Прикладные задачи анализа данных

Семинар 11

Рекомендательные системы

Национальныи Исследовательский Университет
Высшая Школа Экономики

17 мая 2018

Постановка задачи

- Есть база пользователей U; есть база объектов I
- Хотим предлагать пользователю объекты из **нек. множества**, которые ему больше всего понравятся.
- Примеры того, что можно рекомендовать:
  - товары в интернет-магазине
  - контент соцсетей
  - фильмы, музыка.
- Будем считать, что для некоторых пар $u \in U$, $i \in I$ известны оценки r_{ui}, которые отражают степень заинтересованности в товаре (объекте).
- Примеры оценок:
  - для товаров в интернет-магазине: покупки, просмотры
  - для контента соцсетей: время прсмотра, клики, лайки, репосты
  - для фильмов: явные оценки пользователей.

Методы подбора рекомендаций

- Коллаборативная фильтрация (на основе похожести пользователя и товара)
- Модели со скрытыми переменными
- Контентные модели
- Факторизационные машины

Коллаборативная фильтрация — 1

- Два пользователя похожи, если они ставят товарам одинаковые оценки.
- $I_{uv} = \{i \in I | \exists r_{ui} \& \exists r_{vi}\}$ - множество товаров, для которых известны оценки пользователей u и v.
- $U_{ij} = \{u \in U | \exists r_{ui} \& \exists r_{vi}\}$ - множество пользователей.

Коллаборативная фильтрация — 2

Обозначения:

- $\hat{r}_u$, $\hat{r}_v$ — средние рейтинги по множеству пользователей I_{uv}.
- $\hat{r}_i$, $\hat{r}_j$ — средние рейтинги по множеству товаров U_{ij}.

Коллаборативная фильтрация — 3

- Сходство двух пользователей через корреляцию Пирсона:

$$w_{uv} = \frac{\sum\limits_{i \in I_{uv}} (r_{ui} - \hat{r}_u)(r_{vi} - \hat{r}_v)}{\sqrt{\sum\limits_{i \in I_{uv}} (r_{ui} - \hat{r}_u)^2} \sqrt{\sum\limits_{i \in I_{uv}} (r_{vi} - \hat{r}_v)^2}}.$$

- Сходство двух товаров через корреляцию Пирсона:

$$w_{ij} = \frac{\sum\limits_{u \in U_{ij}} (r_{ui} - \hat{r}_i)(r_{uj} - \hat{r}_j)}{\sqrt{\sum\limits_{u \in U_{ij}} (r_{ui} - \hat{r}_u)^2} \sqrt{\sum\limits_{u \in U_{ij}} (r_{vi} - \hat{r}_v)^2}}$$

User-based подход

- Ищем пользователей, похожих на данного u_0 — $U(u_0) = \{v \in U | w_{u_0 v} > \alpha\}$.
- Для каждого товара вычисляется, как часто он покупался пользователями из $U(u_0)$:

$$p_i = \frac{|\{u \in U(u_0) | \exists r_{ui}\}|}{|U(u_0)|}$$

- Пользователю рекомендуются k товаров с наибольшими значениями p_i.

Item-based подход

- Множество товаров, похожих на те, что интересовали данного пользователя u_0 — $I(u_0) = \{i \in I | \exists r_{u_0 i}, w_{i_0 i} > \alpha\}$.
- Для каждого товара вычисляется его сходство с пользователем:

$$p_i = max_{i_0 : \exists r_{u i_0}} w_{i_0 i}.$$

- Пользователю рекомендуются k товаров с наибольшими значениями p_i.

Рекомендации фильмов

- Есть известные оценки, которые пользователи поставили фильмам, которые уже просмотрели
- Есть 2 разные задачи:
  - спрогнозировать оценки, которые поставили бы пользователи фильмам, которые они еще не смотрели
  - порекомендовать пользователям то, что им больше понравится
- см. ноутбук Recsys problem 1

Измеряем качество

- RMSE, MAE - этими метриками мы оцениваем качество прогноза, а НЕ рекомендации!
- А нужно оценивать качество рекомендаций

Контентные модели — 1

- Хотим показать блок из 5 баннеров на сайте интернет-магазина
- Хотим максимизировать деньги, а не просто хорошо предсказывать оценку купит/не купит
- Хотим использовать признаковое описание объекта, пользователя и комбинированные признаки (а не только что-то)
- Хотим обучать любые древесные и линейные модели
- Объект выборки: (пользователь, товар)
- Нет негативных примеров: будем сэмплировать (negative sampling)

Контентные модели — 2

- Объект (пользователь, товар, таймстэмп)
- Задача классификации: купит/не купит (обучаем на logloss, предсказываем вероятность того, что пользователь купит этот товар)
- Признаковое описание товара, пользователя, таймстэмпа и комбинированные признаки

Как максмизировать деньги

- Хотим порекомендовать пользователю сколько-то объектов.
- Само значение метрики logloss (на которую обучались) ни о чем нам не говорит.
- Пусть мы выбрали товары для размещения. Полученные для этих товаров вероятности (по модели) — p_k.
- Мат. ожидание денег $= \sum_k p_k c_k$, где c_k — стоимость товара. Это и хотим максимизровать.

Отбор кандидатов

- Популярные
- Из тех же категорий
- Часто покупаемые с уже просмотренными
- Из заранее подготовленных списков похожих товаров

Что взять в качестве негативных примеров?

- Добавить к каждому позитивному примеру весь каталог как негативный (не реально)
- Случайные, с вероятностями, пропорциональными популярности объекта
- Самые популярные
- Товары, которые рекомендовал бы какой-то алгоритм, но они не были куплены

Метрики качества

- Precision@k = купленное из рекомендованного/k
- Recall@k = купленное из рекомендованного/количество покупок
- Эти метрики взвешиваем по сессиям внутри пользователя и получаем Average Precision@k и Average Recall@k
- Взвешенный ценами Recall@k = стоимость купленного из рекомендованного / стоимость покупок

Задание семинара

Скачиваем тут

- рекомендация фильмов (работа со спарс матрицами) Recsys problem 1.ipynb
- рекомендации товаров Recsys problem 2.ipynb

При подготовке семинара использовались

- материалы лекции Ильи Ирхина по курсу DMIA
- материалы лекции Евгения Соколова по курсу МО на ФКН ВШЭ