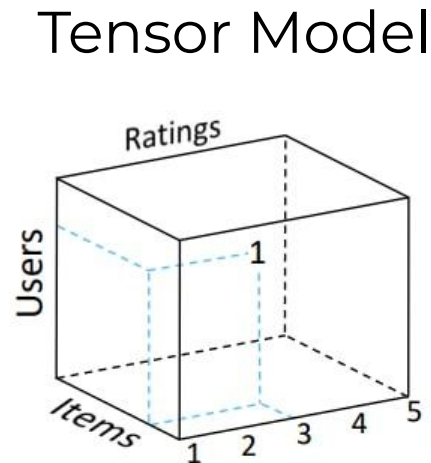
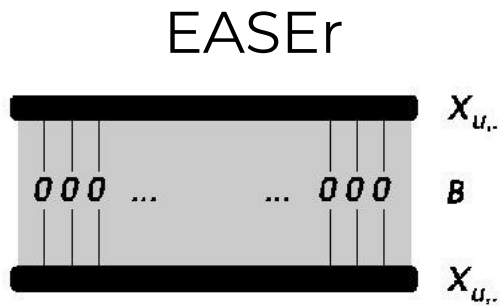
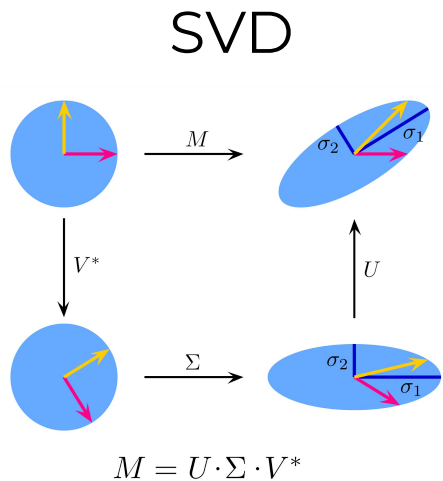


TWO-STAGE RECOMMENDER SYSTEM

Степанов Даниил
Бредихин Александр
Конюшенко Юлия
Яскевич Александр
Ачарйа Кирилл

Фролов Евгений - Дирижёр

First-stage models



- Options
- Directly generate top-10 candidates for scoring
 - Generate candidates for second-stage

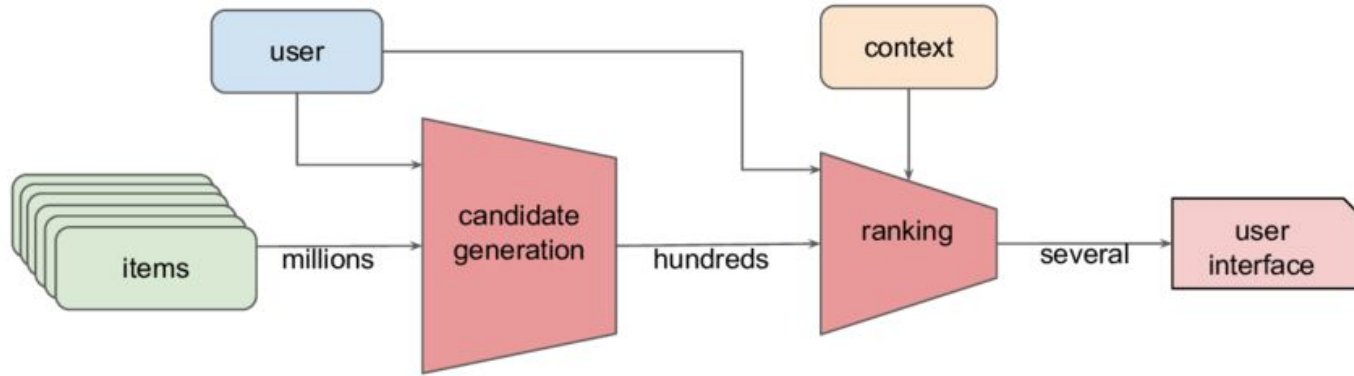
$$\| \mathcal{A}_0 - \mathcal{R} \|_F^2 \rightarrow \min$$

$$\mathcal{R} = \mathcal{G} \times_1 U \times_2 V \times_3 W$$

Why Two-stage Recommender system?

Many real-world recommender systems need to be highly scalable: matching millions of items with billions of users, with milliseconds latency. The scalability requirement has led to widely used two-stage recommender systems, consisting of efficient candidate generation model(s) in the first stage and a more powerful ranking model in the second stage.

What is this?



This is Problems, Always

Big data

Low metrics

Validation leakage

No open-source
evfro - exception

Our plan

1. Honestly split the data
2. Train and test 1st stage models
3. Make some feature generation
4. Train and test 2nd stage models
5. Tune hyperparameters for combined model

YouTube

"была ли у вас какая-то тактика с самого начала , которой вы придерживались" ?

"была ли у вас какая-то тактика с самого начала , которой вы придерживались" ?

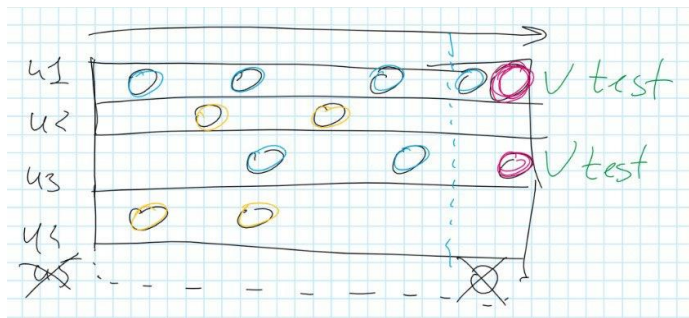


Data preprocessing

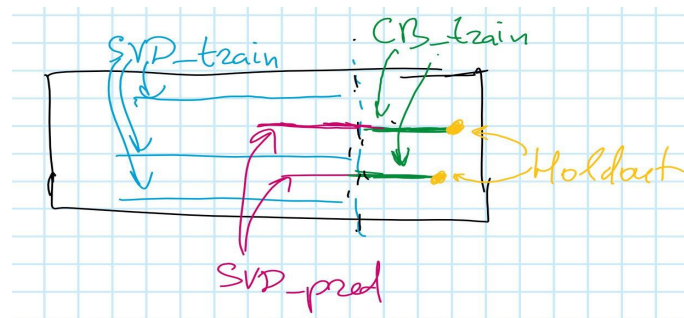
1. First split: train - 95% and test - 5%
2. Remove test users from train
3. Split train on stage1_train, stage2_predict, stage2_train, stage2_holdout
4. Add generated features to stage2_train
5. Finally, split test on test_predict and test_holdout

Data by time: stage1_train, stage2_predict, stage2_train, stage2_holdout, final_test, final_train

train



from this



to this

Feature Engineering

Main concept - Sequential features

Your browser history?

Interactions with objects

SVD-embeddings



Metadata of items

BERT-embeddings

Second-stage model



How to fit?

1. Get candidates from the first-level model
2. Merge with `stage2_train` to generate labels
3. Add generated features
4. Train model

How to predict?

1. Repeat 1, 2, 3 steps from fit for `final_test`, but without labels
2. Score candidates and choose top-10
3. Calculate metrics on `final_holdout`

Results

Baseline models results

model	hit rate	mrr	cov
SVD	0.0841	0.0325	0.2670
EASER	0.0912	0.0365	0.1861

Two-stage model results

model	hit rate	mrr	cov
SVD + CB	0.0859	0.0332	0.2459

Качественные результаты

Разработана схема валидации двухуровневой модели

Проверены следующие гипотезы:

1. Двухуровневая модель лучше одноуровневой
2. Динамические признаки лучше статических
3. Тензорная модель уточняет предсказания

Sources

- Блоги про двухуровневый подход:

<https://habr.com/ru/company/okko/blog/454224/>

<https://habr.com/ru/company/tinkoff/blog/454818/>

<https://habr.com/ru/company/avito/blog/439206/>

- Литература по моделям:

<https://dl.acm.org/doi/10.1145/1864708.1864721>

<https://arxiv.org/abs/1607.04228>

- Генерация фичей:

<https://habr.com/ru/post/447376/>

https://github.com/aprotopopov/retailhero_recommender

https://github.com/mike-chesnokov/x5_retailhero_2020_recs

<https://arxiv.org/abs/1610.04850>

- Примеры:

https://github.com/evfro/recsys19_hybridsvd/tree/master/data

<https://github.com/skoltech-ai/Recommender-Systems-Intro-Sber-2022/blob/main/Evaluation.ipynb>

https://github.com/sharthZ23/your-second-recsys/blob/master/lecture_5/tutorial_hybrid_model.ipynb/