clustering:

https://www.kaggle.com/code/cdeotte/recommend-items-purchased-together-0-021

https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations/discussion/316903

more

Recall methods:

H&M Personalized Fashion Recommendations
kaggle.com

poteman TOPIC AUTHOR
Posted a year ago · 16th in this Competition

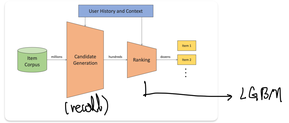
update
price_popular: calculate the mean price of user, and get the most popular item in every price bin,
for every user, find its bin and recommend most popular items in this bin.

A customer mean price = $75

Products $ 10 - 80

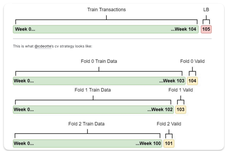
1. X
2. Y
3. Z

https://github.com/xei/recommender-system-tutorial



H&M Personalized Fashion Recommendations
kaggle.com

CV:



MMR Risk for ranking model (n_candidates x n_features)

Pawel Jankiewicz
Posted a year ago · 2nd in this Competition

Good summary. In my case my train features are generated before 2020-04-01 and move week by week until 2020-09-22. I noticed that the ranking model doesn't need much data. For example to train the ranking model I use only labels generated from last 5 months. I think there are some gains to be made when using more data for training but I would need to invest to computation. It is too easy to just add RAM much harder to optimize the code and one's thinking.

MMR Trick:

H&M Personalized Fashion Recommendations
kaggle.com

https://www.kaggle.com/competitions/otto-recommender-system/discussion/372976

Winning solutions summary from H&M Recommendations comp

OTTO – Multi-Objective Recommender System
kaggle.com

specials:

1st place: logistic CLT & candidates

feature engineering, candidates generation → 2nd: lng similarity - mobilenet

3rd:

Samples

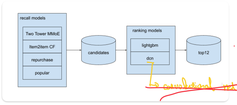
- The generated candidates which the customer didn't purchase in the target week are labeled with 0 and those which the customer purchased are labeled with 1.
- Because of big imbalance (sorry-my-is about 1:1000, downsampling is adopted. I just keep the negative samples with amount of 20*purchase_weekcount.
- The pos samples are only the ones which are recalled in the candidates and the other items that the user purchased in actual are not included.

negative down-sampling exploration to deal with the imbalance

similarity (user-item) features added

BPR Matrix factorization

4th:



~~to simulate real network~~

deep & cross network

Two Tower MMoE

↳ demographic features to handle cold start also

(LGBM single model) 5th: focus on recall - embedding (using transformer) to find similar articles and customers

users

items: text, image (EfficientNet)

word2vec

6th: gap feature old item - new item

product type, department

categorical features

model	objective	average num of candidates for each user	cv map@12
catboost	binary	120	0.0403
lightgbm	binary	120	0.0395
catboost	binary	220	0.0402
lightgbm	binary	220	0.0396
lightgbm	lambdarank	220	0.0400
lightgbm	lambdarank	1000	0.0381

The result of our best cv is the best one on PB. Trust your local cv is really importance in kaggle.

H&M Personalized Fashion Recommendations
kaggle.com

common questions: 1. Negative down-sampling

2. **1. For each customer generate candidates that they can buy.**

These candidates are your negative and positive examples. I'm saying this because someone on the forum asked where are the negative observations. And the answer is you generate them.

The strategies to generate those candidates are probably the most important aspect of this competition. The strategies can be based on:

- item to item similarities between customer previous baskets - so image similarity is not out of the question as somebody suggested
- user based collaborative filtering
- last baskets (with the hope that the user will buy them again)
- model based predictions with customers without the transaction history
- etc

based on what?

Ranking → LGBM

3. Sampling
   Test items