



Machine Learning with Graphs (MLG)

HW3: Model Design & Comparison for Recommendation Models

Deadline: 2023.05.14 (Sun.) 23:59

Submission: Code (.py/.ipynb) and Report (PDF)

The Main Expected Performance Table

	MovieLens			Yelp			Douban Book		
	RMSE	Recall@10	NDCG@10	RMSE	Recall@10	NDCG@10	RMSE	Recall@10	NDCG@10
UCF-s									
UCF-p									
ICF-s									
ICF-p									
MF									
FM									
BPR-MF									
BPR-FM									
GBDT+LR									
XGB+LR									
FNN									
IPNN									
OPNN									
PIN									
CCPM									
NeuMF									
WD									
DeepCross									
NFM									
DeepFM									
5選3 (自行加rows)									
Your own NN model									

10 Typical RecSys Methods

10 NN-based RecSys Methods

3 Recent NN-based Methods

你自己設計的NN方法

RecSys Models to be Compared

10 Typical Approaches

- 1) User-based CF [**UCF-s**] (cosine as similarity)
- 2) User-based CF [**UCF-p**] (Pearson correlation as similarity)
- 3) Item-based CF [**ICF-s**] (cosine as similarity)
- 4) Item-based CF [**ICF-p**] (Pearson correlation as similarity)
- 5) Matrix Factorization [**MF**] (varying k-dim latent factor)
- 6) Factorization Machine [**FM**] (varying k-dim latent factor)
- 7) Matrix Factorization with BPR [**BPR-MF**]
- 8) Factorization Machine with BPR [**BPR-FM**]
- 9) Pre-training via GBDT for LR [**GBDT-LR**]
- 10) Pre-training via XGBoost for LR [**XGB-LR**]

Lecture 6

Lecture 7

RecSys Models to be Compared

- **10** NN-based approaches

- 1) FM-supported Neural Networks [**FNN**]
- 2) Product-based Neural Networks
 - Inner-Product NN [**IPNN**]
 - Outer-Product NN [**OPNN**]
 - Product-network in Network [**PIN**]
- 3) Convolutional Click Prediction Model [**CCPM**]
- 4) NCF Neural Matrix Factorization [**NeuMF**]
- 5) Wide&Deep [**WD**]
- 6) Deep Crossing [**DeepCross**]
- 7) Neural Factorization Machine [**NFM**]
- 8) Deep Factorization Machine [**DeepFM**]

Lecture 8

RecSys Models to be Compared

Select 3 from 5 Recent NN-based approaches:

- 1) **Attentional Factorization Machines (AFM)**
Learning the Weight of Feature Interactions via Attention Networks (IJCAI 2017)
- 2) **Collaborative Memory Networks (CMN)**
Collaborative Memory Network for Recommendation Systems (SIGIR 2018)
- 3) **xDeepFM**
Combining Explicit and Implicit Feature Interactions for RecSys (KDD 2018)
- 4) **Deep Interest Network (DIN)**
Deep Interest Network for Click-Through Rate Prediction (AAAI 2019)
- 5) **DeepGBM**
A Deep Learning Framework Distilled by GBDT for Online Prediction Tasks (KDD 2019)

Dataset 1: MovieLens 100K

Entity	#Entity
User	943
Age	8
Occupation	21
Movie	1,682
Genre	18

Relation	#Relation
User - Movie	100,000
User - User (KNN)	47,150
User - Age	943
User - Occupation	943
Movie - Movie (KNN)	82,798
Movie - Genre	2,861

Dataset 2: Yelp

Entity	#Entity
User	16,239
Business	14,284
Compliment	11
Category	511
City	47

Relation	#Relation
User - Business	198,397
User - User	158,590
User - Compliment	76,875
Business - City	14,267
Business - Category	40,009

Dataset 3: Douban Book

Entity	#Entity
User	13,024
Book	22,347
Group	2,936
Location	38
Author	10,805
Publisher	1,815
Year	64

Relation	#Relation
User - Book	792,062
User - Group	1,189,271
User - User	169,150
User - Location	10,592
Book - Author	21,907
Book - Publisher	21,773
Book - Year	21,192

Evaluation Settings

- Evaluation metrics:
(1) **RMSE (real-valued)**, (2) **Recall@10 (binary)**,
(3) **NDCG@10 (binary)**
- **Data filtering: remove users whose #interactions < 3**
- **Data splitting: 5-Fold Cross Evaluation**
 - **Test data: Randomly select 20% user-item interactions**
 - Randomly split the remaining data into training (70%) and validation (10%) sets
 - Validation set is used for hyperparameter tuning
 - **Report the average scores of RMSE, Recall, NDCG over 5 testing sets**
- For Recall & NDCG, transform the ratings into **binary implicit feedback as ground truth**, indicating whether the user has interacted with the specific item
- **[Important!]** Be sure to **fairly** do all comparisons **under the same experimental settings**

Task Requirements

- Q1: Compared with the typical methods, can our NN-based approaches achieve comparable accuracy? Why?
 - Are recent NN-based methods even better? Why?
- Q2: Are there any hyperparameters in each model that significantly affect the performance?
 - You may need to conduct hyperparameter studies for some models, find the best hyperparameters, and explain why such settings are good
- Q3: Can you create a new end-to-end NN that combine the advantages of nicely-performed methods to beat all methods?
 - You **MUST** at least design one novel NN-based method by yourself, and have it compared with all methods
 - No matter you beat them or fail, explain the possible reasons
- Q4: What if I cannot successfully complete **some** (e.g., 8, 7, 2) of 10 typical, 10 NN-based, 3/5 recent methods?
 - Just try your best to have 13 compared methods. Do as many as you can.
 - Your own method is definitely required

Reference Packages (but not limited)

- Surprise <https://github.com/NicolasHug/ Surprise>
- Spotlight <https://github.com/maciejkula/spotlight>
- LightFM <https://github.com/lyst/lightfm/>
- DeepCTR <https://github.com/shenweichen/DeepCTR>
- NeuRec <https://github.com/wubinzzu/NeuRec>
- RecQ <https://github.com/Coder-Yu/RecQ>
- **Bonus:** implement all required models by yourself using deep learning packages like Tensorflow/PyTorch
- We recommend you to **read the original papers of all required models** so that you can understand to come up with your own method, and make systematic and correct comparison
 - At least you can find the key hyperparameters in the papers

Submission

- Report + Code submission via **Moodle**
 - Deadline: **May 14 (Sun) 2023, 23:59**
 - Submit your code: **.py** or **.ipynb** (preferred)
 - Submit report (PDF): **≥10** pages (you cannot include code in report)
- Content in the report
 - **1) Introduction**
 - **2) Methodology:** briefly describe all of the compared methods, and describe the details of your own method
 - **3) Experimental analysis**, along with analysis and insights
 - Report your experimental settings, hyperparameter setting of each method
 - Compare and report the required methods and your own method
 - Explain WHY your prediction is so GOOD or so BAD!
 - Present any insights based on your results
 - Do hyperparameter analysis
 - Refer to the slide “Task Requirements”
 - **4) Conclusions**
 - Explain the **novelty** of your method, summarize your findings
 - Point out how to improve in the future
 - **5) Citations** (if you use any methods or papers)