Рекомендательные системы

Сергей Николенко

Лаборатория Интернет-исследований,
НИУ Высшая школа экономики, Санкт-Петербург

Лаборатория математической логики,
ПОМИ РАН, Санкт-Петербург

Deloitte Analytics Institute, Москва

20 февраля 2016 г.

Preview

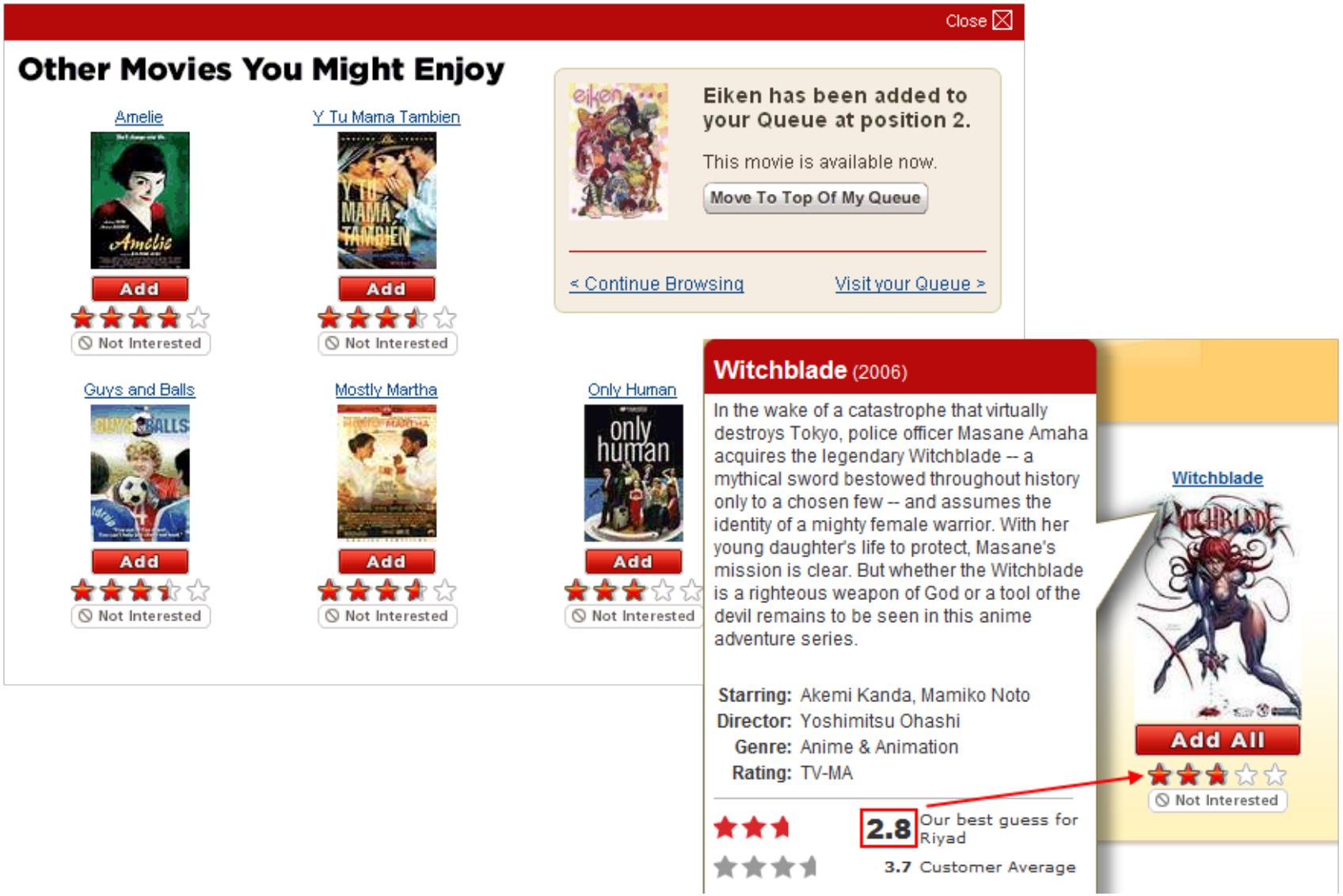
- Три основные идеи:
  1. классика коллаборативной фильтрации: ближайшие соседи и как их масштабировать;
  2. матричные разложения: зачем они нужны, какие бывают, при чём тут рекомендации;
  3. расширения: что ещё можно добавить в рекомендательную систему; в частности, рекомендации с анализом контента.

Outline

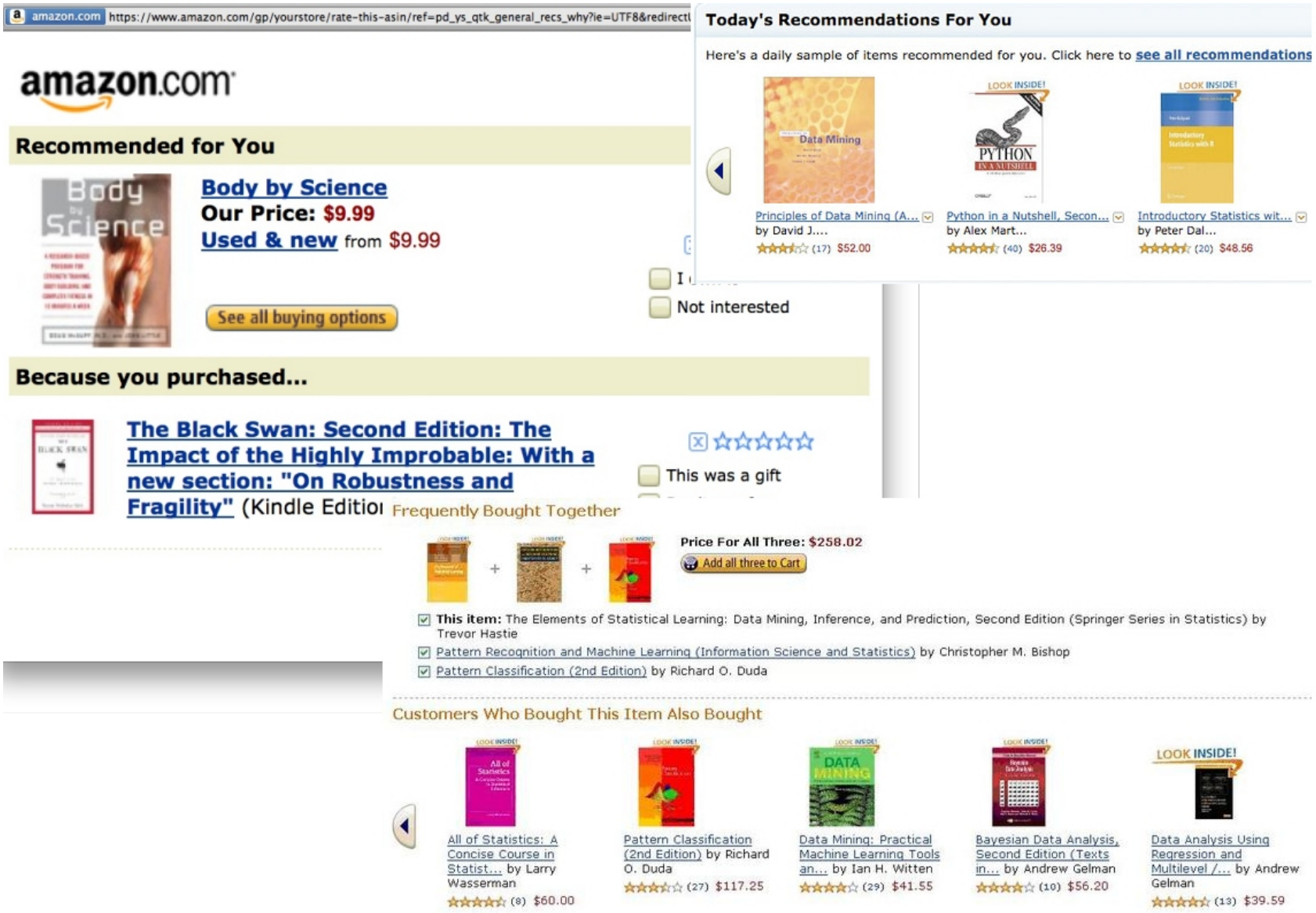
Рекомендательные системы

- Рекомендательные системы анализируют интересы пользователей и пытаются предсказать, что именно будет наиболее интересно для конкретного пользователя в данный момент времени. Варианты:
  1. мы «продаём» что-то онлайн; 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, любой интернет-магазин;
  2. мы предоставляем контент и хотим, чтобы пользователь подольше был на нашем сайте (зарабатываем рекламой); StumbleUpon, Surfingbird, любой контент-провайдер;
  3. мы делаем деньги тем, что размещаем рекламу; нужно показать правильную рекламу правильным людям; Google, Яндекс, Yahoo (немного другая тема).

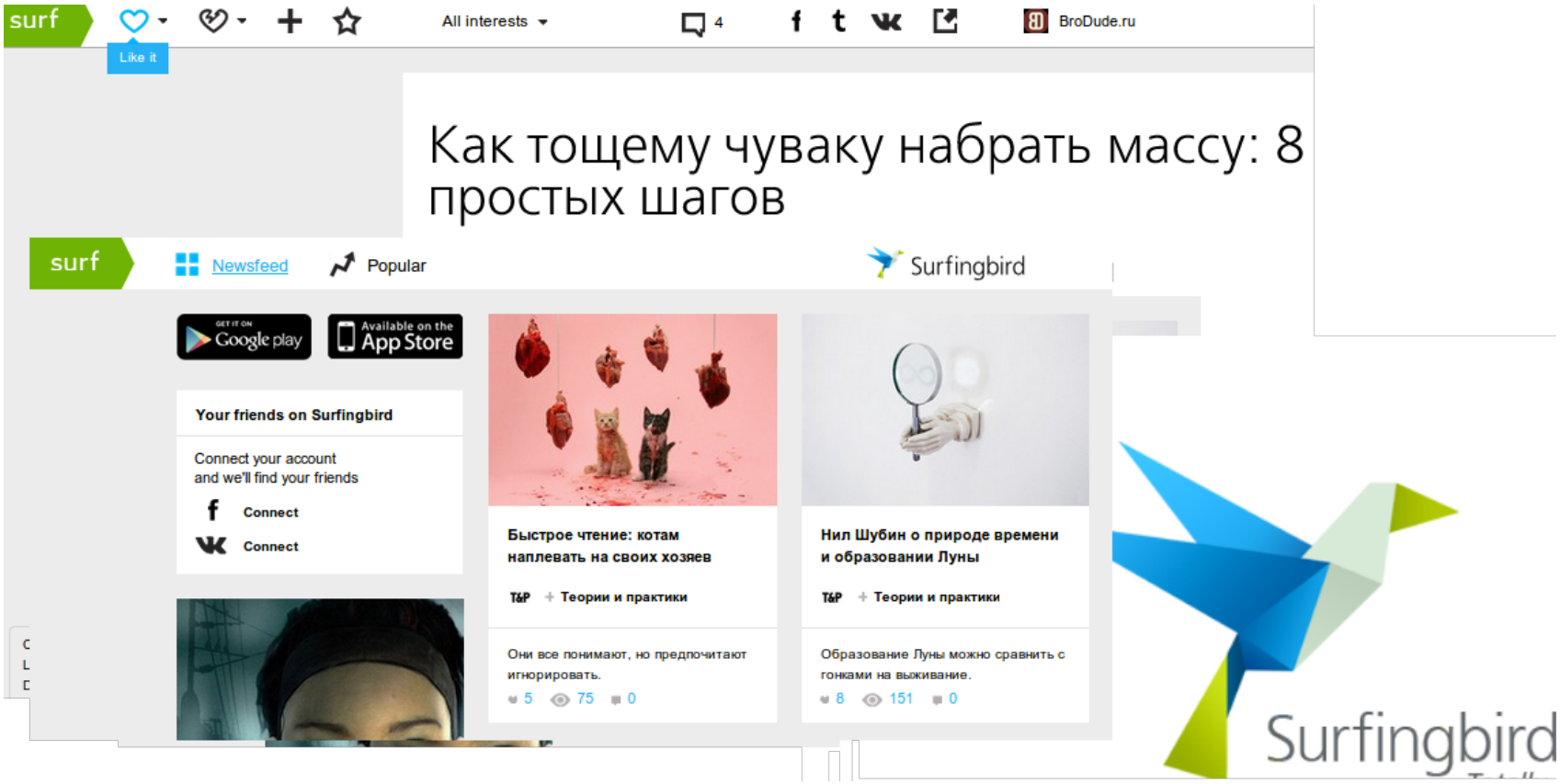
Netflix


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&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



GroupLens

- Метод ближайших соседей: давайте введём расстояние между пользователями и будем рекомендовать то, что нравится вашим соседям.
- 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.

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.

Когда ближайшие соседи перестают работать

- 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).
- Другой основной подход к рекомендациям – это поиск *скрытых признаков* (latent features), которые описывали бы в сжатом виде предпочтения пользователей.
- Это фактически основная идея современных рекомендательных систем, и делается это разными вероятностными моделями; давайте рассмотрим простейшую из них подробно.

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) $b_{i,a}$ для отдельных пользователей b_i и отдельных продуктов b_a:

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

- Чтобы их найти, можно просто решить задачу оптимизации (регуляризация!).

Вероятностные модели

- Но тут пока не было совсем никакой персонализации!
- 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

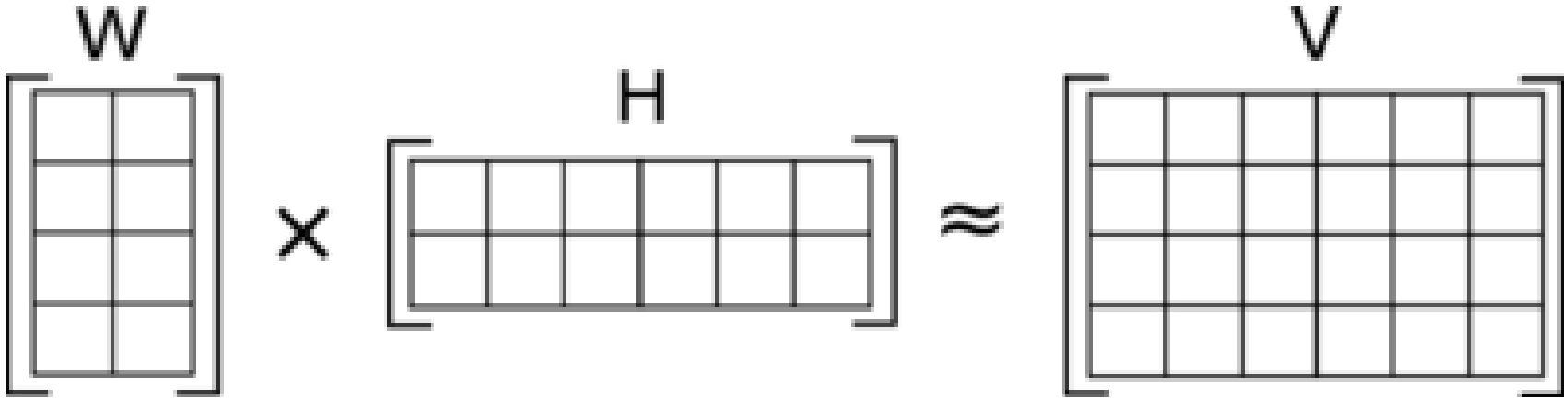
- Предпочтение в итоге будем подсчитывать просто как скалярное произведение $q_a^\top p_i = \sum_{j=1}^{f} q_{a,j} p_{i,j}$.
- Таким образом, добавляя теперь сюда baseline-предикторы, получаем следующую модель предсказаний рейтингов:

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

- Её теперь можно обучать, максимизируя правдоподобие (минимизируя функцию ошибки):
  - SGD – стохастический градиентный спуск;
  - ALS – попеременные наименьшие квадраты.

Матричные разложения

- Мы только что описали метод SVD (singular value decomposition) – разложение матрицы X в произведение матриц маленького ранга.



Матричные разложения

- PCA (анализ главных компонент) – один из самых популярных методов сокращения размерности.
- PCA пытается объяснить как можно больше дисперсии исходного датасета.
- Однако часто направления на кластеры в данных не ортогональны, а признаки PCA трудно интерпретировать.

Матричные разложения

- SVD (сингулярное разложение) мы уже рассматривали.
- Оно делает ровно то, что нам нужно, когда доступны рейтинги:
  - максимизирует правдоподобие имеющихся рейтингов (минимизирует ошибку предсказания рейтингов);
  - умеет работать с разреженными матрицами (минимизирует только по имеющимся рейтингам).
- Но что делать, когда рейтингов никаких нет, а есть только факт использования? SVD тогда ничего хорошего не сделает...

Матричные разложения

- NMF (неотрицательное разложение): раскладываем по-прежнему в произведение

$$X \approx UV^\top,$$

где U размера $n \times f$, V размера $m \times f$, и f гораздо меньше n и m.

- Но теперь требуем, чтобы элементы U и V были неотрицательны.
- Результат – признаки, которые лучше интерпретируются и могут иметь больше смысла.

NMF для рекомендаций

- NMF можно использовать для рекомендаций, когда рейтингов нет.
- Берём матрицу X, в которой единицы – продукты, использованные пользователем, остальное нули.
- И раскладываем по NMF (методом ALS, нужно итеративно решать системы размерности $n \times f$ и $m \times f$).
- Однако задача перестаёт быть разреженной.

Outline

Холодный старт

- Матричные разложения – основа современных рекомендательных систем.
- Но для матричных разложений необходимо, чтобы в строках и столбцах матрицы было достаточно много известных значений.
- Возникает проблема *холодного старта* (cold start) – одна из главных проблем любой рекомендательной системы.
- Но есть и другие проблемы и задачи.

Метрики разнообразия

- Filter bubble: как вывести человека за его привычный круг, а то ведь можно до конца жизни рекомендовать одно и то же. Можно оптимизировать другие 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 многого из 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.
- Но в любом случае данных с контекстом гораздо меньше, холодный старт ещё актуальнее.

Расширения SVD

- Можно добавлять дополнительную информацию в модель SVD. Например, введём вектор факторов для пользователей d_c, которые будут характеризовать не самого пользователя, а целую группу пользователей c (большой кластер, например демографический).
- Модель после этого принимает вид

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

где пользователь i попадает в кластер c.

- Это позволяет выделить общие интересы групп пользователей, что затем можно использовать для холодного старта.

Расширения SVD

- Другое важное расширение – давайте введём ещё вектор 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++.

Время в коллаборативной фильтрации

- А ещё можно, например, добавить время – будем рассматривать базовые предикторы и характеристики пользователя как функции от времени:

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

- Один из таких вариантов, timeSVD++, был 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.
- Это может работать для холодного старта.

Другие постановки задачи

- TopN-рекомендации: часто нужно выдать список.
- Shared accounts: понять, кто перед нами, или просто выдать что-то для всех.
- Рекомендации с дополнительными факторами: что если нам платят за promotions?
- Онлайн-рекомендации: как быстро понять, нужно ли рекомендовать новый продукт (например, новость); совсем другие методы, reinforcement learning.
- Как объяснить рекомендации? Было бы хорошо говорить, почему мы рекомендуем это, но это часто очень сложно.

Рекомендации с контентом

- Другое важное направление – рекомендации с анализом контента.
- Surfingbird – анализ текста веб-страниц на основе LDA и других моделей.
- Расширения LDA с переменными отклика: supervised LDA и другие.

Профилирование

- Важное направление, смежное с рекомендациями: профилирование пользователей.
- Демографические профили: пол, возраст, доход и т.п.
- Но самое интересное – как построить профиль интересов.
- Потом профили используются, естественно, для рекомендаций/рекламы.

Профилирование и NLP

- Если есть готовые теги у продуктов или текстов, это проще, можно просто тематику из них определить.
- Но и просто из текстов можно строить профили интересов; два подхода:
  - тематическое моделирование (topic modeling);
  - семантическое представление текстов (word2vec, document2vec).

Спасибо!

Спасибо за внимание!

GroupLens

- Начнём с постановки задачи. 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

- Метод ближайших соседей: давайте введём расстояние между пользователями и будем рекомендовать то, что нравится вашим соседям.
- 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.

Когда ближайшие соседи перестают работать

- 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.

Вероятностное разложение матриц

- Пусть мы хотим построить разложение матрицы рейтингов на матрицы меньшего ранга:

$$\hat{R} = U^\top V.$$

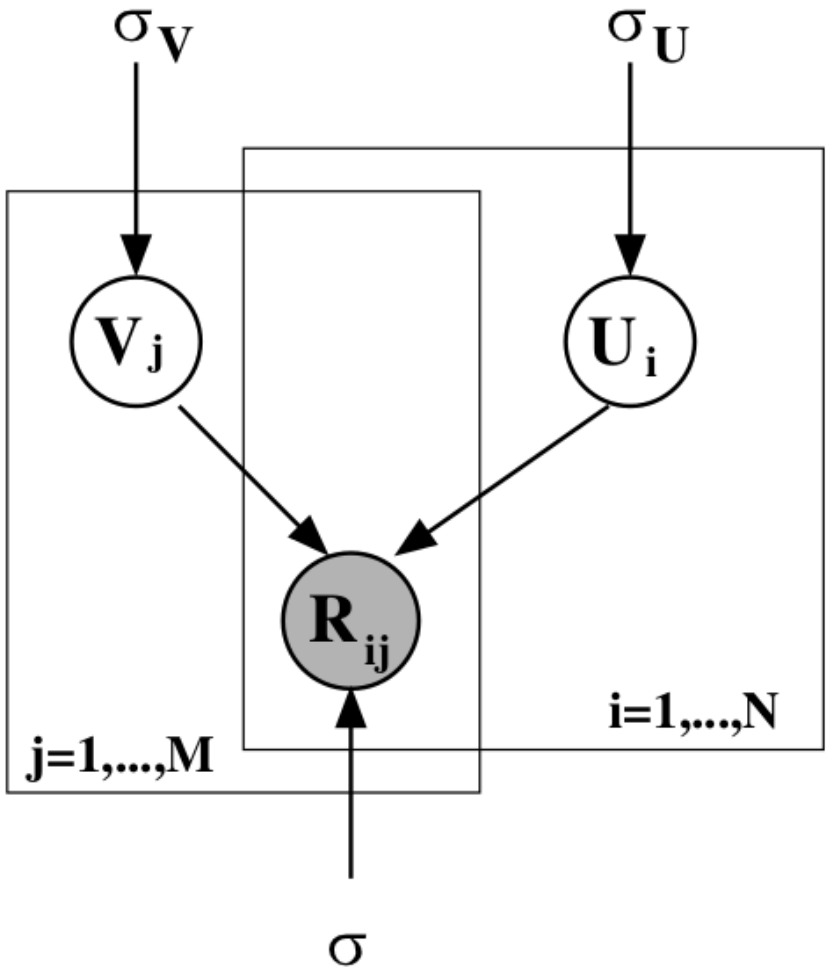
- Вероятностно мы имеем правдоподобие

$$p(R \mid U, V, \sigma^2) = \prod_i \prod_a \left(\mathcal{N}(r_{i,a} \mid u_i^\top v_j, \sigma^2)\right)^{[i \text{ оценил } a]}.$$

- Добавим гауссовские априорные распределения на U и V:

$$p(U \mid \sigma_U^2) = \prod_i \mathcal{N}(U_i \mid \mathbf{0}, \sigma_U^2 \boldsymbol{I}), \quad p(V \mid \sigma_V^2) = \prod_a \mathcal{N}(V_a \mid \mathbf{0}, \sigma_V^2 \boldsymbol{I}).$$

Графическая модель



Вероятностное разложение матриц

- Если просто зафиксировать σ^2, σ_V^2 и σ_U^2, то они будут играть роль регуляризаторов, и нет никакого отличия от «обычного» SVD.
- Разница здесь в том, что теперь мы можем автоматически найти оптимальные $\boldsymbol{\sigma} = (\sigma^2, \sigma_V^2, \sigma_U^2)$, максимизируя общее правдоподобие модели:

$$\boldsymbol{\sigma}^* = \arg\max_{\boldsymbol{\sigma}} p(R \mid \boldsymbol{\sigma}) = \arg\max_{\boldsymbol{\sigma}} \int p(R, U, V \mid \boldsymbol{\sigma}) dU dV$$

EM-алгоритмом:

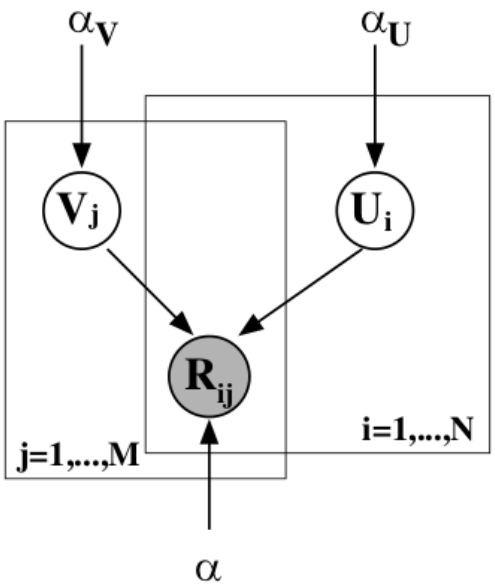
- сначала зафиксируем $\boldsymbol{\sigma}$ и найдём

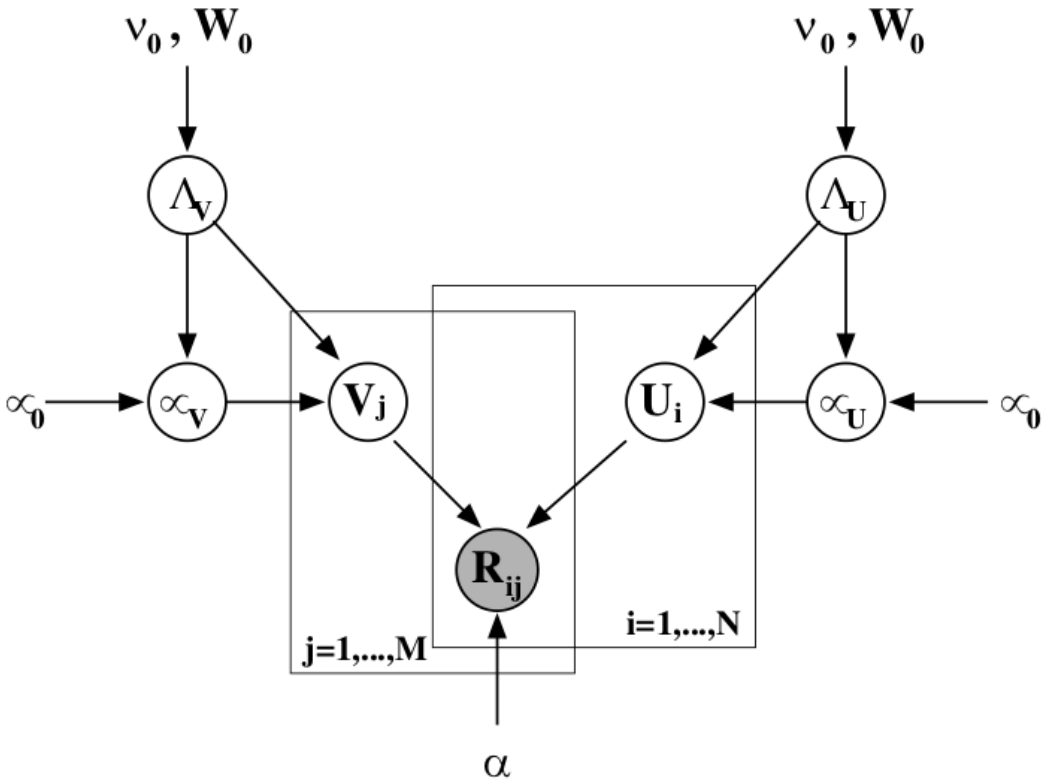
$$f(\boldsymbol{\sigma}) = \mathbf{E}_{U,V|R,\boldsymbol{\sigma}} \left[\log p(R, U, V \mid \boldsymbol{\sigma})\right];$$

- потом максимизируем

$$\boldsymbol{\sigma} := \arg\max_{\boldsymbol{\sigma}} f(\boldsymbol{\sigma}).$$

Графическая модель





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.