Курсовая работа

Система построения ожидаемых оценок товара пользователем на основе уже существующих

Введение

- Существует множество сервисов, предоставляющих пользвателям возможность купить (прослушать, прочитать,…) тот или иной товар (музыку, книгу, …). Как правило, количество предоставляемых товаров значительно превышает объём, с которым человек может справиться своими силами, к тому же это количество со временем растёт
- В связи с этим возникает нужда в рекомендательных системах, которые бы предоставляли пользователю меньшую выборку товаров, с которой он уже мог бы справить своими силами

Введение

- Не все сервисы имеют налаженную возможность оценивать схожесть товаров / пользователей по их характеристикам, не связанным с проставленными оценками, так что определённого внимания заслуживают алгоритмы предсказания ожидаемой оценки, работающие на основе только проставленных оценок

Цель работы

- Разработать инструмент, на основе исключительно данных об уже выставленных оценках предсказывающий, какую оценку пользователь поставит новому для него товару, для возможности дальнейшего использования в алгоритмах рекомендаций

Постановка задачи

- U — множество пользователей ресурса (читателей, …)
- B — множество товаров (книг, …), предлагаемых пользователям
- F — матрица, где f_{ij} есть оценка (целое число из интервала [1..10]), поставленная пользователем i товару j

Постановка задачи

- Predict: U*B $\rightarrow$ R, т. е. predict(user, object) = f^*_{ij}, где f^*_{ij} — ожидаемая оценка, которую пользователь i поставит объекту j, посчитанная на основе имеющихся данных об оценках, данных пользователями объектам

Основные подходы

- Корреляционные модели: хранится вся матрица F, близость u´ к u определяется корреляцией столбцов, близость b´ к b — строк
- Латентные модели: хранятся профили пользователей и объектов, близость смотрится по сходству профилей, не требуется хранения всей F
- Для реализации были выбраны корреляционные модели, потому что без дополнения матрицы предпочтений «внешними» данными (тегами, …), латентные методы теряют своё основное преимущество: решение проблемы «холодного старта»

Примеры существующих алгоритмов

- Общий топ (каждому пользователю просто рекомендуются N товаров с наибольшей средней оценкой)
- Топ похожих товаров (для каждого товара находятся наиболее похожие, которые и рекомендуются на странице этого товара каждому пользователю)
- Общий недостаток перечисленных: отсутсвие персонализации

Примеры существующих алгоритмов

- User-based коллаборативная фильтрация: при предсказывании оценки пользователем i товара j, выбираются наиболее близкие к i пользователи (т. е. оценившие те же товары на похожие оценки, либо N ближайших, либо которые ближе чем некоторый порог), уже поставившие оценку j, берётся средневзвешенная по их оценкам (вес — близость)

Примеры существующих алгоритмов

- Item-based коллаборативная фильтрация: как User-based, только рассматривается близость товаров вместо пользователей
- Недостаток коллаборативной фильтрации: новые и нетипичные пользователи и товары

Реализация

- Для реализации инструмента предсказания были выбраны алгоритмы User-based и Item-based коллаборативной фильтрации. Каждый из них был реализован в 2 вариантах: для случая хранения матрицы предпочтений как матрицы и для случая хранения её в виде списков оцененных товаров для каждого пользователя (списков оценивших пользователей для каждого товара)

Реализация

- В качестве меры близости для первого варианта была выбрана косинусная мера как стандартная для задач предсказания оценок;
- Для второго же варианта в качестве меры близости использовалась косинусная мера между общими частями списков оценок, домноженная на долю общей части от объединения списков

Реализация

- Итоговый алгоритм предсказания оценки в обоих вариантах сводится к следующим шагам:
  - Для каждого пользователя (товара) выбираются другие, близость которых превышает указанный порог и у которых есть оценка в интересующей ячейке матрицы предпочтений
  - Предсказанной оценкой становилась средневзвешенная, где весом оценки была мера близости пользователя, поставившего её (товара, которому поставил интересующий нас пользователь)

Реализация

- Пусть sim:U*U→R — функция близости пользователей (для объектов, соответственно, sim:B*B→R). Тогда шаги алгоритма можно записать в следующем виде:
  - Similar(i) = {u∈U:sim(u,i)>α && f_{uj}!=0}
  - Predict(i,j) = sum(f_{uj}*sim(u,i):u∈Similar(i)) / sum(sim(u,i):u∈Similar(i))

Тестирование

- Для тестирования разработанного алгоритма была взята база данных оценок с сайта shikimori.one, само тестирование проходило следующим образом: для каждойуже известной оценки f_{ij} высчитывалось predict(i,j), модуль разности между предсказанной и актуальной оценкой, потом эти модули суммировались и делились на количество известных оценок
- Т.е. алгоритмы характеризовались средним отклонением предсказания от актуальной оценки

Тестирование

- Test = sum(|predict(i,j)-f_{ij}|:f_{ij}!=0) / |{(i,j):f_{ij}!=0}|
- В итоге среднее смещение для User-based алгоритмов оказалось примерно 1.205, а для Item-based 1.058

Пример работы

- Рассмотрим пользователя 58, его актуальные оценки, а также ожидаемые алгоритмами User-based и Item-based с установленным порогом 0 (т. е. рассматривались все пользователи/объекты)

Актуальные оценки	10	10	10	10	10	5
Предсказание User-based	6.8930	8.0	8.7338	0.0	9.0	7.5877
Предсказание Item-based	8.8979	9.0277	9.0805	8.9386	9.0559	9.9999

Итоги

- Реализован инструмент, способный двумя альтернативными способами вычислять ожидаемую оценку, которую пользователь i поставит объекту j, оба способа в среднем дают смещение порядка 1,2.