Рекомендательные системы

Сергей Николенко

Лаборатория Интернет-исследований,
Национальный исследовательский университет
Высшая школа экономики – Санкт-Петербург

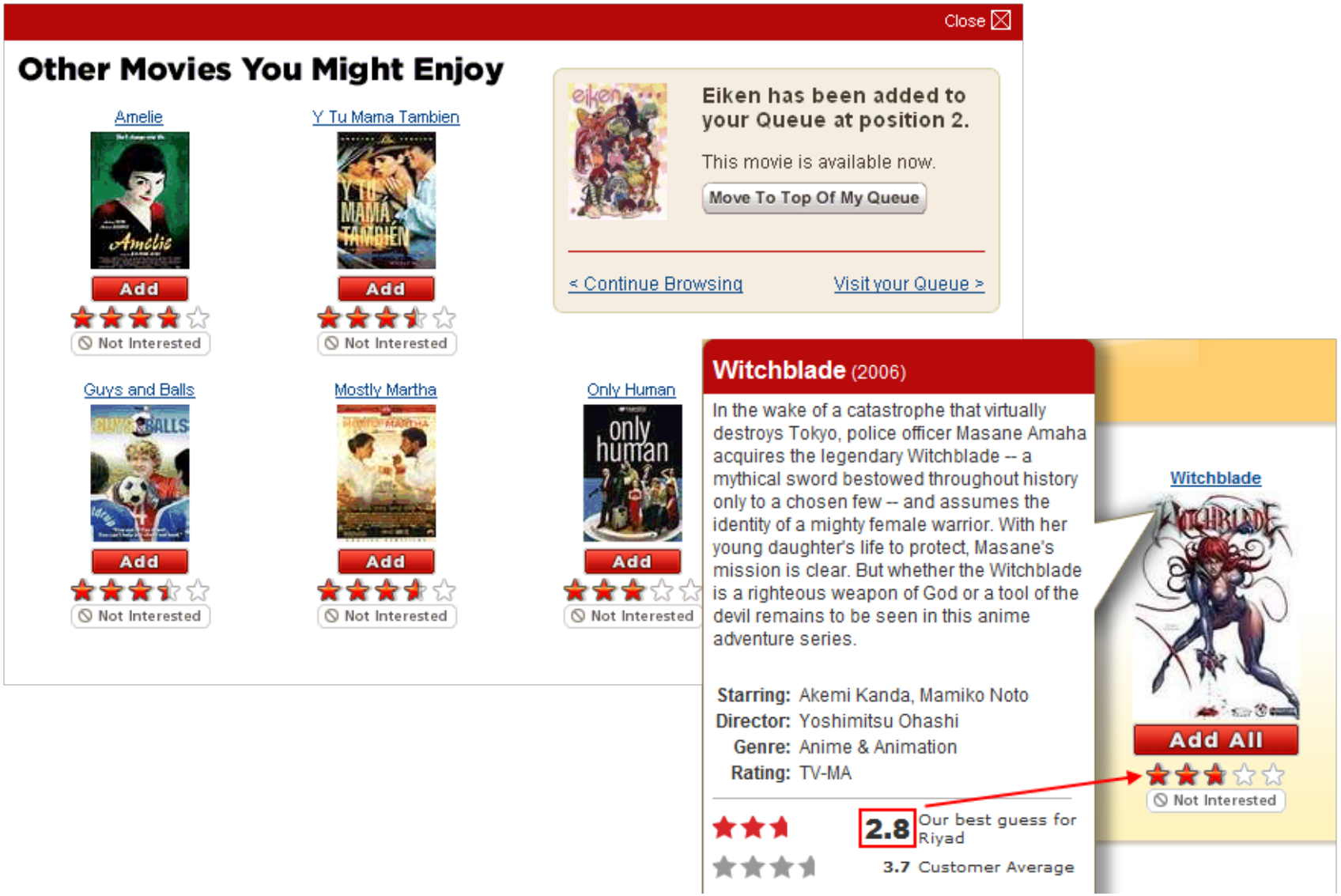
10 декабря 2013 г.

Outline

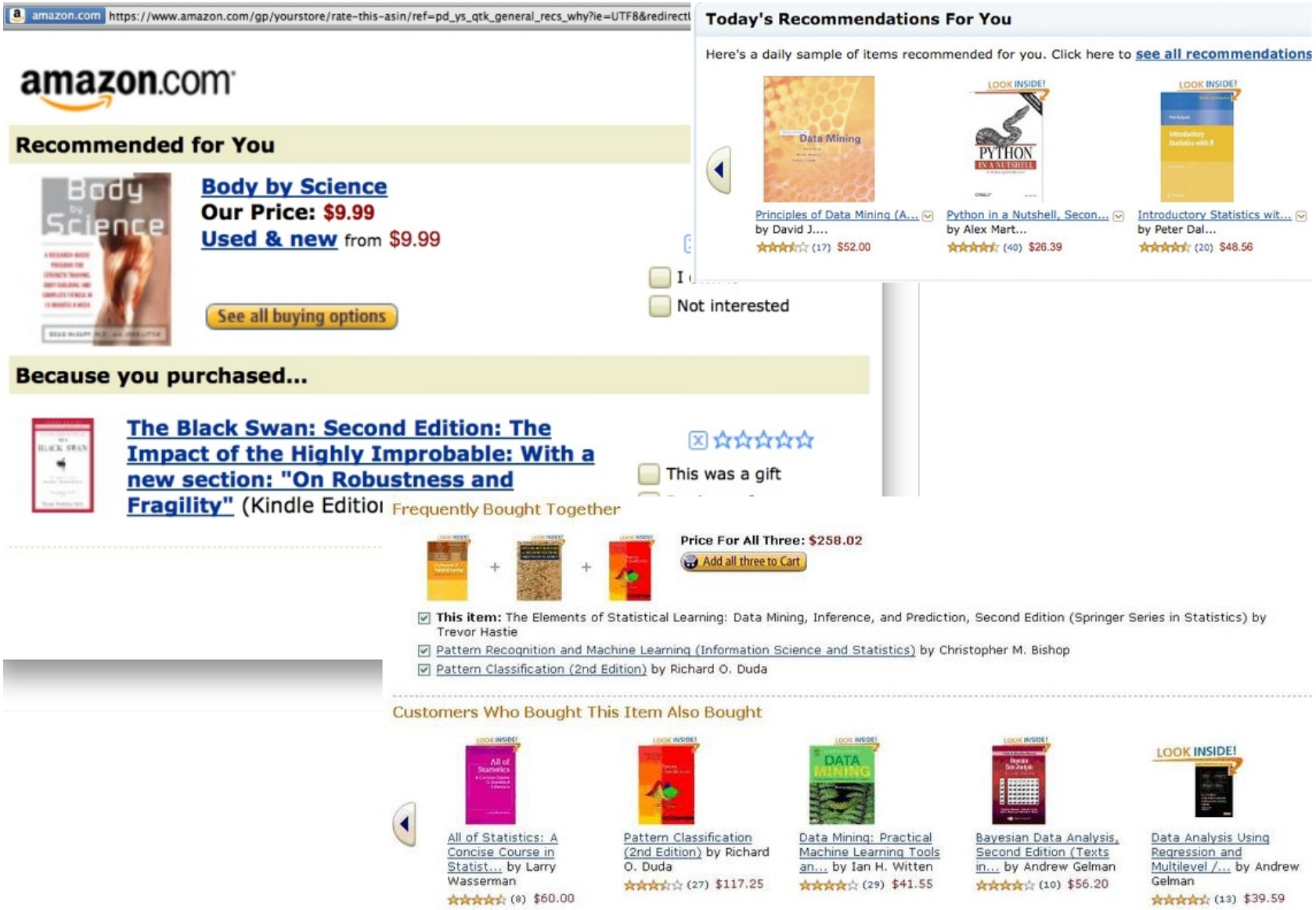
Рекомендательные системы

- 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, Surfingbird;
  2. мы – портал, делаем деньги тем, что размещаем рекламу, надо разместить ссылки, по которым пользователи захотят переходить (и видеть ещё больше вкусной рекламы); Yahoo!, Google, Яндекс, большинство новостных сайтов.

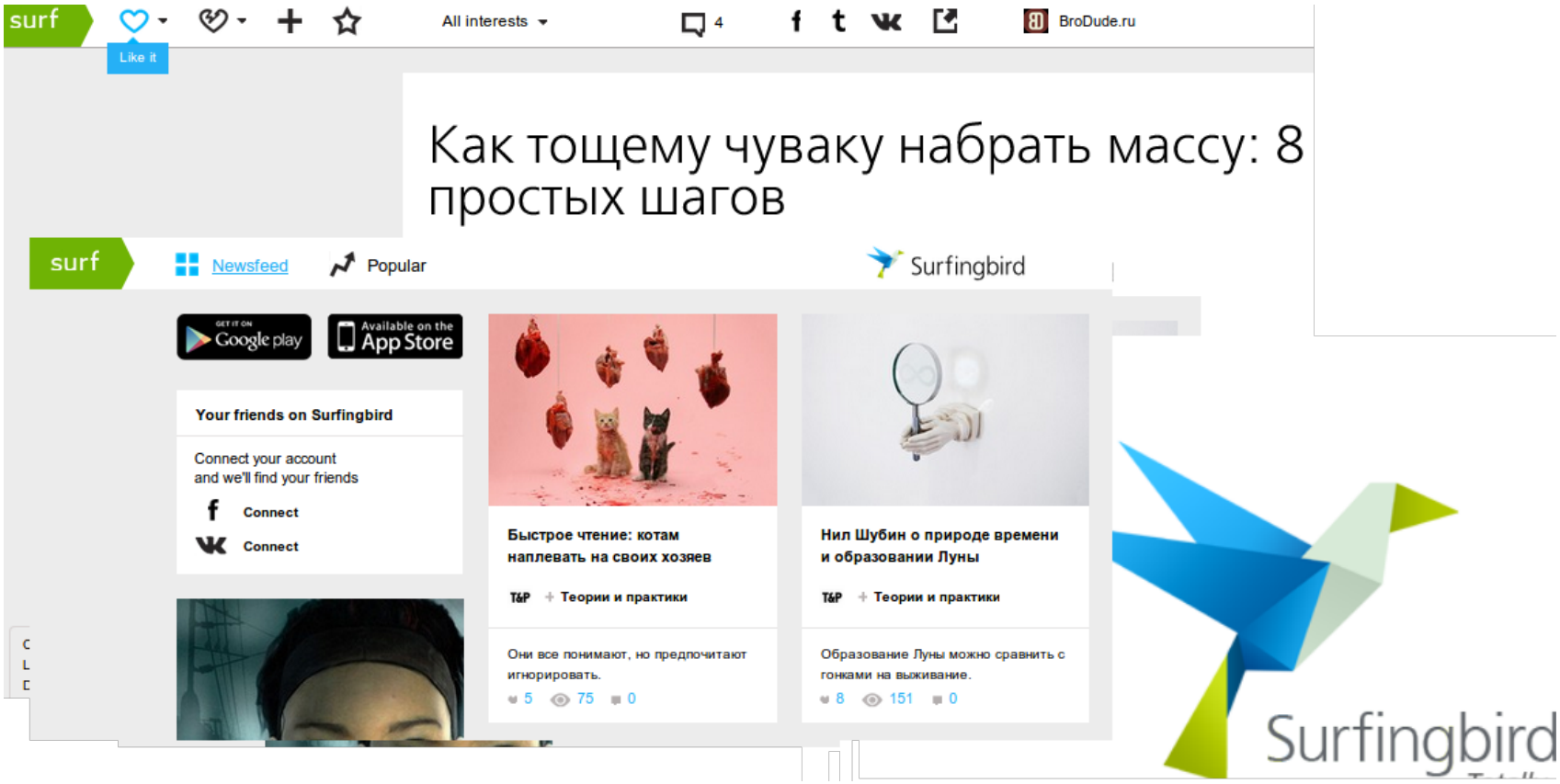
Netflix


Close
Other Movies You Might Enjoy
Amelie
Add
Not Interested
Y Tu Mama Tambien
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 >
Guys and Balls
Add
Not Interested
Mostly Martha
Add
Not Interested
Only Human
Add
Not Interested
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&redirect
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



GroupLens

- Начнём краткий обзор разных рекомендательных систем 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, $a = 1..M$);
  - x_i – набор (вектор) признаков (features) пользователя, x_a – набор признаков предмета;
  - 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 $\hat{r}_{i,a}$.

GroupLens

- Начнём с небайесовских методов. Метод ближайших соседей: давайте введём расстояние между пользователями и будем рекомендовать то, что нравится вашим соседям.
- Расстояние:
  - коэффициент корреляции (коэффициент Пирсона)

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

    где $\bar{r}_a$ – средний рейтинг продукта a среди всех пользователей;
  - косинус угла между векторами рейтингов, выставленных i и j, т.е.

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

GroupLens

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

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

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

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

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.

Вероятностные модели

- Из чего складывается рейтинг пользователя i, который он выдал продукту a?
- Вполне может быть, что пользователь добрый и всем подряд выдаёт хорошие рейтинги; или, наоборот, злой и рейтинг зажимает.
- С другой стороны, некоторые продукты попросту лучше других.
- Поэтому мы вводим так называемые *базовые предикторы* (baseline predictors) $b_{i,a}$, которые складываются из базовых предикторов отдельных пользователей b_i и базовых предикторов отдельных продуктов b_a, а также просто общего среднего рейтинга по базе μ:

$$b_{i,a} = \mu + b_i + b_a.$$

Вероятностные модели

- Чтобы найти предикторы, уже нужен байесовский подход: надо добавить нормально распределённый шум и получить модель линейной регрессии

$$r_{i,a} \sim \mathcal{N}\left(\mu + b_i + b_a, \sigma^2\right).$$

- Можно ввести априорные распределения и оптимизировать; или просто найти среднеквадратическое отклонение с регуляризатором:

$$b_* = \arg\min_b \sum_{(i,a)} (r_{i,a} - \mu - b_i - b_a)^2 + \lambda_1 \left(\sum_i b_i^2 + \sum_a b_a^2 \right).$$

Вероятностные модели

- С тем, чтобы напрямую обучать оставшуюся матрицу предпочтений вероятностными методами, есть одна очень серьёзная проблема – матрица X, выражающая рейтинги, содержит $N \times M$ параметров, гигантское число, которое, конечно, никак толком не обучить.
- Более того, обучать их и не надо – как мы уже говорили, данные очень разреженные, и «на самом деле» свободных параметров гораздо меньше, проблема только с тем, как их выделить.
- Поэтому обычно число независимых параметров модели необходимо уменьшать.

Вероятностные модели

- Метод SVD (singular value decomposition) – разложим матрицу X в произведение матриц маленького ранга.
- Зафиксируем некоторое число f *скрытых факторов*, которые так или иначе описывают каждый продукт и предпочтения каждого пользователя относительно этих факторов.
- Пользователь – вектор $p_i \in \mathbb{R}^f$, который показывает, насколько пользователь предпочитает те или иные факторы; продукт – вектор $q_a \in \mathbb{R}^f$, который показывает, насколько выражены те или иные факторы в этом продукте.

Вероятностные модели

- Предпочтение в итоге будем подсчитывать просто как скалярное произведение $q_a^\top p_i = \sum_{j=1}^{f} q_{a,j} p_{i,j}$.
- Таким образом, добавляя теперь сюда baseline-предикторы, получаем следующую модель предсказаний рейтингов:

$$\hat{r}_{i,a} \sim \mu + b_i + b_a + q_a^\top p_i.$$

Вероятностные модели

- Можно добавлять и дополнительную информацию в эту модель. Например, введём дополнительный набор факторов для продуктов y_a, которые будут характеризовать пользователя на основе того, что он просматривал, но не оценивал.
- Модель после этого принимает вид

$$\hat{r}_{i,a} = \mu + b_i + b_a + q_a^\top \left(p_i + \frac{1}{\sqrt{|V(i)|}} \sum_{b \in V(i)} y_b \right),$$

где $V(i)$ – множество продуктов, которые просматривал этот пользователь ($\frac{1}{\sqrt{|V(i)|}}$ контролирует дисперсию).

- Это называется SVD++.

Outline

Время в коллаборативной фильтрации

- Пример: давайте добавим время, т.е. будем рассматривать базовые предикторы и характеристики пользователя как функции от времени:

$$\hat{r}_{i,a} = \mu + b_i(t) + b_a(t) + q_a^\top p_i(t),$$

где

$$\begin{aligned} b_a(t) =& b_a + b_{a,\mathrm{Bin}(t)}, \\ b_i(t) =& b_i + \alpha_i \mathrm{dev}_i(t) + b_{i,t}, \\ p_{i,f}(t) =& p_{i,f} + \alpha_{i,f}\mathrm{dev}_i(t) + p_{i,f,t} + \frac{1}{\sqrt{|V(i)|}} \sum_{b \in V(i)} y_b, \\ \mathrm{dev}_i(t) =& \mathrm{sign}(t - t_i)\,|t - t_i|^\beta . \end{aligned}$$

- Это называется timeSVD++, и эта модель была одним из основных компонентов модели, взявшей Netflix Prize.

Социальные сети

- Предположим, что пользователи приходят из социальной сети.
- Т.е. есть друзья, есть социальный граф (его часть) и т.д. Это тоже можно добавить в рекомендательную модель:
  - фильтр/перевзвешивание в методе ближайших соседей;
  - дополнительные слагаемые в разложение типа SVD;
  - разложение матрицы доверия (из социального графа) вместе с матрицей рейтингов, меняем априорное распределение для PMF и т.д.

Метрики разнообразия

- Filter bubble: как вывести человека за его привычный круг.
- Можно до конца жизни рекомендовать одно и то же; метрики:
  - diversity – разнообразие, мера похожести элементов списка;
  - novelty – новизна для пользователя, распространённость продукта, доля его рейтингов;
  - serendipity – неожиданность, сюрприз, похожесть на историю пользователя;
  - temporal diversity (разнообразие во времени) – чтобы пользователю не было скучно.
- Для всего этого нужно уметь распознавать похожесть контента рекомендованных товаров.

Контекстно-зависимые рекомендации

- CARS (context-aware recommender systems) – мы рекомендуем в контексте:
  - временном;
  - ситуативном;
  - географическом;
  - предшествующего поведения пользователей и т.д.

Контекстно-зависимые рекомендации

- Формально контекст – это новые измерения в матрице предпочтений.
- Получается “гиперкуб” данных, есть методы тензорного разложения, аналогичного SVD.
- Но часто не хуже работают простые решения – отфильтровать контексты и обучить модели только по этим данным, добавить полученные модели и сам контекст как факторы в бленд.

Контент

- У объектов, которые мы рекомендуем, может быть своё собственное содержание, которое можно анализировать.
- Например, веб-страницы в Surfingbird содержат текст.
- Можно совмещать рекомендательные модели с моделями анализа текстов (text mining), например LDA (latent Dirichlet allocation).

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.

Онлайн-модели

- Данных тут недостаточно, чтобы такие изменения можно было поймать методами коллаборативной фильтрации.
- Поэтому онлайн-методы обычно меньше персонализированы, индивидуальных данных не набёрется просто.
- Если мы хотим быстро оценивать средний рейтинг, то мы попадём в ситуацию обучения с подкреплением: есть набор продуктов, их надо рекомендовать, исход заранее неизвестен, и мы хотим оптимизировать суммарный рейтинг.
- Это называется задачей о *многоруких бандитах* (multiarmed bandits), и нам нужен её вариант, где выплаты меняются со временем.

Thank you!

Спасибо за внимание!