Введение в рекомендательные системы

Гиздатуллин Данил

Мы перегружены информацией

- Тысячи новостей и постов публикуются каждый день
- Миллионы фильмов, книг и песен просматриваются, читаются и слушаются каждый день
- Каждый день мы видим тысячи рекламных объявлений

Может ли Google помочь?

- Да, но только если мы точно знаем, что нам нужно
- Что если мы хотим какой-то интересный фильм?
  - Что значит «Интересный»?

Может ли Facebook помочь?

- Да, мне нравится контент моих друзей
  - Что если у меня мало друзей?
  - Что если мне не всегда нравится то, что они любят?

Могут ли эксперты помочь?

- Да, но это не масштабируется
  - Каждый получит одни и те же рекомендации
- Это то, что им нравится, не мне!
  - Если фильмы нравятся экспертам, это не значит что они понравятся обычным зрителям

Основная идея рекомендательных систем

- Рекомендовать нам то, что возможно нам понравится
  - Это не обязательно должно быть что-то популярное
  - Люди любят то, что находится в длинном хвосте
- Как?
  - На основе истории пользования сервисом
  - На основе того, что нравится другим людям

Примеры

- Amazon
- Netflix
- Google news
- Spotify
- Etc.

Важность рекомендаций

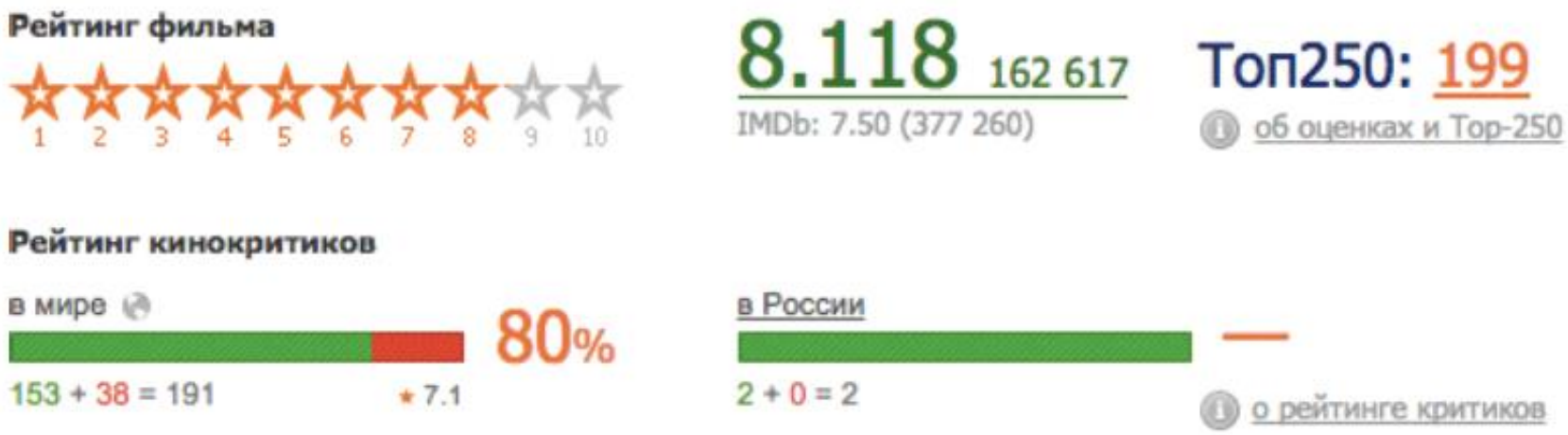
- Netflix
  - 2/3 просмотров происходят из рекомендации
- Google News
  - CTR на рекомендациях более 38%
- Amazon
  - 35% продаж происходит через рекомендации

Виды рекомендательных систем

1. Неперсонализованные
   - Используя агрегированные данные (например, средний рейтинг)
2. Content-based
   - Пользователю рекомендуются объекты, похожие на те, с которыми пользователь уже взаимодействовал
   - Похожие оцениваются по признакам содержимого объектов
   - Сильная зависимость от предметной области, полезность рекомендаций ограничена
3. Коллаборативная фильтрация
   - Для рекомендаций используется история оценок как самого пользователя, так и других пользователей
   - Более универсальный подход, часто дает лучший результат
   - Есть свои проблемы (например, холодный старт)

Неперсонализованные рекомендации

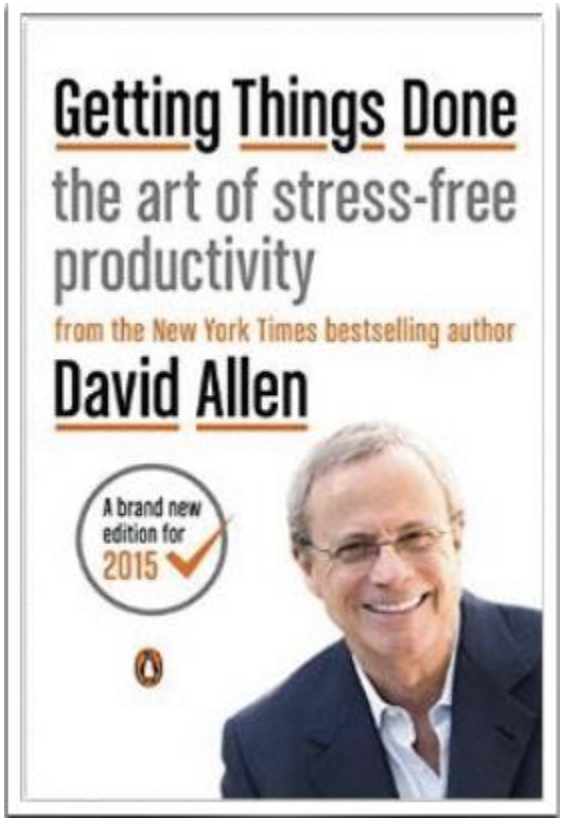
- Рейтинги kinopoisk.ru



Гарри Поттер и филосовский камень

Неперсонализованные рекомендации

- Рейтинг amazon.com



Проблемы с рейтингами

- Явные рейтинги
  - Разные шкалы
  - Разброс рейтингов
- Неявные рейтинги
  - Покупки (понравилось или нет?)
  - Время на сайте (а если отвлекся?)
  - Клик (является ли клик сигналом, а что после клика?)
- Накрутки

Формулы для рейтингов

- $P_i = \frac{\sum_{u=1}^{n} r_{ui}}{n}$
  - Что если у объекта мало рейтингов?
- $P_i = \frac{\sum_{u=1}^{n} r_{ui} + k\mu}{n+k}$ где
  - μ-глобальное среднее
  - k-нужно подбирать

User-based и Item-based (Memory-based)

- $u \in U$ – множество пользователей
- $i \in I$ – множество объектов
- $(r_{ui}, u, i, \dots) \in D$ – множество событий

Хотим предсказать:

- Предпочтение: $\hat{r}_{ui} = Predict(u, i, \dots) \approx r_{ui}$
- Персональные рекомендации:
  $u \to (i_1, \dots, i_k) = Recommend_k(u, \dots)$
- Похожие объекты:
  $i \to (i_1, \dots, i_M) = Similar_M(i)$

Меры сходства между пользователями и между объектами

- Коэффициент корреляции Пирсона:
  - Для пользователей
    - $I_{uv} = \{i \in I | \exists r_{ui} \& \exists r_{vi}\}$
    - $w_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$
  - Для айтемов
    - $U_{ij} = \{u \in U | \exists r_{ui} \& \exists r_{uj}\}$
    - $w_{ij} = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}}$

Упражнение

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	5	4	5			
User 2	4		5			
User 3		3	5		4	
User 4				3	4	
User 5			4	2	4	
User 6	3					5

- Вычислить similarity между User4 и User5
- Вычислить similarity между Item2 и Item3

User-based

$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U_i} sim(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in U_i} sim(u,v)}$, где U_i - топ N пользователей, наиболее похожих на пользователя u

Недостатки:

- Нечего рекомендовать новым или нетипичным пользователям
- Холодный старт. Новые объекты никому не рекомендуются

Item-based

$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in I_u} sim(i,j)(r_{uj} - \bar{r}_j)}{\sum_{j \in I_u} sim(i,j)}$, где I_u - топ N айтемов, наиболее похожих на айтем i

Недостатки:

- Холодный старт. Новые объекты никому не рекомендуются
- Рекомендации часто тривиальны

Проблемы memory-based подходов

- Проблема холодного старта
- Плохие предсказания для новых/нетипичных пользователей/объектов
- Тривиальность рекомендаций
- Ресурсоемкость вычислений. Для того чтобы делать предсказания нам нужно держать в памяти все оценки всех пользователей

Модели со скрытыми переменными (Latent Factor Models)

SVD (Singular Value Decomposition)

$$A_{n*m} = U_{n*n}\Sigma_{n*m}(V_{m*m})^T$$

$$UU^T = I_n; VV^T = I_n;$$
$$\Sigma = diag(\lambda_1, \dots, \lambda_{\min(n,m)})$$
$$\lambda_1 \geq .. \geq \lambda_{\min(n,m)} \geq 0$$

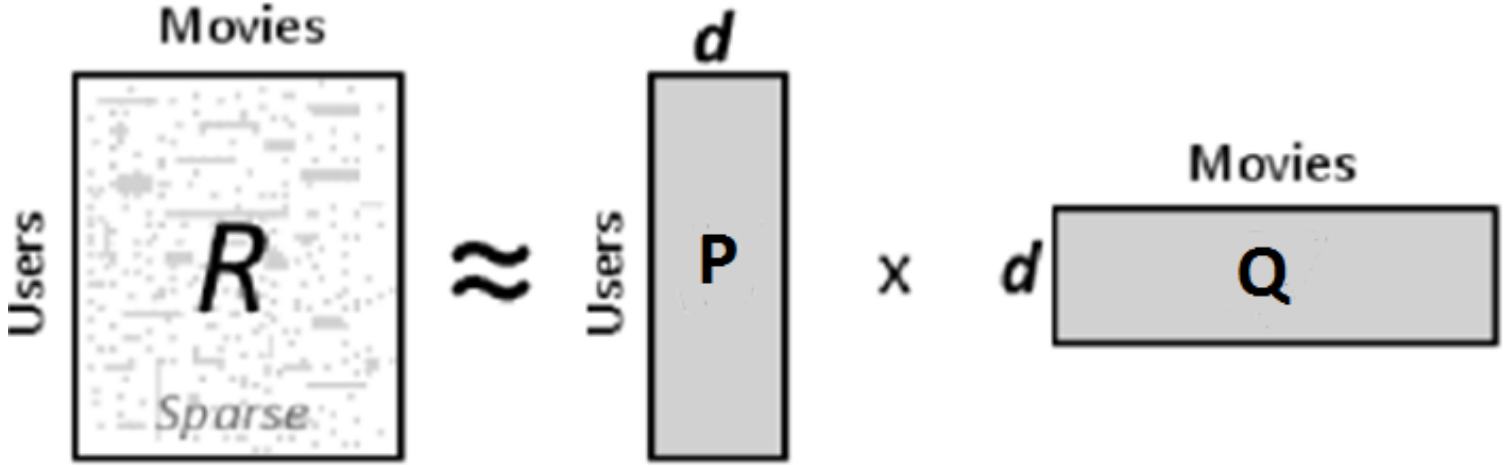
SVD (Singular Value Decomposition)

$$\lambda_{d+1}, .., \lambda_{\min(n,m)} = 0$$

$$A'_{n*m} = U'_{n*d} \Sigma'_{d*d} (V'_{d*m})^T$$

Лучшее низкоранговое приближение по RMSE

SVD для рекомендаций



$$\hat{r}_{ui} = \langle p_u, q_i \rangle$$

Как обучить?

- Запишем функционал ошибки

$$\sum_{(u,i)\in R} \left(r_{ui} - \bar{r}_u - \bar{r}_i - \langle p_u, q_i \rangle\right)^2 \to \min_{P,Q}$$

- Добавим регуляризацию

$$\sum_{(u,i)\in R} \left(r_{ui} - \bar{r}_u - \bar{r}_i - \langle p_u, q_i \rangle\right)^2 + \lambda \sum_{u\in U} \|p_u\|^2 + \mu \sum_{i\in I} \|q_i\|^2 \to \min_{P,Q}$$

Численная оптимизация

Для простоты немного перепишем наш функционал

$$J(\Theta) = \sum_{(u,i)\in\mathcal{D}} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i - r_{ui}\right)^2 + \lambda\left(\sum_u \|\boldsymbol{p}_u\|^2 + \sum_i \|\boldsymbol{q}_i\|^2\right)$$

Можем использовать градиентный спуск, но он работает медленно.

ALS (Alternating Least Squares)

$$\boldsymbol{p}_u^*(\Theta) = \arg\min_{\boldsymbol{p}_u} J(\Theta) = (\boldsymbol{Q}_u^T \boldsymbol{Q}_u + \lambda \boldsymbol{I})^{-1} \boldsymbol{Q}_u^T \boldsymbol{r}_u,$$

$$\boldsymbol{q}_i^*(\Theta) = \arg\min_{\boldsymbol{q}_i} J(\Theta) = (\boldsymbol{P}_i^T \boldsymbol{P}_i + \lambda \boldsymbol{I})^{-1} \boldsymbol{P}_i^T \boldsymbol{r}_i.$$

$$\forall u \in U \quad \boldsymbol{p}_u^{2t+1} = \boldsymbol{p}_u^*(\Theta_{2t}),$$

$$\forall i \in I \quad \boldsymbol{q}_i^{2t+2} = \boldsymbol{q}_i^*(\Theta_{2t+1}).$$

Метрики оценки качества рекомендаций

- Если модель предсказывает рейтинги
  - MSE, RMSE, MAE …
- Предсказание события (клик, просмотр)
  - Precision, Recall, F-мера, ROC-AUC
- Обычно пользователь видит только k товаров
  - hitrate@k, precision@k, recall@k

Специфичные для рекомендаций метрики

- Покрытие товаров
- Покрытие пользователей
- Новизна
- Прозорливость(serendipity)
- Разнообразие

Спасибо за внимание!