Московский Государственный Университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Фоминская Галина
кафедра ММП, группа 317
2018

Рекомендательные системы

1 Введение

В данной работе был проведен анализ работы алгоритма Latent Factor Model для решения задачи рекомендации книг пользователям. Для обучения использовался метод ALS, который был реализован на языке python. Эксперименты были проведены на датасете из соревнования cmcmsu317spring2018recommendation на платформе kaggle. В качестве меры качества использовалась метрика RMSE. Для локального тестирования использовалась валидация на отложенной выборке.

2 Постановка задачи

Даны множества пользователей и книг. Каждой книге некоторые пользователи проставили рейтинги. Требуется для каждого пользователя и для каждой книги восстановить рейтинг, который он бы поставил этой книге.

3 Эксперименты

3.1 Выбор числа компонент

На рисунке 1 приведены графики с результатами экспериментов для разного числа компонент. Видно, что метод быстрее всего сходится и дает наименьшее значение RMSE на отложенной выборке при 2 компонентах. Она была выбрана в качестве лучшей. Видно, что модели с большим числом компонент показывают плохое качество. Это может объясняться тем, что в них много параметров и поэтому они легко переобучаются и требуют дополнительной настройки коэффициентов регуляризации.

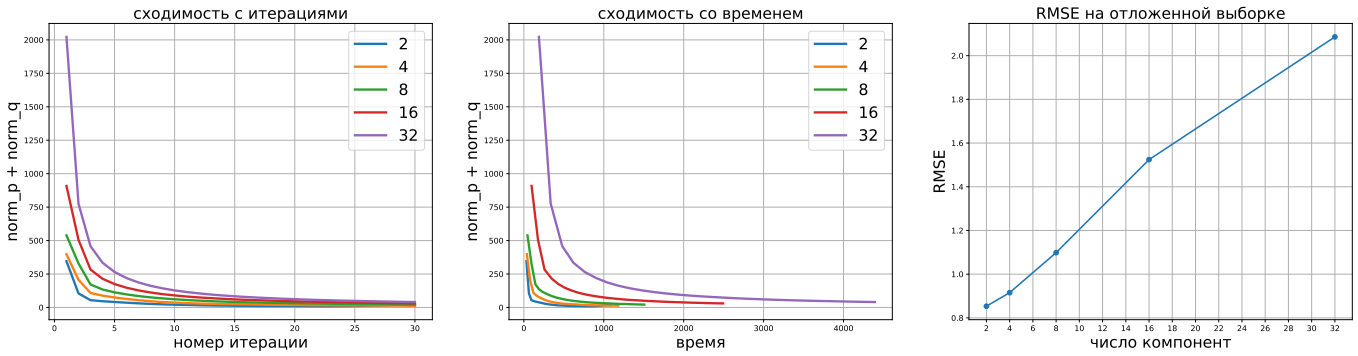

Рис. 1: Базовая модель

3.2 Центрирование оценок

Попробуем улучшить модель, учтя средние оценки пользователей и товаров: $x_{ui}^{new} = x_{ui} - \hat{x_u} - \hat{x_i}$. Сравнение качества этой модели с качеством предыдущей приведено в таблице 1. Качество оценивалось на валидационной выборке и по результату на public лидерборде. Видно, что новая модель в обоих случаях дает улучшение качества.

модель	**RMSE, валидация**	**RMSE, public**
LFM ALS	0.85340	0.84511
LFM ALS with means	**0.85294**	**0.84439**

Таблица 1: Качество моделей

3.3 Добавление других моделей

Пока мы никак не учитывали никакие признаки, кроме взаимодействия пользователя и книги. Попробуем построить модель Extra Random Forest на признаках из датасета и усреднить полученный результат с уже имеющимися. Признак author закодируем one-hot, так как он категориальный. Полученная матрица очень большая, поэтому будет учитывать только авторов, у которых больше 10 книг. На локальной валидации получился результат 0.83672, на лидерборде 0.83898.