

Programming HW2

Web Retrieval and Mining Spring 2020

2020/5/22



Introduction

- ❖ In this homework, you are asked to implement **two small Learning-to-rank recommendation models**.
- ❖ We will give you **only** the historical interacted item list of each users, and your task is to find and recommend top 50 most **relevant** items (**except the training positive items**) for them individually.
- ❖ You should implement **two designated methods** based on the **latent factor model(Matrix Factorization)**. Details will be included in the following pages.



Method 1 (MF with BCE)

- ❖ You should regard this as **binary classification task**. All data in *train.csv* represents the positive pairs $(user, item_{pos})$, so you should sample the negative pairs $(user, item_{neg})$ for each user from the entire item set by yourself.

Note : you can adjust the ratio of positive pairs and negative pairs in your own way.

- ❖ You have to use **binary cross entropy** loss for your hidden factor model(MF).

$$BCELoss := -\left[\sum_{(u,i) \in D^+} \log(\sigma(u^T i)) + \sum_{(u,j) \in D^-} \log(1 - \sigma(u^T j)) \right]$$

Method 2 (MF with BPR)

- ❖ You should regard this as **ranking task**. All data in *train.csv* represents the positive pairs $(user, item_{pos})$, so you should sample the negative pairs $(user, item_{neg})$ for each user from the entire item set by yourself.

Note : you can adjust the ratio of positive pairs and negative pairs in your own way.

- ❖ You have to use **Bayesian personalized ranking** loss for your hidden factor model(MF).

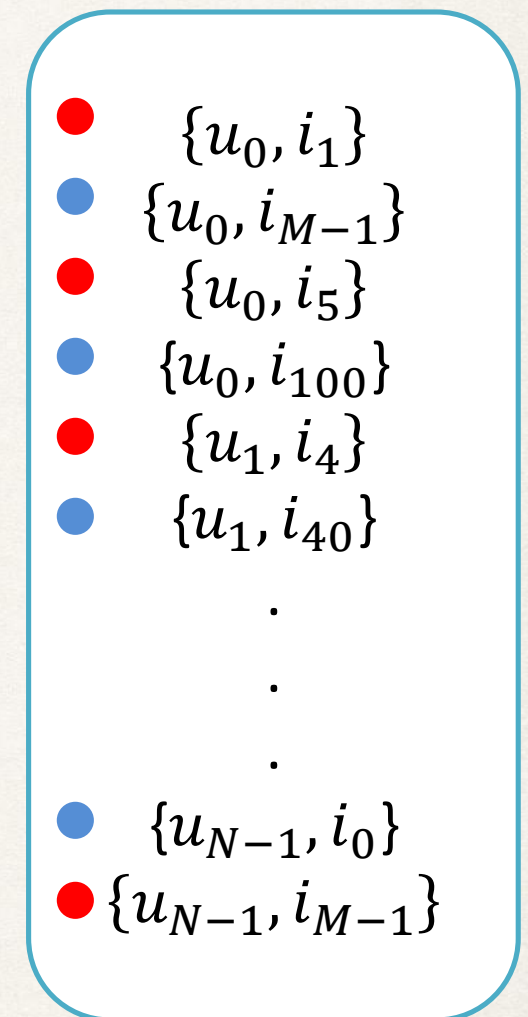
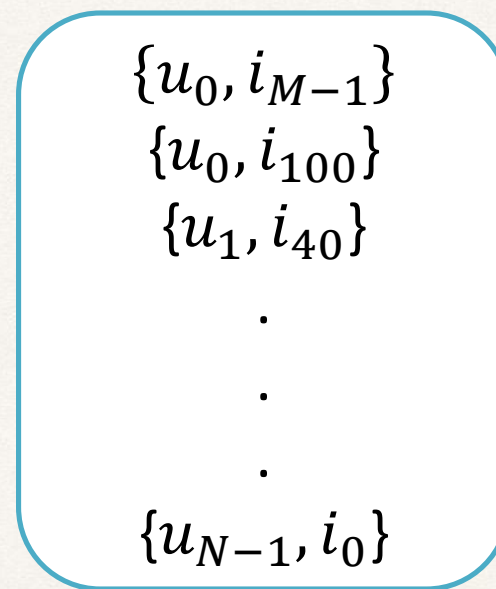
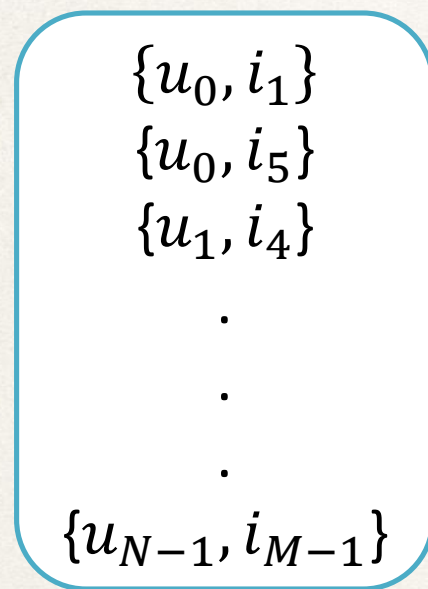
$$BPRLoss := \sum_{(u,i,j) \in D} \ln \sigma(u^T i - u^T j) + \lambda \|\Theta\|^2$$

BPR paper link: <https://arxiv.org/ftp/arxiv/papers/1205/1205.2618.pdf>



Negative Sampling

Totally N users, M items



Positive pair (u, i) from *train.csv*

Negative pair (u, j) from *negative sampling*

Total training data

All user number: $\max(\text{user_id}) + 1$

All item number: $\max(\text{item_id}) + 1$



Program IO

- ❖ Your program is required to support input of a Input File, and output a **ranking list**. (Please see Input File Format and Ranking List Format next pages)
- ❖ There is no restriction to the programming language you use, but make sure your program is **executable on R217 workstation**.
- ❖ Using the third party tools directly for MF or BPR is prohibited. But you can use packages like Keras, Tensorflow, or PyTorch.



Input file format: *train.csv*

- ❖ This is the **only** provided file in this homework!
- ❖ Each line in this file describes the historical interacted items for one user **chronologically** and the format of each line is as following:
[user_id], [item ids of user_id]

User 0

id of interacted items of
user 0

```
1 UserId,ItemId
2 0,1938 490 128 1197 2893 2983 1861 1307 2547 2312 2568 1386 697 275 1755 1812 1200 1113 1548 307 1135 745 2971 2086 2351 1260 2359 1857
3 74 515 1529 1295 2107 3142 1034 3158 3186 2855 1194 2827 1293 2537 122 845 706 2173 830 2538
4 1,465 716 2885 3083 1352 2898 496 1868 2312 1907 1575 2013 1455 2547 898 865 3248 2974 1170 3181 3147 2100 1657 1770 968 3220 1113 2821
5 22 780 18 2031 1793 1135 1593 1786 1579 945 380 443 2299 1156 584 1218 3086 1230 329 572 1268 862 3176 2328 988 3148 132 494 1560 3246 2
6 0 683 269 2614 2862 1068 491 2521 402 965 1803 1258 2840 515 1558 1672 2174 505 3186 3079 1195 471 2291 2795 2726 1784 1200 2320 1128 34
7 783 100 2470 1567 2398 2006 2472 2875 48 2976 2551 2932 2883 2441 964 3059 1973 2308 1964 765 2937 3052 2044 1975 3064 2051 448 603
8 2,865 1575 2385 1156 1812 1199 2946 1505 3205 68 988 1386 1875 2022 607 465 2855 2750 2147 2885 2350 949 2470 800 2976 115 1402 2198 280
9 1695 186 2163 808 964 457 3142 231 1673 277 770 502 1197 3037 2299 471 2902
10 3,2885 2351 3248 2470 2616 1386 2022 465 3209 1102 1755 1200 1357 3022 517 1060 783 1956 380
```



Output Ranking List Format

- ❖ The first line includes two column names: “*UserId*”, “*ItemId*”
- ❖ First column: *UserId* : the set of user_id in ranking list should be equal to the set of user_id in *train.csv*.
- ❖ Second column: *ItemId* : ranking list of top 50 items of each user.
- ❖ The two columns should be separated by a **comma**.
- ❖ Item ids should be separated by spaces.
- ❖ **Note that retrieved items should be sorted by their relevant score.**
- ❖ You can retrieve **at most 50 items** for each user.



Program Execution

- ❖ You should write your own shell script (*run.sh*) to compile and run your program.
- ❖ When testing your program, we will execute the following commands on **R217 workstation**, please make sure your program is executable on the workstation.
- ❖ `$/run.sh [path of your output ranking list]`

Note : (When testing your program, we will use the same testing file as it on Kaggle. i.e. You should get the same result as it on Kaggle.)

We will **only use CPU** to test your program, so please check your program could output the ranking list within 5 minutes using CPU.



Restrictions

- ❖ Using the third party tools directly for MF or BPR is prohibited.
- ❖ If you are not sure packages you used is legal or not, please inquiry TA by e-mail.
- ❖ Your program should finish in 5 minutes using only CPU.
- ❖ Do not copy other's code. Those who copy code and those who allow others to copy his/her code will be punished seriously.
- ❖ If your model is too big to compress, please upload it to an accessible link and attach this link to README.md.



Evaluation

- ❖ We will use the **Mean Average Precision (MAP)** value to evaluate your ranking list.
- ❖ We will **not** provide testing data so you have to split *train.csv* into training and validation data.

Note: Testing items of each user will be the latest interaction of them.

- ❖ The number of testing positive items is around 11% of the training positive items **for every user**.
e.g. User 0 has 48 interacted items in *train.csv*, there will be 5 positive items for user 0 in *test.csv*.



Report

- ❖ Please write your report as a **report.pdf** and put it into the zipped file. The report should contain the following content:
 - ❖ Q1 : Describe your **MF with BCE** (e.g. parameters, loss function, negative sample method and MAP score on Kaggle public scoreboard)
 - ❖ Q2 : Describe your **MF with BPR** (e.g. parameters, loss function, negative sample method and MAP score on Kaggle public scoreboard)
 - ❖ Q3 : Compare your results of Q1 and Q2. Do you think the BPR loss benefits the performance? If do, write some reasons of why BPR works well; If not, write some reasons of why BPR fails.
 - ❖ Q4 : Plot the MAP curve on testing data(Kaggle) for hidden factors $d = 16, 32, 64, 128$ and describe your finding.
 - ❖ (Bonus 10%) Q5 : Change the ratio between positive and negative pairs, compare the results and discuss your finding.



Submission

- ❖ Please put report, scripts and code into the directory named your **student ID**. Package this folder into a zip file and submit it to NTU COOL, following is the structure and content of the **zip**:
 - ❖ For example: R07922XXX.zip
 - ❖ +---R07922XXX(directory) (with **R** in uppercase)
 - ❖ +---report.pdf
 - ❖ +---run.sh
 - ❖ +---Your model(Any types of extensions as long as your program can load!)
 - ❖ +---requirements.txt (Specify all the packages you need.)
 - ❖ +---All the other files and source code required by your program
 - ❖ +---README.md (**not necessary**)
- (Note that if your model is too big to compress, upload it to an accessible link and attach this link to README.md)



Scoring

- ❖ 60% for your report.
- ❖ 10% for performance better than simple baseline on public leaderboard
- ❖ 10% for performance better than simple baseline on private leaderboard
- ❖ 10% for performance better than strong baseline on public leaderboard
- ❖ 10% for performance better than strong baseline on private leaderboard
- ❖ Note that you'll get 0 for performance if you don't have any submit record on Kaggle.



Competition on kaggle



Join Competition

- ❖ This is individual homework. One person in each team.
- ❖ The link of the competition is below:
- ❖ <https://www.kaggle.com/c/wm-2020-personalized-recommendation>



WM 2020 - Personalized Item Recommendation

web mining 2020 programming hw2

Host [Overview](#) Data Notebooks Leaderboard Rules Team

[My Submissions](#)

 This competition hasn't been launched. Only hosts and Kaggle admins can see it.

Overview

[Edit](#)

Description

Evaluation

Timeline

Prizes

Kernels Requirements

[+ Add Page](#)

In this homework, you will learn to deal with the task of personalized recommendation with **implicit feedback**.

You have to implement **two** designated method based on the **latent factor model** (Matrix Factorization).

You should try implement the model using **BCE** (binary cross-entropy error) loss and **BPR** (Bayesian Personalized Ranking) loss.

For more details, please consult the slides.



Leaderboard

- ❖ Public/Private leaderboard
- ❖ 50% / 50% users for public and private respectively
- ❖ Best on public \neq best on private



Rules

- ❖ One account per participant
- ❖ The name on the leaderboard **must** be your student ID(with upper case).
- ❖ You may select up to 2 final submissions for judging.
- ❖ You may submit a maximum of 5 entries per day.



Deadlines

- ❖ Kaggle submission Deadline:
2020/06/14 23:59:59 (UTC+8)
- ❖ Report submission Deadline:
2020/06/15 23:59:59 (UTC+8)
- ❖ Or email to TAs: irlab.ntu@gmail.com
- ❖ Late policy: 10% per day

