JOHANNES KEPLER UNIVERSITÄT LINZ
Institute of Computational Perception
MULTIMEDIA MINING AND SEARCH GROUP

UE Learning from User-generated Data S2024
MRS Challenge

Oleg Lesota (oleg.lesota@jku.at)

Agenda

- Evaluate the course
- MRS challenge
- Implementation ideas, guidelines


EPIC
MUSIC
RECOMMENDER
SYSTEMS
CHALLENGE
@santaepphotography

The Challenge

- Recommend songs to users of Last FM
- Goal: maximize overall nDCG

Data: LFM2B

- Bigger (~8 times more items) LFM2b sample
- Contents:
  - lfm-challenge.user — id, country, age, gender, registration date
  - lfm-challenge.item — id, artist, track, country of origin
  - lfm-challenge.inter_train — user id, item id, listening events
  - lfm-challenge.inter_test — user id, item id, listening events
  - lfm-challenge.musicnn — item features
  - test_indices.txt — ids of test users

Submission – two files

- rec_<matr.num.>_<name>.tsv – the file with your recommendations, where each line corresponds to one target user:

<user_id>\t<item_id>,<item_id>,…

10 **not seen** items for each user (no spaces between), example:

rec_k00000007_Bond_James.**tsv**	
0	0,1,2,3,4,5,6,7,8,9
7	15,1,2,3,4,0,6,8,9,804
9	4,15,10,2,3,5,7,8,9,190
…	

- report _<matr.num.> _<name>.txt – text file describing your approach to the recommendation part of the task: algorithms used, training settings, … No more than ½ A4; Specify your name and matr. number within as well

Evaluation

- The results will be judged by the **overall nDCG** performance and the report
- The exercise is worth **10 points** with up to 10 bonus points available
  - Confidently beat POP-recommender – 6 points
  - ItemKNN performance – 8 points
  - Standard report – 2 points
  - Additional points for better nDCG, interesting and sound methodology, deep analysis of the result, high quality reports

Timing

- Challenge publication: 5.06.24
- Deadline: 25.06.2023
- Occasional (~weekly) score board updates
- Possible Sync/QA meeting(s)

Questions so far?


HOW?
@santaepphotography

Establish experimental framework

1. Create additional data splits train-(val)-test (at least 3 in total)
2. Setup the nDCG evaluation pipeline
3. Produce & evaluate random recommendations
4. Produce & evaluate POP recommendations
5. Begin experiments...

Experiments

For each investigated model:

1. Perform hyper-parameter search
   (try out different embedding sizes, number of neighbors etc.)
2. Evaluate nDCG of each configuration of each model
   on all 3+ data splits. Compare the average to the two baselines
3. Document all experiments:
   motivation, hyper-parameters, results. Save results produced by costly methods

Approaches

1. Try approaches from the UE (some will require optimization)
2. Look up hints in literature
3. Investigate performance of the models, find weak spots. E.g. what users get the lowest nDCG? What can be done for them?
4. Hybridize, use content, try different things, have fun.
5. Keep your experiments logical and clearly motivated.
6. Use recommendation frameworks at your own risk. Make sure you understand approaches you use (every report will be checked).

Report

1. Keep it informative but concise
2. Describe your experimental setup
3. Explain tried approaches and motivation of moving from one to another
4. Report performance of all systems including the baselines
5. The detail should be enough to reproduce the results. Omit code and standard implementation details
6. Use LLMs at your own risk – uninformative reports will be penalized.

Ideas

1. Souped up POP recommender
2. ItemKNN
3. Explore Matrix Factorization
4. Explore more approaches

Extended POP recommender

Demographic filtering:

Hypotheses:

- Users with similar demographics (age, country, gender) have similar tastes
- Users enjoy a certain portion of their local music

Filtering by preference:

- Use more sophisticated ways to cluster users
- Users of each cluster have their own distinct top-pop

Additional data:

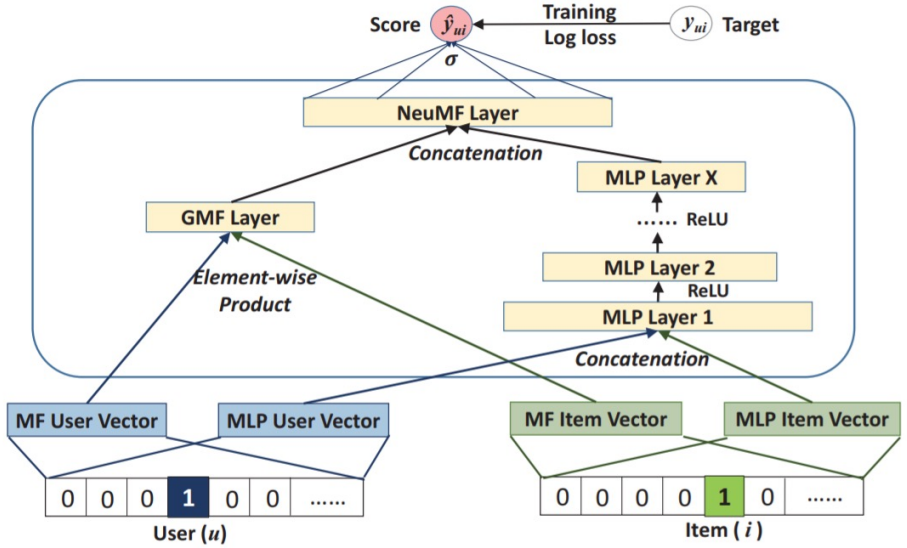
- Assign genres to tracks, make POP recommender aware of user preference genre-wise

Item KNN

- Solid recommender for LFM-2B
- Will require optimization for the new dataset
- Find publication introducing Item KNN
- Check optimization suggestions

Matrix Factorization

- Optimize our implementations
- Check published approaches (SVD, ALS-MF, BPR-MF, ...)
- ~~Neural Matrix Factorization~~
- Shallow autoencoders
- Content: Factorization machines, Inductive Matrix Factorization



Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17).

In a nutshell

- Start small (first try simplest approaches, get your experimental framework going)
- Tune hyper-parameters (N of neighbors, embedding sizes, ...)
- Compare your implementations to each other and baselines (e.g. POP, random recommender, different versions of your system)
- Use multiple train/test splits to make sure one system consistently outperforms the other (we only supply one split)
- Once you figured out your optimal parameters feel free to train your system on the whole data (train + test) and then produce final recommendation
- Optimize, use Google Colab or sample the data if your machine struggles with it
- Follow the submission format closely and don't forget the report!

Thank you & good luck!