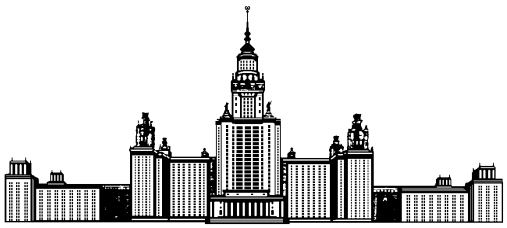


Московский государственный университет
имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Хомутов Никита Юрьевич

Системы прогнозирования предпочтений пользователей на основе их действий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф.-м.н., ведущий н.с. ВЦ РАН
О.В. Сенько

Москва, 2015

Содержание

1 Введение

В данной работе рассматривается задача коллаборативной фильтрации в условиях сильной разреженности данных и их большой размерности. Для решения задачи предлагается использовать модель низкорангового матричного разложения, а так же композиционную модель, использующую модели матричных разложений в качестве базовых.

В работе показана применимость данных моделей в задаче коллаборативной фильтрации. Показана необходимость и полезность введения регулязиции при решении задачи восстановления сильно разреженной матрицы большой размерности. Исследовано поведение обобщающей способности моделей при различных значениях внешних параметрах. Так же проведено сравнение исследуемых моделей с другими моделями.

1.1 Рекомендательные системы

Рекомендательные системы, или системы прогнозирования предпочтений – одно из наиболее популярных приложений интеллектуального анализа данных и машинного обучения в сфере интернет-бизнеса. Система анализирует поведение пользователей интернет-сервиса или магазина, после чего может прогнозировать оценку предпочтения пользователем того или иного объекта рекомендаций. На основе построенной модели поведения пользователей, функциональность сервиса может быть адаптирована для каждого конкретного пользователя. Такую особенность сервиса принято называть *персонализацией*.

Рекомендательные системы помогают пользователю ориентироваться в большом количестве ассортимента, предлагаемого сервисом. В ряде случаев, это необходимая функциональность. Например, по оценке крупнейшей базы данных о кинематографе IMDB[1], на данный момент создано более чем 3 миллиона кинофильмов и телесериалов. На сервисе Яндекс.Музыка[2] размещено (по грубым оценкам) более 50 лет непрерывного аудио-потока. Объём фондов интегрированной электронной библиотеки[3] Российской Государственной Библиотеки составляет более чем 880 тысяч уникальных названий. На практике пользователь не может оценить весь предлагаемый ассортимент, поэтому поиск интересующего контента пользователи осуществляют, используя советы от друзей, единомышленников, популярных теле- и радио-ведущих, читая газетны, рекламу.

[1] `http://imdb.com`

[2] `http://mucis.yandex.ru`

[3] `http://elibrary.rsl.ru`

Результатом работы рекомендательной системы является модель, способная предсказывать поведение каждого конкретного пользователя, основываясь на совокупности действий всех пользователей системы. Полученная модель может быть использована для различных практических задач.

Наиболее популярная задача, решаемая рекомендательной системой, это предсказание списка наиболее релевантного контента для каждого конкретного пользователя. Исключительное значение эта задача имеет для интернет-магазинов, так как предлагая пользователю товар, который его заинтересует, магазин может увеличить свой объём продаж.

Так же может стоять задача для заданного контента по выделению наиболее заинтересованных в данном контенте пользователей. Такая задача возникает у магазинов при повышении эффективности проведения промо-акций. На основании совершённых покупок промо-акция проводится только для наиболее заинтересованных пользователей.

Ещё одно практическое применение рекомендательных систем – планирование. Например, интернет-магазин Amazon использует прогноз предпочтений пользователей, чтобы привезти на склад товары, наиболее интересные пользователям в данной географической местности ещё до факта заказа товара со стороны пользователей, чтобы сократить будущие логистические издержки.

На практике задачу прогнозирования предпочтений необходимо решать в случае сильной разреженности данных

1.2 Построение рекомендательных систем

Распространены две основные стратегии создания рекомендательных систем:

- Фильтрация содержимого
- Коллаборативная фильтрация.

Фильтрация содержимого. Данных подход основан на использовании признаковых описаний, и предполагает, что про пользователей и про контент известно достаточно много информации. Например, пользователи заполняют анкеты, предоставляют демографическую информацию или ответы на определённый набор вопросов. Для объектов экспертами составляется подробное признаковое описание. Этот подход используется в проекте Music Genome Project[4] компании Pandora Media: музы-

[4] `http://pandora.com/about/mgp`

кальный аналитик оценивает каждую композицию по сотням различных музыкальных характеристик, которые могут использоваться для выявления музыкальных предпочтений пользователя. Имея подробное признаковое описание пользователей и объектов, зная историю взаимодействия пользователей и объектов, задача прогнозирования предпочтений может быть сведена к задаче обучения по прецедентам.

На практике данный подход не получил широкого распространения, так как сбор описательной информации о пользователях и объектах рекомендации, как правило, дорогостоящая процедура. Зачастую, невозможно организовать сбор данных о пользователях без ущерба качеству использования сервиса.

Коллаборативная фильтрация В данном подходе используется информация о совершённых действий пользователей в прошлом, например, информация о покупках или оценках, выставленных пользователями для объектов. Построение прогноза происходит при этом исключительно на основании взаимодействия пользователей с объектами.

Сильной мотивацией в исследовании математических моделей коллаборатиной фильтрации послужил конкурс Netflix Prize [1], проводившийся компанией Netflix. Основной деятельностю компании является кинофильмов на DVD. Целью проводимого конкурса являлось улучшение качества предсказываемой оценки пользователя к фильму. Набор данных для конкурса содержал список оценок, поставленных пользователями к фильмам. Пользователи выставляли оценки от 1 до 5. Обучающие данные содержат 100'480'507 оценок, которые 480'189 пользователей поставили 17'770 фильмам. Таким образом, для обучающих данных известно только около 1.1% возможных оценок пользователей к фильмами.

На практике приходится иметь дело с сильно разреженными данными, что обусловлено большим количеством объектов и ограниченным временем пользователя на изучение ассортимента. Например, база данных сервиса рекомендаций фильмов MovieLens[5] содержит только 0.33% возможных оценок пользователей к фильмам.

В данной работе рассматривается подход, основанный на коллаборативной фильтрации.

2 Задача коллаборативной фильтрации

В данной работе будем использовать следующие обозначения:

[5] `http://grouplens.org/datasets/movielens/`

- $\mathcal{U}$ – множество пользователей
- $\mathcal{I}$ – множество объектов
- n – количество пользователей
- m – количество объектов
- k – выборанный ранг матричного разложения
- r_{ui} – оценка, поставленная пользователем $u \in \mathcal{U}$ к объекту $i \in \mathcal{I}$
- $\mathbf{R} \in \mathbb{R}^{n \times m}$ – матрица оценок предпочтений пользователей к объектам
- $\mathcal{R} = \{(u,i) | u \in U, i \in I, r_{ui} \text{ – известна}\} \subseteq \mathcal{U} \times \mathcal{I}$ – множество пар пользователей и объектов, для которых известна оценка предпочтения
- $c_{ui} = \begin{cases} 1, (u,i) \in \mathcal{R} \\ 0, (u,i) \notin \mathcal{R} \end{cases}$ – индикатор известности оценки пользователя $u \in \mathcal{U}$ к объекту $i \in \mathcal{I}$
- $\mathbf{C} = (c_{ui})_{u=1\ldots n}^{i=1\ldots m}$ – матрица индикаторов известности оценки предпочтения
- $\mathcal{N}_u(u) = \{i \in \mathcal{I} | (u,i) \in \mathcal{R}\}$ – множество объектов, для которых известна оценка предпочтения пользователя $u \in \mathcal{U}$
- $\mathcal{N}_i(i) = \{u \in \mathcal{U} | (u,i) \in \mathcal{R}\}$ – множество пользователей, для которых известна оценка предпочтения к объекту $i \in \mathcal{I}$

По имеющимся данным — известным оценкам предпочтения пользователей — требуется восстановить все оценки предпочтения. На задачу коллаборативной фильтрации можно смотреть как на задачу заполнения пропущенных значений в матрице.

Формальная постановка задачи. Пусть задано множество пользователей $\mathcal{U}$, множество объектов $\mathcal{I}$. Пусть некоторого множества пар пользователей и объектов $\mathcal{R}$ известны оценки предпочтения. Требуется по обучающему набору $\{\mathcal{R}, \{r_{ui} | (u,i) \in \mathcal{R}\}$ построить регрессионую модель $\hat{r}(u,i)$, предсказывающую оценку предпочтения для произвольной пары $(u,i) \in \mathcal{U} \times \mathcal{I}$

Для решения задачи коллаборативной фильтрации существуют различные алгоритмы, которые используется во многих рекомендательных системах. В данном подходе для активного пользователя (или предмета) подбирается подгруппа пользователей (или предметов) схожих с ним. Комбинация весов и оценок подгруппы используется для прогноза оценок активного пользователя. Ярким примером алгоритма коллаборативной фильтрации является взвешивание оценки предпочтения по пользователям (used-based):

$$\hat{r}_{ui} = \bar{r}_u + \frac{1}{\sum_{v \in \mathcal{N}_i(i)} |\rho(u,v)|} \sum_{v \in \mathcal{N}_i(i)} \rho(u,v)(r_{vi} - \bar{r}_v)$$

и объектам (item-based):

$$\hat{r}_{ui} = \bar{r}_i + \frac{1}{\sum_{j \in \mathcal{N}_u(v)} |\rho(i,j)|} \sum_{j \in \mathcal{N}_u(i)} \rho(i,j)(r_{uj} - \bar{r}_j)$$

Здесь $\rho \in \mathcal{U} \times \mathcal{U}$ (для used-based), $\rho \in \mathcal{I} \times \mathcal{I}$ (для item-based) – заранее заданные метрики схожести для пользователей и объектов соответственно. Так же

- $\bar{r}_u = \frac{1}{\mathcal{N}_u(u)} \sum_{i \in \mathcal{N}_u(u)} r_{ui}$ – среднее значение оценки предпочтения для данного пользователя
- $\bar{r}_i = \frac{1}{\mathcal{N}_i(i)} \sum_{u \in \mathcal{N}_i(i)} r_{ui}$ – среднее значение оценки предпочтения для данного объекта

Мера схожести $\rho(u,v)$ (и аналогичная для объектов) вычисляется по матрице оценок R, либо с использованием дополнительной информации о пользователях (и объектах соответственно). Мера схожести является важным параметов алгоритма. Наиболее употребимые метрики схожести – корреляция Пирсона:

$$\rho(u,v) = \frac{\sum_{i \in \mathcal{N}_u(u) \cap \mathcal{N}_u(v)} (r_{ui} - \hat{r}_u)(r_{vi} - \hat{r}_v)}{\sqrt{\sum_{i \in \mathcal{N}_u(u) \cap \mathcal{N}_u(v)} (r_{ui} - \hat{r}_u)^2 \sum_{i \in \mathcal{N}_u(u) \cap \mathcal{N}_u(v)} (r_{vi} - \hat{r}_v)^2}}$$

и косинусное расстояние соответствующих строк (столбцов) матрицы оценок предпочтений:

$$\rho(u,v) = \frac{\sum_{i \in \mathcal{N}_u(u) \cap \mathcal{N}_u(v)} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{N}_u(u) \cap \mathcal{N}_u(v)} r_{ui}^2} \sqrt{\sum_{i \in \mathcal{N}_u(u) \cap \mathcal{N}_u(v)} r_{vi}^2}}$$

Подобные алоритмы хороши для однократного вычисления рекомендаций на распределённом кластере, хорошо адаптируются на вычислительную архитектуру MapReduce. Но такие алгоритмы плохо подходят для оперативного обновления рекомендаций при поступлении новых данных. Настройка меры схожести для задачи из конкретной предметной области является скорее искусством, нежели отлаженной технологией. Существенным недостатком данного подхода является неинтерпретируемость прогнозируемых оценок предпочтения.

3 Оценка качества

Определим критерии качества решения задачи коллаборативной фильтрации (или задачи восстановления матрицы разреженной).

Пусть задан метод обучения

$$\mu : \mathbb{R}^{m,n} \times 2^{\mathcal{U}\times\mathcal{I}} \to (\mathcal{U} \times \mathcal{I} \to \mathbb{R})$$

То есть μ принимает на вход известные элементы матрицы $\mathbf{R}$, а на выходе получаем регрессионную модель, определённую на индексах матрицы $\mathbf{R}$.

Пусть задана матрица $\mathbf{R}$, множество известных элементов $\mathcal{R}$. Тогда, разделяя множество $\mathcal{R} = \mathcal{R}_{train} \cup \mathcal{R}_{test}$ на обучающую и тестовую части, можем оценить потери качества восстановления матрицы $\mathbf{R}$ на множестве $\mathcal{R}_{test}$, если на обучении для модели было доступно множество известных элементов $\mathcal{R}_{train}$.

$$L(\mu, \mathbf{R}, \mathcal{R}_{train}, \mathcal{R}_{test}) = \sum_{(u,i)\in\mathcal{R}_{test}} (\mu(\mathbf{R}, \mathcal{R}_{train})(u,i) - r_{ui})^2$$

Данный подход разделения выборки на обучающую и тестовую распостранён при оценке качества модели. В дальнейшем для этих целей будем использовать скользящий контроль.

4 Низкоранговое разложение

Рассмотрим матрицу $\mathbf{R} \in \mathbb{R}^{n\times m}$. Если ранг матрицы $rank(\mathbf{R}) \leq k$, то найдутся такие $X \in \mathbb{R}^{n\times k}, Y \in \mathbb{R}^{m\times k}$, такие что:

$$\mathbf{R} = \mathbf{X}\mathbf{Y}^T$$

В том случае, если $rank(\mathbf{R}) > k$, то *приближением* ранга k матрицы $\mathbf{R}$ будем называть

$$\mathbf{X}\mathbf{Y}^T = \hat{\mathbf{R}} \approx \mathbf{R}$$

,

где $\mathbf{X} \in \mathbb{R}^{n \times k}, \mathbf{Y} \in \mathbb{R}^{m \times k}$ – матрицы-факторы приближения. Приближение ранга k позволяет представить в памяти вычислительного устройства матрицу размера $n \times m$, используя объём памяти $O(k(n+n))$ вместо $O(nm)$. Рассмотрим два способа получения матричного приближения заданного ранга k

Сингулярное разложение Оптимальное приближение ранга k с точки зрения суммы квадратичных отклонений всех элементов матрицы R находится из решения оптимизационной задачи

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(r_{ij} - \mathbf{x}_i^T\mathbf{y}_j)^2 = \|\mathbf{R} - \mathbf{X}\mathbf{Y}^T\|_2^2 \to \min_{\mathbf{X},\mathbf{Y}}$$

здесь r_{ij} – элемент i-й строки, j-го столбца матрицы $\mathbf{R}$; $\mathbf{x}_i$ – i-я строка матрицы $\mathbf{X}$; $\mathbf{y}_j$ – j-я строка матрицы $\mathbf{Y}$

Метод, позволяющий построить данное матричное разложение, называется сингулярным разложением (SVD). Известно [3], что все локальные оптимумы данной оптимизационной задачи являются глобальными. Матрицы $\mathbf{X}, \mathbf{Y}$ при этом определены с точностью до невырожденного линейного пребразования $A \in \mathbb{R}^{k \times k}$, так как:

$$(\mathbf{X}\mathbf{A})(\mathbf{Y}\mathbf{A}^{-T})^T = \mathbf{X}(\mathbf{A}\mathbf{A}^{-1})\mathbf{Y}^T = \mathbf{X}\mathbf{Y} \tag{1}$$

Если известны все элементы матрицы $\mathbf{R}$, для которой мы хотим найти низкоранговое разложение, то применим стандартный метод SVD. При этом мы можем получить сингулярные числа матрицы, и по ним определить оптимальный ранг разложения, допускающее квадратичную ошибку не больше наперёд заданной. Но если же не все элементы матрицы известны, то метод сингулярного разложения не применим для поиска низкорангового приближения.

В рассматриваемой задаче коллаборативной фильтрации, как правило, матрица $\mathbf{R}$ сильно разрежена, поэтому на практике приходится использовать другие методы поиска низкорангового разложения для прогнозирования оценки предпочтения.

Взвешенное низкоранговое разложение В задаче коллаборативной фильтрации матрица $\mathbf{R}$ не известна полностью (то есть содержит пропуски). В таком случае имеет смысл от задачи 1 перейти к задаче

минимизации суммы квадратичных отклонений для известных элементов матрицы:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}(r_{ij} - \mathbf{x}_i^T\mathbf{y}_j)^2 = ||\mathbf{C} \odot (\mathbf{R} - \mathbf{X}\mathbf{Y}^T)|| \to \min_{X,Y} \quad (2)$$

Здесь $\odot$ – оператор поэлементного матричного множения.

Из результатов исследования низкоранговых приближений в работе [3] известно, что в случае $rank(\mathbf{C}) = 1$ в задаче 2 все локальные оптимумы являются глобальными. Однако, если $rank(\mathbf{C}) > 1$, то задача может иметь сколько угодно локальных оптимумов, не являющихся глобальными.

Именно случай $rank(\mathbf{C}) > 1$ представляет практически значимый интерес. С учётом сложности глобальной оптимизации задачи 2, при её решении ограничиваются поиском локального оптимума.

4.1 Случаный выбор элементов матрицы

В работе [4] доказана следующая теорема:

Теорема 1. Пусть имеется матрица $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$. Пусть ранг матрицы $rank\mathbf{M} = r$. Зафиксируем m – число известных элементов матрицы $\mathbf{M}$. Пусть

$$\mathcal{O}_m = \{\{(i_k, j_k)\}_{k=1}^m | \forall p \neq q : (i_p, j_p) \neq (i_q, j_q), \forall p : i_p \in \mathbb{N}_{n_1}, \forall p : j_p \in \mathbb{N}_{n_2}\}$$

– множество различных наборов длины m из индексов матрицы M. Здесь $\mathbb{N} = \{1, \ldots, d\}$ – первые d натуральных чисел.

Пусть на множестве M задана вероятностная мера, соответствующая равномерному распределению , то есть

$$\forall \Omega_1 \in \mathcal{O}_m \forall \Omega_2 \in \mathcal{O}_m : P(\Omega_1) = P(\Omega_2)$$

Обозначим $n = max\{n_1, n_2\}$. Введём случайную величину $\xi(\Omega \in \mathcal{O}_m)$, принимающую значение 1, если при известных элементах матрицы $\mathbf{M}$ заданных множеством Ω можем в точности и единственным образом восстановить все неизвестные значения матрицы $\mathbf{M}$ (при этом зная ранг матрицы), и 0 – в противном случае, когда не существует единственного решения.

Тогда утверждение теоремы состоит в следующем. Пусть выполнено неравенство:

$$m \geq Cn^{5/4} r \log n$$

Тогда

$$P(\xi = 1) \geq 1 - cn^{-3} \log n$$

Здесь C, c – константы, определённые автором теоремы.

Таким образом, в предположении низкого ранга, матрица большого размера может быть восстановлена из полностью случайно выбранных элементов. Данное утверждение можно рассматривать как мотивацию для использования низкоранговых приближений для задачи восстановления матриц, содержащих пропуски.

5 Модель матричного разложения

Рассмотрим модель низкорангового разложения для решения задачи коллаборативной фильтрации. Пусть задан ранг k. Пусть заданы матрицы $\mathbf{P} \in \mathbb{R}^{n \times k}, \mathbf{Q} \in \mathbb{R}^{m \times k}$, $\Theta = (\mathbf{P}, \mathbf{Q})$. Определим прогноз оценки предпочтения как:

$$\hat{r}(u, i,) = \mathbf{p}_u^T \mathbf{q}_i$$

Здесь $\mathbf{p}_u$ – u-я строка матрицы P_u, $\mathbf{q}_i$ – i-я строка матрицы Q. Вектора $\mathbf{p}_u$ и $\mathbf{q}_i$ – вектора *скрытых (латентных)* предпочтений для пользователя и объекта соответственно.

Подобное разложение встречается в задаче тематического моделирования, где стоит задача по заданному корпусу документов выделить скрытые тем. Здесь же наблюдается подобных эффект: каждый столбец матрицы $\mathbf{P}$ и $\mathbf{Q}$ соответствует одной тематике. Тематики выделяются автоматически, и столбцы матрицы P и Q могут быть использованы исследователями при изучении свойств предоставленных данных. К примеру, фильмы могут быть разделены по жанрам, и для каждой пары фильма и жанра можно указать, насколько данный фильм соответствует данному жанру. Аналогично, зритель может предпочитать одни жанры сильнее чем другие, а некоторые жанры может воспринимать негативно.

Введём регуляризацию квадратичную регуляризацию для данной модели

$$\Lambda(\Theta) = \sum_{u \in \mathcal{U}} \|\mathbf{p}_u\|^2 + \sum_{i \in \mathcal{I}} \|\mathbf{q}_i\|^2$$

Необходимость регуляризации связана с большой размерностью параметров модели, что может приводить к переобучению. При построении регрессионных моделей хорошей практикой является введение регуляризации в таких случаях. Приведённые ниже формулы можно использовать и при отсутствии регуляризации, задав коэффициент регуляризации равным нулю. Далее, в экспериментальной части, мы изучим необходимость введения регуляризации.

Будем обучать модель, оптимизируя квадратичную ошибку на обучающих данных с учётом регуляризации:

$$L(\Theta) = \sum_{(u,i)\in\mathcal{R}} (\mathbf{p}_u^T \mathbf{q}_i - r_{ui})^2 + C\Lambda(\Theta) \to \min_{\Theta} \tag{3}$$

Будем решать данную оптимизационную задачу методом покоординатного спуска. При фиксированном $\mathbf{Q}$ задача 3 разбивается на независимые по $\mathbf{p}_u$ подзадачи:

$$\sum_{i:(u,i)\in\mathcal{R}} (\mathbf{p}_u^T \mathbf{q}_i - r_{ui})^2 + C\|\mathbf{p}_u\|^2 \to \min_{\mathbf{p}_u}, \forall u \in \mathcal{U} \tag{4}$$

Данная задача сводится к решению линейного уравнения:

$$\left(\sum_{i:(u,i)\in\mathcal{R}} \mathbf{q}_i \mathbf{q}_i^T + C\mathbf{I} \right) \mathbf{p}_u - \sum_{i:(u,i)\in\mathcal{R}} \mathbf{q}_i^T r_{ui} = \mathbf{A}_u \mathbf{p}_u - \mathbf{b}_u = 0 \tag{5}$$

Аналогично, совершив координатный спуск по переменной $\mathbf{P}$, совершаем спуск по переменной $\mathbf{Q}$. Задача так же разбивается на независимые подзадачи.

$$\left(\sum_{u:(u,i)\in\mathcal{R}} \mathbf{p}_u \mathbf{p}_u^T + C\mathbf{I} \right) \mathbf{q}_i - \sum_{u:(u,i)\in\mathcal{R}} \mathbf{p}_u^T r_{ui} = \mathbf{A}_i \mathbf{q}_i - \mathbf{b}_i = 0 \tag{6}$$

В результате, задача оптимизации 3 может быть решена итерационным алгоритмом.

5.1 Программная реализация

Алгоритмы обучения и предсказания для данной модели были реализованы на языке Python с использованием следующих сторонних библиотек:

- NumPy – поддержка операций линейной алгебры

Исходные параметры: $\mathbf{R}, \mathcal{R}, k, C, D$
Результат: $\mathbf{P}, \mathbf{Q}$
Инизиализировать параметры $\mathbf{P}, \mathbf{Q}$;
$d := 1$;
цикл $d \leq D$ **выполнять**
  $u := 1$;
  цикл $u \leq n$ **выполнять**
    $\mathbf{A}_u := \sum_{i:(u,i)\in\mathcal{R}} \mathbf{q}_i\mathbf{q}_i^T + C\mathbf{I}$;
    $\mathbf{b}_u = \sum_{i:(u,i)\in\mathcal{R}} \mathbf{q}_i^T r_{ui}$ $p_u := \text{МНК}(\mathbf{A}_u, \mathbf{b}_u)$;
    $u := u + 1$
  конец цикла
  $i := 1$;
  цикл $i \leq m$ **выполнять**
    $\mathbf{A}_i := \sum_{u:(u,i)\in\mathcal{R}} \mathbf{p}_u\mathbf{p}_u^T + C\mathbf{I}$;
    $\mathbf{b}_i = \sum_{u:(u,i)\in\mathcal{R}} \mathbf{p}_u^T r_{ui}$ $q_i := \text{МНК}(\mathbf{A}_i, \mathbf{b}_i)$;
    $i := i + 1$
  конец цикла
конец цикла

Алгоритм 1: Алгоритм обучения модели матричного разложения

- SciPy - подмодуль sparse – поддержка операций с разреженными матрицами
- mkl, accelerate – оптимизация модулей NumPy и SciPy для работы на многопоточных системах

Так как на каждом шаге координатного спуска исходная задача разбивается на множество подзадач, которые не имеют зависимости по данным, то эти подзадачи могут быть решены параллельно, что и было реализовано.

6 Ансамбль моделей матричного разложения

Данный подход использует схему бустинга для построения модели оценки предпочтения, где в качестве базовых моделей используются модели матричного разложения. В [5] предложен подход к построению композиции.

Общая схема бустинга. Ансамбль моделей строится в ходе итерационного процесса. На первом шаге строится модель на исходных данных. На k-м шаге происходит процесс перевзвешивания выборки, веса вычисляются по результатам полученных ошибок коллективного (ансамбля) предиктора (модели прогнозирования предпочтения), построенного на предыдущем, $k-1$-м шаге. Затем обучается индивидуальный предиктор на перевзвешенной выборке. Далее выбирается вес для нового построенного индивидуального предиктора, с которым он будет учитываться в коллективном предикторе.

Введём обозначения:

- $\hat{r}_{u,i}^{(k)}$ – прогнозируемая индивидуальным предиктором, построенным на k-м шаге, оценка предпочтения пользователя $u \in \mathcal{U}$ к предмету $i \in \mathcal{I}$
- $\hat{r}^{(k)}$ – модель индивидуального предиктора, построенного на k-м шаге
- $\tilde{r}_{u,i}^{(k)}$ – прогнозируемая коллективным предиктором, построенным на k-м шаге, оценка предпочтения пользователя $u \in \mathcal{U}$ к предмету $i \in \mathcal{I}$
- $\tilde{r}^{k}$ – модель коллективного предиктора, построенного на k-м шаге
- γ_k – вес k-го индивидуального предиктора в ансамбле
- $w_{ui}^{(k)}$ – вес объекта $r_{ui}, (u,i) \in \mathcal{R}$ в перевзвешенной на k-й итерации построения ансамбля выборке

Соотношение между коллективным и индивидуальными предикторами следующее:

$$\tilde{r}_{ui}^{(k)} = \frac{1}{\sum_{t=1}^{k} \gamma_t} \sum_{l=1}^{k} \gamma_l \hat{r}_{ui}^{(l)}$$

В процессе исследований была предложена следующая схема модель. Пусть в процессе построения композиции строится T моделей.

1. На первом шаге веса элементов выборки полагаются одинаковыми и равными $w_{u,i}^{(1)} = 1$
2. На k-м шаге построения композиции происходит выбор весов для элементов выборки:

$$w_{ui}^{(k)} = 1 + \eta |r_{ui} - \hat{r}_{ui}^{(k-1)}|$$

Здесь η – свободный параметр, который дальше будем называть *коэффициентом несглаживания.*

3. Затем обучается модель матричного разложения заданного ранга на взвешенной выборке

$$Q^{(k)}(\tilde{r}) = \sum_{(u,i)\in\mathcal{R}} w_{ui}^{(k)}(\tilde{r}_{ui} - r_{ui})^2 \to \min_{\tilde{r}}$$

Модель обучается с учётом регуляризации

4. Выбирается γ_k исходя из решения оптимизационной задачи

$$\sum_{(u,i)\in\mathcal{R}} \left(\frac{1}{1+\gamma_k} \left(\hat{r}_{ui}^{(k-1)} + \gamma_k \tilde{r}_{ui}^{(k)} \right) - r_{ui} \right)^2 \to \min_{\gamma_k}$$

Пусть

$$\alpha_k = \min(1-\varepsilon, \left| \frac{\sum_{(ui)\in\mathcal{R}} (\tilde{r}_{ui}^{(k)} - \hat{r}_{ui}^{(k-1)})}{\sum_{(ui)\in\mathcal{R}} (r_{ui} - \hat{r}_{ui}^{(k-1)})} \right|)$$

Где ε – малая величина. В дальнейших экспериментах выбрано $\varepsilon = 0.1$. Тогда выберем $\gamma_k = \frac{\alpha_k}{1-\alpha_k}$

5. Шаги 2-4 повторяются, пока не создана модель композиции из T моделей.

6. Все веса индивидуальных предикторов γ_k устанавливаем одинаковыми и равными $\frac{1}{T}$.

Таким образом, в процессе построения модели, индивидуальные предикторы учитываются с различными весами, рассчитаными исходя из получаемой ошибки коллективного предиктора и отклонения нового индивидуального предиктора от коллективного. Затем, при построении финальной композиционной модели, индивидуальные предикторы уже учитываем равнозначно. Экспериментально такая схема показала лучшую обобщающую способность по сравнению со схемой, где веса γ индивидуальных предикторов сохранялись неизменными с того момента, как были рассчитаны на этапе построения последовательности индивидуальных предикторов.

7 Эксперименты с прогнозированием оценки предпочтения

Данные Для проведения экспериментов были взяты реальные данные MovieLens 100k[6]. Данные содержат 100 000 оценок от 943 пользователей на 1682 фильмов. Данный набор отобран так, что каждый пользователь оценил не менее 20 фильмов. Оценки градуируются шкалой от 0.5 до 5 с шагом 0.5.

Для оценки качества модели исходная выборка была разбита на 5 равных частей, из чего было составлено 5 пар выборок для обучения и тестирования (в пропорции 80% к 20%). Качество модели оценивается на скользящем контроле на данных разбиениях.

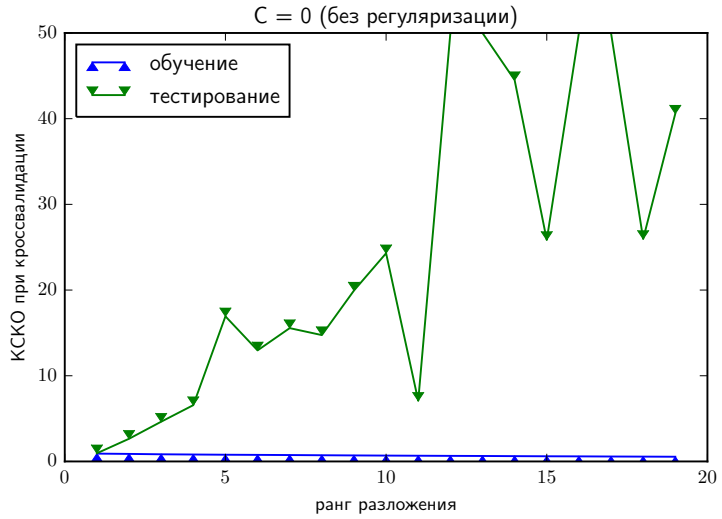
Чтобы обеспечить повторяемость результатов экспериментов, параметры $\mathbf{P}$ и $\mathbf{Q}$ для модели матричного разложения инициализировались при заранее заданном инициализационном зерне генератора псевдослучайных чисел. Элементы матрицы $\mathbf{P}$ и $\mathbf{Q}$ инициализировались с помошью семлирования из равномерного распределения на отрезке $[-\sigma; +\sigma], \sigma = 0.3$

7.1 Влияние параметра регуляризации на качество обучения

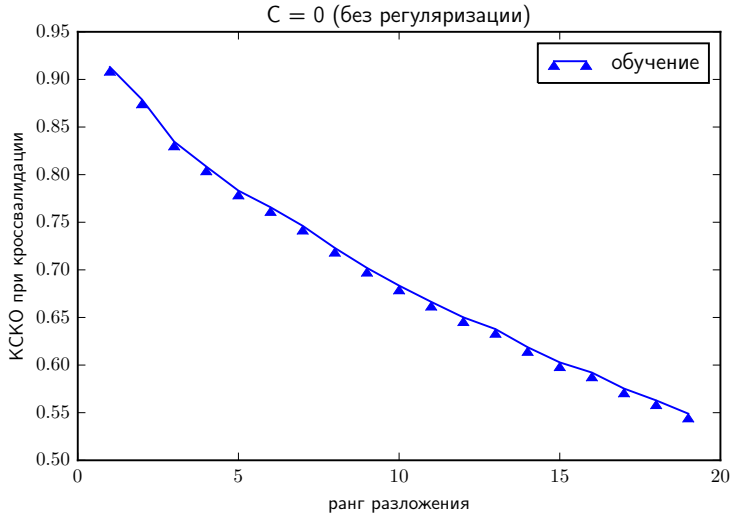
Обучим модель матричного разложения с параметром регуляризации $C = 0$ при различных значениях ранга разложения. Каждая модель обучалась с помощью 20-кратного применения шагов покоординатного спуска. На рис. 1 отображены результаты эксперимента. На оси ординат показано среднее значение функционала ошибки КСКО (корня из среднеквадратичного отклонения), полученное на скользящем контроле. На верхнем графике отображены ошибки на обучающей выборке и на тестовой. Как видим, ошибка на обучающей выборке получилась неустойчивой. Все элементы матрицы принимают значения от 0.5 до 5, но значение КСКО значительно превышает эти значения. Следовательно, нерегуляризованная модель неустойчива к прореживанию элементов матрицы, и не способна предсказать новые значения.

[6] `http://grouplens.org/datasets/movielens/`

Рис. 1: Значение ошибки при отсутствии регуляризации



(a) Ошибка на тестовой и обучающей выборке



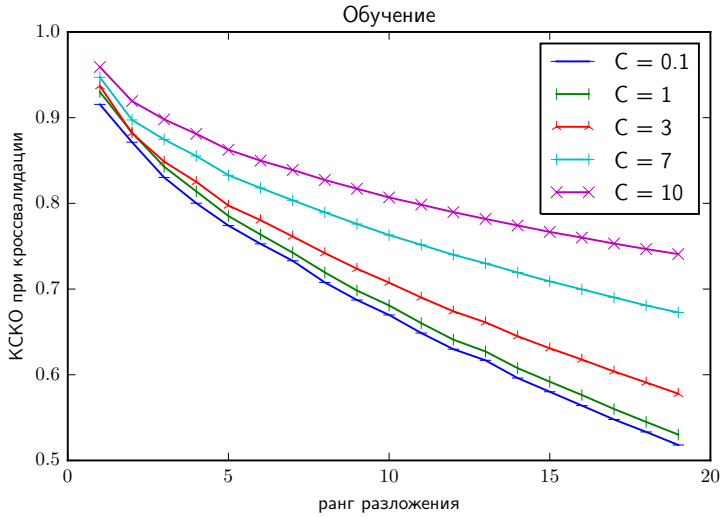
(b) Ошибка на обучающей выборке

На нижнем графике показана ошибка на обучении без ошибки на тесте. Как видим, при увеличении ранга, ошибка стабильно уменьшается. Поэтому низкоранговое нерегуляризованное матричное приближение может быть использовано для компактного хранения разреженных матриц, но не для предскания неизвестных элементов.

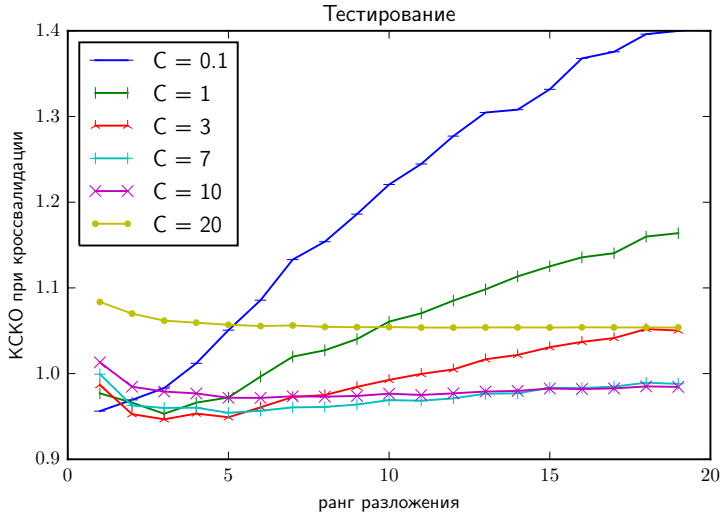
В итоге, при отсутствии регуляризации мы получили неустойчивое к

прореживанию элементов матрицы решение. То есть нерегуляризованная модель обладает плохой обобщающей способностью.

Рис. 2: Значение ошибки при различных параметрах регуляризации регуляризации



(a) Ошибка на обучающей выборке



(b) Ошибка на тестовой выборке

Исследуем поведение ошибки на обучающей и тестовой выборке при различных параметрах регуляризации. Результаты эксперимента приве-

дены на рис. 2. Как видим, введение квадратичного регуляризатора помогло значительно уменьшить ошибку на тестовой выборке в сравнении с ситуацией без регуляризатора. При этом с ростом параметра регуляризации растёт ошибка на обучении, что объяснимо: увеличивается регуляризационный штраф в функционале оптимизационной задачи на обучении, смещая точку его оптимума ближе к нулевой, и дальше от точки оптимума нерегуляризованной задачи.

Как результат, показана необходимость и полезность введения регуляризации в модель матричного разложения.

Введение квадратичной регуляризации не решает окончательной проблемы переобучения, что видно на всёх трёх линиях на втором графике рис. 2. Тем не менее, ошибка на тестовой выборке заметно ниже, чем без введения регуляризатора. Для всех трёх рассмотренных значений ошибка на тестовой выборке наблюдается яма, что говорит о том, что каждому значению C соответствует оптимальный ранг. При этом значение оптимального ранга растёт с ростом C.

При каждом фиксированном значении C при росте ранга разложения мы наблюдаем эффект переобучения.

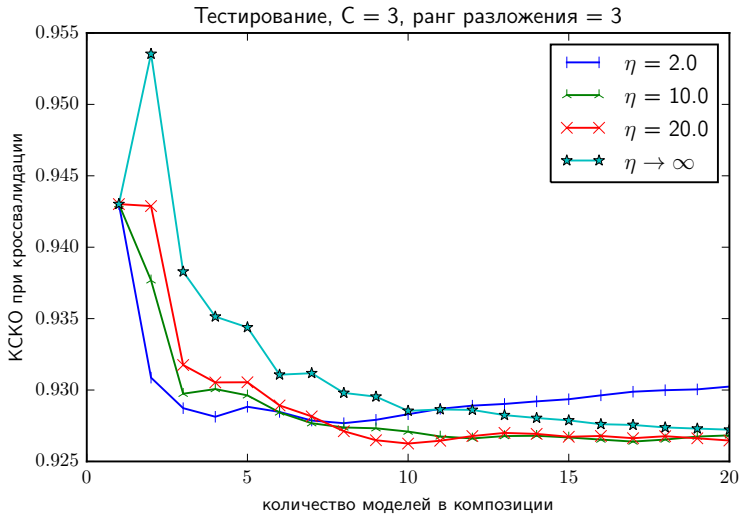
7.2 Эксприменты с моделью бустинга над матричными разложениями

На рис. 3 показаны результаты эксперимента, в котором оценивалась ошибка на тестовой и обучающей части при различных значениях коэффициента несглаживания и количества моделей в композиции.

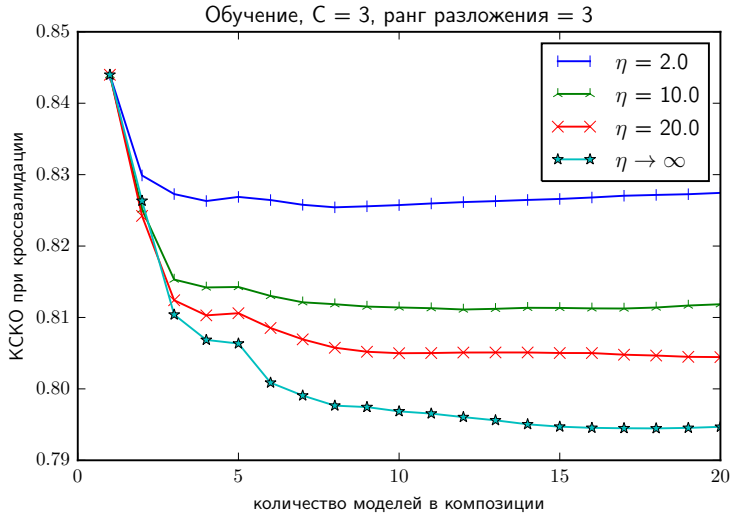
Как видно, увеличение параметра несглаживания помогает лучше подстроить модель под данные (данный параметр позволяет определить соотношение между оптимальным набором весом и набором весом, где все элементы равнозначны). Но с другой стороны, при небольшом количестве моделей композиции, обобщающая способность выше у моделей с более низким значением коэффициента несглаживания. Данный результат может быть интересен, если стоит задача построить более простую модель, обладающую хорошей обобщающей способностью. Однако, при увеличении количества моделей, обобщающая способность будет выше у моделей с более высоким значением параметра несглаживания.

При фиксирванном значении $\eta < \infty$ может наблюдаться ухудшение обобщающей способности при росте количества моделей в композиции.

Рис. 3: Модель бустинга над матричными разложениями



(a) Ошибка на тестовой выборке



(b) Ошибка на обучающей выборке

7.3 Сравнение с другими моделями

Для корректного сравнения моделей была применена следующая схема. Исходная выборка была разбита в отношении 90% к 10%. Используемое разбиение поставляется вместе с набором данных Movie Lens-100K (называется ua.base И ua.test). Первая часть использовалась для настрой-

ки моделей, вторая часть – для валидации. Далее, 90%-часть исходной выборки разбивается случайным образом на 5 равных подвыборок, из которых генерируются пять пар наборов данных вида "обучение"и "тестирование"(в отношении 80% к 20%), используемых при кроссвалидации. Внешние параметры моделей (такие как ранг матричного разложения, параметры регуляризации, количество моделей в ансамбле бустинга, параметр сглаживания) настраиваются с помощью кроссвалидации на полученных пяти парах наборах данных: внутренние параметры моделей оценивались на обучающих подвыборках, качество оценивалось на тестовых подвыборках. Затем выбирались оптимальные значения внешних параметров. Затем, при выбранных значениях внешних параметров, настраивалась итоговая модель, внутренние параметры которой были настроены по 90% разбиению исходной выборки. Затем оценивалось качество итоговой модели на валидационной подвыборке.

Модель I. Данная модель прогнозирования предпочтения исходит из следующей гипотезы: в среднем пользователь, не наблюдавший рассматриваемый предмет, поставит такую же оценку, как и в среднем – оценившие этот предмет. То есть:

$$\hat{r}_{item}(u,i) = \begin{cases} r_{ui}, & (u,i) \in \mathcal{R} \\ \frac{1}{|\mathcal{N}_i(i)|} \sum_{v \in \mathcal{N}_i(i)} r_{vi} & (u,i) \notin \mathcal{R} \end{cases}$$

Модель U. Данная модель прогнозирования предпочтения исходит из следующей гипотезы: пользователь оценивает предметы, которые не наблюдал, так, как в среднем оценил увиденные предметы. То есть:

$$\hat{r}_{user}(u,i) = \begin{cases} r_{ui}, & (u,i) \in \mathcal{R} \\ \frac{1}{|\mathcal{N}_i(u)|} \sum_{j \in \mathcal{N}_u(u)} r_{uj} & (u,i) \notin \mathcal{R} \end{cases}$$

Модель I и модель U не требуют настройки внешних параметров.

Модель baseline. В [6] описана модель, объединяющая модель I и модель U. Прогноз оценки предпочтения строится без учёта взаимодействия между пользователем и предметом. Для каждого оценивается, насколько он влияет на итоговую оценку предпочтения. То же оценивается и для каждого предмета.

Данная модель имеет следующие параметры:

- $\mu \in \mathbb{R}$ – средняя оценка предпочтения

- $\{b_u \in \mathbb{R}\}_{u \in \mathcal{U}}$ – множество смещений оценок предпочтения для каждого пользователя
- $\{b_i \in \mathbb{R}\}_{i \in \mathcal{I}}$ – множество смещений оценок предпочтения для каждого предмета

Прогноз оценки предпочтения в данной модели для заданного пользователя $u \in \mathcal{U}$ и предмета $i \in \mathcal{I}$:

$$r_{baseline}(u, i) = \mu + b_u + b_i$$

В [7] проведены исследования для данной модели при том же разбиении выборки на валидационную (ua.test) и невалидационную (ua.base) часть, что и в других экспериментах, описанных в данной секции. Итоговое качество, корень среднего квадрата ошибки на валидационной части, составило 0.9665.

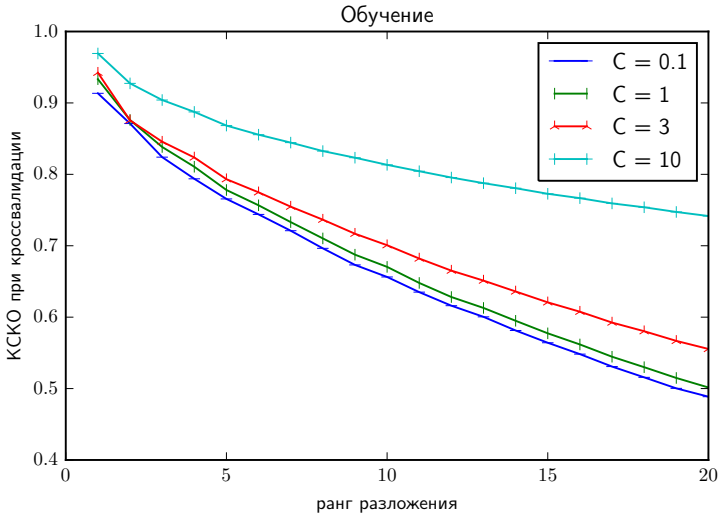
Модель матричного разложения. Настроим модель матричного разложения. Из рис. 4 видно, что можно взять параметры: ранг равный 3, параметр регуляризации $C = 3$ как оптимальный. Модели более низкого ранга, как правило, обладают лучшей обобщающей способностью.

Модель бустинга над матричными разложениями. Настоим параметры бустинга над матричными разложениями. Для каждой модели матричного разложения выполнялось 40 шагов оптимизации. Из рис. 5 видно, что в качестве оптимальных параметров можем выбрать 10 моделей в композиции при параметре несглаживания $\eta = 20$.

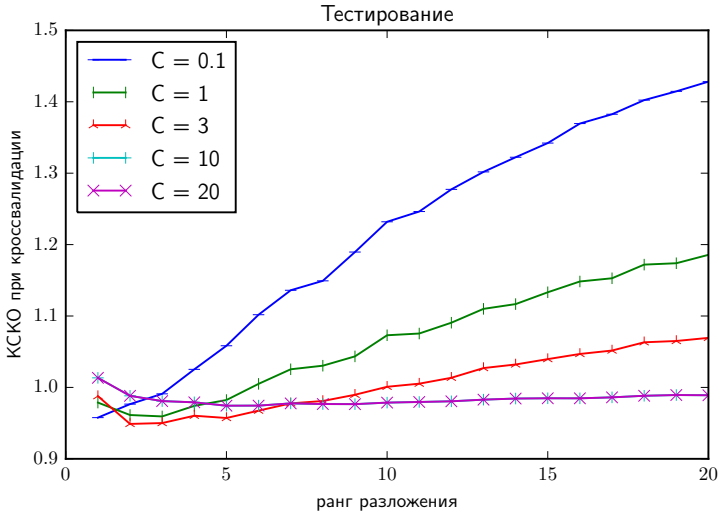
Сравнение. Сравним рассмотренные выше модели при зафиксированных ранее внешних параметрах, которые были выбраны после 5-блочной процедуры скользящего контроля на выборке ua.base. Обучим модели на невалидационной части выборки (90% от исходного объёма выборки, ua.base), и оценим качество на валидационной (10% от исходного объёма выборки, ua.test). На таб. 1 показаны результаты данного эксперимента.

Как видим, модель матричного разложения лучше эвристических моделей U, I, а так же лучше модели baseline, не моделирующей взаимодействие между пользователем и предметом.. Модель бустинга существенно улучшает модель матричного разложения.

Рис. 4: Значение ошибки при различных параметрах регуляризации регуляризации и ранге разложения. Кроссвалидация на 90% данных

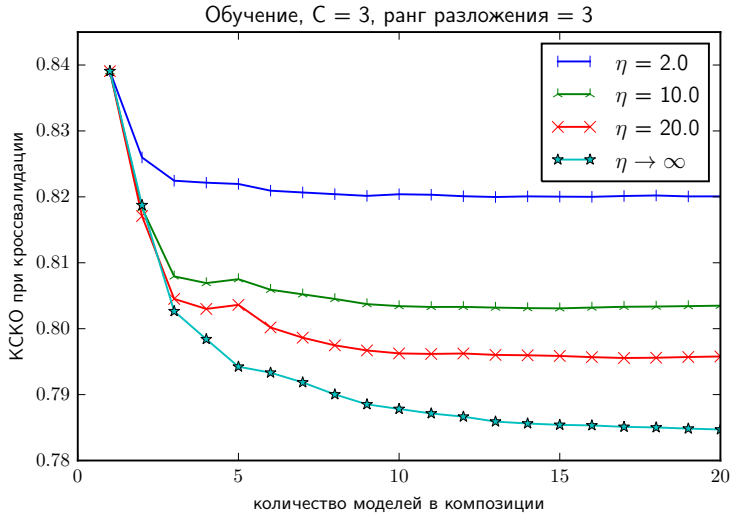


(a) Ошибка на обучающей выборке

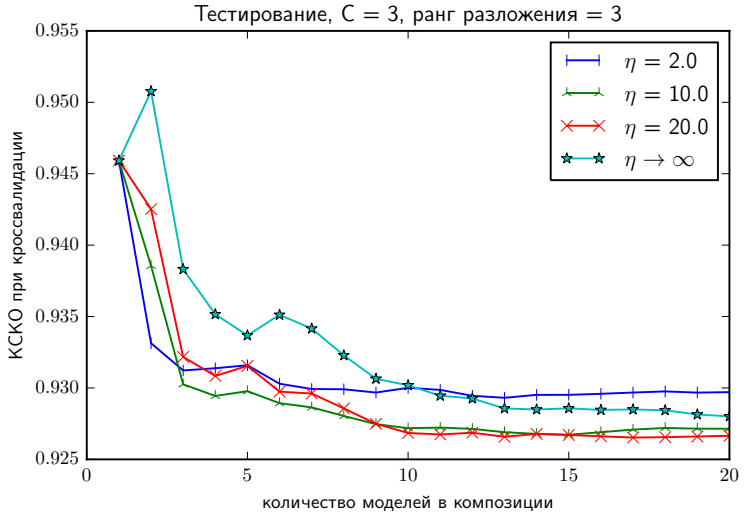


(b) Ошибка на тестовой выборке

Рис. 5: Значение ошибки при различных значениях параметра несглаживания и количества моделей в композиции. Кроссвалидация на 90% данных



(a) Ошибка на обучающей выборке



(b) Ошибка на тестовой выборке

Модель	RMSE on CV
Модель I	1.0417
Модель U	1.0431
Модель baseline	0.9665
Матричное разложение, $k = 3, C = 3$	**0.9547**
Бустинг на матричных разложениях $k = 3, C = 3, m = 10, \eta = 20.0$	**0.9417**

Таблица 1: Сравнение качества моделей прогнозирования оценки предпочтения на валидационной выборке (ua.test)

8 Заключение

- Была рассмотрена задача восстановления матрицы по её известным элементам. Рассмотрены применяющиеся на практике подходы.
- Была исследована модель низкорангового матричного разложения для задачи коллаборативной фильтрации. Было проведено иссследование по применимости данной модели в задаче восстановления матрицы большой размерности.
- Экспериментально на реальных данных показана необходимость введения регуляризации для модели низкорангового разложения.
- Исследована применимость схемы бустинга над матричными разложениями в задаче восстановления матрицы большой размерности.
- Исследована зависимость обобщающей способности модели бустинга в зависимости от параметра несглаживания и количества моделей в композиции.
- Проведено сравнение обобщающей способности моделей как между собой, так и с другими моделями.

Список литературы

[1] James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on,* 17(6):734–749, 2005.

[3] Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In ICML, volume 3, pages 720–727, 2003

[4] Emmanuel J Cand‘es and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics,* 9(6):717–772, 2009.

[5] Xiaotian Jiang, Zhendong Niu, Jiamin Guo, Ghulam Mustafa, Zihan Lin, Baomi Chen, Qian Zhou. Novel Boosting Frameworks to Improve the Performance of Collaborative Filtering. *Journal of Machine Learning Research : Workshop and Conference Proceedings* volume 29, pages 87-99, 2013

[6] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B Kantor. Recommender systems handbook. 2011

[7] Zhouxiao Bao, Haiying Xia. Movie Rating Estimation and Recommendation. 2015.