Hybrid Models in the Task of Building Recommender Systems

Abstract

The aim of this thesis is to investigate the ways of improving the performance of a Recommender System for movie recommendation by supplying it with information on users and items on top of the user-item ratings. Both classic collaborative and content-based approaches to movie recommendation can suffer from the cold-start problem, in case not enough ratings for the corresponding user or item are provided, since they both rely on the nearest neighbor models. For the sake of utilizing different kinds of extra information, thereby drastically improving performance, a hybrid meta-model is proposed.

The objective to incorporate such data as user's age, movie genre, its release year, etc. is fulfilled by adding features to a supervised meta-model, receiving a Collaborative Filtering model based on SVD of the user-rating matrix as the input. Novel features were engineered to incorporate all the available data. To further enrich the used MovieLens datasets movie plots were acquired from the public domain. The plots are utilized with the help of features based on the results of Topic Modelling. Random Forest regression model serves as the meta-model. According to the performed experiments, the performance of the resulting model surpassed that of the classic approaches by a wide margin in terms of RMSE and MAE.

Министерство образования и науки Российской Федерации
Московский физико-технический институт
(государственный университет)
Факультет инноваций и высоких технологий
Кафедра «Алгоритмы и технологии программирования»

Соколова Евгения Александровна

Гибридные модели в задаче построения рекомендательных систем

Выпускная квалификационная работа магистра

Научный руководитель:
асс. каф. АТП ФИВТ Эмели Драль

Москва
2016

Содержание

Аннотация

Решается задача предсказания оценок пользователями фильмов на основании набора имеющихся оценок, а также дополнительной информации о пользователях и фильмах. Для решения данной задачи применяется мета-модель с использованием метода машинного обучения с учителем. Также предложены методы построения признаков на основе различных типов информации о пользователях и фильмах. Тестовые датасеты были дополнены новыми видами информации о фильмах. Проанализировано качество работы разработанного метода. Эксперименты подтверждают как преимущество мета-модели над стандартными методами, так и улучшение качества ее работы при добавлении предложенных признаков.

1 Введение

Интенсивное увеличение количества информации в сети интернет, в частности, доступных через него товаров и услуг, повлекло за собой бум развития рекомендательных систем, позволяющих эффективно использовать большие объемы данных при минимальном привлечении экспертов в области какой-либо группы продуктов. Данные системы могут, с одной стороны, использоваться клиентами для максимизации положительных результатов потребления предлагаемых продуктов, с другой - для максимизации прибыли фирмами посредством таргетирования продаж. Исследования подтверждают эффективность рекомендательных систем для увеличения продаж товаров и количества их просмотров на веб-сайтах [1].

Стандартной задачей является предсказание значений оценок для пар пользователь-предмет. Выделяют два основных стандартных подхода к рекомендации: коллаборативная фильтрация и рекомендация на основе контента. При коллаборативной фильтрации оценка строится на основании откликов других пользователей. Она опирается на неявное предположение, что предсказываемый рейтинг будет близок к рейтингу того же предмета другим пользователем, если он с текущим пользователем оценивает другие предметы схожим образом [2]. Подход основанный на контенте опирается на имеющиеся оценки от текущего пользователя для составления профиля его интересов, предметы при этом должны быть снабжены текстовыми и/или признаковыми описаниями [3].

Оба подхода могут страдать от так называемой проблемы холодного старта – недостатка информации о целевом пользователе или предмете. Тем не менее, подход на основе контента способен дать оценку релевантности нового предмета для пользователя, при условии, что предмет снабжен описанием. В свою очередь, коллаборативные методы не требуют наличия текстового описания и способны производить «нестереотипные» рекомендации, основывающиеся на закономерностях пользовательских вкусов, а не схожих признаковых описаниях [4].

Для борьбы с вышеперечисленными проблемами начали разрабатываться гибридные рекомендательные системы, сочетающие применение различных подходов. В одних из первых гибридных систем было предложено использование профилей пользователей для определения схожести их интересов [5], а также использование коллаборативной информации в качестве признаков [6].

1.1 Общая информация

В работе будет рассматриваться только использование явных оценок пользователями предметов, такие оценки так же называются рейтингами.

Рейтинги предметов пользователями представляются в виде матрицы R, где элемент r_{ui} – значение оценки предмета i пользователем u. Рейтинги представляют собой целые числа от 1 до 5. Отсутствию рейтинга соответствует 0.

Предсказываемый рейтинг для целевых (u, i) обозначается $\hat{r}_{ui}$.

Природа данных – оценка пользователями фильмов.

1.2 Требования

Реализовать гибридный метод, способный предсказывать рейтинг для пары (u, i), $\hat{r}_{ui} \in [1, 5]$ с учетом следующей дополнительной информации:

- дата выпуска фильма
- жанр фильма
- возраст пользователя
- пол пользователя
- профессия пользователя

Рассмотреть возможность использования других видов дополнительной информации.

1.3 Возможности применения

Предсказание рейтинга.

С помощью разрабатываемого метода возможно предсказывать рейтинги пользователей для ранее не оцененных фильмов.

Top-N рекомендация.

Данные предсказания могут быть использованы для ранжирования и выдачи пользователю краткого списка рекомендуемых ему фильмов, которые наиболее вероятно будут им высоко оценены.

2 Постановка задачи

Целью данной работы является исследование и разработка гибридных методов предсказания оценок пользователями фильмов, при наличии дополнительной информации о фильмах и пользователях. Для достижения поставленной цели необходимо:

1. Исследовать существующие гибридные методы предсказания оценок для пар пользователь-предмет
2. Разработать и реализовать гибридный метод предсказания оценки для пары пользователь-фильм с учетом имеющейся дополнительной информации о фильмах и пользователях
3. Обогатить имеющиеся данные большим количеством дополнительной информации и разработать методы её использования
4. Выполнить экспериментальную оценку качества разработанного метода
5. Оценить вклад использования различных типов информации о фильме в качество работы гибридного алгоритма

3 Обзор существующих методов

3.1 Базовые методы

Рассмотрим основные виды получения базовой оценки, описанные в [7]. Данные методы не учитывают взаимосвязь между целевыми пользователем и предметом. Базовая оценка обозначается b_{ui}. Простейшим подходом является взятие среднего всех известных оценок μ. Также можно брать средний по целевому пользователю или предмету рейтинг, $\overline{r_u}$ или $\overline{r_i}$. Отметим, что данная оценка применима для нового предмета или пользователя соответственно. В общем виде базовую оценку обычно представляют как

$$b_{ui} = \mu + b_u + b_i$$

где b_u и b_i – базовые оценки основанные на рейтингах данного пользователя или предмета. Их можно определить как среднее отклонение от μ и от $\mu + b_u$ соответственно

$$b_u = \frac{1}{|I_u|} \sum_{i \in I_u} (r_{ui} - \mu)$$

$$b_i = \frac{1}{|U_i|} \sum_{u \in U_i} (r_{ui - b_u - \mu})$$

Также производят регуляризацию b_u и b_i, вводя сглаживающие члены β_u и β_i, таким образом базовая оценка b_{ui} приближается к глобальному среднему μ при малом количестве рейтингов у пользователя или предмета (чем больше известно рейтингов, тем точнее оценка среднего пользовательского или предметного рейтинга).

3.2 Методы, основанные на соседстве

Основой данных методов является определение близости между векторами, соответствующими пользователям или предметами. Исходя из предположения, что пользователи со схожими вкусами будут похожим образом оценивать предметы, или что схожие предметы будут похожим образов оценены одним и тем же пользователем, итоговая оценка для целевой пары (u, i) определяется как взвешенная оценка ближайших соседей пользователя или предмета соответственно.

3.2.1 Меры близости

Наиболее известными мерами близости являются косинусная мера

$$cosine(x, y) = \frac{\mathbf{r}_x^\top \mathbf{r}_y}{||\mathbf{r}_x||_2 ||\mathbf{r}_y||_2},$$

корреляция Пирсона для пользователей и предметов соответственно

$$pearson(u, v) = \frac{\sum\limits_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \cdot \sum\limits_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

$$pearson(i, j) = \frac{\sum\limits_{u \in U_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum\limits_{u \in U_{ij}} (r_{ui} - \bar{r}_i)^2 \cdot \sum\limits_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}}$$

и ранговая корреляция Спирмена для пользователей и предметов соответственно

$$spearman(u, v) = \frac{\sum\limits_{i \in I_{uv}} (k_{ui} - \bar{k}_u)(k_{vi} - \bar{k}_v)}{\sqrt{\sum\limits_{i \in I_{uv}} (k_{ui} - \bar{k}_u)^2 \cdot \sum\limits_{i \in I_{uv}} (k_{vi} - \bar{k}_v)^2}}$$

$$spearman(i, j) = \frac{\sum\limits_{u \in U_{ij}} (k_{ui} - \bar{k}_i)(k_{uj} - \bar{k}_j)}{\sqrt{\sum\limits_{u \in U_{ij}} (k_{ui} - \bar{k}_i)^2 \cdot \sum\limits_{u \in U_{ij}} (k_{uj} - \bar{k}_j)^2}}, \text{ здесь } k \text{ – ранг}$$

Данные меры применяются к векторам, состоящим из рейтингов для пользователей или предметов, соответственно. При использовании косинусной меры отсутствию рейтинга ставится в соответствие нулевое значение. Вычисление корреляции производится по индексам, на которых присутствуют значения обоих рейтингов.

Также было предложено множество вариантов перечисленных мер, в том числе с использованием центрирования и нормирования рейтингов [3].

3.2.2 User-User

Для предсказания оценки для целевой пары (u, i) необходимо произвести следующее [8]:

1. Определить близость всех пользователей относительно u
2. Отобрать подмножество пользователей, относительно которых будет производиться предсказание

3. Произвести предсказание на основе взвешивания их оценок

Процедура также может включать в себя нормировку рейтингов.

Одна из формул взвешивания

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in N_i(u)} s_{uv} \cdot r_{vi}}{\sum\limits_{v \in N_i(u)} s_{uv}}, \text{ где}$$

$N_i(u)$ – соседи u, оценившие i; s – значение функции схожести пользователей [3].

Первая система, выполнявшая автоматизированную коллаборативную фильтрацию методом соседства, GroupLens, использовала для определения близости пользователей корреляцию Пирсона, взвешивание производилось по формуле

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum\limits_{v \in N_i(u)} s_{uv} \cdot (r_{vi} - \bar{r}_v)}{\sum\limits_{v \in N_i(u)} s_{uv}} \text{ [9]}$$

В сравнительно недавнем методе, описанном в [10] достигнуто существенное улучшение качества работы User-User метода посредством использования генетического алгоритма для оптимального подбора весов входящих в выражение для близости пользователей.

3.2.3 Item-Item

Для получения предсказания необходимо [11]:

1. Определить близость для всех пар предметов, чьи множества оценивших пользователей пересекаются

2. Скомбинировать рейтинги предметов-соседей всех оцененных пользователем предметов с целью формирования предсказаний для данного пользователя

Возможное взвешивание

$$\hat{r}_{ui} = \frac{\sum\limits_{j \in N_u(i)} s_{ij} \cdot r_{uj}}{\sum\limits_{j \in N_u(i)} s_{ij}}, \text{ где}$$

$N_u(i)$ – соседи i, оцененные u; s – значение функции схожести предметов [3].

3.3 SVD

Известная техника матричной факторизации SVD может быть применена в коллаборативной фильтрации. Она позволяет получить разложение матрицы рейтингов $\mathbf{R}$ размера $m \times n$ и ранга r в следующем виде: $\mathbf{R} = \mathbf{USV}$, где матрицы $\mathbf{U}$ и $\mathbf{V}$ ортогональные размера $m \times r$ и $r \times n$ соответственно, а матрица $\mathbf{S}$ диагональная $r \times r$, с положительными собственными числами матрицы $\mathbf{R}$, расположенными на диагонали в порядке убывания. Свойством данного разложения является то, что оно ведет к наилучшему по норме Фробениуса низкоранговому приближению матрицы $\mathbf{R}$, для получения такого приближения достаточно оставить k наибольших значений матрицы $\mathbf{S}$. Матрицы $\mathbf{U}$ и $\mathbf{V}$ домножаются на $\mathbf{S}^{1/2}$ справа и слева соответственно. Предсказание производится посредством вычисления скалярного произведения соответствующих строки и столбца получившихся матриц [12]. Эти матрицы можно рассматривать как описания пользователей и предметов в виде k скрытых признаков.

3.4 Гибридные методы

Подробный обзор гибридных методов произведен в работе [13]. Различные методы, используемые при построении рекомендательных систем обладают своими достоинствами и недостатками. Они в разной степени испытывают проблемы предсказания оценки для нового пользователя или предмета, нуждаются для успешной работы в различном количестве и типах информации. Естественной идеей является комбинирование методов для получения более точных предсказаний рейтингов.

Простейшим вариантом является взвешивание с различной настройкой весов. Недостатком такой комбинации является невозможность учитывать различную способность включенных методов производить оценку для отдельных пар пользователь-предмет. Так, переключение между коллаборативным и основанным на контенте методами позволит получить лучшую оценку в зависимости от имеющегося количества информации одного из типов.

Способы построения гибридных методов также включают в себя:

- добавление признаков на основе контента с последующей коллаборативной фильтрацией

- использование дополнительного метода для выявления различий между близкими полученными оценками
- использование полученных оценок как признаков для следующего метода
- использование построенной модели как входа для следующего метода

3.4.1 Обзор гибридных методов

Для задачи типа потребительской корзины в работе [14] был предложен гибридный метод на основе классификации. В данной задаче $r_{ui} \in \{0, 1\}$, требуется для пары (u, i) предсказать факт приобретения товара. Информация о всех известных приобретениях товаров используется для построения бинарного логистического классификатора. Для более эффективного построения модели количество признаков уменьшается с использованием PCA.

В статье [15] предлагается итеративное построение User-User модели до сходимости. При этом изначально известные рейтинги не меняются. Получающаяся в итоге модель может быть использована для непосредственного получения искомых оценок либо передана для использования в мета-модели, в частности, Item-Item. В работе [16] для заполнения отсутствующих рейтингов использована SlopeOne модель, в [17] в качестве такой модели использована SVD, в обеих работах основная модель User-User.

В работе [18] предложено учитывать информацию о жанрах, режиссерах и актерах при вычислении близости фильмов. Делается это следующим образом: пусть у пары фильмов (i, j) есть соответствующие множества жанров режиссеров и актеров (G_i, D_i, A_i), (G_j, D_j, A_j). Векторы, соответствующие i и j состоят из $|G_i \cup G_j| + |D_i \cup D_j| + |A_i \cap A_j|$ индикаторных признаков и вычисляется сглаженная косинусная близость:

$$\omega_{ij} = \frac{1 + \langle f_i, f_j \rangle}{||f_i||_2 \cdot ||f_j||_2} \text{ при } \langle f_i, f_j \rangle > k$$

$$\omega_{ij} = \frac{1}{||f_{max}||_2 \cdot ||f_j||_2} \text{ иначе, где } f_{max} \text{ соответствует фильму с } |f|_{max}$$

Полученные таким образом веса используются при вычислении взвешенной корелляции Пирсона в User-User методе.

3.5 Графические модели

Графические вероятностные модели [19] – удобные схематические представления вероятностных распределений в виде графа, где узлы отвечают случайным переменным, а ребра – вероятностным отношениям между ними. Получается, что граф отражает, каким образом совместное распределение надо всеми случайными переменными может быть представлено как произведение членов, зависящих лишь от подмножеств переменных. Графические вероятностные модели обладают следующими полезными свойствами:

- являются простым способом визуализации структуры вероятностных моделей и могут быть использованы для проектирования и обоснования новых моделей
- упрощают понимание свойств моделей, таких как свойства условной независимости, через рассмотрение графа

Графические модели могут быть использованы для получения оценок рейтингов. Они также предоставляют широкие возможности для включения дополнительной информации различного рода в процесс получения оценки.

Так, в [20] предложена графическая модель, описывающая принадлежность пользователей и фильмов к соответствующим множествам классов, от которой зависит оценка.

Модель в [21] включает в себя пользователей, предметы, а также описания предметов посредством бинарных признаков. Байесовская сеть используется для представления как коллаборативной, так и основанной на контенте компонент. Вклад каждой из компонент оценивается автоматически, предложенная модель демонстрирует превосходство над коллаборативной моделью, переключающейся на основанную на контенте в случае невозможности произведения оценки.

В работе [22] совместно моделируются коллаборативная информация, аспекты и текстовые отзывы пользователей. Предполагается, что пользователи и фильмы имеют распределение по аспектам. Отзывы пользователей о фильмах формируются в соответствии с этими распределениями. То есть, упоминание какого-либо аспекта фильма с положительной или отрицательной тональностью, вероятно, будет присутствовать в отзыве, если пользователь заинтересован в этом аспекте и он присутствует в фильме. Языковая модель отзывов выделяет следующие группы слов:

- фоновые, имеют одинаковые распределения во всех отзывах
- специфичные фильму, такие как имена героев
- аспектные, например "сюжет"или "саундтрек"
- аспектно-тональные, выражают оценку аспекта
- обще-тональные, выражают тональность мнения о фильме в целом

Предполагается, что итоговая оценка зависит от комбинации частичных оценок, связанных с различными аспектами фильма. Сообщается, что данный подход превосходит подход, связывающий скрытые признаки с тематической моделью, описывающей отзывы в условиях наличия малого количества отзывов от пользователя.

3.6 Тематическое моделирование

Для использования текстовых данных, соответствующих фильмам, в данной работе будут использоваться методы тематического моделирования (см. подглаву 4.4.4).

К методам тематического моделирования относятся методы анализа корпусов текстов, позволяющие определять принадлежность каждого из текстов к некоторому набору тем, извлекаемому из корпуса. Тема в данном случае характеризуется определенным распределением слов.

В работе [23] предлагается следующая формальная постановка задачи тематического моделирования.

Дано множество документов D, каждый из которых является последовательностью элементов множества W, называемого словарем (элементы этого множества называются словами): $\forall d \in D : d = \{w_i\}_{i=1}^{n_d}, \forall w_i \in W$, где n_d - число слов в документе.

Также заранее задается число искомых тем $|T|$, множество искомых тем обозначают как T.

3.6.1 PLSA

В работе Хоффмана [24] предлагается простейшая тематическая модель – PLSA.

Параметрами модели являются матрицы $\Theta \in \mathbb{R}^{|D|\times|T|}$ и $\Phi \in \mathbb{R}^{|W|\times|T|}$. Матрица Θ задает принадлежность тем документам, а матрица Φ задает принадлежность слов темам.

На матрицы Φ и Θ накладываются условия нормировки и неотрицательности

$$\forall d \in D : \sum_{t \in T} \theta_{dt} = 1 \text{ и } \forall d \in D, \forall t \in T : \theta_{dt} \geqslant 0 \tag{1}$$

$$\forall t \in T : \sum_{w \in W} \varphi_{wt} = 1 \text{ и } \forall w \in W, \forall t \in T : \varphi_{wt} \geqslant 0 \tag{2}$$

Формально, PLSA предполагает независимость генерации документов, а также что для каждого документа d для каждой позиции $1 \leqslant i \leqslant n_d$ были сгенерированы независимые одинаково распределенные случайные величины $z_{di} \sim Mult(\theta_{d\cdot})$, а затем сгенерирована последовательность слов $w_{di} \sim Mult(\varphi_{z_i\cdot})$.

Таким образом, сначала в соответствии с вектором $\theta_{d\cdot}$ для каждого слова определяется, из какой темы оно будет сгенерировано, а затем в соответствии с матрицей Φ генерируются сами слова.

Для нахождения значений параметров $\hat{\Phi}$ и $\hat{\Theta}$ предлагается использовать метод максимума правдоподобия:

$$\begin{gathered}(\hat{\Theta}, \hat{\Phi}) = \arg\max_{\Theta,\Phi} L(\Theta, \Phi) \\ \text{при условиях (1) и (2)}\end{gathered} \tag{3}$$

Данную оптимизационную задачу предлагается решать с помощью EM-алгоритма [24] [25] (См. алгоритм 1)

Логарифм правдоподобия модели PLSA выписывается следующим образом

$$L(\Theta, \Phi) = \ln \mathbb{P}(D|\Theta, \Phi) = \sum_{d \in D} \sum_{w \in W} n_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} \tag{4}$$

3.6.2 Регуляризованный PLSA

Также была предложена регуляризованная версия алгоритма PLSA [26]. Данный подход позволяет получать разреженные параметры, а также увеличить различность извлекаемых тем.

Предлагается от оптимизационной задачи (3) перейти к ее регуляризованной версии

Алгоритм 1 EM-алгоритм для PLSA.

Вход: корпус D,

число тем $|T|$,

начальные приближения Θ и Φ

Выход: параметры Θ и Φ;

повторять
  для всех $d \in D,\ w \in W,\quad t \in T$
    $n_{wt} = 0,\ n_{dt} = 0$
  для всех $d \in D,\ w \in d$
    $Z = \sum_{t \in T} \varphi_{wt}\theta_{td}$
    для всех $t \in T$
      $\delta = n_{dw}\frac{\varphi_{wt}\theta_{td}}{Z}$
      $n_{wt}\ +\!= \delta,\ n_{dt}\ +\!= \delta$
  $\varphi_{wt} := n_{wt}\ \ \forall w \in W\ \ \forall t \in T$;
  $\theta_{td} := n_{dt}\ \ \forall d \in D\ \ \forall t \in T$;
  отнормировать Φ и Θ в соответствии с условиями (2) и (1)
пока $L(\Theta, \Phi)$ не стабилизируется.

$$(\hat{\Theta}, \hat{\Phi}) = \arg\max_{\Theta,\Phi} L(\Theta, \Phi) + R(\Theta, \Phi)$$
$$\text{при условиях (1) и (2)} \tag{5}$$

где $R(\Theta, \Phi)$ обозначает регуляризационный член.

3.6.3 LDA

В работе [23] была предложена модель LDA (скрытое распределение Дирихле). Она определяется как PLSA с априорным распределением Дирихле на строки матриц Φ и Θ

$$\varphi_{w\cdot} \sim Dir(\varphi_{w\cdot}|\alpha)$$

$$\theta_{d\cdot} \sim Dir(\theta_{d\cdot}|\beta)$$

где

Алгоритм 2 EM-алгоритм для регуляризованного PLSA.

Вход: корпус D,

число тем $|T|$,

начальные приближения Θ и Φ;

Выход: параметры Θ и Φ;

повторять

  для всех $d \in D,\ w \in W,\quad t \in T$

    $n_{wt} = 0,\ n_{dt} = 0$

  для всех $d \in D,\ w \in d$

    $Z = \sum_{t \in T} \varphi_{wt}\theta_{td}$

    для всех $t \in T$

      $\delta = n_{dw}\frac{\varphi_{wt}\theta_{td}}{Z}$

      n_{wt} += $\delta,\ n_{dt}$ += δ

  $\varphi_{wt} := (n_{wt} + \varphi_{wt}\frac{\partial R(\Theta,\Phi)}{\partial \varphi_{wt}})_+ \ \ \forall w \in W \ \ \forall t \in T;$

  $\theta_{td} := (n_{dt} + \theta_{td}\frac{\partial R(\Theta,\Phi)}{\partial \theta_{td}})_+ \ \ \forall d \in D \ \ \forall t \in T;$

  отнормировать Φ и Θ в соответствии с (2) и (1)

пока $L(\Theta, \Phi) + R(\Theta, \Phi)$ не стабилизируется.

$$Dir(x|\alpha) = \frac{1}{Z}\prod_{i=1}^{n} x_i^{\alpha_i - 1}$$

Заметим, что регуляризованный PLSA с некоторым $R(\Theta, \Phi)$ эквивалентен модели PLSA с выбранным априорным распределением. Действительно

$$e^{L(\Theta,\Phi)+R(\Theta,\Phi)} = \mathbb{P}(D|\Theta, \Phi)\underbrace{e^{R(\Theta,\Phi)}}_{Z\mathbb{P}(\Theta,\Phi)}$$

То есть соответствие регуляризации априорному распределению устанавливается как

$$\mathbb{P}(\Theta, \Phi) = \frac{1}{Z}e^{R(\Theta,\Phi)}, \text{ где } Z \text{ - нормирующий множитель}$$

И PLSA с регуляризационным членом

$$R(\Theta, \Phi) = \sum_{w \in W} \ln Dir(\varphi_{w\cdot}|\alpha) + \sum_{t \in D} \ln Dir(\theta_{d\cdot}|\beta) \tag{6}$$

3.6.4 Робастный PLSA

Также была предложена робастная модификация алгоритма PLSA [25], допускающая генерацию слов не только "из тем но также из специфичному каждому документу шума и общему для всей коллекции фона.

Данная модель добавляет к параметрам следующие распределение

$$\pi_w = p(w|\text{фон})$$

$$\pi_{wd} = p(w|\text{шум}, d)$$

Как и на матрицы Φ и Θ, накладываются ограничения неотрицательности и нормировки

$$\forall d \in D : \sum_w \pi_{wd} = 1 \text{ и } \forall d \in D, \forall w \in W : \pi_{wd} \geqslant 0 \tag{7}$$

$$\sum_w \pi_w = 1 \text{ и } \forall w \in W : \pi_w \geqslant 0 \tag{8}$$

Робастный PLSA также предполагает, что наперед заданы значения ε и γ, определяющие вероятность слова быть сгенерированным из фона как $\frac{\varepsilon}{1+\varepsilon+\gamma}$ и из шума как $\frac{\gamma}{1+\varepsilon+\gamma}$.

Слова из шума генерируются как $w_{di} \sim Mult(\pi_{d,\cdot})$, из фона как $w_{di} \sim Mult(\pi_{\cdot})$, а из так же, как и в модели PLSA.

Метод максимума правдоподобия приводит к следующей оптимизационной задаче

$$\begin{gathered}(\hat{\Theta}, \hat{\Phi}) = \arg\max_{\Theta,\Phi,\Pi} L(\Theta, \Phi\Pi) \\ \text{при условиях (1), (2), (7) и (8)}\end{gathered} \tag{9}$$

Логарифм правдоподобия робастного PLSA:

$$L(\Phi, \Theta, \Pi) = \ln \mathbb{P}(D|\Theta, \Phi, \Pi) = \sum_{d\in D} \sum_{w\in d} n_{dw} \ln \sum_{t=1}^{T} \frac{\varphi_{wt}\theta_{td} + \epsilon\pi_w + \gamma\pi_{dw}}{1 + \epsilon + \gamma} \tag{10}$$

Работой [25] отдельно подчеркивается невозможность оптимизации по всем параметрам Φ, Θ, Π, ε и γ, поскольку это приведет к бесконечному росту параметра γ, что соответствует генерации всех слов из шума и вероятности обучающего набора документов, близкой к единице, что устраняет обобщающую способность получаемой модели. В статье рекомендуются значения $\varepsilon = 0.01$ и $\gamma = 0.3$.

Работа [25] предлагает версию EM-алгоритма, пригодную для решения такой оптимизационной задачи – см. алгорим 3.

Алгоритм 3 EM-алгоритм для Робастной модели PLSA.

Вход: корпус D, число тем $|T|$, начальные приближения Θ, Φ;

Выход: параметры Θ, Φ и Π;

$\pi_{dw} = n_{dw}\ \forall d \in D\ \forall w \in W$

$\pi_w = \sum_{d \in D} n_{dw}\ \forall w \in W$

Нормализовать Π в соответствии с (7) и (8)

повторять

  $n_{td} = 0;\ n_{wt} = 0;\ n_w = 0,\ \nu_d = 0$

  для всех $d \in D$

    $Z_w = \sum_t \varphi_{wt}\theta_{td} + \gamma\pi_{dw} + \varepsilon\pi_w\ \forall w \in d$

    для всех $w \in d$

      $\nu_d\ += n_{dw}\gamma\pi_{dw}/Z_w$

      для всех $t \in T$

        $\delta = n_{dw}\theta_{td}\varphi_{wt}/Z_w$

        $n_{td}\ += \delta, \quad n_{wt}\ += \delta, \quad n_w\ += \varepsilon\delta$

      $\pi_{dw} = \left(\pi_{dw} + \frac{n_{dw}}{\nu_d} - \frac{Z_w}{\gamma}\right)_+$

  $\varphi_{wt} = n_{wt}\ \ \forall w \in W\ \ \forall t \in T;$

  $\theta_{td} = n_{td}\ \ \forall d \in D\ \ \forall t \in T;$

  $\pi_w = n_w$

  отнормировать Φ, Θ и π_w в соответствии с (2), (1) и (8)

пока $L(\Theta, \Phi, \Pi)$ не стабилизируется.

4 Исследование и построение решения задачи

4.1 Использование классификатора для предсказания оценки

Первым способом улучшения качества работы является использование для предсказания классификатора над моделью SVD. Данный гибридный метод в [13] классифицируется как модель мета-уровня.

Обоснование использования такого метода следующее: в стандартной SVD модели с приближенной матрицей $\hat{\mathbf{R}} = \mathbf{PQ}$ предсказание производится как

$$\hat{r}_{ui} = \mathbf{p_{u\cdot}} \mathbf{q_{\cdot i}} \tag{11}$$

Очевидно, что предсказание вида (11) является частным случаем класса предсказаний вида

$$\hat{r}_{ui} = f(\mathbf{p_{u\cdot}}, \mathbf{q_{\cdot i}^T}, \mathbf{p_{u\cdot}} * \mathbf{q_{\cdot i}^T}) \text{ для некоторой функции } f : \mathbb{R}^k \times \mathbb{R}^k \to [1, 5] \tag{12}$$

где $\cdot * \cdot$ – поэлементное произведение.

Для восстановления функции f используются методы машинного обучения с учителем. При использовании нелинейных (и даже линейных) классификаторов возможно получить приближение (11) с достаточно высокой точностью, что дает основание полагать, что результат предсказания будет, по крайней мере, не хуже.

Плюсом выбранного подхода является простота добавления дополнительных признаков на основе контента.

4.2 Использование статистик рейтингов

Основываясь на использовании средних в базовых предсказаниях (см. 3.1), предположим, что в статистиках рейтингов целевых u и i может быть заключена информация, которая позволит нам улучшить качество предсказания оценки. Под рейтингами u или i подразумеваются те известные рейтинги, что относятся к данному пользователю

$$\mathcal{R}_u^{\mathcal{U}} = \{R_{uj} | j \in I \wedge R_{uj} \neq 0\}$$

или предмету

$$\mathcal{R}_i^{\mathcal{I}} = \{R_{vi} | v \in U \wedge R_{vi} \neq 0\}$$

Добавим в качестве признаков следующие статистики $\mathcal{R}_u^{\mathcal{U}}$ и $\mathcal{R}_i^{\mathcal{I}}$

- среднее
- медиана
- мода

Также может быть полезным регуляризовать базовые оценки, исходя из того, что чем больше имеется рейтингов, тем точнее оценка среднего [7]. Руководствуясь данным наблюдением добавим в качества признаков количества рейтингов предмета и пользователя:

- $|\mathcal{R}_u^{\mathcal{U}}|$
- $|\mathcal{R}_i^{\mathcal{I}}|$

4.3 Использование информации, встроенной в датасет

Далее описываются разработанные способы использования изначально имеющейся в тестовых датасетах информации о предметах и пользователях (см. 6.1). Информацию о почтовом индексе пользователя было решено не использовать, так как в обоих датасетах количество уникальных индексов составляет более половины количества пользователей, что ведет к необходимости группировки индексов по географической близости, определение которой затруднено недостаточным описанием природы почтовых индексов в справочной информации.

4.3.1 Жанр

Пусть есть множество жанров $\mathbb{G}$. Каждому фильму i поставлено в соответствие множество его жанров $G_i \subset \mathbb{G}$. Рассмотрим набор троек T пользователь-жанр-рейтинг (u, g, r), включающий в себя элемент тогда и только тогда когда существует фильм i, с жанром $g \in G_i$, которому пользователем u был поставлен рейтинг r. Обозначим множество троек, относящихся к пользователю u и жанру g как $T_{ug} = \{(u, g, r) \in T\}$ Для каждой пары (u, g) определим величину R_{ug}^G как средний рейтинг по множеству T_{ug}. Для случая $T_{ug} = \emptyset$ доопределим $R_{ug}^G = 0$.

Теперь, применяя SVD, получим декомпозицию матрицы $\mathbf{R}^G$:

$$\hat{\mathbf{R}}^G = \mathbf{P}^G \mathbf{Q}^G$$

По аналогии предлагается добавить векторы $p^G_{u\cdot}$ и

$$s^G_{ui} = \frac{1}{|G_i|} \sum_{g \in G_i} (q^G_{\cdot g})^T$$

к признакам для обучения классификатора. Также добавим и значение

$$a^G_{ui} = \frac{1}{|G_i|} \sum_{g \in G_i} m_{ug}$$

где m_{ug} средняя оценка пользователя u фильму жанра g. Таким образом, решение будем искать в виде

$$\hat{r}_{ui} = f\left(p_{u\cdot}, q^\top_{\cdot i}, p_{u\cdot} * q^\top_{\cdot i}, p^G_{u\cdot}, s^G_{ui}, p^G_{u\cdot} * s^G_{ui}, a^G_{ui}\right) \tag{13}$$

4.3.2 Год выпуска фильма

Было замечено, что в одном из датасетов в дате выхода в прокат часто допускаются неточности (для большого количества фильмов – первое января), во втором указывается лишь год. Поэтому, было решено использовать год выпуска фильма. Делается это следующим образом: каждому фильму i ставится в соответствие одноэлементное множество Y_i, содержащее номер десятилетия выхода фильма. Соответственно, определим $\mathbb{Y} = \bigcup\limits_i Y_i$. Аналогично предыдущему разделу определим переменные $\mathbf{P}^Y$, $\mathbf{S}^Y$, $\mathbf{A}^Y$. Теперь решене ищется в виде

$$\hat{r}_{ui} = f\left(p_{u\cdot}, q^\top_{\cdot i}, p_{u\cdot} * q^\top_{\cdot i}, p^G_{u\cdot}, s^G_{ui}, p^G_{u\cdot} * s^G_{ui}, a^G_{ui}, p^Y_{u\cdot}, s^Y_{ui}, p^Y_{u\cdot} * s^Y_{ui}, a^Y_{ui}\right) \tag{14}$$

4.3.3 Пол пользователя

Определим вектор $\mathbf{h}_i \in \mathbb{B}^{|\mathbb{G}|}$, такой что $h_{ig} = 1 \Leftrightarrow g \in G_i$. Здесь и далее $\mathbb{B} = \{0, 1\}$.

Так же определим вектор $\mathbf{l}_u \in \mathbb{B}^2$, такой что $l_{u1} = 1$ тогда и только тогда когда пользователь мужчина, а $l_{u2} = 1$ тогда и только тогда когда пользователь женщина.

Предлагается добавить к признакам $\mathbf{l}_u \times \mathbf{h}_i$.

4.3.4 Возраст пользователя

В одном из датасетов для пользователей указывается лишь возрастная группа. Всего групп семь. Предлагается разбить пользователей по тем же группам и в другом датасете. В качестве признаков использовать индикаторы принадлежности к группам.

4.3.5 Профессия пользователя

Для пользователя u известна его профессия $o_u \in O$. Определим для всех профессий $o \in O$ и всех жанров $g \in \mathbb{G}$ вектор из трех элементов $\mu_{o,g}$ – среднего, медианы и матожидания множества оценок, поставленных пользователями профессии o всех фильмов жанра g: $\{R_{ui} | g \in G_i, o_u = o\}$. Далее для заданного фильма i и пользователя u предлагается использовать вектор

$$\frac{1}{|G_i|} \sum_{g \in G_i} \mu_{o_u,g}$$

в качестве признаков.

4.4 Использование информации, собранной самостоятельно

Далее описывается использование самостоятельно собранной информации о фильмах (см. 6.3). Было решено также добавить признаки на основе данной информации, в силу ее более легкой доступности по сравнению с личной информацией о пользователях. Получение последней сопряжено с проблемой конфиденциальности пользователей.

4.4.1 Дата выпуска

В качестве признака также предлагается использовать сезон выхода фильма, поскольку известно, что фильмы, ориентированные на развлечение чаще выходят летом, на новогоднюю тематику – зимой и т. д.

4.4.2 Продолжительность

Продолжительность предлагается использовать в качестве действительного признака.

4.4.3 Бюджет и кассовый сбор

Предположение: с течением времени как затраты на производство фильмов, так и ожидаемая выгода от проката возрастают.

Предлагается для бюджета и кассового сбора использовать в качестве признаков долю фильмов, для которых выбранный денежный признак за текущий год был меньше целевого.

4.4.4 Сюжет

В данной работе используются признаки, основанные на результатах тематического моделирования корпуса сюжетов фильмов при помощи таких методов как PLSA, робастный PLSA и регуляризованный PLSA. В качестве регуляризационных членов использовались пенализации, соответствующие разреживающему распределению Дирихле (с $\alpha, \beta < 1$, см. (6)), а также антикоррелирующий регуляризатор, предложенный в [26]:

$$R_{anti-corr}(\Theta, \Phi) = -\sum_{t_1 \in T, t_2 \in T, t_1 \neq t_2} Cov(\varphi_{t_1 \cdot}, \varphi_{t_2 \cdot})$$

Результатом тематического моделирования являются распределения фильмов по темам $\theta_{i\cdot}$. Для каждого пользователя u определим множество хороших фильмов M_u^G как множество тех фильмов, которым он поставил оценку 5. Множество плохих фильмов M_u^B включает в себя фильмы, оцененные пользователем u как 1 или 2. Определим среднюю тематику плохих и хороших фильмов по мнению пользователя u как

$$\theta_u^G = \frac{1}{|M_u^G|} \sum_{i \in M_u^G} \theta_{i\cdot}$$

$$\theta_u^B = \frac{1}{|M_u^B|} \sum_{i \in M_u^B} \theta_{i\cdot}$$

В случае пустоты множеств M_u^B и/или M_u^G векторы θ_u^G и θ_u^B определяются как среднее распределений фильмов по темам, взятое по всем фильмам.

В качестве признаков для пары (u, i) используются $\theta_u^G * \theta_{i\cdot}$, $\theta_u^B * \theta_{i\cdot}$.

4.5 Метод машинного обучения с учителем

В качестве метода обучения с учителем в данной работе был использован алгоритм Random Forest [27]. Выбор этого алгоритма обусловлен его нелинейностью, устойчивостью к переобучению. Данный алгоритм является агрегацией решающих деревьев, обученных на случайных подмножествах обучающего множества и признаков. Предсказание в случае регрессии производится усреднением по всем построенным деревьям.

Таким образом, Random Forest использует модифицированную технику бэггинга, подходящую для объединения большого количества моделей с малым смещением (bias) и большим разбросом (variance). Решающие деревья представляют собой хороший пример таких моделей. Достоинством решающих деревьев является способность достаточно хорошо описывать данные сложной структуры. Недостатком является склонность к переобучению. Известно, что Random Forest позволяет бороться с переобучением усредненной модели благодаря декорреляции отдельных деревьев в процессе процедуры их построения [28]. Таким образом, при использовании выбранной модели достигается оптимальный баланс между сложностью модели и обобщающей способностью.

Дополнительными достоинствами решающих деревьев, как базовых моделей для агрегации является способность работать как с действительными, так и категориальными признаками, а также устойчивость к выбросам в обучающих данных.

Другим эффективным подходом к агрегации нескольких моделей является бустинг. Основой данного метода является слабый классификатор, ошибка которого лишь немного лучше, чем у случайного. Производится последовательное повторение обучения модели с увеличением или уменьшением весов элементов множества прецедентов, в зависимости от того, допускала ли на них ошибку предыдущая модель. Последовательно получаемые данным образом модели составляют комитет. Было замечено, что в большинстве задач бустинг несколько превосходит бэггинг [28], однако несомненным достоинством последнего является простота в распараллеливании.

5 Детали реализации

5.1 Выбранный инструментарий

Методы предсказания пользовательских оценок реализованы в рамках набора инструментальных средств для рекомендательных систем LensKit [29]. LensKit реализован на языке программирования Java и предоставляет API для методов рекомендации и оценки качества их работы. Среди предлагаемых методов коллаборативная фильтрация на основе соседства, Slope One рекомендация [30] и FunkSVD[1].

В качестве языка программирования был выбран Scala [31], совместимый с Java, что позволяет использовать все многообразие существующих библиотек на Java.

Для использования алгоритмов машинного обучения был выбран набор библиотек Weka [32], также реализованный на Java.

Для получения модели SVD, используемой при вычислении признаков на основе дополнительной информации использована Scala-библиотека Breeze[2].

В качестве реализации методов тематического моделирования использована Scala-библиотека tm [3].

Для токенизации текстов с целью последующего тематического моделирования используется Java-библиотека OpenNLP [4].

5.2 Обучение модели регрессии

В случае возникновения проблем обучения модели на больших объемах данных, вызванных ограниченностью оперативной памяти предлагается производить обучение на случайно выбранном подмножестве обучающего множества.

5.3 Общая схема работы

Работа системы разбивается на 3 шага

[1] http://sifter.org/ simon/journal/20061211.html

[2] https://github.com/scalanlp/breeze

[3] https://github.com/ispras/tm

[4] https://opennlp.apache.org

1. построение лежащих в основе моделей SVD

2. формирование признаков для каждой пары (u, i) с известным рейтингом

3. обучение модели регрессии

5.4 Архитектура программной реализации

Набор программных средств LensKit состоит из модулей, реализующих доступ к данным, части алгоритмов предсказания оценок для фильмов и их ранжирования, а также фреймворка для тестирования их качества. Таким образом, задача сводится к реализации интерфейса основного класса, производящего оценки, ItemScorer.

Архитектура ключевой части выполненной программной реализации приведена на рисунке 1.

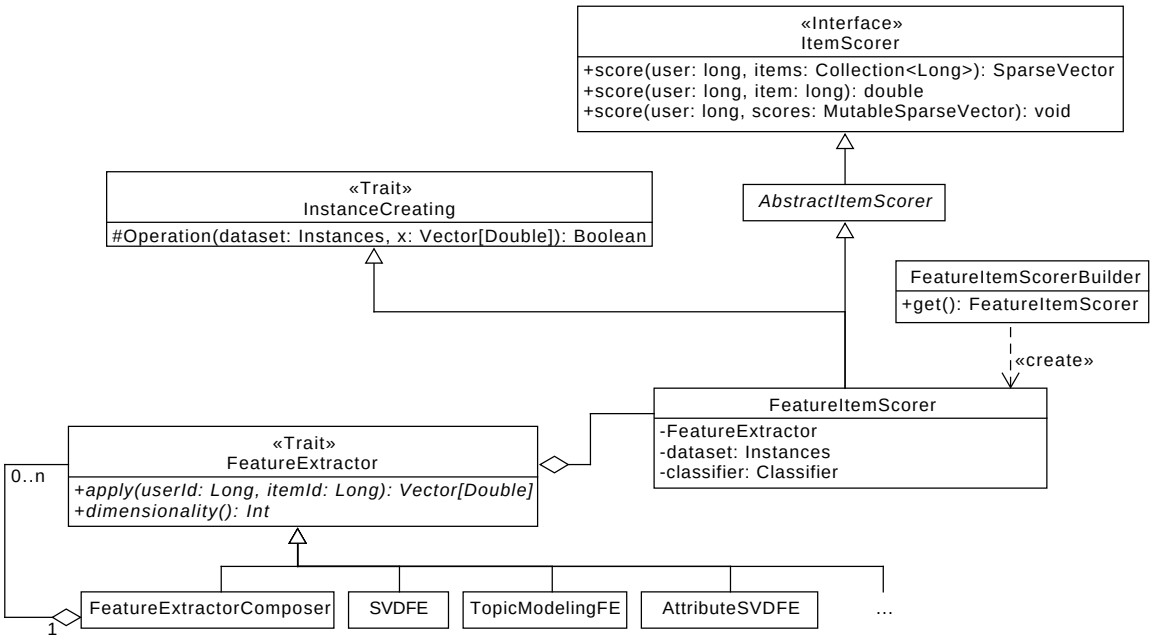

Рис. 1: Архитектура программной реализации

В интерфейсе *ItemScorer* объявлены методы для предсказания рейтинга для пары (u, i), а также сразу для нескольких предметов и заданного пользователя, с возвращением предсказаний и на месте. Главный класс *FeatureItemScorer* реализует

вышеупомянутый интерфейс. Для его инстанцирования с помощью системы внедрения зависимостей GRAPHT [33], используется $FeatureItemScorerBuilder$, таким образом, реализуется шаблон проектирования «Строитель» [34]. Класс $FeatureItemScorer$ использует $Classifier$ из библиотеки Weka и *dataset* в ее формате. Вспомогательный трейт $InstanceCreating$ отвечает за добавление прецедента в датасет. Наследники трейта $FeatureExtractor$ реализуют получение части признаков для данной пары (u, i) на основе определенного вида дополнительной информации. Наследник $FeatureExtractoComposer$ позволяет последовательно применять извлечение различных признаков и конкатенируя их результаты, реализуя шаблон проектирования «Компоновщик» [34].

6 Эксперимент

6.1 Описание данных

В экспериментах были использованы датасеты $MovieLens100K$ и $1M$ [35]. Они были выпущены в 1998 и 2003 годах соответственно и являются одними из наиболее часто используемых при оценке качества новых методов рекомендации. Данные собраны с сайта онлайн-рекомендаций фильмов MovieLens. Каждый пользователь имеет не менее 20 рейтингов. Оба датасета содержат следующую дополнительную информацию о пользователях

- возраст
- пол
- профессия
- почтовый индекс

а также дополнительную информацию о фильмах

- название
- год выпуска
- принадлежность жанрам, в виде бинарного вектора

Плотность матрицы R определяется как доля ее ненулевых элементов. Параметры датасетов представлены в таблице 1

	ml-100k	ml-1m
количество пользователей	943	6040
количество фильмов	1681	3706
количество рейтингов	100 000	1 000 209
плотность матрицы R, %	6.3	4.47

Таблица 1: Параметры датасетов MovieLens

6.2 Сбор дополнительной информации о фильмах

Так как стоит задача использования и исследования влияния на качество оценки различных видов информации, было решено обогатить датасет $MovieLens100K$ большим количеством информации о фильмах из сети Интернет. Кроме того, было замечено, что не вся дополнительная информация о фильмах в оригинальном датасете верна. Отметим, что интернет-страницы не подходят под единый шаблон, что затрудняет их парсинг, вызывая необходимость для корректности вносить ручные исправления, либо составлять сложные правила пропуска исключений. Поэтому для большего датасета $MovieLens1M$ сбор дополнительной информации не производился ввиду нехватки человеческих ресурсов.

6.2.1 Получение интернет-страницы для фильма

Один из датасетов $MovieLens$, $100K$, снабжен файлом со ссылками на соответствующую страницу онлайн-базы IMDB [5] для каждого фильма. К сожалению, они не являются рабочими, в силу возраста датасета. Поиск интернет-страницы, соответствующей фильму производился в интернет-энциклопедии Wikipedia [6], см. алгоритм 4. Скачивание страниц производится средствами стандартной библиотеки языка Scala. В результате имеем не более одной страницы для каждого фильма.

6.3 Отбор типов дополнительной информации

На странице, соответствующей каждому фильму может содержаться информация, соответствующая одному из типов, таких как дата выпуска, по какому произведению снят, страна производства и т.д. Сгруппируем все имеющиеся дополнительные данные по типам. Откажемся от использования тех данных, которые присутствуют для малого количества фильмов. Оставшиеся типы данных:

- дата выпуска
- бюджет
- кассовые сборы

[5] http://www.imdb.com/help/show_leaf?about

[6] https://en.wikipedia.org/wiki/Main_Page

Алгоритм 4 Алгоритм получения страницы Wikipedia для фильма

Вход: название фильма n, год его выпуска y;

Выход: соответствующая ему страница p или $\emptyset$

сформировать поисковую строку r от n, y
для всех $p \in P$, где P множество страниц, соответствующих ответам на r
  для всех $c \in C^{\mathcal{Y}}$, где $C^{\mathcal{Y}}$ условия на страницу, зависящие от года, в порядке убывания приоритета
    если $c(p)$ **верно то**
      вернуть p
  для всех $c \in C^{\bar{\mathcal{Y}}}$, где $C^{\bar{\mathcal{Y}}}$ условия на страницу, не зависящие от года в порядке убывания приоритета
    если $c(p)$ **верно то**
      вернуть p
вернуть $\emptyset$

- продолжительность
- сюжет

Проценты фильмов с имеющейся информацией по ее типам для датасетов $MovieLens100K$ и $1M$ приведен в таблице 2.

	ml-100k	ml-1m
дата выпуска	87	91
продолжительность	91	95
бюджет	56	59
кассовый сбор	74	73
сюжет	79	82

Таблица 2: Проценты фильмов, для которых получена информация данного типа

6.4 Меры качества

В зависимости от области применения рекомендательного метода (см. 1.3) для оценки качества могут использоваться метрики для определения точности оценки или же ранговые метрики.

6.4.1 Точности оценки

Для оценки качества точности оценки разработанного метода были выбраны следующие меры

$$MAE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} |\hat{r}_{ui} - r_{ui}|}$$

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2}$$

Мера $RMSE$ приобрела большую популярность благодаря требованию улучшить ее в известном конкурсе по рекомендации от Netflix [7].

Обе этих меры направлены на измерение точности предсказания рейтинга и являются одними из наиболее часто используемых. Отличие RMSE от MAE состоит в том, что она сильнее реагирует на большие отклонения оценки от истинного рейтинга [3].

6.4.2 Ранговая

Мотиваций используемых для измерения качества ранжирования метрик поощрение появления предметов с высокой релевантностью в начале списка ранжирования.

Для оценки качества ранжирования фильмов была выбрана мера $NDCG$ (Normalized Discounted Cumulative Gain).

Рассмотрим сначала меру Discounted Cumulative Gain, предложенную в [36] для задачи информационного поиска:

$DCG(i) = \begin{cases} G(i), \text{ i} = 1 \\ DCG(i-1) + \dfrac{G(i)}{\log_b i} \text{ иначе} \end{cases}$, здесь i – ранг объекта, G – выигрыш от него

В нашем случае с фильмами роль выигрыша играют истинные рейтинги от пользователя в тестовой выборке. Значения меры усредняются по всем пользователям.

В той же работе [36] предлагается производить сравнение полученного таким образом вектора DCG с наилучшим теоретически возможным.

Используемая мера $NDCG$:

$NDCG(i) = \dfrac{DCG(i)}{IDCG(i)}$, где $IDCG = DCG$, посчитанному на идеальной последовательности

[7] https://en.wikipedia.org/wiki/Netflix_Prize

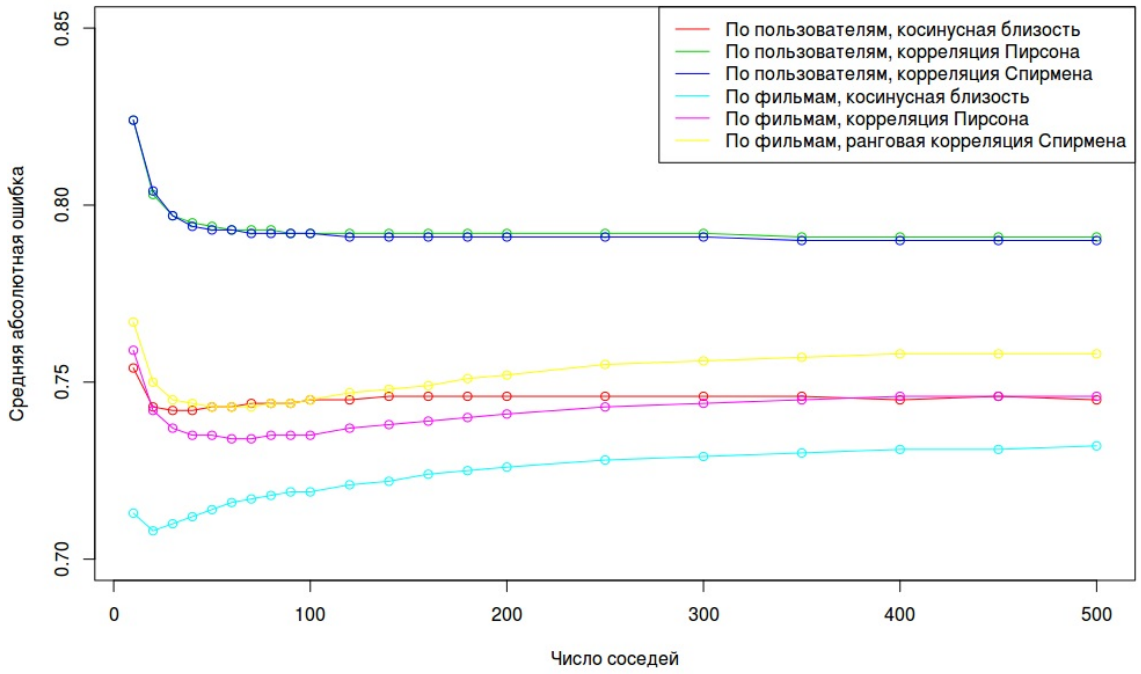

Рис. 2: Зависимость MAE от количества соседей для датасета ML 100K

6.5 Подбор параметров коллаборативных методов

Для произведения сравнительной оценки качества будут использоваться методы, основанные на соседстве, и SVD. Был произведен выбор моделей для сравнения, демонстрирующих наилучший результат. Для методов User-User и Item-Item менялись меры близости, используемые для выбора соседей, а также количество соседей. Оценка качества будет производиться по мерам MAE и $RMSE$. Из экспериментов в разделе 6.6, что ранговая мера при изменении параметров алгоритмов меняется совсем уж незначительно.

Результаты экспериментов для датасета ML 100K представлены на рисунках 2, 3.

Видно, что для методов, основанных на соседстве, на датасете ML 100K лучшие результаты относительно обеих мер достигаются методом Item-Item с косинусной мерой близости, при количестве соседей 20.

Результаты экспериментов для датасета ML 1M для методов, основанных на соседстве представлены на рисунках 4, 5.

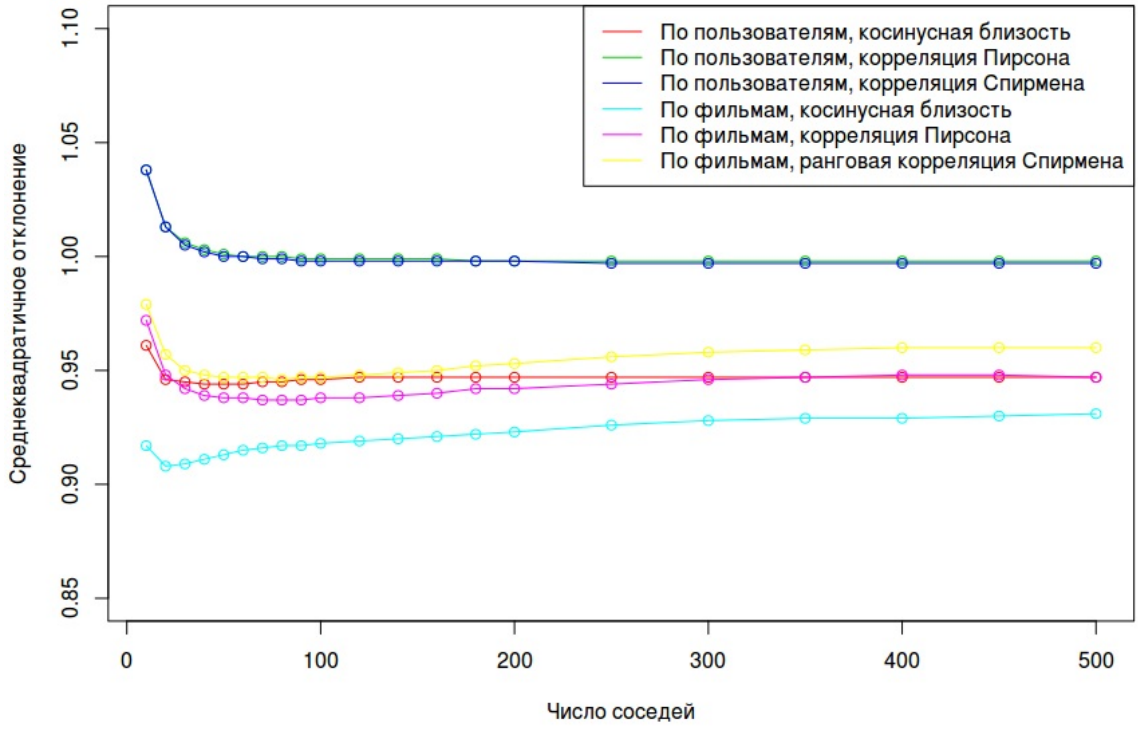

Рис. 3: Зависимость RMSE от количества соседей для датасета ML 100K

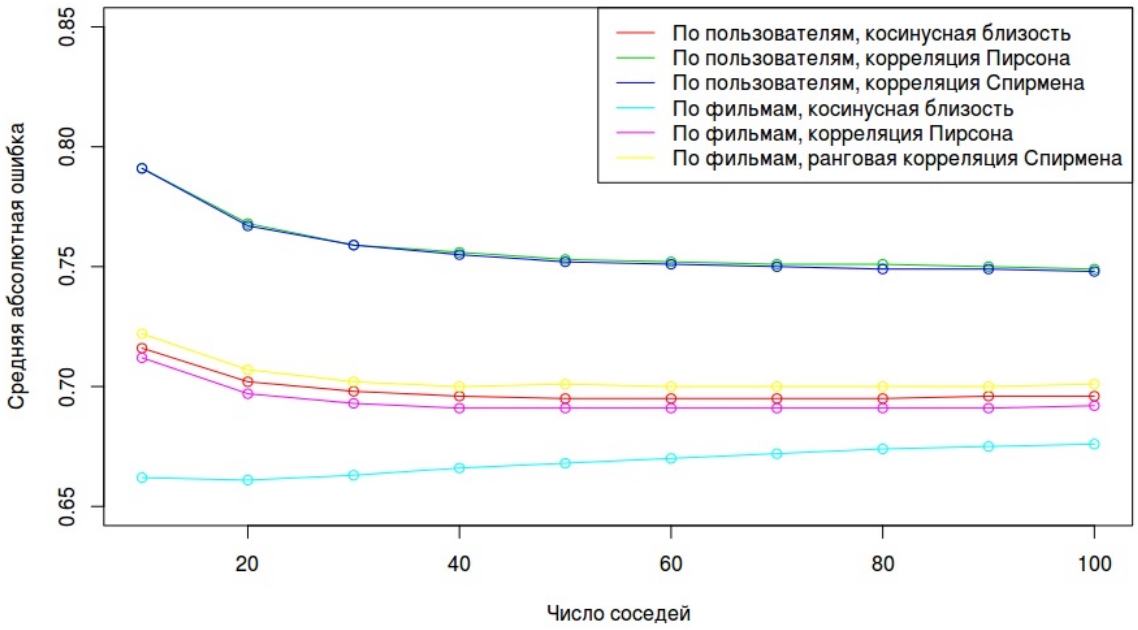

Рис. 4: Зависимость MAE от количества соседей для датасета ML 1M

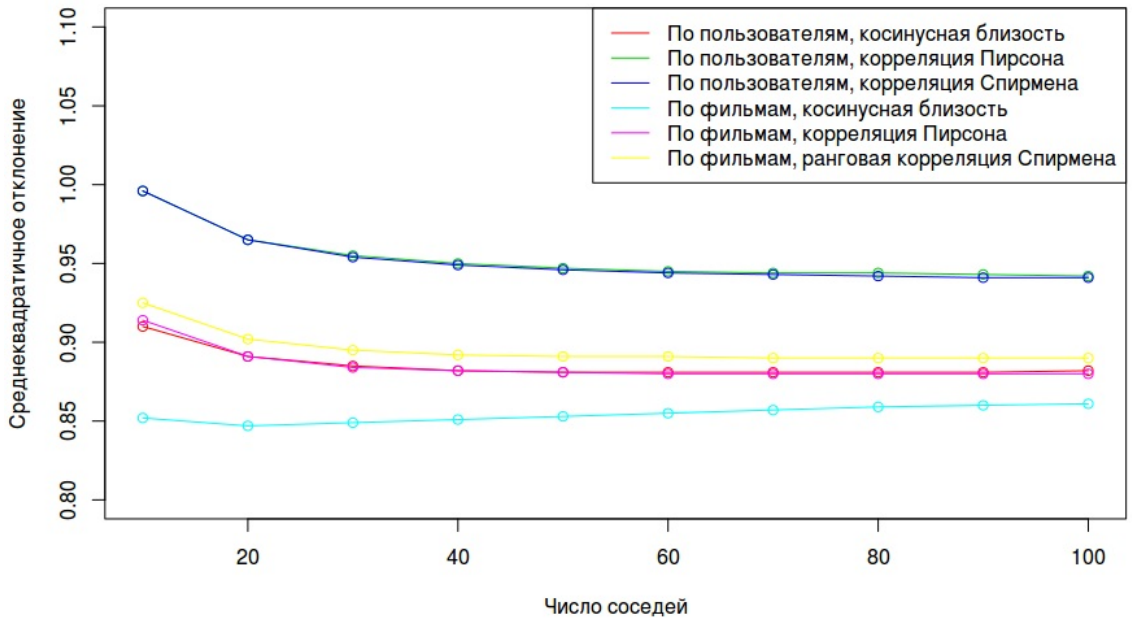

Рис. 5: Зависимость RMSE от количества соседей для датасета ML 1M

Видно, что для методов, основанных на соседстве, на датасете ML 1M лучшие результаты относительно обеих мер достигаются методом Item-Item с косинусной мерой близости, при количестве соседей 20.

В модели SVD изменялось количество признаков.

Результаты экспериментов для датасета ML 100K представлены на рисунках 6, 7.

Для SVD на датасете ML 100K лучшие результаты относительно обеих мер качества достигаются при количестве признаков 80.

Результаты изменения количества признаков в модели SVD для датасета ML 1M представлены на рисунках 8, 9.

Для SVD на датасете ML 1M лучшие результаты относительно обеих мер качества достигаются при количестве признаков 40.

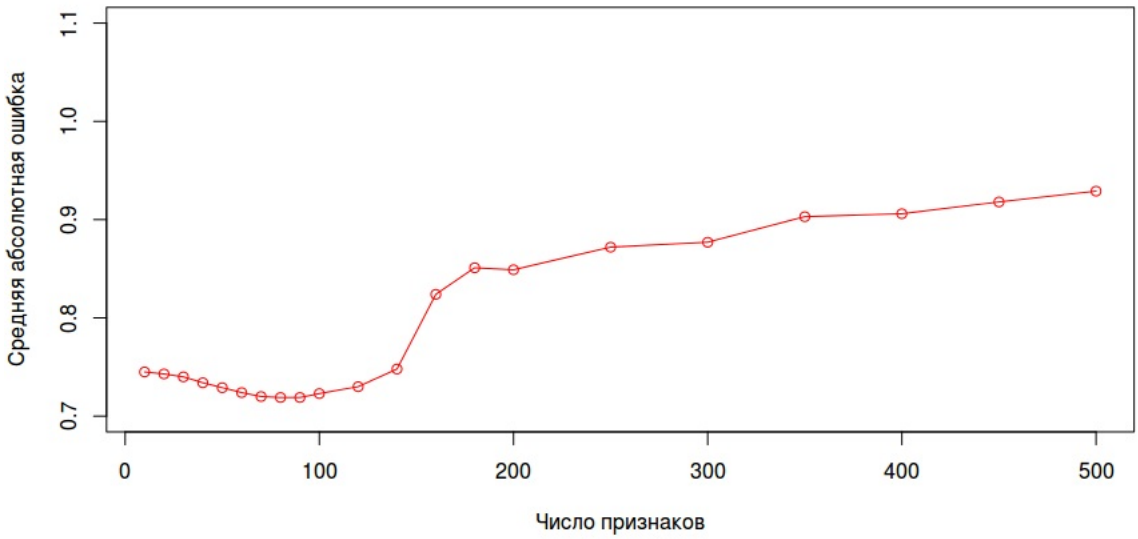

Рис. 6: Зависимость MAE от количества признаков SVD для датасета ML 100K

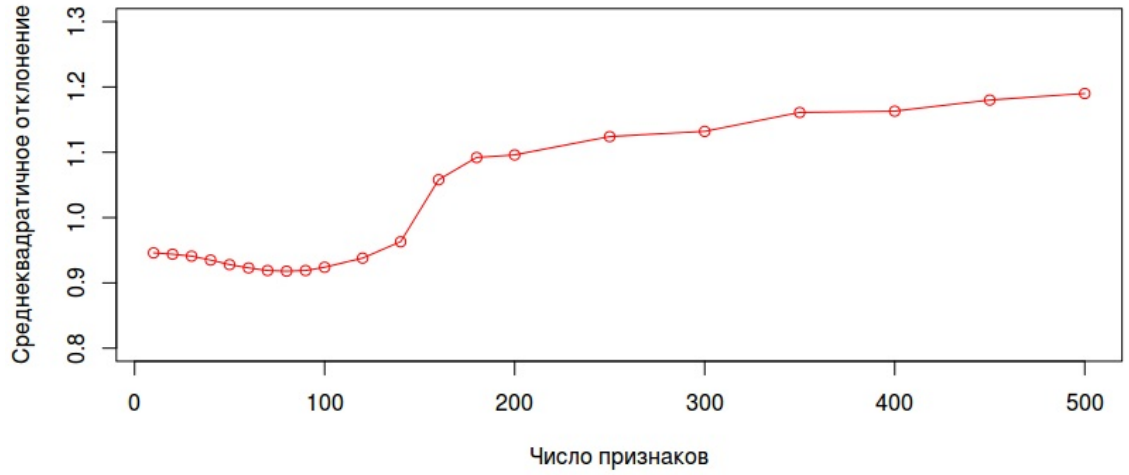

Рис. 7: Зависимость RMSE от количества признаков SVD для датасета ML 100K

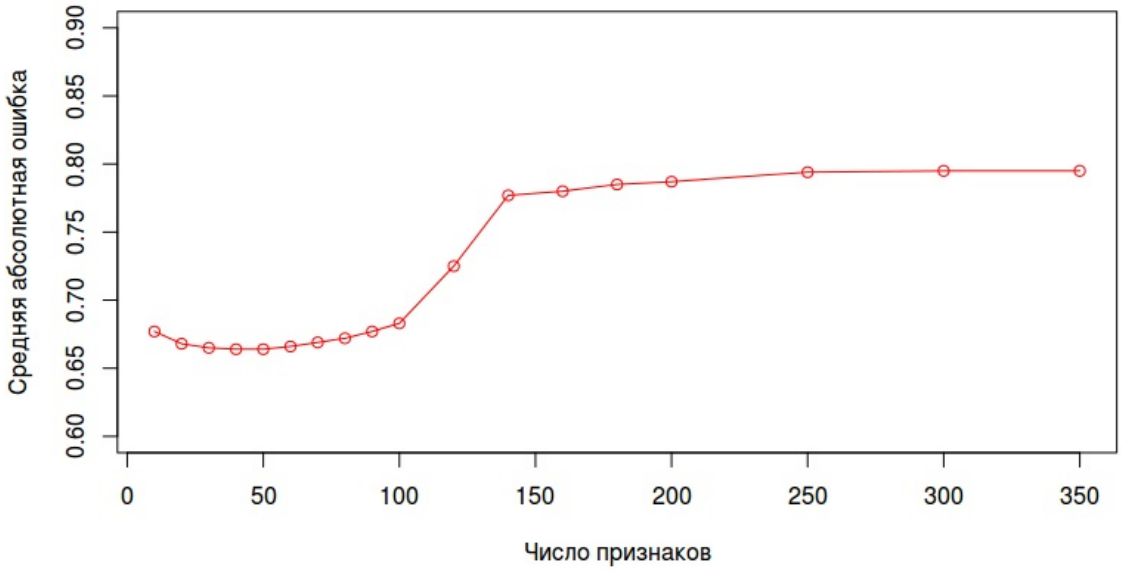

Рис. 8: Зависимость MAE от количества признаков SVD для датасета ML 1M

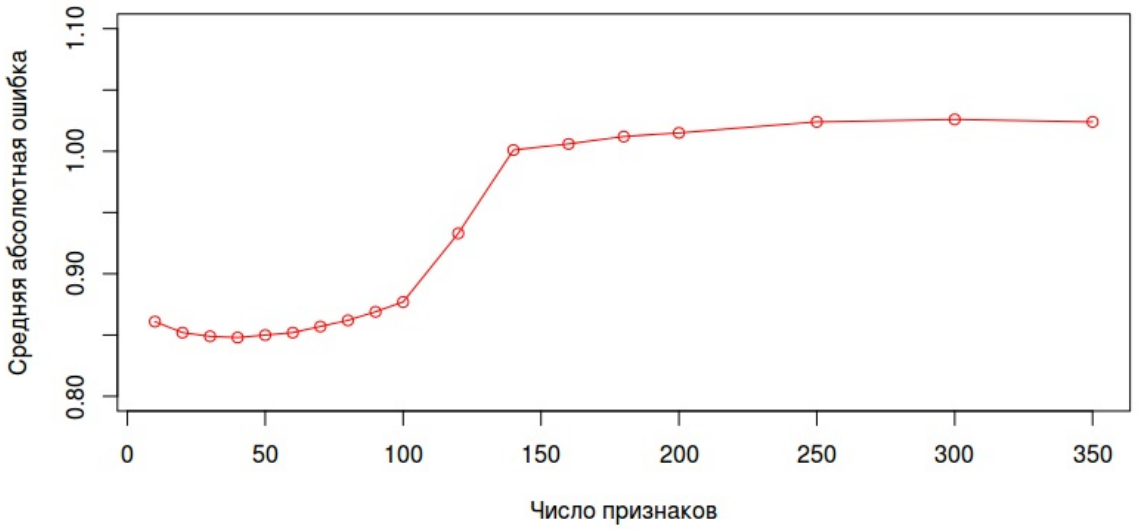

Рис. 9: Зависимость RMSE от количества признаков SVD для датасета ML 1M

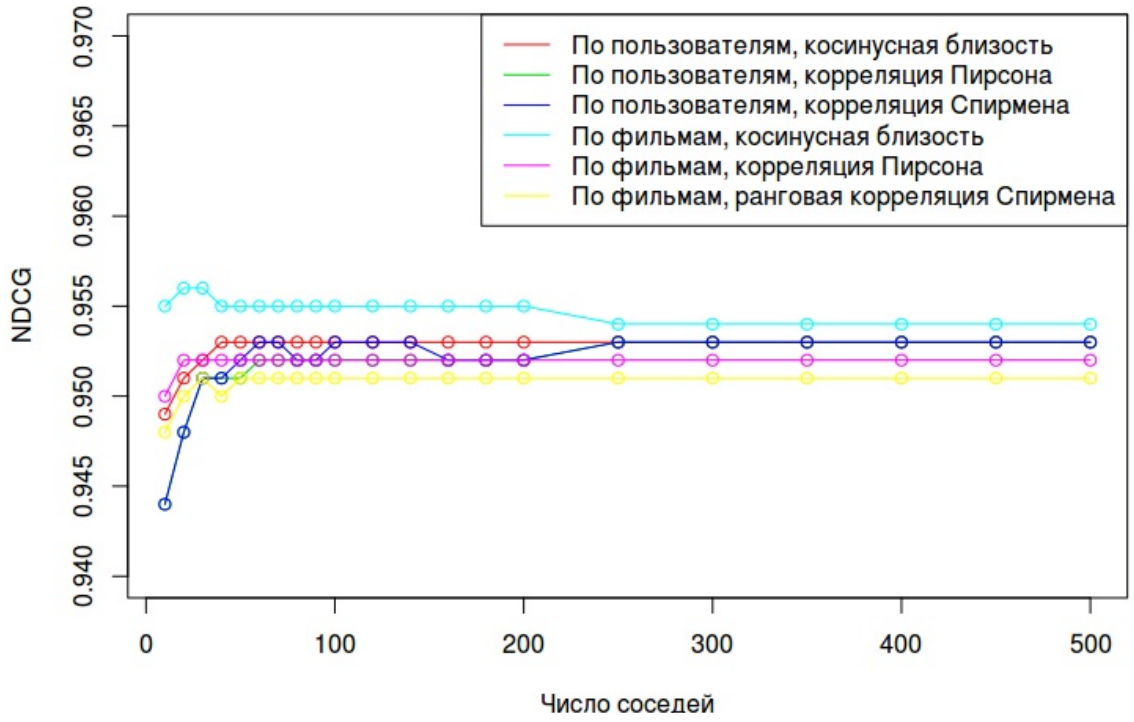

Рис. 10: Зависимость NDCG от количества соседей для датасета ML 100K

6.6 Результаты измерения качества ранжирования для стандартных методов

Результаты экспериментов для методов, основанных на соседстве, для всех фильмов из тестового набора на датасетах ML 100K и 1M представлены на рисунках 10 и 11 соответственно.

Видно, что наилучшие значения меры на обоих датасетах достигаются для методов, основанных соседстве теми же методами при тех же параметрах, что и лучшие значения мер точности оценок рейтингов.

Результаты экспериментов для SVD для всех фильмов из тестового набора на датасетах ML 100K и 1M представлены на рисунках 12 и 13 соответственно.

Наилучшие значения ранговой меры достигаются при 90 признаках для датасета ML 100K и от 30 до 70 признаков для датасета ML 1M

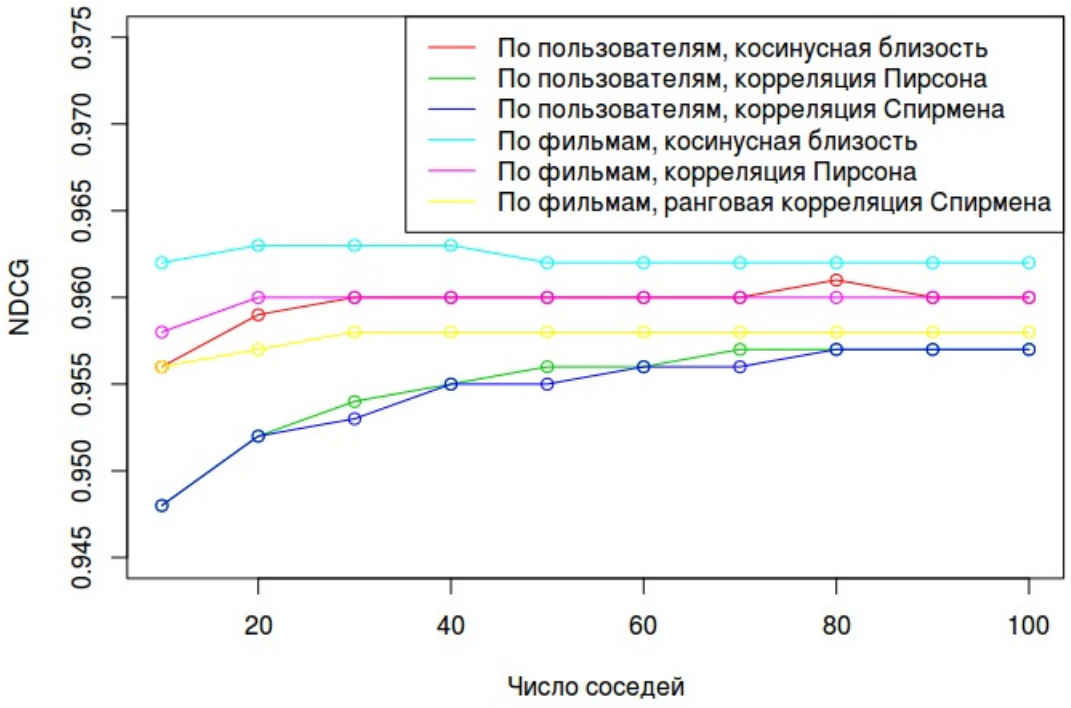

Рис. 11: Зависимость NDCG от количества соседей для датасета ML 1M

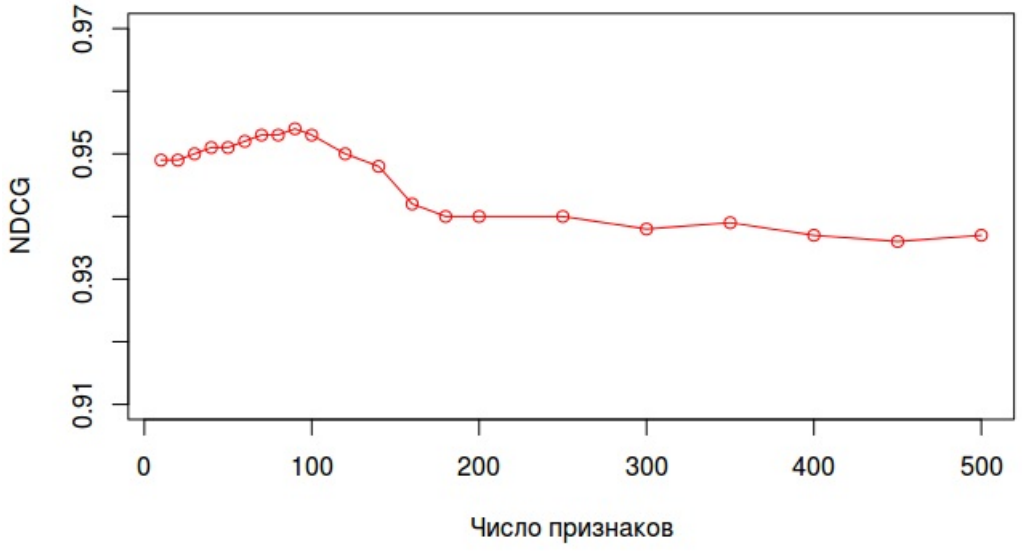

Рис. 12: Зависимость NDCG от количества признаков для датасета ML 100K

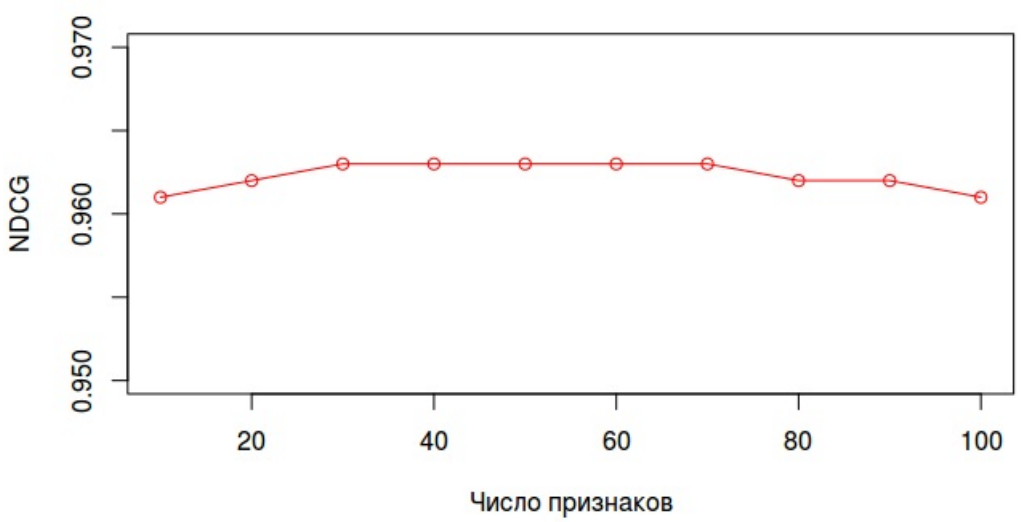

Рис. 13: Зависимость NDCG от количества признаков для датасета ML 1M

6.7 Экспериментальная оценка качества предсказания рейтинга

Были проведены эксперименты, оценивающие качество предсказания рейтинга в сравнении со стандартными коллаборативными методами. Оценка производилась стандартным методом перекрестной проверки с пятью частями (5-fold cross validation). Для получения оценки для каждого из рекомендательных методов использовались одинаковые разбиения. Результаты приведены в таблицах 3 и 4, на датасете MovieLens1M обучение модели регрессии производилось для 20% и 100% обучающих данных.

6.7.1 Доверительные интервалы

Построим доверительные интервалы для мер качества для стандартного алгоритма, демонстрирующего наилучшие результаты и для разработанного метода, запускаемого с использованием всех предложенных признаков на основе дополнительных данных.

Для каждого из методов повторялся эксперимент с тестовой выборкой, составляющей 20% от всех данных. Таким образом была получена выборка из 10 независимых одинаково распределенных реализаций величины: $\xi_i : 1 \leqslant i \leqslant 10$. С использованием

	MAE	RMSE	NDCG
User-User	0.742	0.944	0.953
Item-Item	0.708	0.908	0.956
SVD	0.738	0.939	0.954
RF(SVD)	0.277	0.352	
RF(SVD + Genre + Year)	0.252	0.324	
RF(SVD + Genre + Year + Gender)	0.252	0.322	
RF(Topics)	0.337	0.453	
RF(SVD + Genre + Money + Time + Plot)	0.248	0.317	0.977

Таблица 3: Качество предсказания оценки на датасете ML 100K

	MAE	RMSE	NDCG
User-User	0.695	0.88	0.961
Item-Item	0.66	0.847	0.963
SVD	0.668	0.85	0.963
RF(SVD), 20% обучающих данных	0.679	0.862	
RF(SVD), 100% обучающих данных	0.252	0.317	
RF(SVD + Genre) , 20% обучающих данных	0.645	0.824	
RF(SVD + Genre + Money + Time + Plot), 20% обучающих данных	0.591	0.772	0.979
RF(SVD + Genre + Money + Time + Plot), 100% обучающих данных	0.237	0.301	

Таблица 4: Качество предсказания оценки на датасете ML 1M

полученной выборки строится нормальный доверительный интервал на выборочное среднее $\overline{\xi}$. Для оценки дисперсии выборочного среднего воспользуемся фактом, что

$$\mathrm{D}\overline{\xi} = \frac{1}{10}\mathrm{D}\xi_1 \tag{15}$$

А в качестве несмещенной оценки $\mathrm{D}\xi_1$ возьмем

$$\hat{\mathrm{D}}\xi_1 = \frac{1}{9}\sum_{i=1}^{10}(\xi_i - \overline{\xi})^2 \tag{16}$$

95% доверительный интервал построим как $\left(\overline{\xi} - 1.96\sqrt{\hat{\mathrm{D}}\overline{\xi}}, \overline{\xi} + 1.96\sqrt{\hat{\mathrm{D}}\overline{\xi}}\right)$.

Результаты приведены в таблице 5.

	MAE	RMSE	NDCG
Standard, ML 100K	0.7313 ± 0.023	0.9314 ± 0.025	0.965 ± 0.027
New, ML 100K	0.2459 ± 0.0090	0.3155 ± 0.010	0.999 ± 0.0002
Standard, ML 1M	0.6975 ± 0.0050	0.8847 ± 0.0058	0.9607 ± 0.00097
New, ML 1M	0.2364 ± 0.0022	0.301 ± 0.0027	1 ± 0

Таблица 5: Доверительные интервалы для мер качества

Показано, что даже применение случайного леса к SVD, без использования дополнительных признаков, дает более чем двукратное улучшение качества оценки по обеим используемым мерам точности оценки. Использование предложенных признаков дает дополнительное увеличение точности оценки более 5% на обоих датасетах.

Также продемонстрировано увеличение метрики NDCG при использовании разработанного метода, таким образом, он, возможно, будет иметь преимущество перед стандартными методами в задачах ранжирования рекомендаций.

7 Заключение

1. Исследованы существующие методы предсказания оценки пользователями фильмов с учетом коллаборативной информации, а также дополнительной информации, основанной на контенте.

2. Разработан и реализован собственный метод предсказания оценки с учетом всех имеющихся видов дополнительной информации, а также всех типов информации, которыми удалось дополнить датасеты.

3. Выполнена экспериментальная оценка качества разработанного метода, продемонстрировано его существенное превосходство над стандартными методами по выбранным мерам качества.

Список литературы

[1] Dokyun Lee and Kartik Hosanagar. Impact of recommender systems on sales volume and diversity. In *Proceedings of the International Conference on Information Systems - Building a Better World through Information Systems, ICIS 2014, Auckland, New Zealand, December 14-17, 2014*, 2014.

[2] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. The adaptive web. chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.

[3] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[4] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[5] Marko Balabanović. An adaptive web page recommendation service. In *Proceedings of the First International Conference on Autonomous Agents*, AGENTS '97, pages 378–385, New York, NY, USA, 1997. ACM.

[6] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 714–720, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[7] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2):81–173, February 2011.

[8] Jon Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310, October 2002.

[9] Jon Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310, October 2002.

[10] Majid Hatami and Saeid Pashazadeh. Improving results and performance of collaborative filtering-based recommender systems using cuckoo optimization algorithm.

[11] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[12] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *IN ACM WEBKDD WORKSHOP*, 2000.

[13] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[14] Jong-Seok Lee, Chi-Hyuck Jun, Jaewook Lee, and Sooyoung Kim. Classification-based collaborative filtering using market basket data. *Expert Syst. Appl.*, 29(3):700–704, October 2005.

[15] Buhwan Jeong, Jaewook Lee, and Hyunbo Cho. An iterative semi-explicit rating method for building collaborative recommender systems. *Expert Syst. Appl.*, 36(3):6181–6186, 2009.

[16] D. Zhang. An item-based collaborative filtering recommendation algorithm using slope one scheme smoothing. In *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on*, volume 2, pages 215–217, May 2009.

[17] Y. Ren and S. Gong. A collaborative filtering recommendation algorithm based on svd smoothing. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, volume 2, pages 530–532, Nov 2009.

[18] Niloofar Rastin and Mansoor Zolghadri Jahromi. Using content features to enhance performance of user-based collaborative filtering performance of user-based collaborative filtering. *CoRR*, abs/1402.2145, 2014.

[19] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[20] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *In Proc. of ICML*, pages 704–711. AAAI Press, 2003.

[21] Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *Int. J. Approx. Reasoning*, 51(7):785–799, September 2010.

[22] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 193–202, 2014.

[23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[24] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[25] Anna Potapenko and Konstantin Vorontsov. *Robust PLSA Performs Better Than LDA*, pages 784–787. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[26] Konstantin Vorontsov, Anna Potapenko, and Alexander Plavin. *Additive Regularization of Topic Models for Topic Selection and Sparse Factorization*, pages 193–202. Springer International Publishing, Cham, 2015.

[27] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[28] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[29] Michael D. Ekstrand, Michael Ludwig, Jack Kolb, and John T. Riedl. Lenskit: A modular recommender framework. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 349–350, New York, NY, USA, 2011. ACM.

[30] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining (SDM'05)*, 2005.

[31] Martin Odersky, Lex Spoon, and Bill Venners. *Programming in Scala: A Comprehensive Step-by-step Guide*. Artima Incorporation, USA, 1st edition, 2008.

[32] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations, 1999.

[33] Michael D. Ekstrand and Michael Ludwig. Dependency injection with static analysis and context-aware policy. *Journal of Object Technology*, 15(1):1:1–31, February 2016.

[34] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[35] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, December 2015.

[36] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM.