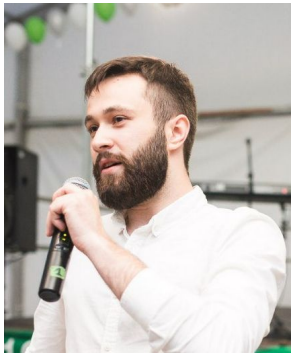
Your First RecSys

Даниил Потапов
Руководитель группы персонализации и рекомендательных систем
МТС BigData



“Куда” и “как” копать дальше

План

- Продвинутые метрики
- Approximate nearest neighbors
- Нейронные сети

Проверка качества

Два этапа тестирования:

- Offline - на ретро данных
  - Метрики: Precision@K, MAP, MRR, NDCG...
- Online - в реальном мире
  - Метрики: Conversion rate, CTR, Retention...

Основная цель - подобрать и оптимизировать такие offline метрики, которые улучшат online метрики.

Продвинутые метрики

- Coverage - покрытие
- Diversity - разнообразие
- Novelty - новизна
- Serendipity - прозорливость

Продвинутые метрики

Coverage

- Сколько наших объектов в итоге попадает в рекомендации?
- Сколько наших пользователей в итоге могут получить рекомендации?

Как считать:

- Построить рекомендации для всей базы
- Посчитать, сколько уникальных объектов попало в наши рекомендации
- Посчитать соотношение рекомендованных уникальных объектов к кол-ву объектов во всей базе

Продвинутые метрики

Diversity

- Сколько категорий объектов представлено в рекомендациях?
- Сколько разных жанров/актеров/режиссеров в рекомендациях для одного человека?

Как считать - так же как и Coverage, только с усреднением по пользователям и в разрезе фичей объектов.

МТС

Продвинутые метрики

Novelty

- Насколько новы наши рекомендации для пользователя?

$$\text{Novelty}(item) = 1 - \frac{\text{count}(\text{users recommended } item)}{\text{count}(\text{all users})}$$

$$\text{Novelty}(item) = 1 - \frac{\text{count}(\text{users recommended } item)}{\text{count}(\text{users not interacted item})}$$

Продвинутые метрики

Serendipity

- Неожиданность рекомендаций при сохранений релевантности
- Чаще всего основная цель рекомендательной системы

$$\text{Serendipity}(user) = \frac{1}{count(recs)} \sum_{i \in recs} \max(P(user, i) - P(allUsers, i), 0)) * rel(user, i)$$

Что такое хорошо и что такое плохо: метрики для рекомендательных систем

Approximate nearest neighbors

В матричных методах процесс построения рекомендаций - это нахождение для вектора пользователя таких векторов объектов, что их скалярное произведение будет максимально возможным.

Если данных много, то такой поиск может занимать слишком много времени. Этот момент можно ускорить и с помощью методов поиск ближайших соседей.

Approximate nearest neighbors

Базовая схема использования:

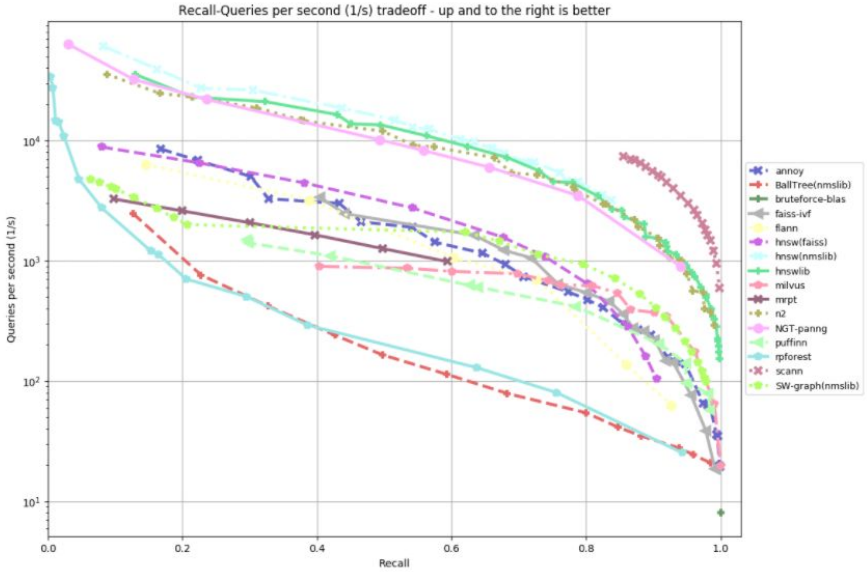
- Построить матричную модель и получить вектора для объектов
- На основе векторов построить индекс в inner product space
- Для каждого пользователя по его вектору получаем из индекса топ объектов с наиболее большим скалярным произведением

Approximate nearest neighbors

Evaluated

- Annoy
- FLANN
- scikit-learn: LSHForest, KDTree, BallTree
- PANNS
- NearPy
- KGraph
- NMSLIB (Non-Metric Space Library): SWGraph, HNSW, BallTree, MPLSH
- hnswlib (a part of nmslib project)
- RPForest
- FAISS
- DolphinnPy
- Datasketch
- PyNNDescent
- MRPT
- NGT: ONNG, PANNG
- SPTAG
- PUFFINN
- N2
- ScaNN

glove-100-angular



https://github.com/erikbern/ann-benchmarks

МТС

Approximate nearest neighbors

Основные три либы:

- faiss - cpu/gpu, quantization, memory mapping
- annoy - cpu, memory mapping
- nmslib - cpu, sparse data, many metrics

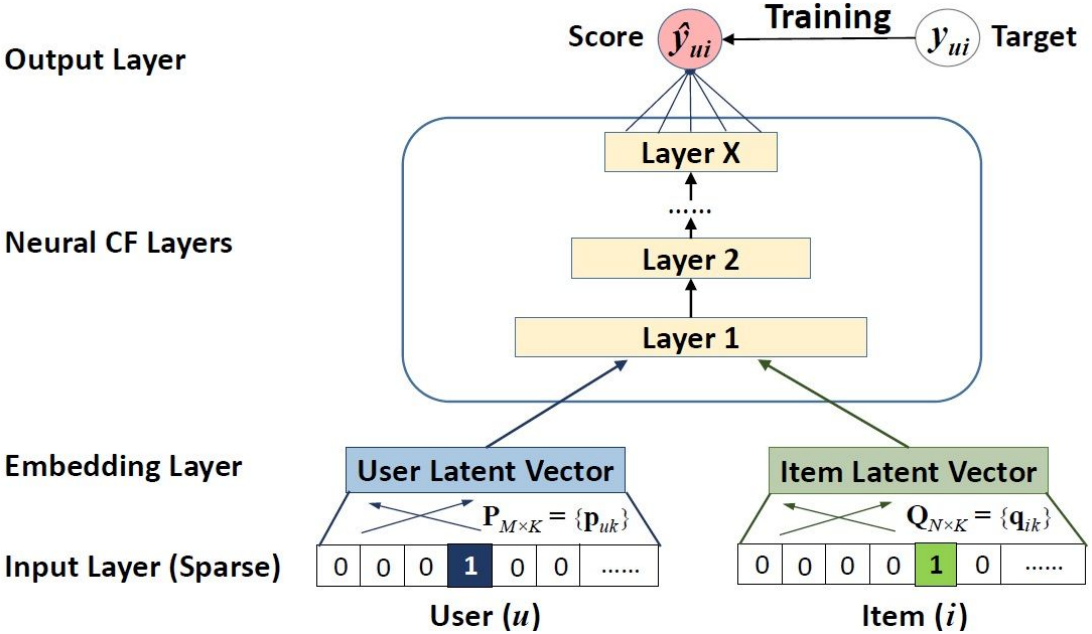
МТС

Approximate nearest neighbors

Как выбрать либу:

- Если векторов порядка 10^5 - sklearn.neighbors.KDTree
- 10^6 - 10^8 - можно подбирать из faiss, annoy, nmslib
- 10^9 и выше - только faiss

Нейронные сети



Neural Collaborative Filtering

Нейронные сети

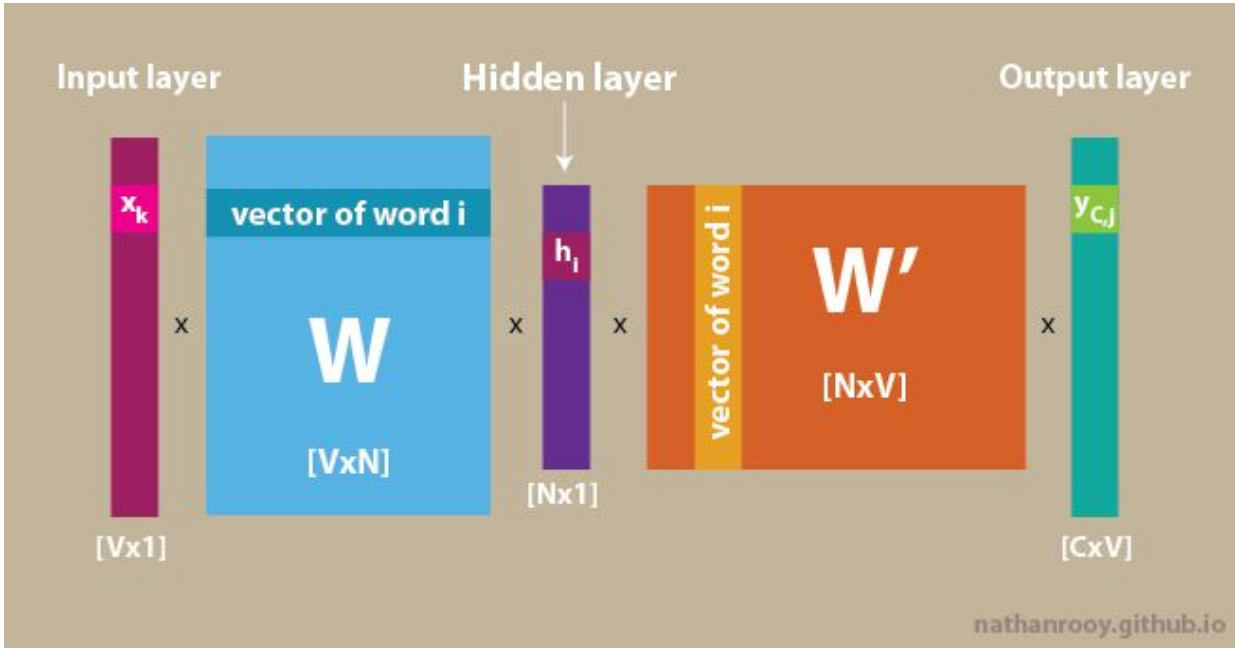
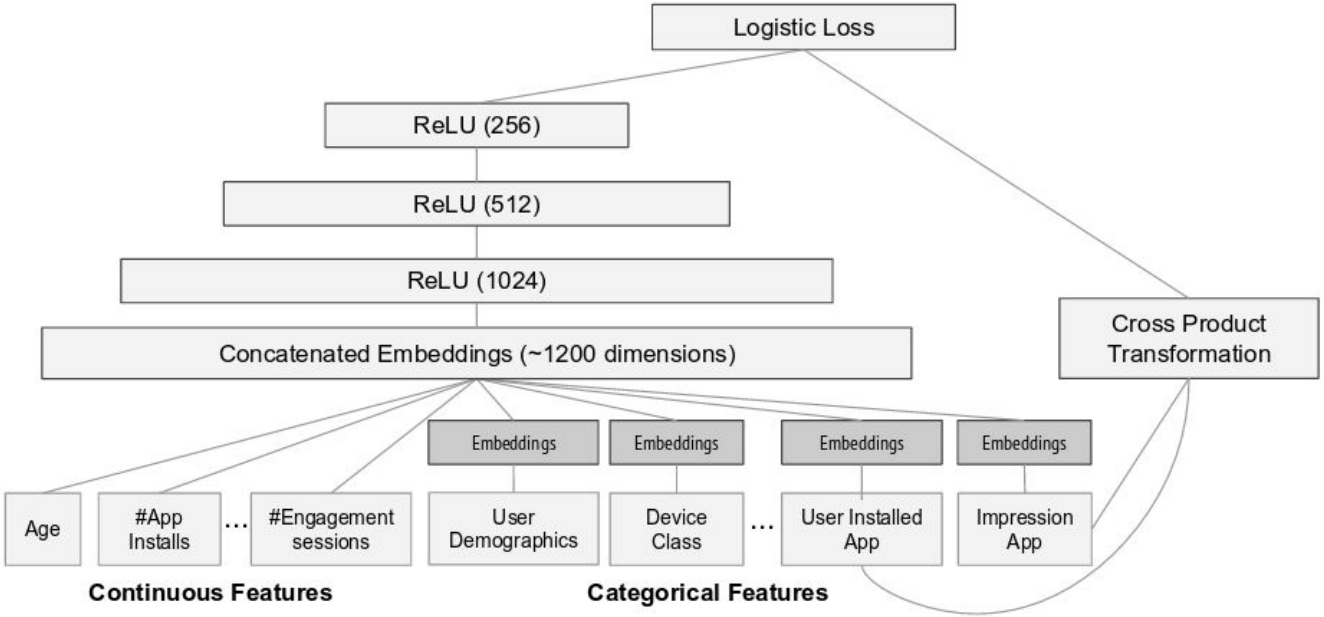


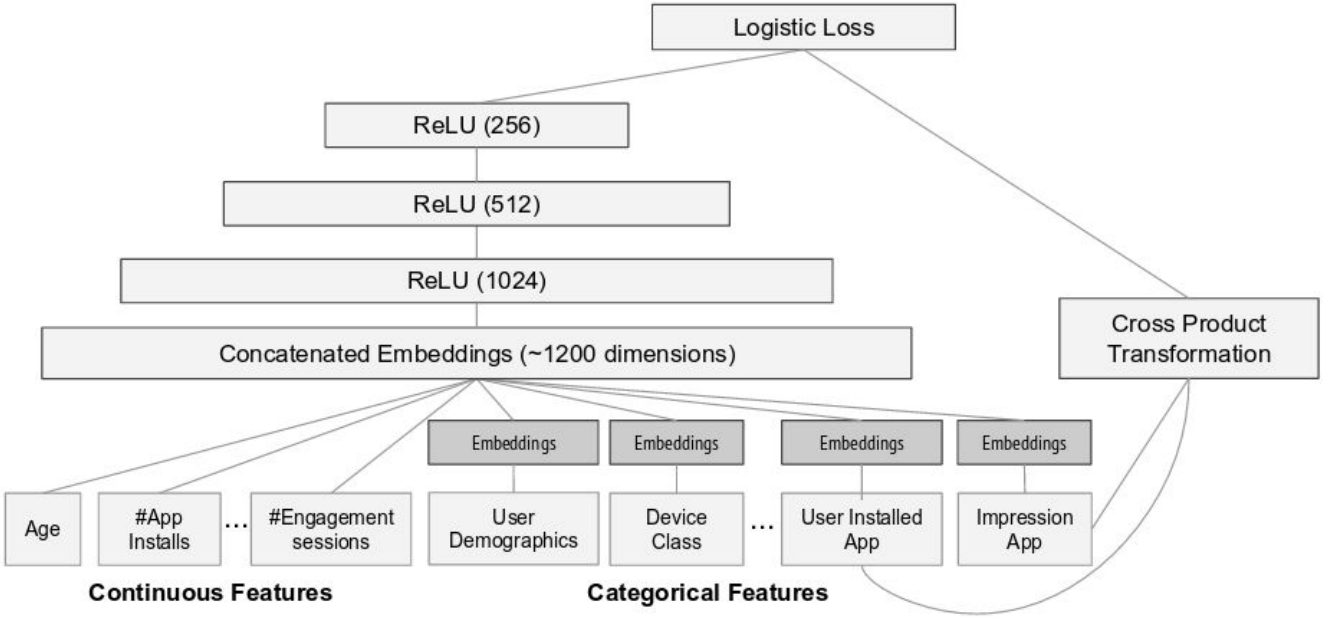
Image credit

Нейронные сети



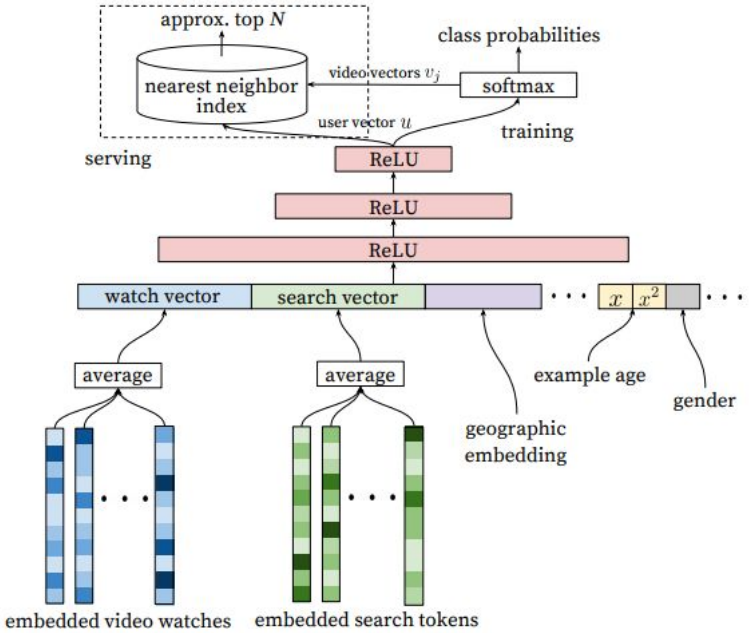
Wide & Deep Learning for Recommender Systems

Нейронные сети



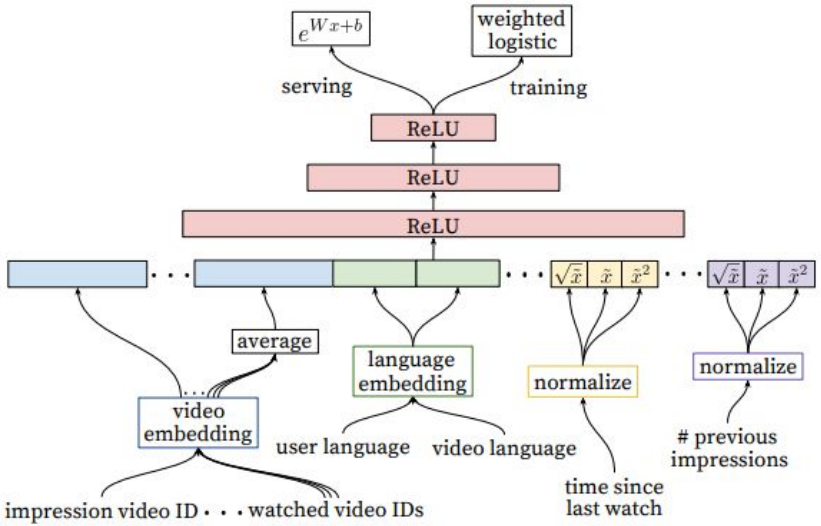
Wide & Deep Learning for Recommender Systems

Нейронные сети. Youtube



Deep Neural Networks for YouTube Recommendations

Нейронные сети. Youtube



[Deep Neural Networks for YouTube Recommendations]

Инструментарий

Фреймворки:

- https://github.com/maciejkula/spotlight
- https://github.com/wubinzzu/NeuRec
- https://github.com/PreferredAI/cornac
- https://github.com/ylongqi/openrec

RL:

- https://github.com/google-research/recsim
- https://github.com/criteo-research/reco-gym

RecSys на Java

- https://github.com/guoguibing/librec

И даже на C++

- https://github.com/cnclabs/smore

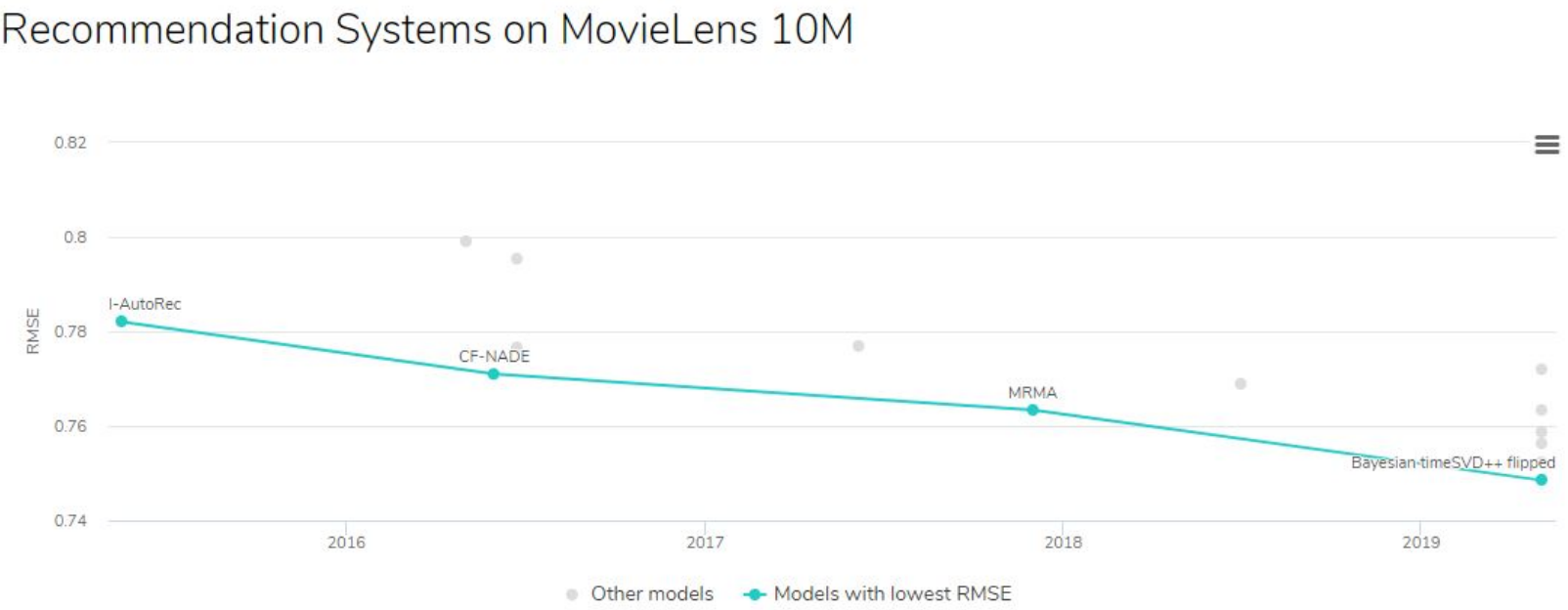
Или Go

- https://github.com/zhenghaoz/gorse

Awesome recsys:

- https://github.com/hongleizhang/RSPapers
- https://github.com/chihming/competitive-recsys

Papers with code



Collaborative Filtering on movielens-10m

Всем спасибо за внимание :)

Вопросы можно задавать здесь

 Telegram чате курса

 sharthZ23