[PMLDL] D1.1 Report

Team

Team name: *16-th Data Science Division*
Team members:
- Bulat Sharipov (DS-01); b.sharipov@innopolis.university
- Dinar Yakupov (DS-01); d.yakupov@innopolis.university
- Danil Fathutdinov (DS-01); d.fathutdinov@innopolis.university

Project Intro

Topic: Creating a Recommendation Systems using Graph Neural Networks and Yambda Dataset. We will basically create a web-service that will recommend you next song to listen to, given your previous preferences.
Importance: During the project execution team aims at developing skills and experience in recommendation systems, as well as understanding the fundamental theory behind it and common approaches in this field. We believe that recommendation systems are one of the most valuable and profitable applications of Machine Learning in Business. To incorporate more new skills and fields into this project, we will try to build our system using Graph Neural Networks. *Currently, **there are no** papers, projects, or benchmarks on using Graph Neural Networks on Yambda Dataset.* Therefore, we will be pioneers in exploring this promising field. We will also contribute to the ML community by providing Medium-like article about our project and our results.
Target users: People who like to listen to music and want to explore more songs. ML Engineers and companies will also benefit from our project, as we will be covering the gap of application of Graph Neural Networks on Yambda dataset.

Github

Link

Dataset

During the project our main focus will be on Yambda Dataset. This is a huge industrial-scale dataset that allows developing and analysing recommendation systems. The dataset was conducted from the Yandex.Music application, and covers 1.000.000 users and 5.000.000.000 interactions. There are several types of possible interactions
- Listens

- Likes
- Dislikes
- Unlikes
- Undislikes

We explore the dataset in more detail in our repository.
[Dataset link]

Current Progress

Exploratory Data Analysis

We explored 50 million version of the dataset using python data analysis and visualization tools. We calculate basic statistics - number of unique users, items, and interactions. We also plotted several distributions, to understand the nature of the underlying data. Finally, we sampled a small number of users from the dataset and have built and plotted a small bipartite graph, representing the relationships in this dataset. You can look at more details by accessing the data exploration .ipynb file (Path: Exploratory_Data_Analysis/Yambda_EDA.ipynb).

Baseline models

We benchmarked MostPop, BPR, and NeuMF on a randomly sampled 10% of the Yambda 50m split (≈ 5,000,000 interactions) using the RecBole Python package. We chose these three to cover complementary families: a non-personalized popularity baseline (MostPop), a classic pairwise matrix-factorization ranking model (BPR), and a neural collaborative filtering baseline (NeuMF). We ran tests with a fixed random seed for reproducibility, and all evaluation metrics were collected and saved to recbole_results.csv.

Model	Recall@10	MRR@10	NDCG@10	Hit@10	Precision@10
Pop	0.0138	0.1133	0.0504	0.2474	0.0441
BPR	0.0150	0.1131	0.0509	0.2595	0.0452
NeuMF	0.0177	0.1708	0.0811	0.3590	0.0735

We attempted to include ItemKNN and LightGCN but deferred them because they were impractically heavy under the current sampling/config: ItemKNN's and LightGCN's benchmarking caused long initialization times.

Other Solutions and Competitors

We will orient on the public solutions that show the best result on Yambda Dataset at this moment. These solutions are composed of only the baseline models that were provided by researchers from Yandex with the dataset, since the dataset was released, no more efficient models have been published. These solutions include itemKNN, DecayPop, and SasRec.

Item-Based k-Nearest Neighbors (ItemKnn)

Classical Item-based collaborative filtering algorithm. Usually, the cosine similarity or adjusted cosine are used as measure of similarity between items. The prediction of the rating for the item from the user is just a weighted sum across the similarity measures of k most similar neighboring items the user interacted with. The results of the method on Yambda-50M Dataset:

- within setup “Listen+”: NDCG@10 = 0.0781, Recall@10 = 0.0373
- within setup “Like”: NDCG@10 = 0.0125, Recall@10 = 0.0199

The method was published in the paper “Item-Based Collaborative Filtering Recommendation Algorithms”: https://www.ra.ethz.ch/cdstore/www10/papers/pdf/p519.pdf

Decay-weighted Popularity (DecayPop)

The effective non-personalized recommendation algorithm. Items are ranked by weighted popularity in the recent time window, where the more recent the time segment, the more the weight. The result of the method on Yambda-50M Dataset:

- Within setup “Listen+”: NDCG@10 = 0.0260, Recall@10 = 0.0122
- Within setup “Like”: NDCG@10 = 0.0180, Recall@10 = 0.0333

The method was introduced in the paper “A Re-visit of the Popularity Baseline in Recommender Systems”: https://arxiv.org/pdf/2005.13829

Self-Attentive Sequential Recommendations (SasRec)

The effective model of sequential recommendations based on self-attention blocks. The number of blocks and additional components can differ for each implementation, the loss function is usually binary cross-entropy. The result of the method on Yambda-50M Dataset:

- Within setup “Listen+”: NDCG@10 = 0.0744, Recall@10 = 0.0322
- Within setup “Like”: NDCG@10 = 0.0103, Recall@10 = 0.0208

The method was introduced in the paper “Self-Attentive Sequential Recommentation”: https://arxiv.org/pdf/1808.09781

Project Success Criteria

Graph neural networks (GNN) are very powerful models in recommendation systems and we believe that research in this direction can imply the GNN model with best relevance metrics NDCG@k and Recall@k among the results of the baseline models on Yambda Dataset.

Our team has bounded computing resources for GNN model training and hyper-parameter tuning. Based on this, we are aiming to build the GNN model that can achieve results comparable with the results of the described baseline models on Yambda-50M Dataset:

- Within setup “Listen+”:
  - NDCG@10 > 0.07, NDCG@100 > 0.075
  - Recall@10 > 0.03, Recall@100 > 0.1
- Within setup “Like”:
  - NDCG@10 > 0.01, NDCG@100 > 0.02
  - Recall@10 > 0.02, Recall@100 > 0.06

Work Distribution

- Bulat Sharipov: Writing readme and report, doing exploratory data analysis
- Dinar Yakupov: Trying baseline models
- Danil Fathutdinov: Researching on current solutions and defining success criteria