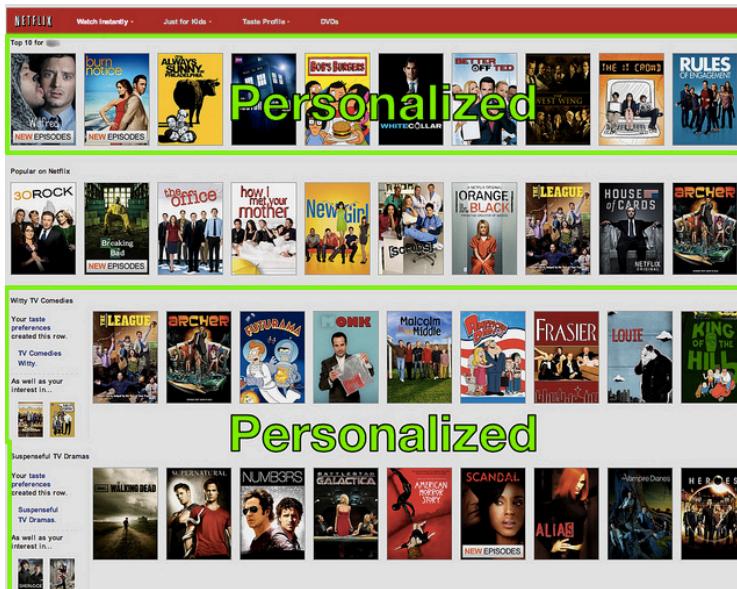


# Обучение построению рекомендаций

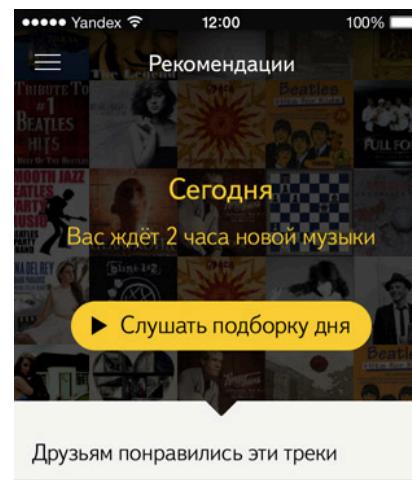
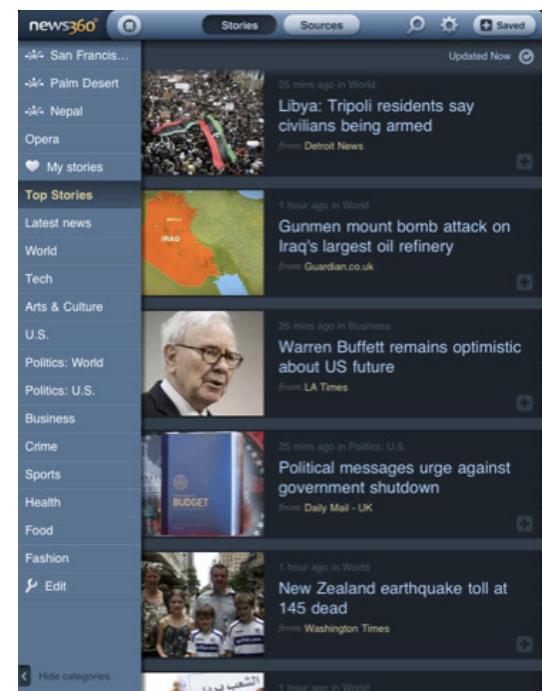
Петя Ромов

# Рекомендательные системы

Netflix: Рекомендация фильмов



News360:  
персональная новостная лента



Яндекс.Музыка:  
персональное интернет-радио

# Рекомендательные системы

- Всяческого рода реклама
- Электронная коммерция
  - «С этим товаром также покупают»
- Персональные почтовые рассылки (спам)
  - Отправить нужное письмо правильному человеку

# Составляющие рекомендательной системы

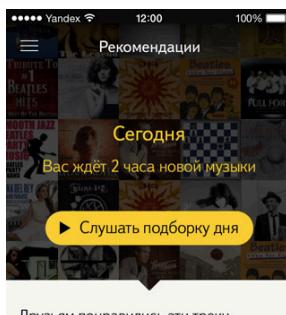
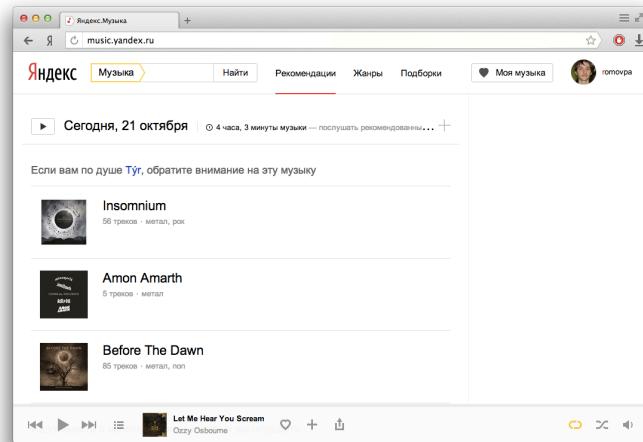
- Пользователи
- Объекты рекомендаций
- Отклики
  - акт взаимодействия пользователя с объектом
- Формат
  - Скорость генерации рекомендаций
  - Число рекомендованных объектов для пользователя

# Рекомендация фильмов

The screenshot shows the Netflix interface. At the top, it displays 'Shaun Dishman | Your Account'. Below the header, there's a navigation bar with 'Browse', 'Recommendations', 'Friends', 'Queue', and 'Buy DVDs'. Underneath are links for 'Home', 'Genres', 'New Releases', 'Netflix Top 100', 'Critics' Picks', and 'Award Winners'. The main content area is titled 'Movies For You'. It starts with a message: 'Shaun, the following movies were chosen based on your interest in: MASH, Scanners, Brick'. Below this is a section titled 'OTHER MOVIES YOU MIGHT ENJOY' featuring three movie cards: 'The Sisterhood of the Traveling Pants', 'Me and You and Everyone We Know', and 'The Producers'. Each card includes a small thumbnail, the movie title, an 'Add' button, and a 5-star rating scale with 'Not Interested' at the bottom.

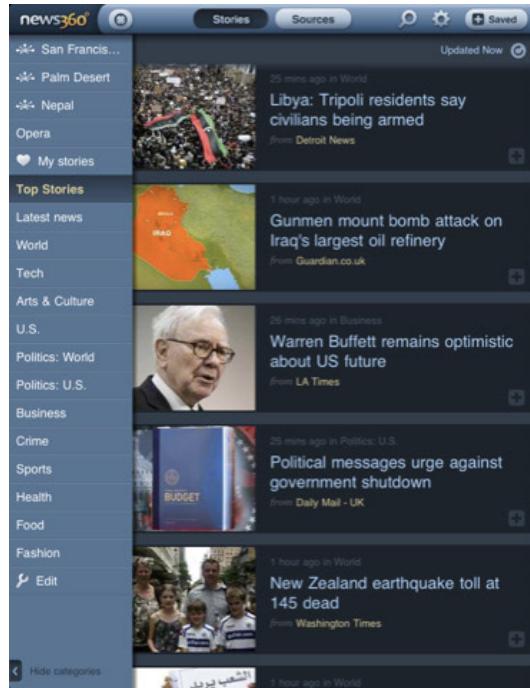
- Пользователь
  - оценки, поставленные фильмам
  - просмотренные фильмы
- Объект (фильм)
  - режиссер, актеры
  - описание от производителя, жанр
- Отклик
  - оценки фильмам по 5-балльной шкале
  - просмотры фильмов, клики
- Особенности
  - Офф-лайн обновление: можно построить рекомендации и крутить весь день / неделю

# Рекомендация музыки



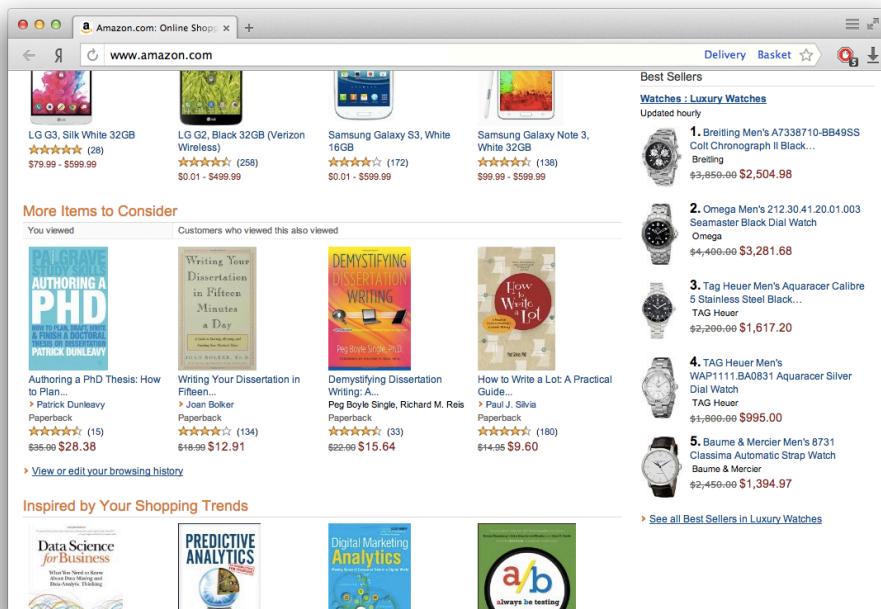
- Пользователь
  - известно поведение на сервисах Яндекса
- Объект (трек, альбом, артист)
  - таксономия
  - лейбл, описание альбомов (скучная информация)
  - признаки из аудио-сигнала
- Отклик (лайк, скип, плейлист, просмотр)
  - **лайк и плейлист** означает положительное предпочтение
  - про остальные — не ясно
- Особенности
  - Один трек можно (нужно!) рекомендовать много раз
  - Он-лайн рекомендации: быстрое обновление
  - Рекомендации идут потоком

# Рекомендация новостей



- Пользователь
  - история переходов по новостным страницам
- Объект (новостная статья)
  - текст, теги
  - источник публикации
- Отклик (переход)
- Особенности
  - Новости очень быстро протухают
  - Интересы пользователей подвержены общему тренду (пример: большое происшествие)

# Рекомендации в e-commerce



- Особенности
  - Рекомендации можно померить в чистых деньгах
  - Необходимо учитывать торговые особенности
- Отклик
  - Клик
  - Покупка

# Идея алгоритмов построения рекомендаций

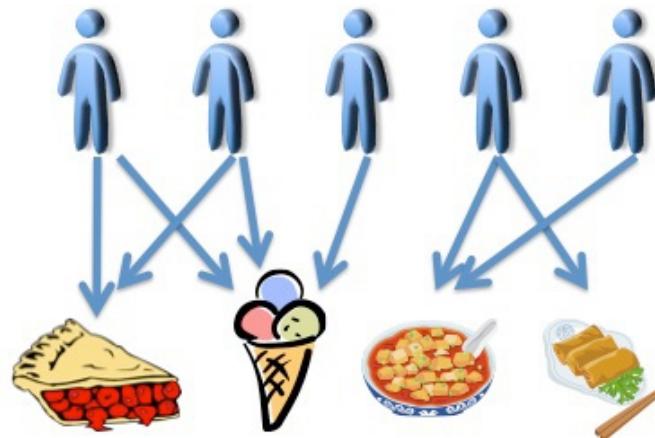
	Фото	Книга	Видео	Игра
Пользователь 1	👍	👎	👍	👍
Пользователь 2		👍	👎	👎
Пользователь 3	👍	👍	👎	
Пользователь 4	👎		👍	
Пользователь 5	👍	👍		👎

# Идея алгоритмов построения рекомендаций

	Фото	Книга	Фильм	Игра
Пользователь 1	👍	👎	👍	👍
Пользователь 2		👍	👎	👎
Пользователь 3	👍	👍	👎	
Пользователь 4	👎		👍	
Пользователь 5	👍	👍	?	👎

# Коллаборативная фильтрация

- Имеются данные о предпочтениях для пары (пользователь, объект)



# Коллаборативная фильтрация

- Требуется построить правило, предсказывающее **оценку предпочтения**
- Оценка предпочтения используется для **ранжирования** объектов

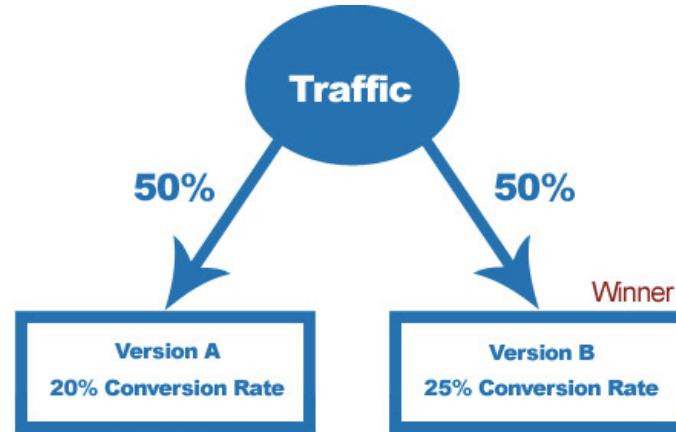
$$(\text{👤}, \text{-avatar}) \xrightarrow{f} \{1, 0\}$$

# Особенности рекомендательного ранжирования

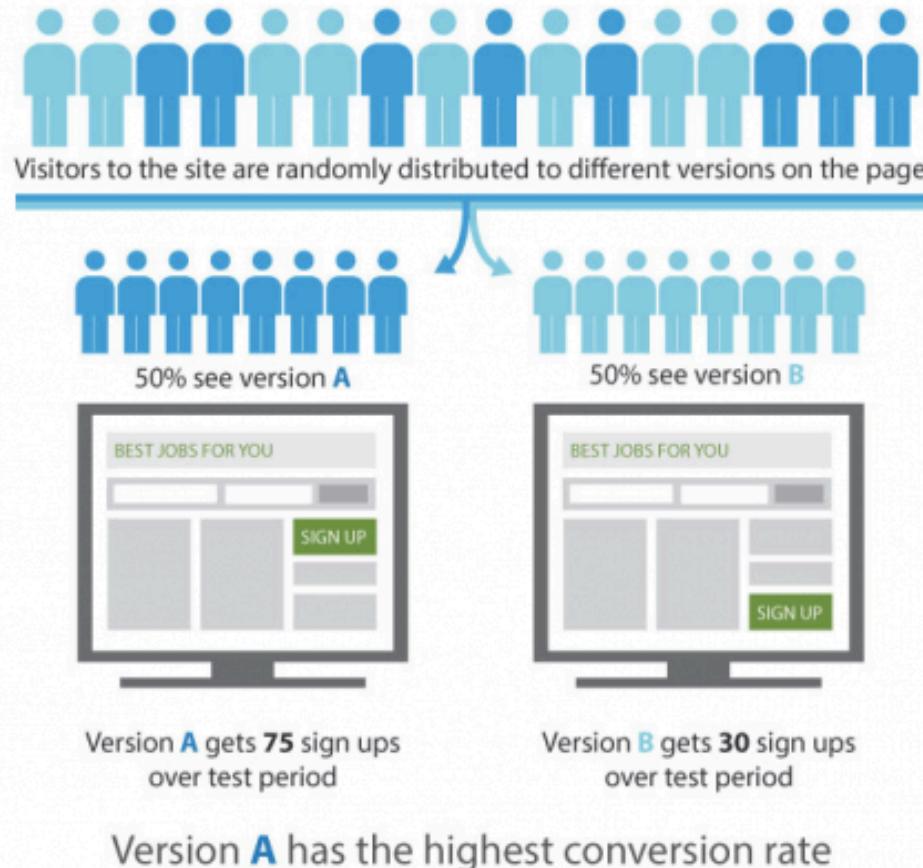
- отсутствует богатое **признаковое описание** объектов, пользователей
- отсутствует (либо очень мало) **отрицательный отклик**
- данные смещены в сторону предложенных рекомендаций
- ассессорское оценивание практически невозможно

# Оценка качества рекомендаций

- **Он-лайн**
  - АВ-тест
  - Честная оценка бизнес-метрик
  - Дорогой эксперимент
- **Офф-лайн**
  - Оценка метрик ранжирования
    - ожидается, что они коррелируют с бизнес-метрикой
  - Эмуляции работы рекомендательной системы **(протокол)**
  - Дешевый эксперимент



# А/В Тестирование



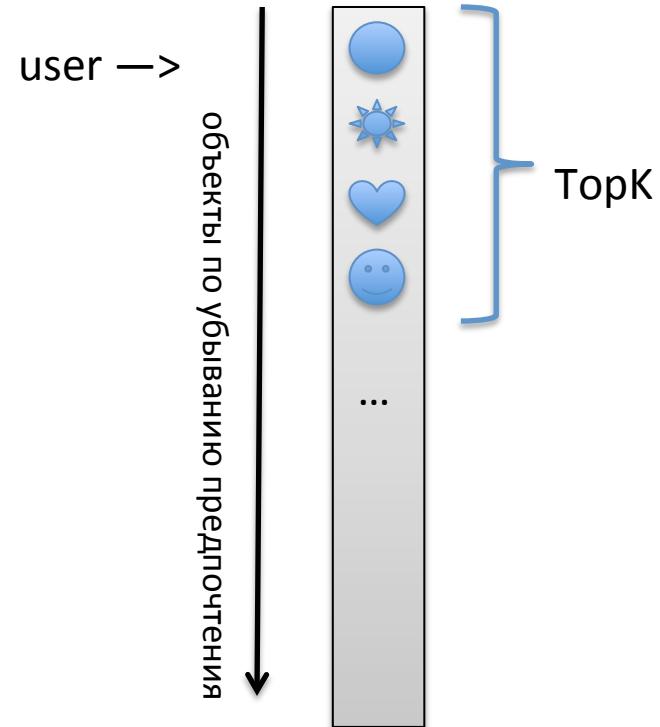
**Пример:** оптимизация числа регистраций на сайте

# Он-лайн оценка: бизнес метрики

- Netflix
  - Суммарное число просмотров
  - Число кликов в блок рекомендаций
- Яндекс.Музыка
  - Суммарное время прослушивания пользователем
- Электронная коммерция
  - Конверсия посетителя сайта в покупателя
- Метрики неуспеха
  - Число жалоб
  - Отток пользователей

# Офф-лайн оценка качества рекомендаций

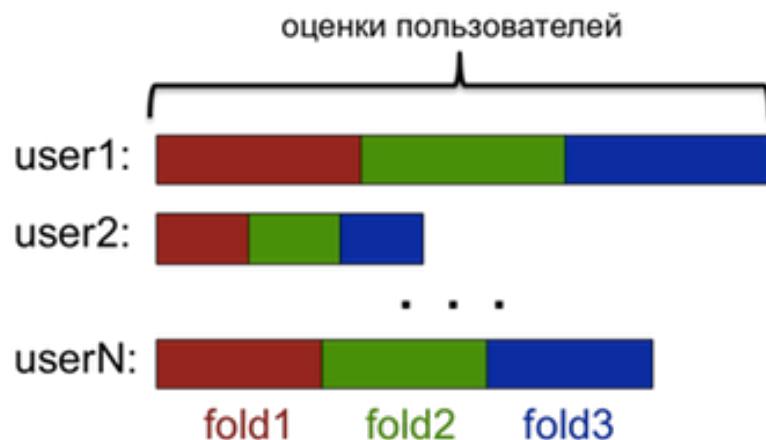
- Метрики
  - HitRate@k
  - Recall@k
  - AUC-Recall@k
  - MAP@k
  - nDCG@k
- Протокол оценки качества
  - разбиение
  - рекомендации
  - учет времени / контекста
  - отклонение известных пользователю объектов



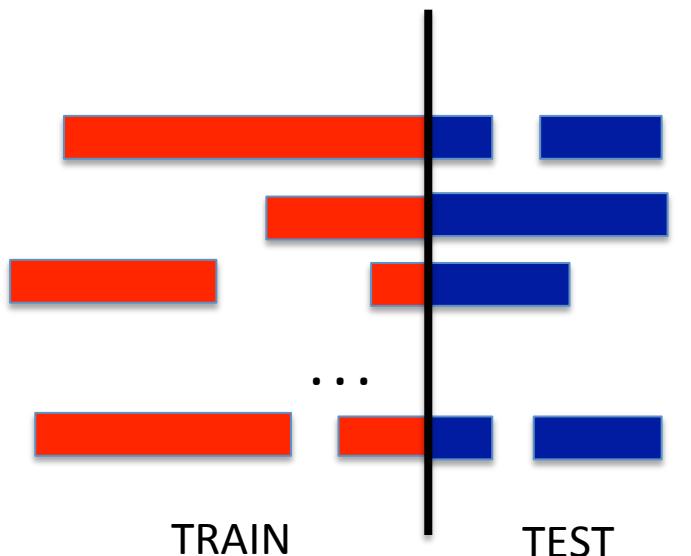
# Офф-лайн оценка: разбиение

Разбиение множества оценок пользователей на Train/Test или фолды  
для Кросс-валидации

независимо от времени



всеобщий временной порог



# Метрики рекомендательного ранжирования

$$MAP@k = \frac{1}{|U|} \sum_{u \in U} AP@k$$

$$AP@k = \sum_{i=1}^K \frac{\text{Precision}(i)}{i}$$

Precision(k) — доля релевантных среди Top(k)

Recall(k) — доля релевантных в Top(k) среди всех релевантных

# Метрики рекомендательного ранжирования

Различная степень релевантности:

$$DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

Бинарная релевантность:

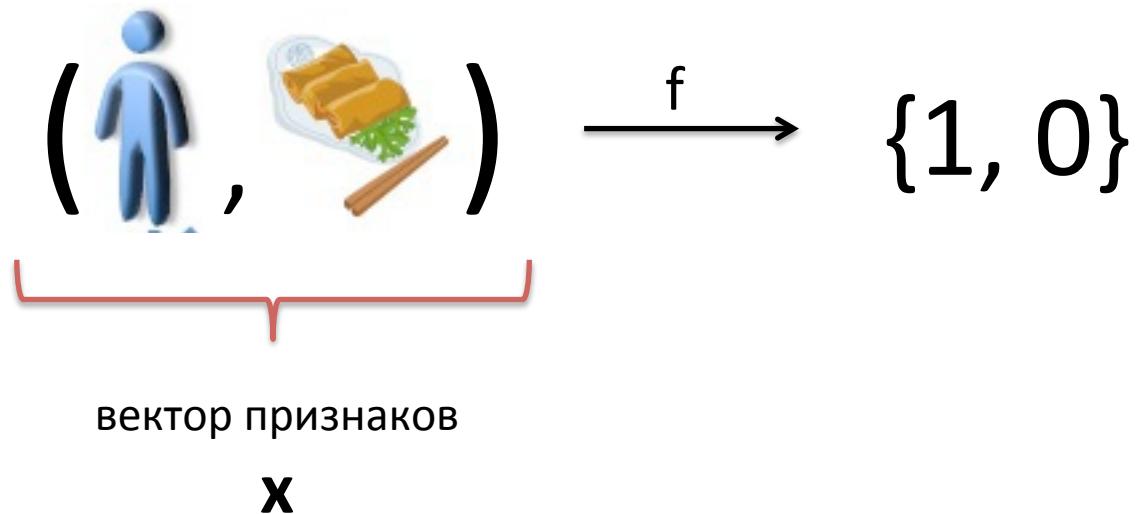
$$DCG@k = \sum_{i=1}^k \frac{rel_i}{\log_2(i + 1)}$$

$$nDCG = DCG / IdealDCG$$

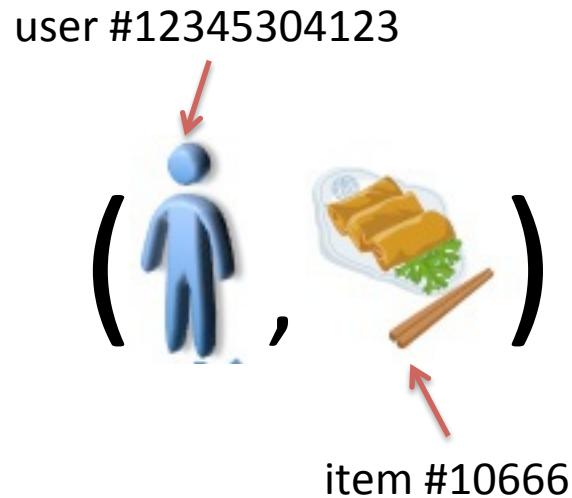
# Алгоритмы построения рекомендаций

- Memory-based
  - User/Item similarity
- Model-based (это про нас, про ML)
  - Weighted-SVD
  - Factorization Machines
  - Композиции алгоритмов

# Коллаборативная фильтрация



# Коллаборативная фильтрация

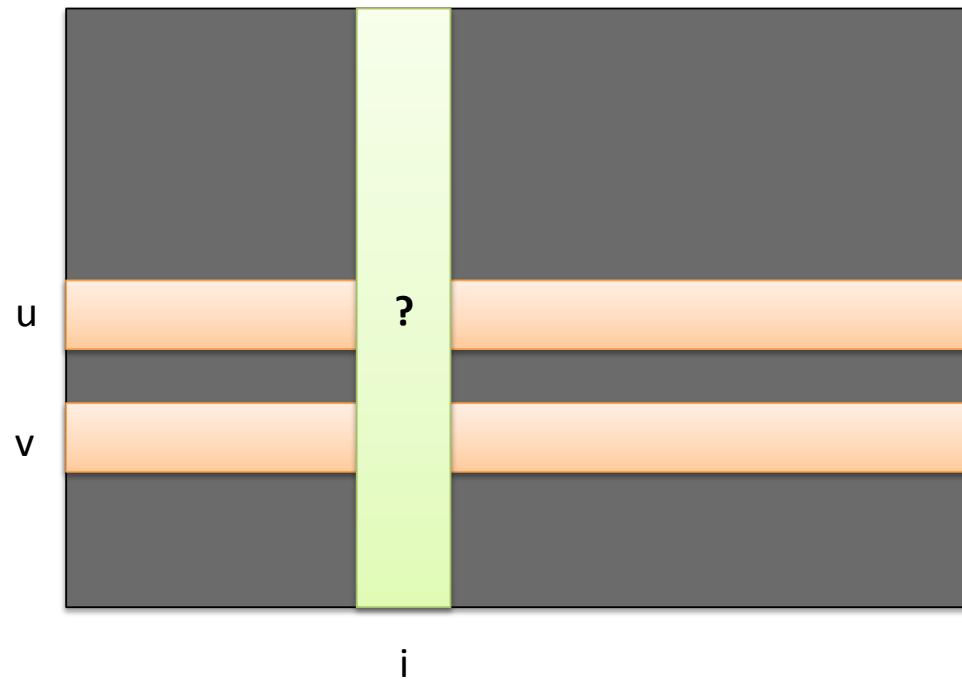


# Коллаборативная фильтрация: введем обозначения

- Пользователи  $u \in U$
- Объекты  $i \in I$
- События  $(r_{ui}, u, i, \dots)$
- Предсказание  $\hat{r}_{ui}$
- Множество событий (оценок)  $\mathcal{R}$
- Объекты, имеющие пользовательскую оценку  $\mathcal{R}(u)$

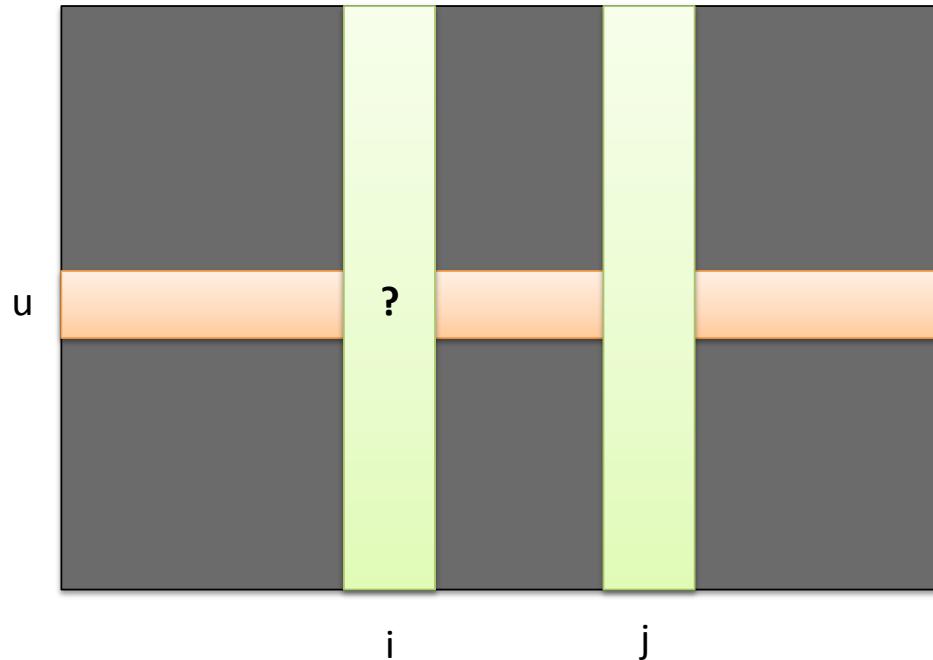
# User-based Filtering

$$\hat{r}_{ui} = \bar{r}_u + \sum_{v \in U_i} sim(u, v)(r_{vi} - \bar{r}_v)$$



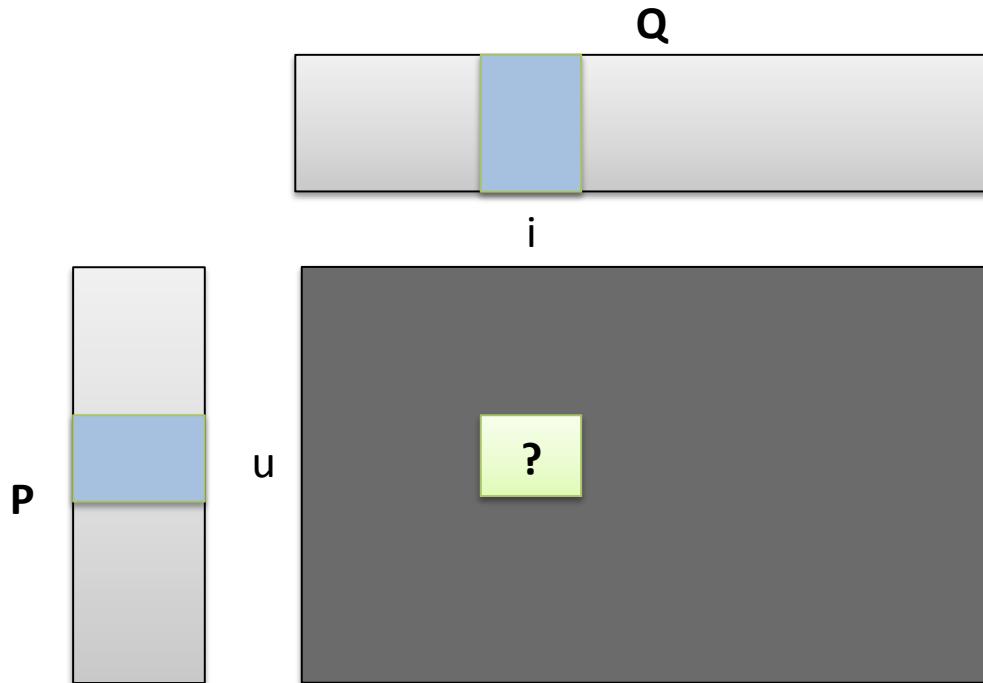
# Item-based Filtering

$$\hat{r}_{ui} = \bar{r}_i + \sum_{j \in I_u} sim(i, j)(r_{uj} - \bar{r}_j)$$



# Model-based

$$\hat{r}_{ui} = \mu + b_u + b_i + \langle \vec{p}_u, \vec{q}_i \rangle$$



# Модель SVD

$$\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i$$

$$\sum_{(u,i) \in \mathcal{R}} (r_{ui} - \mathbf{p}_u^T \mathbf{q}_i)^2 + \lambda \left( \sum_u \|\mathbf{p}_u\|^2 + \sum_i \|\mathbf{q}_i\|^2 \right) \rightarrow \min_{\Theta}$$

Оптимизация при фиксированных векторах объектов Q

$$\sum_u \left\{ \sum_{i \in R(u)} (r_{ui} - \mathbf{p}_u^T \mathbf{q}_i)^2 + \lambda \|\mathbf{p}_u\|^2 \right\} \rightarrow \min_{\{\mathbf{p}_u\}}$$

# Модель SVD: обновление профиля пользователя

$$\mathbf{p}_u^* = \underbrace{(\mathbf{Q}_u^T \mathbf{Q}_u + \lambda \mathbf{I})^{-1}}_{\mathbf{W}_u} \underbrace{\mathbf{Q}_u^T \mathbf{r}_u}_{\mathbf{d}_u} = \mathbf{W}_u \mathbf{d}_u = \sum_{j \in \mathcal{R}(u)} \mathbf{W}_u \mathbf{q}_j r_{uj}$$

$$\hat{r}_{ui} = \mathbf{q}_i^T \mathbf{p}_u^* = \sum_{j \in \mathcal{R}(u)} \mathbf{q}_i^T \mathbf{W}_u \mathbf{q}_j r_{uj} = \sum_{j \in \mathcal{R}(u)} r_{uj} \langle \mathbf{q}_i, \mathbf{q}_j \rangle \mathbf{W}_u$$

$$\mathbf{W}_u = \mathbf{V}_u^T \mathbf{V}_u \quad \hat{r}_{ui} = \sum_{j \in \mathcal{R}(u)} r_{uj} \langle \mathbf{V}_u \mathbf{q}_i, \mathbf{V}_u \mathbf{q}_j \rangle$$

# Implicit Feedback: есть только положительный отклик

- Имеются только тройки  $(u, i, r=1)$
- Какую модель обучит алгоритм SVD?

# Implicit Feedback: есть только положительный отклик

- Имеются только тройки  $(u, i, r=1)$

$$\sum_u \sum_i c_{ui} (r_{ui} - p_u^T q_i)^2 + \lambda \Omega(P, Q) \rightarrow \min_{\Theta}$$

- Пытаемся приблизить все элементы матрицы R
  - В случае, если отклика нет — он считается отрицательным ( $r=0$ )
- Элементы, которые не наблюдаем берем с малым весом
- Много слагаемых!
  - Алгоритм I-ALS при помощи предпосчета работает со сложностью обычного ALS

# Усложнение модели SVD

Пусть предметы имеют теги:

$$i \mapsto T(i) = \{t_1, t_2, \dots\} \subseteq \mathcal{T}$$

Учет тегов в модели: введем латентные вектора тегов  
 $t \mapsto \mathbf{x}_t \in \mathbb{R}^d$ .

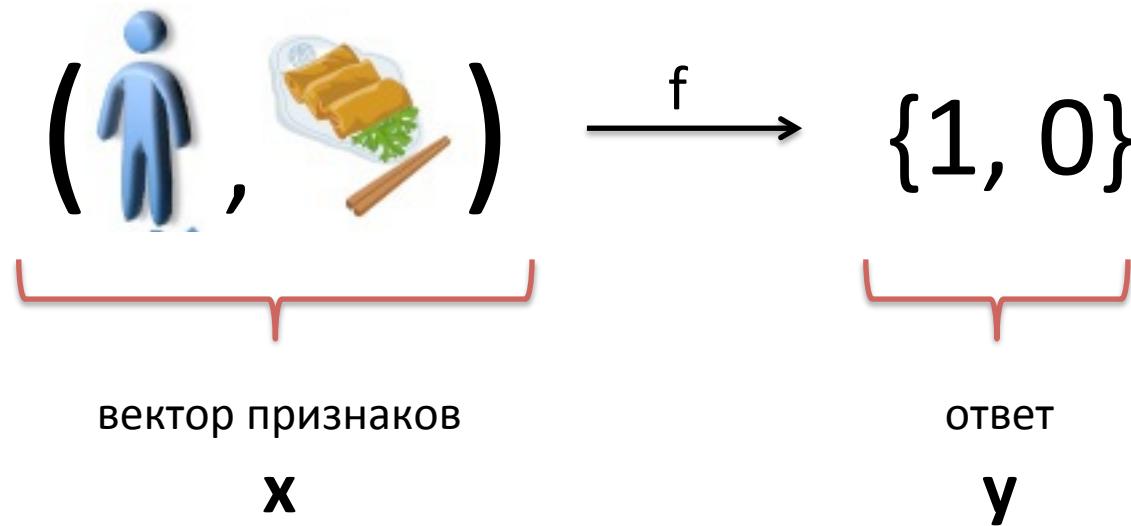
$$\hat{r}_{ui} = \mu + b_u + b_i + b_t + \langle \mathbf{p}_u, \mathbf{q}_i + \sum_{t \in T(i)} \mathbf{x}_t \rangle$$

# Таксономическая модель SVD

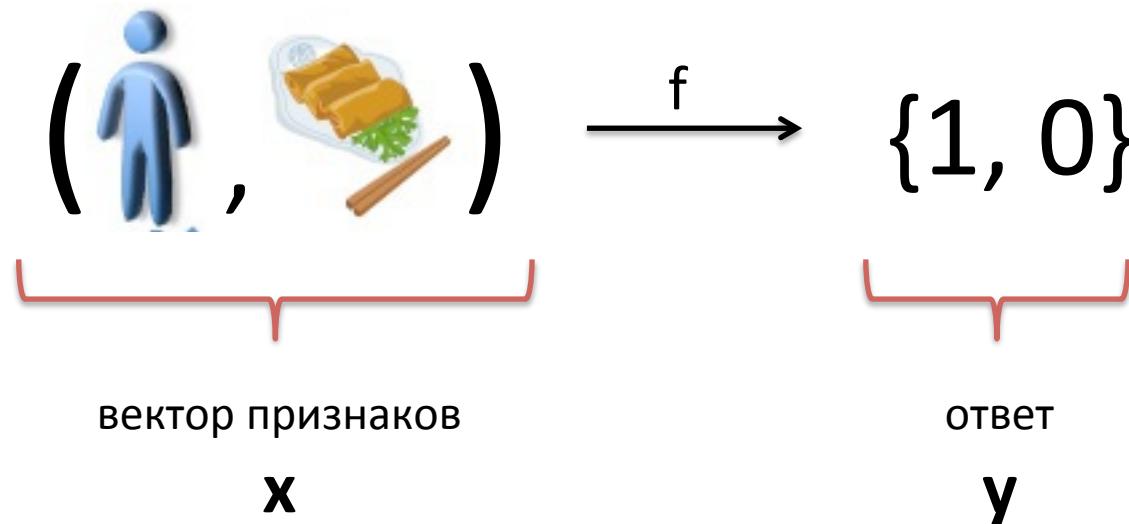
- ▶ Треки, альбомы и артисты являются предметами в модели
- ▶  $\text{type}(i) \in \{\text{track}, \text{album}, \text{artist}\}$
- ▶  $\text{album}(i)$  — предмет, являющийся альбомом предмета  $i$
- ▶  $\text{artist}(i)$  — предмет, являющийся артистом предмета  $i$
- ▶  $\tilde{b}_{ui} = \mu + b_u + b_{u,\text{type}(i)} + b_i + b_{\text{album}(i)} + b_{\text{artist}(i)}$
- ▶  $\tilde{\mathbf{q}}_i = \mathbf{q}_i + \mathbf{q}_{\text{album}(i)} + \mathbf{q}_{\text{artist}(i)}$

$$\hat{r}_{ui} = \tilde{b}_{ui} + \langle \mathbf{p}_u, \tilde{\mathbf{q}}_i \rangle$$

# Факторизационные машины



# Факторизационные машины



# Факторизационные машины

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{j=1}^p w_j x_j + \sum_{j=1}^p \sum_{j'=j+1}^p x_j x_{j'} \langle \mathbf{v}_j, \mathbf{v}_{j'} \rangle$$

$$(u, i) \mapsto \mathbf{x}, \quad \hat{y}(\mathbf{x}) = w_0 + w_u + w_i + \langle \mathbf{v}_u, \mathbf{v}_i \rangle$$

Feature vector  $\mathbf{x}$

$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...

A B C ... User      TI NH SW ST ... Movie

Target y

5	$y^{(1)}$
3	$y^{(2)}$
1	$y^{(3)}$
4	$y^{(4)}$
5	$y^{(5)}$
1	$y^{(6)}$
5	$y^{(7)}$

# Факторизационные машины

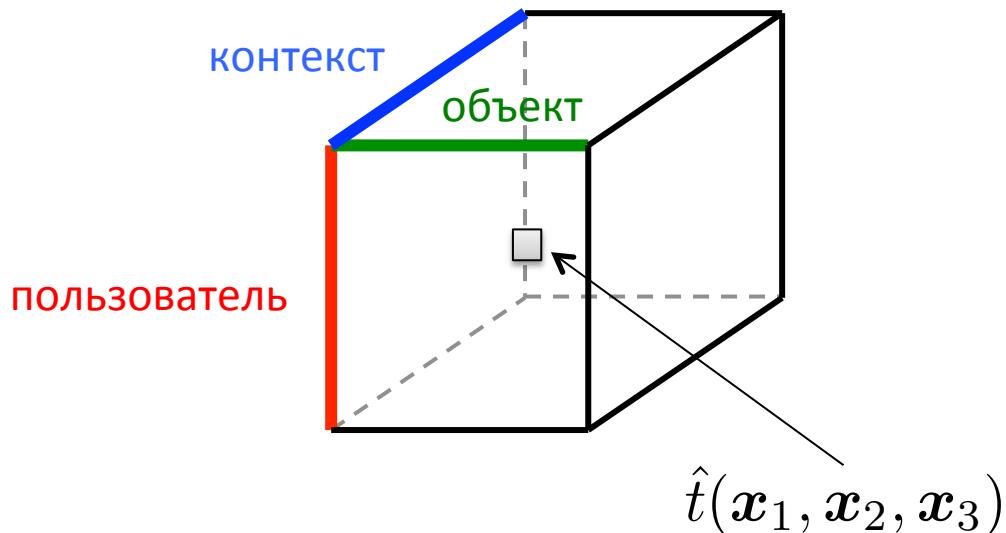
$$\hat{y}(\mathbf{x}) = w_0 + \sum_{j=1}^p w_j x_j + \sum_{j=1}^p \sum_{j'=j+1}^p x_j x_{j'} \langle \mathbf{v}_j, \mathbf{v}_{j'} \rangle$$

Feature vector $\mathbf{x}$													Target $y$										
$\mathbf{x}_1$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y_1$	
$\mathbf{x}_2$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y_2$	
$\mathbf{x}_3$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y_3$	
$\mathbf{x}_4$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y_4$	
$\mathbf{x}_5$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y_5$	
$\mathbf{x}_6$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y_6$	
$\mathbf{x}_7$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y_7$	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...			
	User				Movie					Other Movies rated						Last Movie rated							

# Факторизационные машины

- Позволяют добавить в модель коллаборативной фильтрации
  - обычные признаки
  - информацию о контексте рекомендаций
  - таксономию объектов
  - теги
  - ...
- Эффективный и масштабируемый метод обучения
- Хорошие результаты на Kaggle (!)

# Тензорные разложения



- Обобщение алгоритма Implicit-ALS

Hidasi, Balázs, and Domonkos Tikk. "Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback." Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2012. 67-82.

# Инструменты: mrec

- Python-библиотека
- Фреймворк для оценки качества
  - подготовка/разбиение выборки
  - вычисление метрик (HitRate, Recall, MAP, ...)
- Набор стандартных алгоритмов
  - WRMF
  - KNN (User/Item-based)
  - Popularity baseline
  - SLIM, WARM (не очень стандартные)

<http://mendeley.github.io/mrec/>

# Инструменты: GraphLab Collaborative Filtering Toolkit

- Большое число эффективно реализованных факторизационных моделей
  - SVD (ALS, SGD)
  - SVD++
  - Weighted-ALS
  - Non-negative Matrix Factorization
  - CCD++
- Алгоритмы представлены отдельными бинариками с простым консольным интерфейсом

[http://docs.graphlab.org/collaborative\\_filtering.html](http://docs.graphlab.org/collaborative_filtering.html)

# Инструменты: Vowpal Wabbit

- Создаем несколько Namespas-ов  
**1 |user 1 |item a |producer P**
- Режим Matrix Factorization  
**--rank 10 -q ui -q up**
- Можно вывести компоненты разложения
- Алгоритм: SGD
  - Никак не сделать обработку Implicit

[https://github.com/JohnLangford/vowpal\\_wabbit/wiki/Matrix-factorization-example](https://github.com/JohnLangford/vowpal_wabbit/wiki/Matrix-factorization-example)

# Инструменты: LibFM

- Эффективная реализация модели Факторизационных машин
- Алгоритмы: SGD, ALS
  - МСМС: круто работает, но нельзя сохранить модель
- Эффективная обработка реляционной структуры матрицы объект-признак

The screenshot shows a web browser displaying a Kaggle competition page. The title of the competition is "Private Leaderboard - What Do You Know?". The page indicates that the competition is completed with a total prize of \$5,000 and 239 teams participated. The date range for the competition was from Fri 18 Nov 2011 to Wed 29 Feb 2012 (2 years ago). The dashboard tab is selected. Below it, the private leaderboard for the competition is shown. The table has columns for rank, team name, score, entries, and last submission UTC. The top entry is circled in red.

#	Δ1w	Team Name *in the money*	Score	Entries	Last Submission UTC (Best - Last Submission)
1	—	Steffen Rendle *	0.24598	16	Sun, 26 Feb 2012 08:05:07
2	↑1	Alexander D'yakonov *	0.24729	38	Wed, 29 Feb 2012 22:38:54 (-0.9h)

<http://libfm.org/>