РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ

Сергей Николенко

СПбГУ — Санкт-Петербург
1 декабря 2017 г.

Random facts:

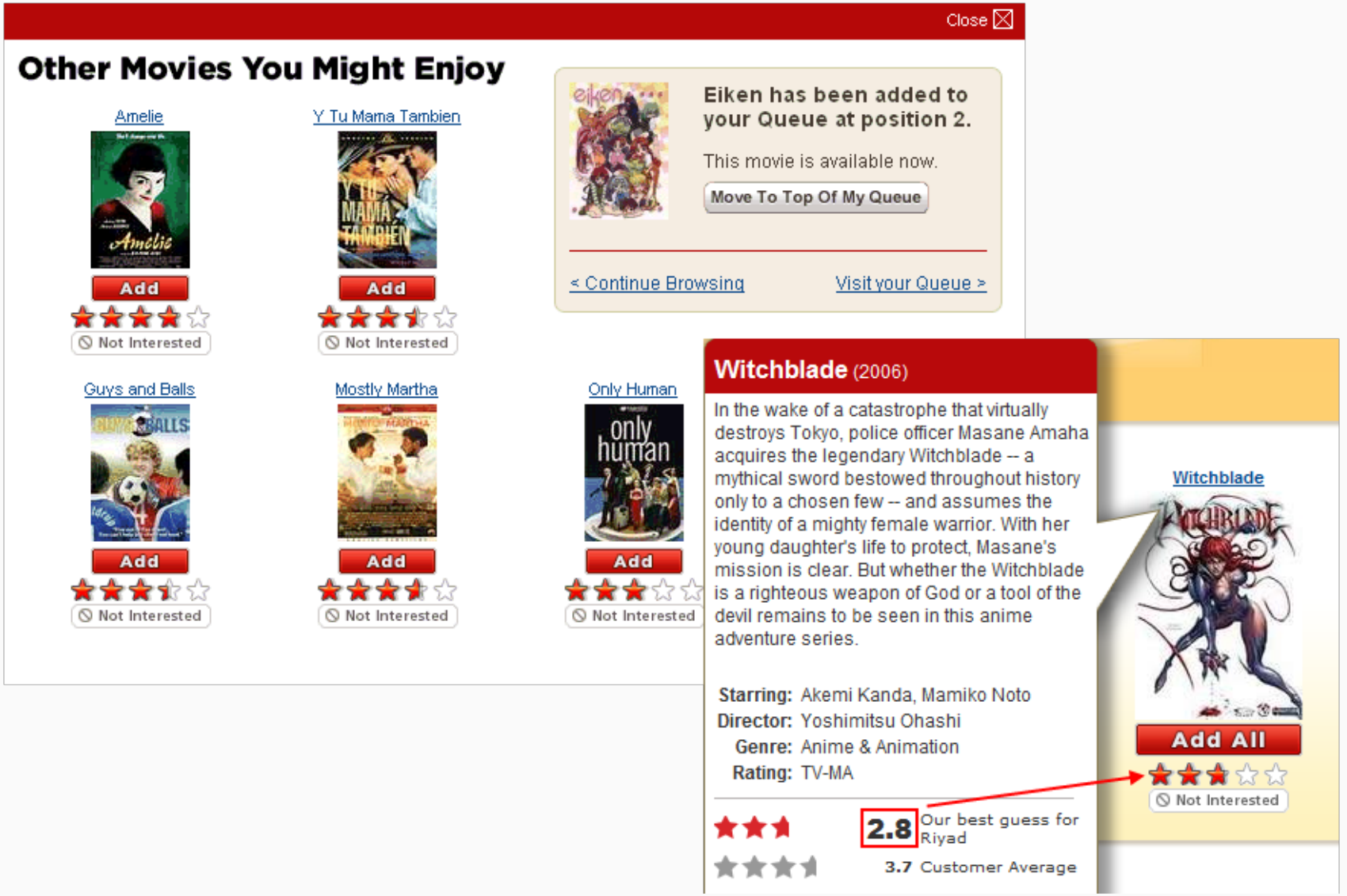
- 1 декабря 1887 г. в свет вышел первый детектив о Шерлоке Холмсе «Этюд в багровых тонах», а 1 декабря 1903 --- первый вестерн «Большое ограбление поезда»
- 1 декабря 1891 г. студентам YMCA из Спрингфилда, штат Массачусетс, было очень скучно на занятиях физкультурой; в результате их преподаватель Джеймс Нейсмит придумал баскетбол
- 1 декабря 1959 г. заключен Договор об Антарктике, запрещающий использование Антарктиды в военных целях

ВВЕДЕНИЕ

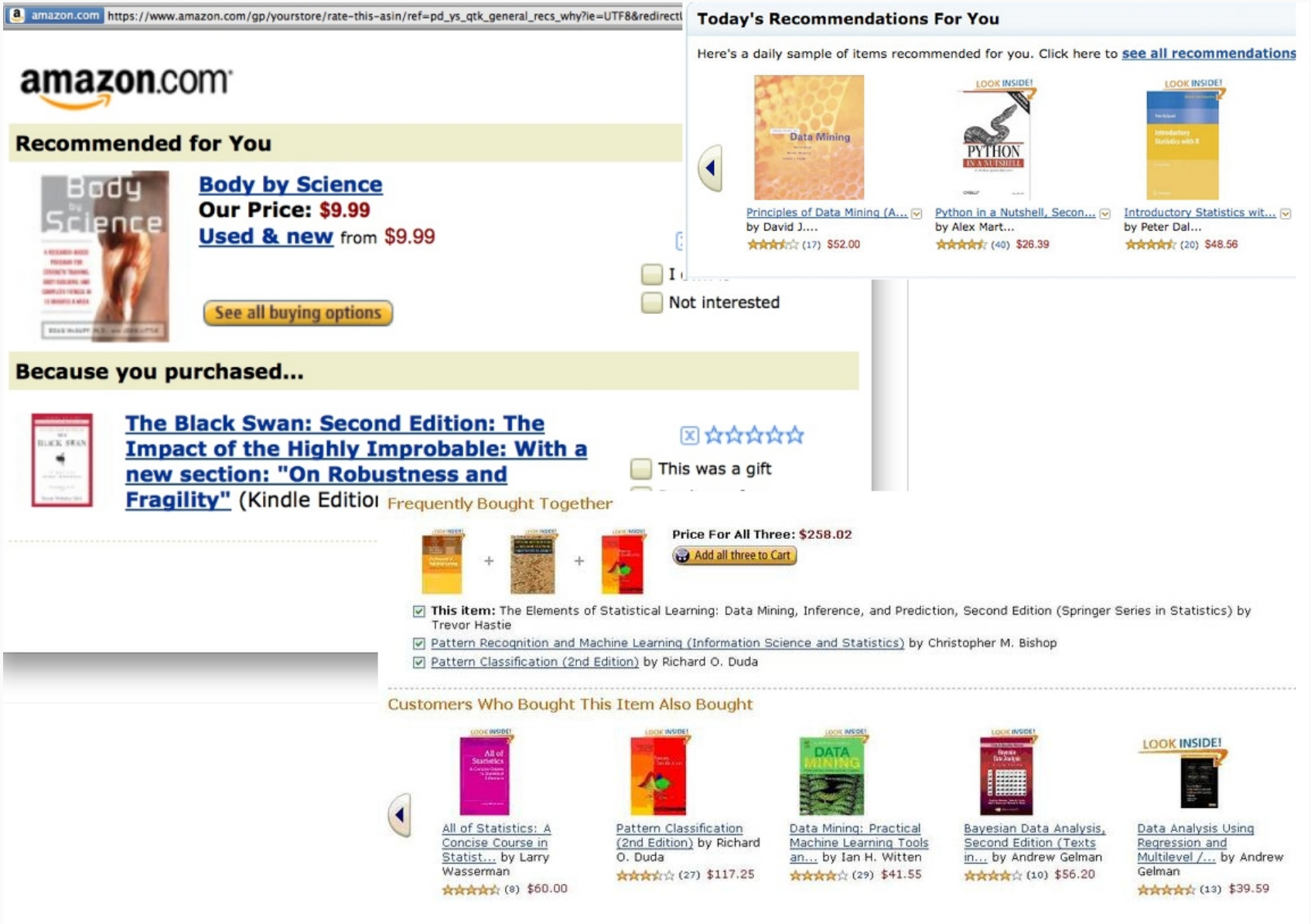
- Основные идеи:
  (1) классика коллаборативной фильтрации: ближайшие соседи и как их масштабировать;
  (2) матричные разложения: зачем они нужны, какие бывают, при чём тут рекомендации;
  (3) расширения: что ещё можно добавить в рекомендательную систему; в частности, рекомендации с анализом контента;
  (4) неперсонализованные рекомендации: что, если нам нужно быстро реагировать на меняющийся контекст, и персональная статистика не успевает?

РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ

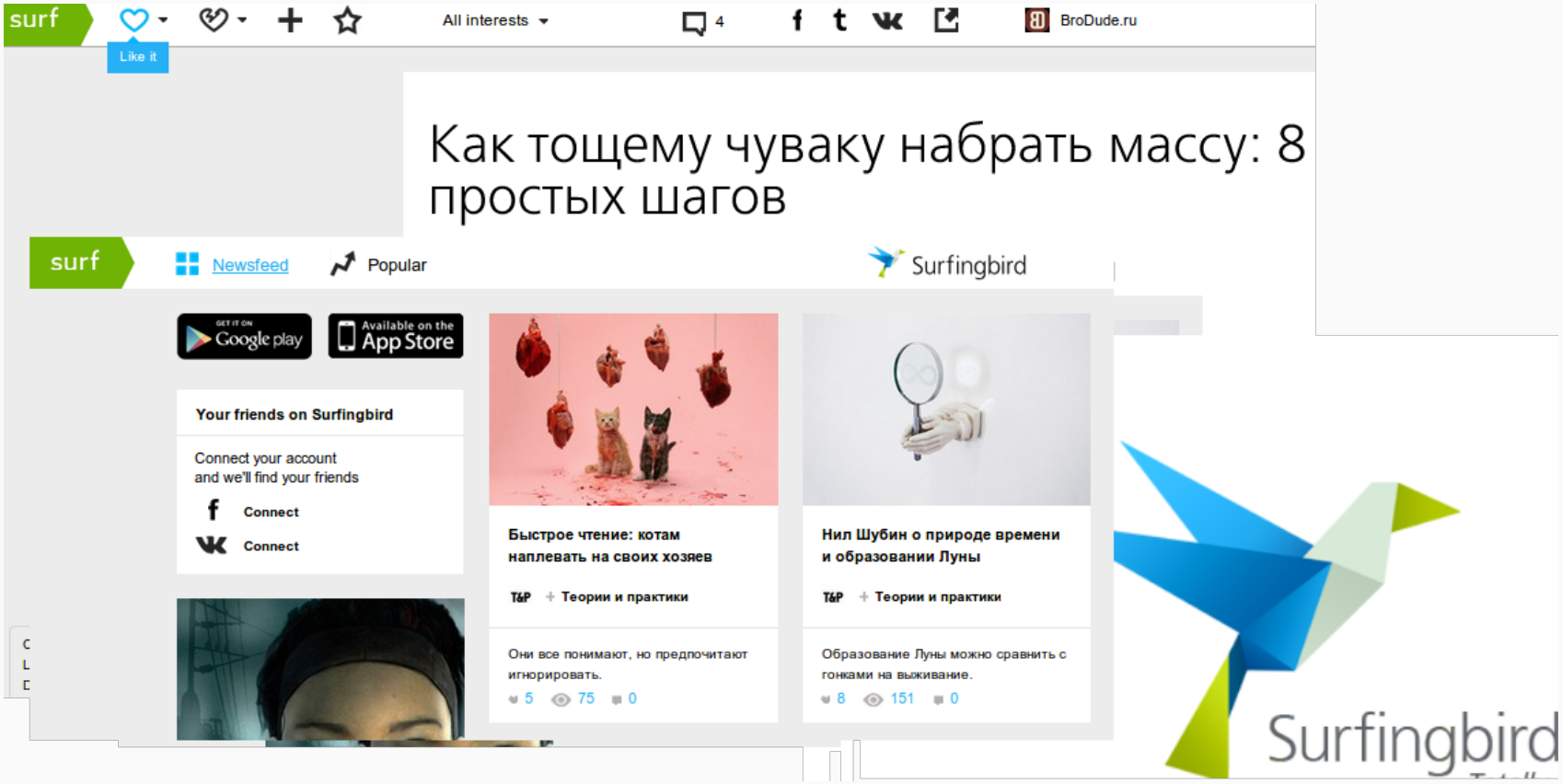
- Рекомендательные системы анализируют интересы пользователей и пытаются предсказать, что именно будет наиболее интересно для конкретного пользователя в данный момент времени.
- Компании–лидеры в рекомендательных системах в основном делятся на две категории:
  (1) мы «продаём» какие-то товары или услуги онлайн; у нас есть пользователи, которые либо явно оценивают товары, либо просто что-то покупают, а что-то нет; интересно порекомендовать товар, который данному покупателю максимально понравится; Netflix, Amazon;
  (2) мы – портал, делаем деньги тем, что размещаем рекламу, надо разместить ссылки, по которым пользователи захотят переходить (и видеть ещё больше вкусной рекламы); Mail.Ru, Yahoo!, Google, Яндекс, контент-провайдеры, новостные сайты.


Close
Other Movies You Might Enjoy
Amelie
Add
Not Interested
Y Tu Mama Tambien
Add
Not Interested
Guys and Balls
Add
Not Interested
Mostly Martha
Add
Not Interested
Only Human
Add
Not Interested
Eiken has been added to your Queue at position 2.
This movie is available now.
Move To Top Of My Queue
< Continue Browsing
Visit your Queue >
Witchblade (2006)
In the wake of a catastrophe that virtually destroys Tokyo, police officer Masane Amaha acquires the legendary Witchblade -- a mythical sword bestowed throughout history only to a chosen few -- and assumes the identity of a mighty female warrior. With her young daughter's life to protect, Masane's mission is clear. But whether the Witchblade is a righteous weapon of God or a tool of the devil remains to be seen in this anime adventure series.
Starring: Akemi Kanda, Mamiko Noto
Director: Yoshimitsu Ohashi
Genre: Anime & Animation
Rating: TV-MA
2.8 Our best guess for Riyad
3.7 Customer Average
Witchblade
Add All
Not Interested

AMAZON


amazon.com https://www.amazon.com/gp/yourstore/rate-this-asin/ref=pd_ys_qtk_general_recs_why?ie=UTF8&redirectl
amazon.com
Recommended for You
Body by Science
Our Price: $9.99
Used & new from $9.99
See all buying options
Not interested
Because you purchased...
The Black Swan: Second Edition: The Impact of the Highly Improbable: With a new section: "On Robustness and Fragility" (Kindle Editio
This was a gift
Today's Recommendations For You
Here's a daily sample of items recommended for you. Click here to see all recommendations
Principles of Data Mining (A... by David J.... (17) $52.00
Python in a Nutshell, Secon... by Alex Mart... (40) $26.39
Introductory Statistics wit... by Peter Dal... (20) $48.56
Frequently Bought Together
Price For All Three: $258.02
Add all three to Cart
This item: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) by Trevor Hastie
Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop
Pattern Classification (2nd Edition) by Richard O. Duda
Customers Who Bought This Item Also Bought
All of Statistics: A Concise Course in Statist... by Larry Wasserman (8) $60.00
Pattern Classification (2nd Edition) by Richard O. Duda (27) $117.25
Data Mining: Practical Machine Learning Tools an... by Ian H. Witten (29) $41.55
Bayesian Data Analysis, Second Edition (Texts in... by Andrew Gelman (10) $56.20
Data Analysis Using Regression and Multilevel /... by Andrew Gelman (13) $39.59

SURFINGBIRD



Онлайн vs. оффлайн

- У рекомендательной системы есть два разных «уровня», на которых она должна работать:
  - глобальные оценки, медленно меняющиеся особенности и предпочтения, интересные страницы, зависимость от user features (география, пол etc.) и т.д.;
  - кратковременные тренды, hotness, быстрые изменения интереса во времени.

Онлайн vs. оффлайн

- Это очень разные задачи с разными методами, поэтому различают два класса моделей.
  - *Оффлайн-модели* выявляют глобальные закономерности (обычно это и называется коллаборативной фильтрацией). Цель зачастую в том, чтобы найти и рекомендовать человеку то, что ему понравится, из достаточно редких вещей, работать с «длинными хвостами» распределений интересов людей и веб-страниц.
  - *Онлайн-модели* должны реагировать очень быстро (поэтому там обычно подходы попроще, как правило, не индивидуализированные), они выявляют кратковременные тренды, позволяют рекомендовать то, что hot прямо сейчас.

Классическая коллаборативная фильтрация

- Начнём с коллаборативной фильтрации. Обозначения:
  - индекс i всегда будет обозначать пользователей (всего пользователей будет N, $i = 1..N$);
  - индекс a – предметы (сайты, товары, фильмы...), которые мы рекомендуем (всего M, $a = 1..M$);
  - когда пользователь i оценивает предмет a, он производит отклик (response, rating) $r_{i,a}$; этот отклик – случайная величина, конечно.
- Наша задача – предсказывать оценки $r_{i,a}$, зная признаки x_i и x_a для всех элементов базы и зная некоторые уже расставленные в базе $r_{i',a'}$. Предсказание будем обозначать через $\hat{r}_{i,a}$.

- Метод ближайших соседей: давайте введём расстояние (похожесть) между пользователями и будем рекомендовать то, что нравится вашим соседям (похожим на вас).
- Расстояние:
  - коэффициент корреляции (коэффициент Пирсона)

$$w_{i,j} = \frac{\sum_a \left(r_{i,a} - \bar{r}_a\right)\left(r_{j,a} - \bar{r}_a\right)}{\sqrt{\sum_a \left(r_{i,a} - \bar{r}_a\right)^2}\sqrt{\sum_a \left(r_{j,a} - \bar{r}_a\right)^2}},$$

    где $\bar{r}_a$ – средний рейтинг продукта a среди всех пользователей;
  - косинус угла между векторами рейтингов:

$$w_{i,j} = \frac{\sum_a r_{i,a} r_{j,a}}{\sqrt{\sum_a r_{i,a}^2}\sqrt{\sum_a r_{j,a}^2}}.$$

GROUPLENS

- Простейший способ построить предсказание нового рейтинга $\hat{r}_{i,a}$ – сумма рейтингов других пользователей, взвешенная их похожестью на пользователя i:

$$\hat{r}_{i,a} = \bar{r}_a + \frac{\sum_j \left(r_{j,a} - \bar{r}_j\right) w_{i,j}}{\sum_j |w_{i,j}|}.$$

- Это называется GroupLens algorithm – так работал дедушка рекомендательных систем GroupLens.
- Чтобы не суммировать по всем пользователям, можно ограничиться ближайшими соседями:

$$\hat{r}_{i,a} = \bar{r}_a + \frac{\sum_{j \in \mathrm{knn}(i)} \left(r_{j,a} - \bar{r}_j\right) w_{i,j}}{\sum_{j \in \mathrm{knn}(i)} |w_{i,j}|}.$$

- Естественно предположить, что продукты, которые любят или не любят практически все пользователи, не слишком полезны в определении ближайшего соседа.
- Поэтому естественно взвесить продукты по тому, как часто их уже оценивали пользователи; такая метрика называется iuf – inverse user frequency, обратная частота пользователей: $f_a = \log \frac{N}{N_a}$, где N – общее число пользователей, N_a – число оценивших продукт a. Получается

$$w_{i,j}^{\text{iuf}} = \frac{\sum_a f_a \sum_a f_a r_{i,a} r_{j,a} - \left(\sum_a f_a r_{i,a}\right)\left(\sum_a f_a r_{j,a}\right)}{\sqrt{\sum_a f_a \left(\sum_a f_a r_{i,a}^2 - \left(\sum_a f_a r_{i,a}\right)^2\right)} \sqrt{\sum_a f_a \left(\sum_a f_a r_{j,a}^2 - \left(\sum_a f_a r_{j,a}\right)^2\right)}}$$

а для косинуса

$$w_{i,j}^{\text{iuf}} = \frac{\sum_a f_a^2 r_{i,a} r_{j,a}}{\sqrt{\sum_a (f_a r_{i,a})^2} \sqrt{\sum_a (f_a r_{j,a})^2}}.$$

ITEM-ITEM CF

- Симметричный подход – item-based collaborative filtering. Считаем похожесть между продуктами, выбираем похожие продукты.
- Amazon: customers who bought this item also bought...
- Преимущество – может быть эффективнее за счёт того, что похожесть продуктов всегда можно считать оффлайн, пара новых оценок не повлияет на неё совсем радикально.
- Считаем похожесть между парами продуктов, у которых есть общий оценивший пользователь.

Как масштабировать ближайших соседей

- Искать ближайших соседей довольно сложно алгоритмически (k-d-деревья в больших размерностях плохо работают).
- В действительно больших рекомендательных системах используют приближённые методы.
- Например, LSH (locality sensitive hashing) на min-hash:
  - заведём несколько хеш-функций, посчитаем их от каждого продукта;
  - для каждого пользователя вычислим минимальное значение хеш-функций на его продуктах;
  - будем соседей искать только среди тех пользователей, у которых есть одинаковые значения хотя бы в одном хеше.

Что делать с лайками?

- Заметим, что мы до сих пор рассматривали только системы, в которых есть явно поставленные рейтинги, некоторые хорошие, некоторые плохие.
- Но в реальной ситуации часто есть только множества «потреблённых» продуктов I и J для пользователей i и j:
  - лайки (дизлайки мало кто ставит);
  - купленные товары без явных рейтингов.
- Это называется implicit feedback. Что делать?

ЧТО ДЕЛАТЬ С ЛАЙКАМИ?

- Надо определить расстояние между множествами; похожесть по Жаккару

$$w_{i,j} = \mathrm{Jaccard}(I, J) = \frac{|I \cap J|}{|I \cup J|}.$$

- Можно так ввести веса между пользователями и дальше рекомендовать по GroupLens.

Что делать с лайками?

- Но ещё чаще похожесть по Жаккару используется для поиска других продуктов, «похожих на».
- Определим похожесть между a и b через множества потребивших их пользователей A и B:

$$w_{a,b} = \text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

- Часто хорошо работает, но есть тонкие эффекты.
- Например, что если продукт a редкий, а b более популярный, и так получилось, что почти все $i \in A$ также потребили и b? (это очень реальная ситуация)

Что делать с лайками?

- Похожесть по Жаккару плохо подходит в несбалансированных случаях, потому что она симметричная.
- Давайте сделаем несимметрично:

$$w_{a,b} = \frac{|A \cap B|}{|A|}, \quad w_{b,a} = \frac{|A \cap B|}{|B|}.$$

- Теперь с предыдущим примером всё хорошо, но и тут есть тонкие эффекты.
- Что если один из продуктов очень популярен, и почти все его видели? Banana trap.

Что делать с лайками?

- Ещё одна вариация – метод ассоциаций:

$$w_{a,b} = \frac{|A \cap B| \,/\, |A|}{\left|\bar{A} \cap B\right| \,/\, \left|\bar{A}\right|},$$

  где $\bar{A}$ – дополнение A.

- На практике обычно стоит просто попробовать все эти варианты и выбрать лучший.

СПАСИБО!

Спасибо за внимание!