Pretrained models for creating recommendation systems

...

By Max Kmet and Denys Ivanenko

Project Idea

Business Problem:

1. Small amount of data to create an effective recommender systems
2. Expensive creation of a recommender system from scratch

Business Solution:

1. Use cheaper recommender systems, pretrained on large datasets from the same domain

Project Goals

- Create framework for designing and implementing Recommendation Systems in various domains.
- Research usage of pretrained models for Recommendation Systems
- Measure performance of recommender implemented with our framework against State of the art solutions with respect to common metrics.
- Implement convenient way to apply knowledge from another domain to main data.
- Make framework suitable for implementing aggregated ML approaches

Problems we solve in our research:

1. Matching of items and users.
2. Aggregating recommendations from two or more recommenders .
3. Metrics Balancing.
4. Absence of pretrained models, that can be used for creating. recommendation systems with low amount of data.

Solution:

1. Matching algorithm based on own heuristics for users and items.
2. Ranking model, that will take recommendations from all recommenders and rank them in the relevance order.
3. Metrics Prioritizer to fit the model to the needs of domain.
4. Training, storing and shipping pretrained models for stated needs.

DATA

In order to validate our solution, we will focus on one domain - movies.

So the datasets we will choose are:

- Netflix Prize Data
- MovieLens 100K
- IMDb movies
- Anime Recommendations Database

Related Works

- **Transfer Meets Hybrid: A Synthetic Approach for Cross-Domain Collaborative Filtering with Text**
  - https://arxiv.org/pdf/1901.07199.pdf
- **Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems**
  - https://www.microsoft.com/en-us/research/uploads/prod/2019/07/pp004-lian.pdf
- Transfer Learning for Content-Based Recommender Systems Using Tree Matching
  - https://www.ise.bgu.ac.il/faculty/liorr/Naseem.pdf
- Collaborative Recurrent Neural Networks for Dynamic Recommender Systems
  - http://proceedings.mlr.press/v63/ko101.pdf
- Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations
  - https://arxiv.org/pdf/1905.01997.pdf

Competitors

- **Surprise**
  - Surprise is Python scikit for building and analyzing recommender systems that deal with explicit rating data
  - **http://surpriselib.com**
  - Flaws:
    - No pretrained models supplied
    - Small set of metrics
    - Small set of methods
- **Lenskit**
  - Python tools for recommender experiments
  - https://lenskit.org
  - Flaws:
    - No pretrained models supplied
    - Small set of metrics
- LightFM
  - Flaws:
    - No pretrained models supplied
    - Only one model available for training
  - https://making.lyst.com/lightfm/docs/home.html

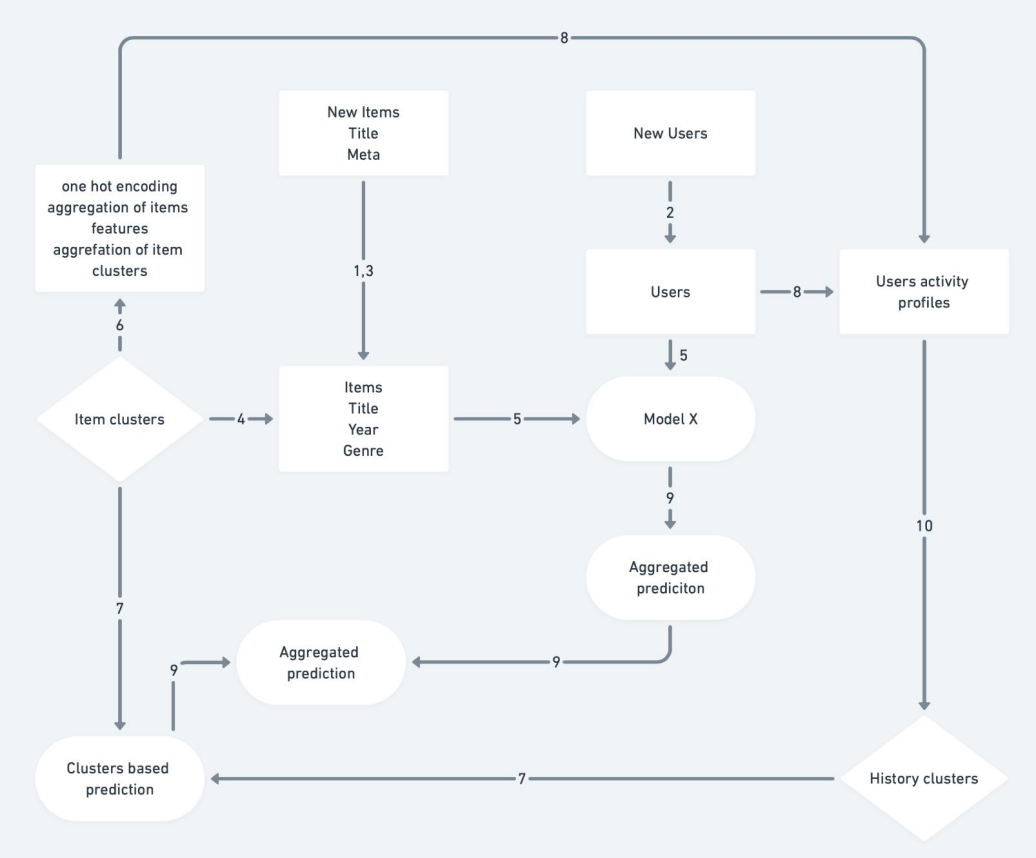
Baseline

Baseline: SVD from surprise package trained on ML100k dataset. Our model, pretrained on Netflix Prize in combination with data from ML100K data, should give better metric on ML10M.

Also we intend to create a method that helps to select the best model based on the most important metrics for specific domain.

Metrics we consider to use are explained in evaluation block.

Architecture



Model X in the Scheme is the model, that is used alongside with clustering. We plan to try different models as Model X: Collaborative filtering, Content-Based filtering, etc.

We have identified three stages of our solution:

- Pretraining
- Model update
- Model usage

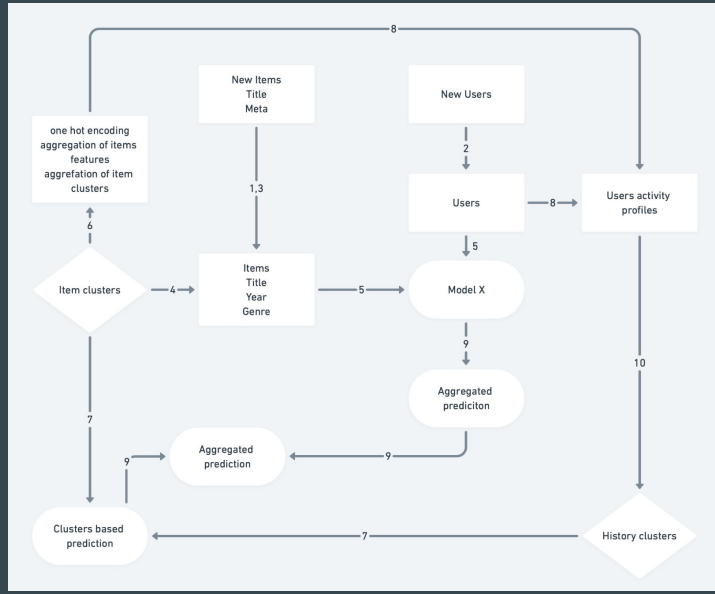
Model update stage should be run periodically.

Following slides explain stated stages with Collaborative Filtering taken as Model X.

Pretraining and model update

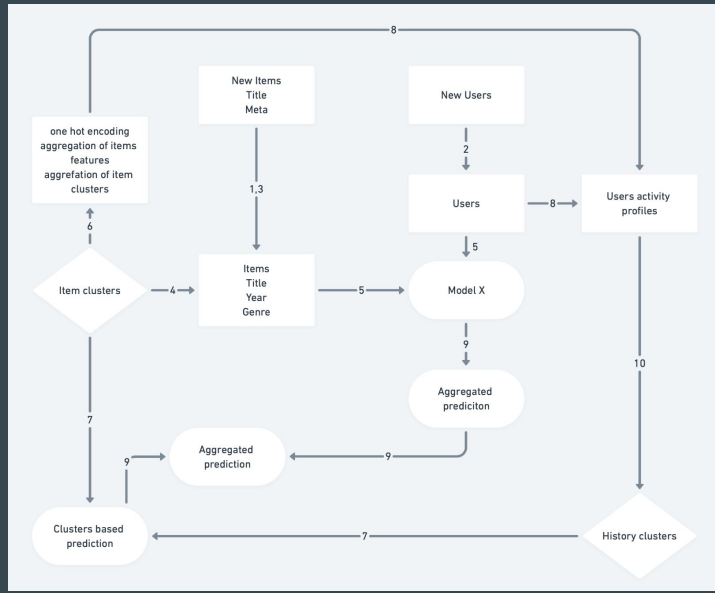
Pretraining:
4 - Create clusters of items based on metadata and most common occurrence in users clusters likings
8 - Aggregate data from user activity in our system and create a user profile
10 - Create clusters of users based on their likings vectors on a large dataset (Netflix Data)

Model update:
8 - Aggregate data from user activity in our system and create a user profile
3 - Add new films



Model Usage

Model usage:
1 - Match items from one dataset to items
to items in other (large one used for pretraining) dataset based on metadata
2 - Match users from one dataset to most similar users
in the large dataset based on their likings vectors
3 - Match new unknown items from the new dataset to corresponding item cluster based on metadata
5 - Perform collaborative filtering on matched data and make predictions
7 - Use our knowledge about user cluster to item clusters matching and make another prediction
9 - Combine predictions from 5 and 7



Evaluation

Metrics

Given list of recommendations:

- MAP@K and MAR@K (Mean Average Precision and Recall at Cutoff k)
- Personalization (1- cosine similarity between user's lists of recommendations)

Outcome

- As the result of our project we will present production-ready recommendation method with the best evaluation metrics for chosen domain
- Method will benefit from pretrained model
- Method will be accepted if it beats baseline in all evaluation metrics

Planned Components

1. CollaborativeFiltering
    a. Memory Based (basic)
    b. Model Based (ensamble based on further components)
2. ItemsVectorizer
    a. description2vector
    b. image2vector
3. UsersSegmenter
    a. predefined segments
    b. generate segments
4. DomainKnowledgeApplier
5. Abstract2Item
6. Metrics prioritizer
7. And much more...

Problems to think over

1) Change of taste over time + temporary interests
2) Recognizing rising trends among all people and sub groups
3) Check how trend influences particular individuals
4) Methods of handling cold start