Assignment 5 – Advanced Recommender Systems

Advanced Data Mining

Task 1 [2 points]

1. Download and upload data from the category Books: https://amazon-reviews-2023.github.io/. Process the reviews to a pd.DataFrame with columns `['user_id', 'item_id', 'rating', 'timestamp']`.
2. Consider the following train-test split procedures. What are their advantages and disadvantages?
   - Randomly select 20% of all interactions.
   - Select 20% interactions for each user at random.
   - Select the most recent 20% interactions for each user.
   - Select a fixed number of the most recent interactions for each user so as to obtain around 20% of the data.
   - Select the most recent 20% of all interactions.
3. Provide plots for a basic data analysis:
   (a) number of interactions for each item,
   (b) distribution of users' number of interactions,
   (c) other plots which you find interesting.
4. Perform a train-test split by selecting 2 most recent items for each user for testing.

Task 2 [10 points]

1. Provide an implementation (suggested using PyTorch) of:
   - the Matrix Factorization model,
   - the LightGCN model (you can base on https://github.com/gusye1234/LightGCN-PyTorch/tree/master).
2. Enable using following loss functions:
   - MSE (like in Probabilistic MF):

$$\mathcal{L}_{MSE} = \sum_{(u,i)} \mathbb{1}_{ui} \left(\frac{r_{ui} - 1}{R_{max} - 1} - \sigma(\mathbf{p_u} \times \mathbf{q_i}) \right)^2 + \lambda_U \sum_u \|\mathbf{p_u}\|^2 + \lambda_I \sum_i \|\mathbf{q_i}\|^2$$

- BPR:

$$\mathcal{L}_{BPR} = \sum_{(u,i,j)} \ln \sigma(\mathbf{p_u} \times \mathbf{q_i} - \mathbf{p_u} \times \mathbf{q_j}) - \lambda_\Theta \|\Theta\|^2$$

- Alignment & Uniformity:

$$\mathcal{L}_{AU} = \mathbb{E}_{(u,i)} \|\mathbf{p_u} \times \mathbf{q_i}\|^2 + \lambda \left(\log \mathbb{E}_{(u,v)} \frac{1}{2} e^{-2\|\mathbf{p_u}-\mathbf{p_v}\|^2} + \log \mathbb{E}_{(i,j)} \frac{1}{2} e^{-2\|\mathbf{q_i}-\mathbf{q_j}\|^2} \right)$$

3. Compare their performance in terms of:
   - convergence speed,
   - recall@20,
   - hit-rate@20,
   - Normalized Discounted Cumulative Gain @20,
   - Mean Reciprocal Rank @20.

   For a fair comparison, use the same embedding dimensionality for both models.

Task 3 - bonus [8 points]

Train a TAGNN model (you can base on https://github.com/CRIPAC-DIG/TAGNN) on the Diginetica dataset (https://competitions.codalab.org/competitions/11161#learn_the_details-data2). Propose a methodology for evaluating the popularity bias in your recommendations and a way to mitigate it. How does it influence the accuracy of recommendations?