык программирования Python [4].

Данный выбор обусловлен следующими факторами:

- Большое количество исчерпывающей документации;
- Широкий выбор доступных библиотек для разработки;
- Простота синтаксиса языка и высокая скорость разработки.

При написании программного продукта использовалась среда разработки Visual Studio Code. Данный выбор обусловлен тем, что данная среда распространяется по свободной лицензии, поставляется для конечного пользователя с открытым исходным кодом, а также имеет большое число расширений, ускоряющих разработку.

3.2 Библиотеки

При анализе и обработке датасета, а также для решения поставленных задач использовались библиотеки:

- pandas;
- numpy;
- matplotlib;
- scikit-surprise [5].

Данные библиотеки позволили полностью покрыть спектр потребностей при выполнении работы.

4. Исследовательский раздел

4.1 Условия исследований

Исследование проводилось на персональном вычислительной машине со следующими характеристиками:

- процессор Apple M1 Pro,
- операционная система Ventura 13.5.2,
- 32 Гб оперативной памяти.

Временные затраты определялись с использованием библиотеки time.

Оценки RMSE и MAE определялись внутренними средствами библиотеки scikit-surprise.

4.2 Зависимость времени исполнения алгоритмов от значения параметра регуляризации

На рисунке 4.1 представлен график зависимости времени исполнения алгоритмов от значения параметра регуляризации.

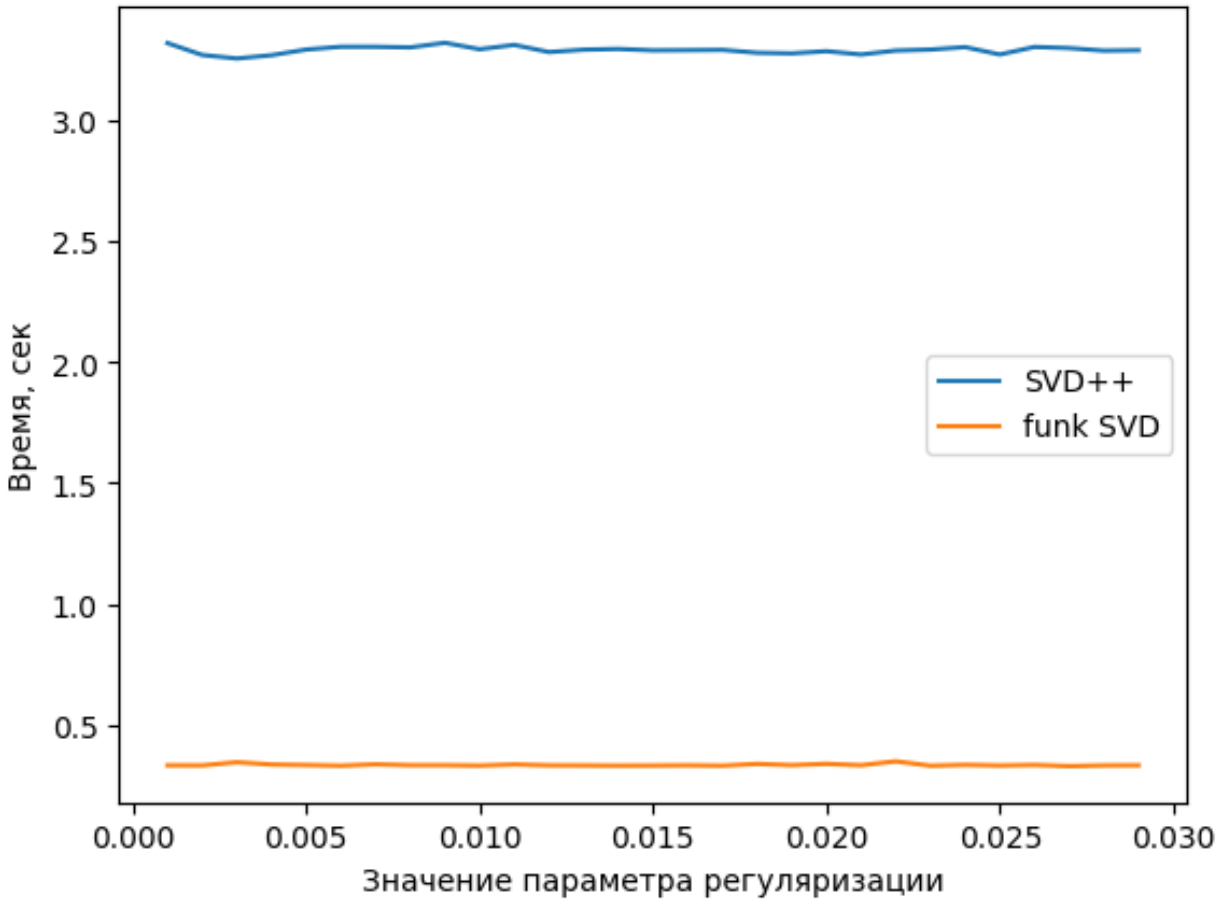

Рис. 4.1: График зависимости времени исполнения алгоритмов от значения параметра регуляризации.

4.3 Зависимость времени исполнения алгоритмов от значения параметра скорости обучения

На рисунке 4.2 представлен график зависимости времени исполнения алгоритмов от значения параметра скорости обучения.

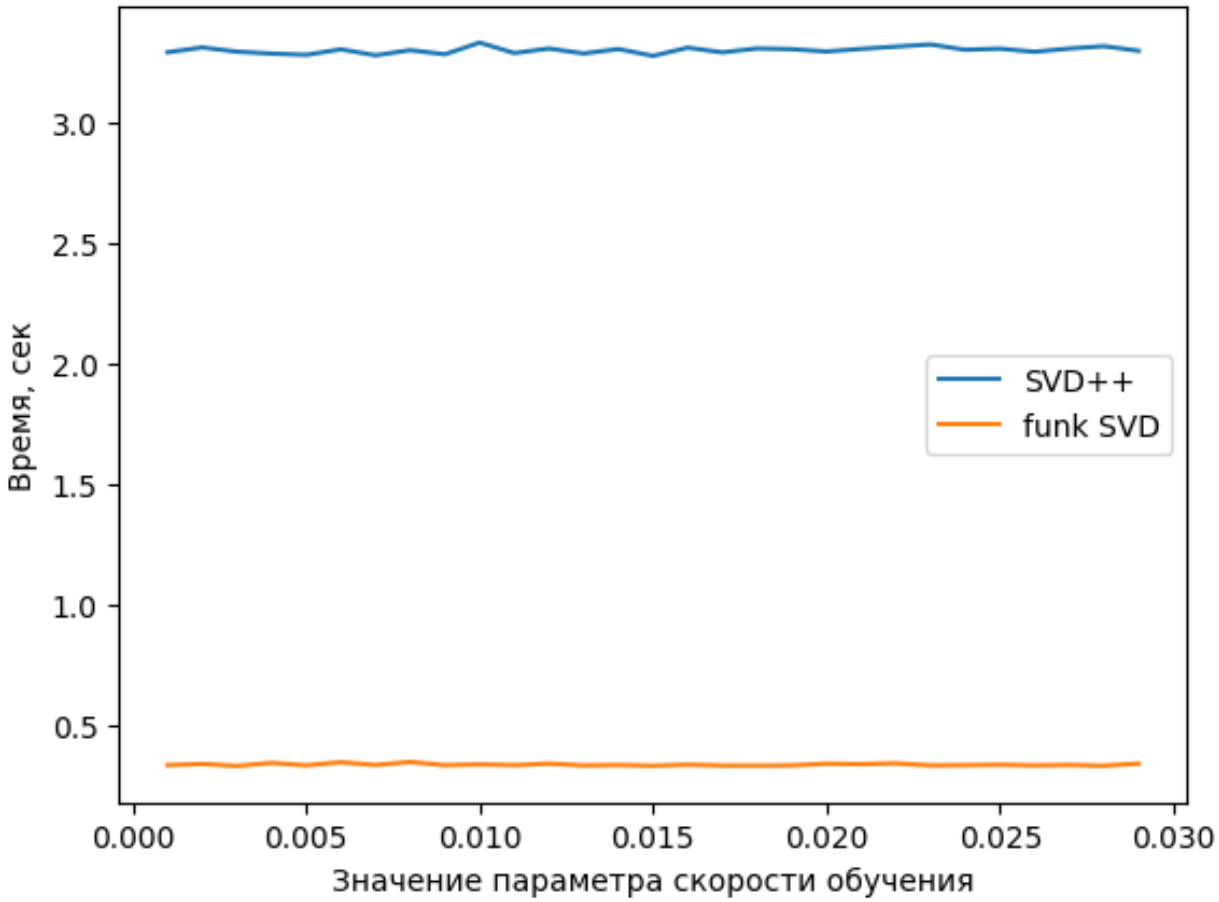

Рис. 4.2: График зависимости времени исполнения алгоритмов от значения параметра скорости обучения.

4.4 Зависимость значения метрики RMSE алгоритмов от значения параметра регуляризации

На рисунке 4.3 представлен график зависимости значения метрики RMSE от значения параметра регуляризации.

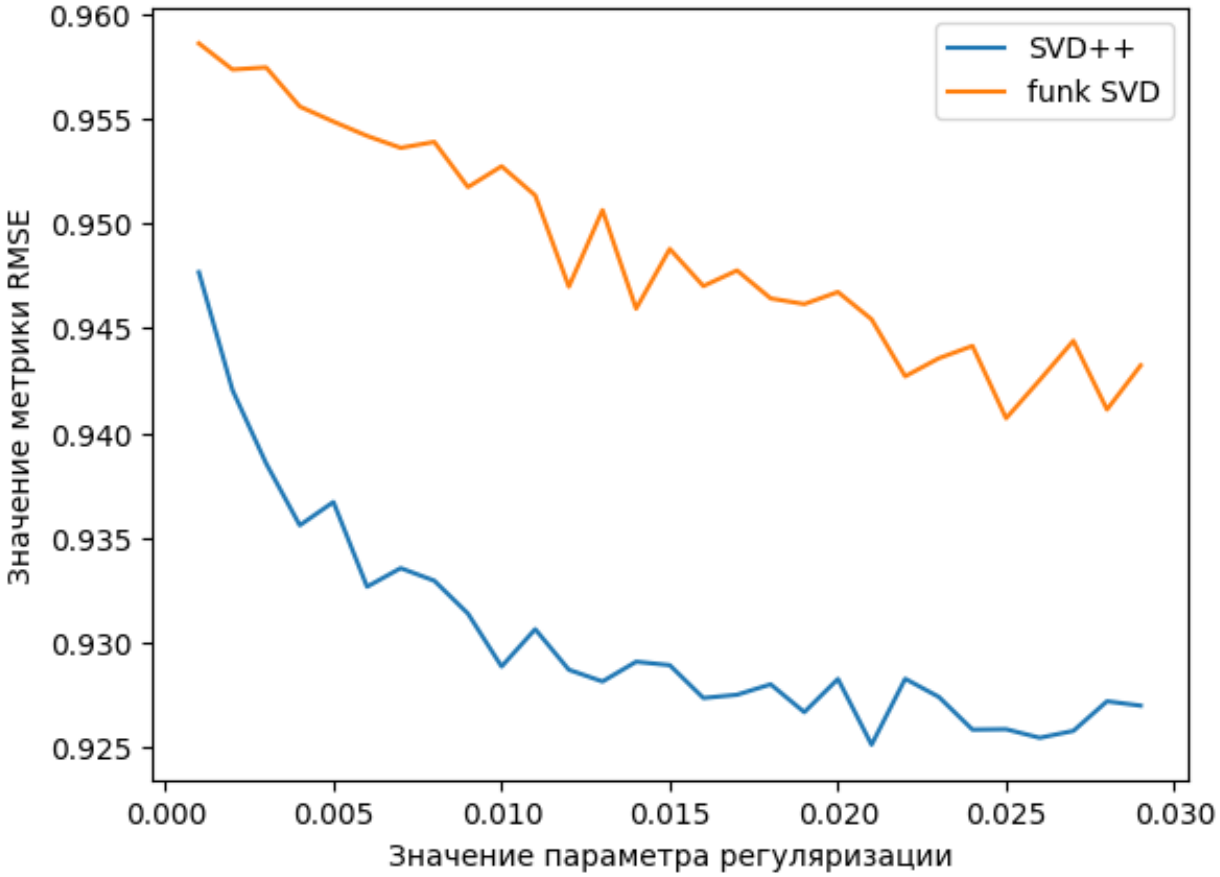

Рис. 4.3: График зависимости значения метрики RMSE от значения параметра регуляризации.

4.5 Зависимость значения метрики RMSE алгоритмов от значения параметра скорости обучения

На рисунке 4.4 представлен график зависимости значения метрики RMSE от значения параметра скорости обучения.

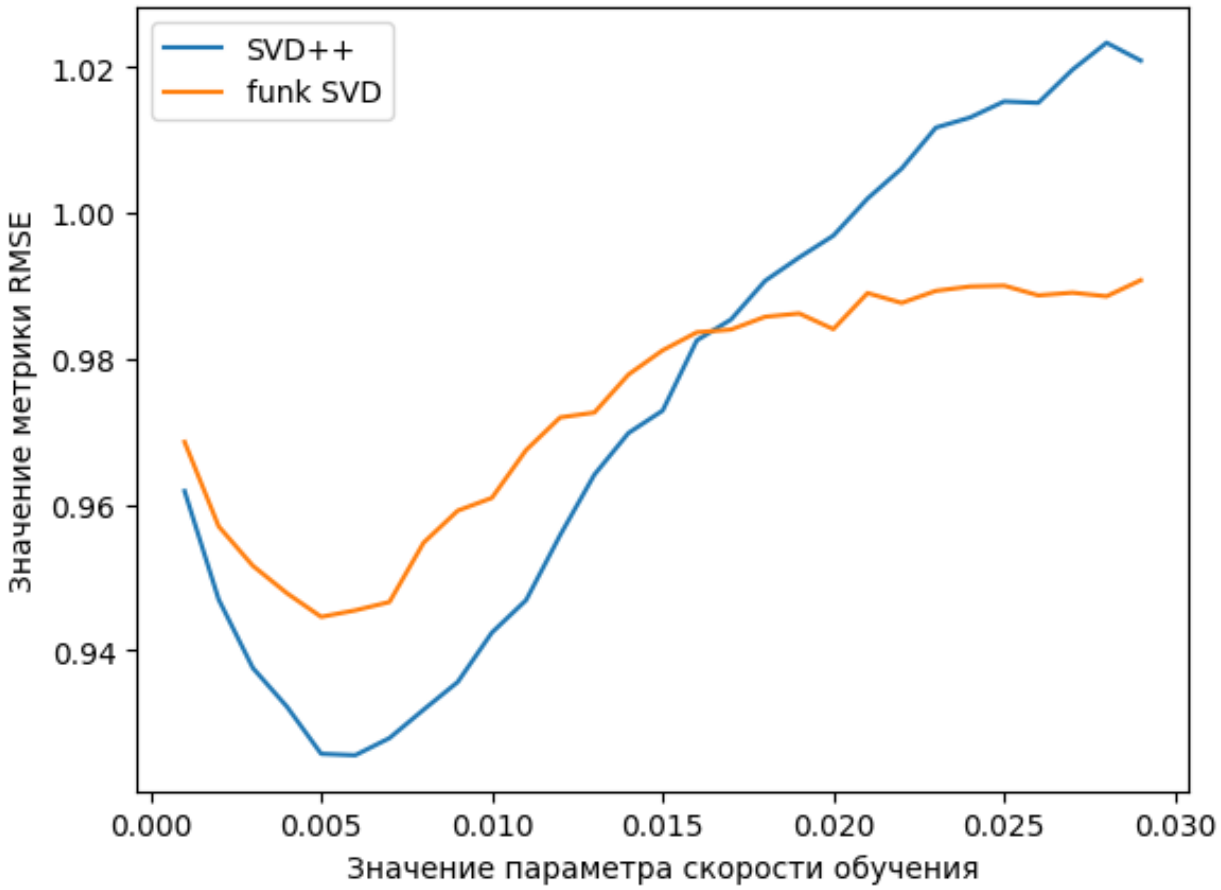

Рис. 4.4: График зависимости значения метрики RMSE от значения параметра скорости обучения.

4.6 Зависимость значения метрики MAE алгоритмов от значения параметра регуляризации

На рисунке 4.5 представлен график зависимости значения метрики MAE от значения параметра регуляризации.

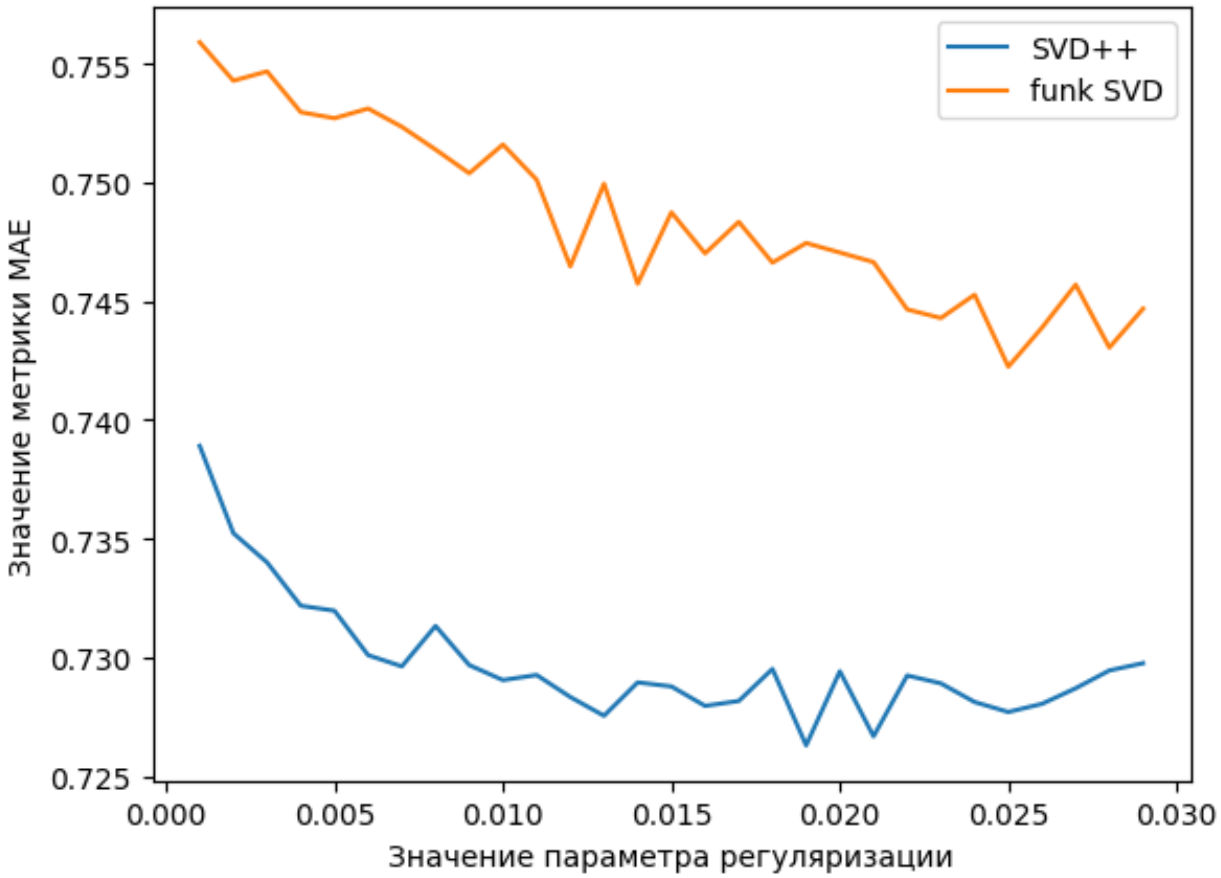

Рис. 4.5: График зависимости значения метрики MAE от значения параметра регуляризации.

4.7 Зависимость значения метрики MAE алгоритмов от значения параметра скорости обучения

На рисунке 4.6 представлен график зависимости значения метрики MAE от значения параметра скорости обучения.

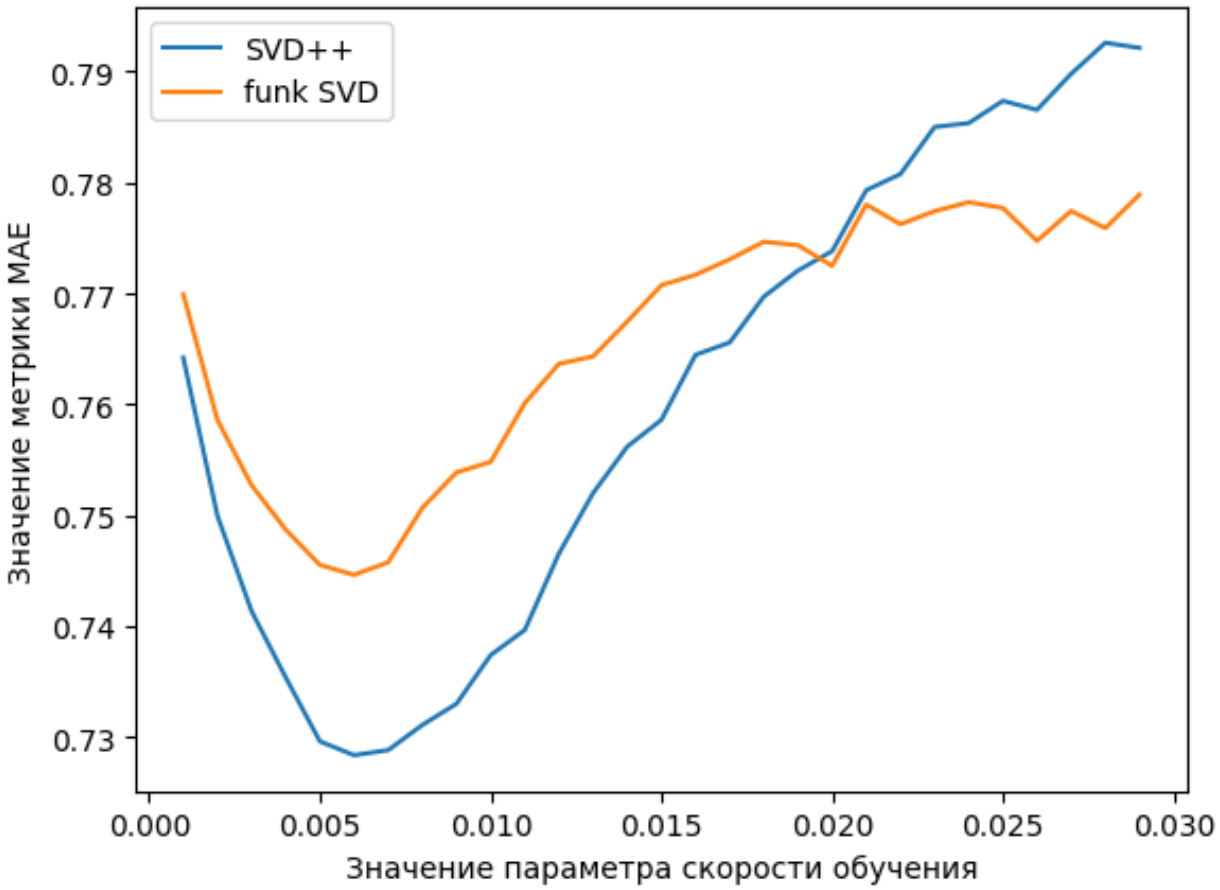

Рис. 4.6: График зависимости значения метрики MAE от значения параметра скорости обучения.

Заключение

В результате проведенных исследований заметно, что SVD++ с включенным кэшированием на заданном датасете работает заметно быстрее, чем Funk SVD, и при изменении значения параметра регуляризации, так и при изменении значения параметра скорости обучения.

Также стоит отметить, что SVD++ при изменении параметра регуляризации показывает метрики RMSE и MAE меньше Funk SVD, однако при изменении параметра скорости обучения на значениях ≈ 0.16 для RMSE и ≈ 0.20 для MAE он начинает уступать по точности Func SVD.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы было проведено сравнение алгоритмов коллаборативной фильтрации по пользователю и по объекту.

Были решены следующие задачи:

- приведено описание алгоритмов;
- приведено описание используемых для исследования данных;
- приведены зависимости скорости работы алгоритмов от заданных параметров.

В результате проведенных исследований стало известно, что SVD++ с включенным кэшированием на заданном датасете работает заметно быстрее, чем Funk SVD, и при изменении значения параметра регуляризации, так и при изменении значения параметра скорости обучения.

Также SVD++ при изменении параметра регуляризации показывает метрики RMSE и MAE меньше Funk SVD, однако при изменении параметра скорости обучения на значениях ≈ 0.16 для RMSE и ≈ 0.20 для MAE он начинает уступать по точности Func SVD.

Список литературы

1. Koren Y. Bell R. Volinsky C. MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS // IEEE Computer. 2009. № 42.

2. Cornell University: An introduction to Matrix factorization and Factorization Machines in Recommendation System, and Beyond by Yuefeng Zhang [Электронный ресурс]. Режим доступа: https://arxiv.org/abs/2203.11026 (дата обращения 16.09.2023).

3. Grouplens: MovieLens 100K Dataset [Электронный ресурс]. Режим доступа: https://grouplens.org/datasets/movielens/100k/ (дата обращения 16.09.2023).

4. Python official page [Электронный ресурс]. Режим доступа: https://www.python.org/ (дата обращения 10.05.2023).

5. Scikit-surprise: official PyPl project page [Электронный ресурс]. Режим доступа: https://pypi.org/project/scikit-surprise/ (дата обращения 16.09.2023).