# Recommendation Systems

Introduction

Eugeny Malyutin / Sergey Dudorov

# Who am I to lecture you?

- Unfinished PhD (in academic vacation)
- No kaggle medals
- 5+ years in recommendation systems
- Data magician at ok_ru and vk_com:
  - Newsfeed
  - PYMK (people you may know)
  - Search engines
  - Streaming news trends
  - Group recommendation
  - …

# Course grade system:

- 35% home work assignments (3 HW + 2 deadlines on each)
- 35% seminar work
- 30% exam (oral)

# Prerequisites:

- Python
- Core ML

# Rec sys in 5 minutes:

Продакты не нужны

**Машинлернинг на коленке**

Машинлернинг и датасаенс специалисты стоят очень дорого. Не исключено, что у них тоже есть свой заговор, как у продактов.

Но что делать, если вам в стартапе нужны умные алгоритмы рекомендации контента, а все деньги вы уже профукали на касдевы и блогеров инфлуенсеров? Рассказываю.

Берёте свой массив объектов и сортируете его по популярности за последний час/сутки/неделю, в зависимости от того, сколько трафика у вас есть. Что считать популярностью - решите сами, это особо ни на что не влияет: просмотры, лайки, попадание во вьюпорт. Пойдёт любая метрика.

Если кто-то предложит «давайте хотя бы старгетируем эту сортировку на мужчин и женщин», можете ударить его.

Этой сортировки хватит на ближайшую пару лет, можете смело забывать про этот кусок и заниматься другими делами. Через несколько лет, когда уже нечем будет заняться, а деньги некуда будет девать, наймёте дата саентистов. Они за полгода напишут алгоритм, который ещё на 10% улучшит эту сортировку. На большом размере бизнеса это будет ощутимо.

9949 изменено 09:31

👍 220

Source: https://t.me/betternotworse

# Everything is a recommendation

Over 80% of what people watch in Netflix comes from recommendations (https://dl.acm.org/doi/pdf/10.1145/2843948)

# OK, what’s more?

E-commerce (Amazon):

# Newsfeed / PYMK

# MOAR:

Google
как попасть в рекомендации тик ток
ALL
VIDEOS
IMAGES
NEWS
MAPS
Когда видео заканчивается, **Тик-Ток** запускает его заново. Если человек смотрит его снова и снова, это знак для алгоритма, что видео зацепило, и тогда оно может **попасть в рекомендации**.
Dec 20, 2019
https://postium.ru › kak-popast-v-re...
Как попасть в рекомендации Тик-Ток: хэштеги, алгоритм, секреты для видео ...
About Featured Snippets
Feedback
https://partnerkin.com › blog › articles
Как в Тик Токе попасть в рекомендации если ты не популярен и нет ...

Умные плейлисты для вас, Евгений
Плейлист дня
Подкасты недели
Премьера
Дежавю
Плейлист дня
Каждый день — новый. Каждый день — ваш!
Обновлён сегодня
Подкасты недели
Слушайте с пользой: подкасты, которые могут вам понравиться
Обновлён в среду
Премьера
Только новинки, подобранные по вашим предпочтениям
Обновлён вчера
Дежавю
Вы ещё не слушали эти треки, но, похоже, вам они понравятся
Обновлён в четверг

in:spam
1–5 из 5
Письма, хранящиеся в спаме более 30 дней, удаляются автоматически
Удалить все письма со спамом
J Editor
Invite You to Submit Papers and Join Editorial Panel/Reviewer Team -- ВЫДЕЛЕНИЕ ТЕКСТОВЫХ ТРЕНД...
25 авг.
SimpleWine
Красное вино для жаркого августа - Что делать, если в жару хочется красного
20 авг.
editorial
Become a part of Editorial Board : (CCCS) - Current Chinese Computer Science 17 August, 2020 Dr. EA Mal...
17 авг.
Роман Юрьевич Царев
Приглашаем подать статью в Journal of Physics: Conference Series (Scopus) с публикацией в IV квар...
11 авг.
Editor Zhang
SCOPE of this conference - Opened in Xi'an (Xi'an Guangcheng Hotel) 2020 International Conference on Ph...
4 авг.

ЧТО?!

# What is recommender systems?

# What is recommender systems?

# What is recommender systems?

**Candidates**

Action-based
Matrix-factorisation
Unclassical embeddings
Deep learning
...

**Ranking engine**

Classical ml
Neural networks
...
+ learning to rank

**Business processing**

Diversity
Debiasing
...

**Ranked candidates**

UI/UX
Decision making (A/B)
Metrics awareness

# Topics list

- Matrix factorisation
- Unclassical embeddings
- Learning to rank
- BI + data metrics
- AB-tests
- Bayesian bandits
- Diversity + debiasing
- Neural Networks
- Deep learning
- SysML (how ~~not~~ to fail prod)

# History: Netflix Prize

- 480189 users
- 17770 movies
- 1004800507 ratings
- 02.10.2006 > 21.09.2009
- 1000000$
- Task: RMSE up on 10% (0.9513 -> 0.8536)

https://www.thrillist.com/entertainment/nation/the-netflix-prize

# Isn’t solved yet? No, it’s hard!

- Every person is unique with a variety of interests
- Help people what they want to they’re not sure hat the want
- Large datasets but small data per user
  … and potentially biased by the output of your system
- Cold-start problems on all sides
- Non-stationarity, context-dependent, mood-dependent
- More than just accuracy: diversity, novelty, freshness, fairness
- …

# Formally:

- U — set of subjects (users)
- R — set of objects (items/goods/posts/…)
- Y — transactions space

**Raw data:** $D = (u_i, r_i, y_i)_{i=1}^{m}$

**Aggregated data:**

$F = |f_{ur}|$ - cross-tabulate matrix $|U| \times |R|$

$$f_{ur} = agg\{(u_i, r_i, y_i) \in D \,|\, u_i = u, r_i = r \,|\,\}$$

**Tasks:**

- Predict unknown cells values $f_{ur}$
- Evaluate similarity metrics: $p(u, u), p(r, r), p(u, r)$
- Discover latent topics $p(t \,|\, u), p(t \,|\, r)$ related to known or unknown topics list t = 1…T

# Example:

- U — all users of vk.com
- R — all (with some exceptions) groups of vk.com

D — sequence of user interaction with communities (transaction data)

F = $|f_{ur}|$ , subscription matrix. Factually agg function - filtering out everything except subscriptions

**Tasks:**

- Form a list of relevant groups for a user s/he hasn't subscribed yet
- Find «similar» groups for a given one

vk.com/groups

# Recommenders ontology:

- **Correlation** models:
  - Keep all matrix F in memory
  - Users similarity — correlation between rows of matrix F
  - Objects similarity — correlation between cols of matrix F
- **Latent** models:
  - Estimate profiles of clients and objects (profile as vector of latent/hidden characteristics)
  - Keep profile instead of matrix F.

# Correlation models:trivial example

Peoples who bought $r_0$ also bought $R(r_0)$ (Amazon)

- $U(r_0) = \{u \in U \,|\, f_{ur_0} \neq \varnothing, u \neq u_0\}$ — «collaboration»
- $R(r_0) = \{r \in R \,|\, B(r) = \dfrac{|\,U(r_0) \cap U(r)\,|}{|\,U(r_0) \cup U(r)\,|}$, B — any similarity measures (not only ^ Jaccard)
- Sort $R(r_0)$ by $B(r)$ descending, take top(N)

# Trivial example

Peoples who bought $r_0$ also bought $R(r_0)$ (Amazon)

- $U(r_0) = \{u \in U \,|\, f_{ur_0} \neq \varnothing, u \neq u_0\}$ — «collaboration»
- $R(r_0) = \{r \in R \,|\, B(r) = \dfrac{|\,U(r_0) \cap U(r)\,|}{|\,U(r_0) \cup U(r)\,|}$, B — any similarity measures (not only ^ Jaccard)
- Sort $R(r_0)$ by $B(r)$ descending, take top(N)

**Problems?**

# Trivial example

**Problems:**

- Trivial recommendations
- Does not take into account the interests of the concrete user
- «Cold start» problem
- Need to keep huge matrix R in memory

# Corr models: user-based

People like $u_0$ also bought $R(u_0)$

- $U(u_0) = \{u \in U \,|\, corr(u, u_0) > \alpha\}$ — collaboration
- $R(u_0) = \{r \in R \,|\, B(r) = \dfrac{|\,U(u_0) \cap U(r)\,|}{|\,U(u_0) \cup U(r)\,|}\}$, B — any similarity metric
- Sort $R(u_0)$ by B desc, take top-N

# Corr models: user-based

People like $u_0$ also bought $R(u_0)$

- $U(u_0) = \{u \in U \,|\, corr(u, u_0) > \alpha\}$ — collaboration
- $R(u_0) = \{r \in R \,|\, B(r) = \dfrac{|\,U(u_0) \cap U(r)\,|}{|\,U(u_0) \cup U(r)\,|}\}$, B — any similarity metric
- Sort $R(u_0)$ by B desc, take top-N

**Pr-oblems?**

# Corr models: user-based

**Pr-roblems:**

- Still matrix R in memory
- Cold start
- Problems with new and unusual users

# Corr models: item-based

- $R(u_0) = \{i \in I \mid \exists i_0 : r_{u_0,i_0} \neq \emptyset, sim(i, i_0) > \alpha\}$ – collaboration
- Sort $R(u_0)$ by B desc and take top-N
- Problems? Still the same.

# Missing values recovery

Non-parametric regression by Nadaray–Watson:

$$\hat{f}_{ur} = \bar{f}_u + \frac{\sum_{u' \in U_\alpha(u)} K(u, u')(f_{u'r} - \bar{f}_{u'})}{\sum_{u' \in U_\alpha(u)} K(u, u')}$$, where $\bar{f}_u$ — average score by user,

- $K(u, u')$ — smoothing kernel, similarity between u and u’
- $U_\alpha(u)$ — collaboration, users in alpha-neighbourhood of u

# Missing values recovery

Non-parametric regression by Nadaray–Watson:

$$\hat{f}_{ur} = \bar{f}_u + \frac{\sum_{u' \in U_\alpha(u)} K(u, u')(f_{u'r} - \bar{f}_{u'})}{\sum_{u' \in U_\alpha(u)} K(u, u')}$$, where $\bar{f}_u$ — average score by user,

- $K(u, u')$ — smoothing kernel, similarity between u and u’
- $U_\alpha(u)$ — collaboration, users in alpha-neighbourhood of u

Problems:

- Cold start
- Still huge matrix in F

# Similarity measures:

- Pearson correlation (also possible Kendal, Spearmen,…):

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

- Cosine similarity: $cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$

- Statistical criteria's based metrics (Fisher, $\chi^2$, etc…)

- Set-theory based: Jaccard, …

- Graph-based metrics: PageRank, ADAMIC-ADAR, …

# Summary:

**Pros** for business applications:

- Easy to interpret
- Easy to realise (+-)

**Cons**:

- No strong theoretical basis
- A lot of ways to estimate similarity
- A lot of hybrid user-item-based methods… and it's not clear which is better
- Keep huge matrix F

Ok, isn’t it enough?

# No, personalisation is hard!

- Every person is unique with variety of interests!
- Large dataset but small data per user
  …and potentially biased by the output of your system
- Cold-start problems on all sides
- Non-stationarity, context- mood- time- dependent
- More than just accuracy: diversity, novelty, freshness, fairness…

# What’s trending now?

1. Deep Learning
2. Causality
3. Bandits & Reinforcement Learning
4. «Fairness»
5. Experience Personalisation

# What is recommender systems?

**Candidates**

Action-based
Matrix-factorisation
Unclassical embeddings
Deep learning
...

**Ranking engine**

Classical ml
Neural networks
...
+ learning to rank

**Business processing**

Diversity
Debiasing
...

**Ranked candidates**

UI/UX
Decision making (A/B)
Metrics awareness

# Trend #1: Deep Learning

- 2012: Becomes popular in Machine Learning (started with CV)
- 2017: Becomes popular in Recommender Systems

What's takes so long?

# Traditional recommendations:

**Collaborative filtering:** choose items that similar users have chosen.

# Matrix factorization view:

**Matrix factorization:** decompose ratings matrix R onto U([num_users x factors]) x V([num_factors x num_items]).

$< u_i, v_j >$ should reconstruct original ratings R.

$$min_{u,v} \sum_{i,j \in R} (r_{i,j} - u_i^T v_j)^2 + \lambda(||u_i||^2 + ||v_j||^2)$$

# A feed-forward network:

$$min_{u,v} \sum_{i,j \in R} (r_{i,j} - u_i^T v_j)^2 + \lambda(||u_i||^2 + ||v_j||^2)$$

# A (deeper) feed-forward view

# …here is a problem

# …but opens up many possibilities

# Treat recommendation as sequence prediction:

- Bogina V., Kuflik T. Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks
  **Input:** sequence of user actions
  **Output:** next action

- Sequence-aware recommendation via attention techniques: https://habr.com/ru/company/mailru/blog/445348/

# Leveraging additional data:

- https://dl.acm.org/doi/abs/10.1145/2959100.2959190
- Two stage ranker: candidate generation (shrinking set of items to rank) and ranking (classifying actual impressions)
- Two feed-forward, fully connected, networks with hundreds of features
- Remark: too hard to implement as a runtime model

# Trend #2: Causation

- Most recommendations algorithms are correlational:
  - Some early recommendation algorithms literally computed user-items correlations
- Did you watch a movie cause you liked it? Or because we showed it to you? Or both?

$$p(Y|X) = p(Y|X, do(R))$$

# Causation

**Feedback loops leads system to reinforce biases.**
**Especially noticeable with CI/CD ml models.**

# Causation

**Closed** loop

**Open** loop

# Trend #3: Bandits and RL

- Uncertainty about user interests and new items
- Sparse and indirect feedback
- Changing trends

- Break feedback loop
- Want to explore to learn
- Interest in long-term reward (retention / timespent / etc)

# Spotify case

- Bandit selecting both items and explanations for spotify homepage
- Factorization machine with eps-greedy explore over personalized candidate set
- Counterfactual risk minimisation to train the bandit

| Explanation | # Impressions |
|---|---|
| Because it's [day of week] | 140.3K |
| Inspired by [user]'s recent listening | 138.4K |
| Because it's a new release | 140.5K |
| Because [user] likes [genre] | 130.7K |
| Because it's popular | 140.5K |
| Mood | 140.7K |
| Focus | 140.5K |

https://dl.acm.org/doi/abs/10.1145/3240323.3240354

# Challenges of RL:

- **High-dimensional actions space:** Single item rec - O(|C|) becomes combinatorial on page construction or ranking, which is combinatorial
- **High-dimensional state space:** Users represented in the state along with long history
- **Off-policy training:** Need to learn from existing policy actions
- **Concurrency:** Don’t observe full trajectories, need to learn simultaneously from many trajectories
- **Changing action space:** New items become available and need to be cold-started
- **No good simulator:** Requires knowing feedback for user on recommended items

# Trend #4: Fairness

- What is fair recommendations?
- Fairing as matching distributions os users interests
- Accuracy as an objective can lead to unbalanced predictions
- Many recommendation algo's exhibit this behaviour pf exaggerating the dominant interests and crowd out less frequent ones

https://dl.acm.org/doi/abs/10.1145/3240323.3240372

# Trend #5: Experience Personalisation

- **Algorithms level:** ideal balance of freshness, popularity, trendiness, interests similarity and etc. may depend on the person
- **Display level:** how you explain your items or explain recommendation can also be personalised
- **Interaction level:** balancing the needs of lean-back users and power users

# Experience Personalisation

# Summary correlation model:

**Pros** for business applications:

- Easy to interpret
- Easy to realise (+-)

**Cons**:

- No strong theoretical basis
- A lot of ways to estimate similarity
- A lot of hybrid user-item-based methods… and it's not clear which is better
- Keep huge matrix F

# Hype summary:

1. Deep Learning
2. Causality
3. Bandits & Reinforcement Learning
4. «Fairness»
5. Experience Personalisation

# Sources:

- Netflix overview: https://slideslive.com/38917692/recent-trends-in-personalization-a-netflix-perspective

- Bit of Vorontsov's course: http://www.machinelearning.ru/wiki/images/archive/9/95/20140413184117%21Voron-ML-CF.pdf