

This tool is to process and collect the results of the experiments. When running our model, there will be a .xlsx file generated to store the results.

data_collection	2020/6/22 11:19	文件夹	
data_process	2020/6/27 11:51	文件夹	
Movielens_LCFN_1592661763571.xlsx	2020/6/20 23:14	Microsoft Excel 工...	39 KB
Movielens_LCFN_1592711290100.xlsx	2020/6/21 13:01	Microsoft Excel 工...	39 KB
Movielens_LCFN_1592728762480.xlsx	2020/6/21 17:51	Microsoft Excel 工...	39 KB
Movielens_LCFN_1592683858826.xlsx	2020/6/21 5:23	Microsoft Excel 工...	39 KB
Movielens_LCFN_1592715678771.xlsx	2020/6/21 14:14	Microsoft Excel 工...	39 KB
Movielens_LCFN_1592670693727.xlsx	2020/6/21 1:44	Microsoft Excel 工...	39 KB

In each .xlsx file, there are three sheet. In the first sheet, we record the dataset, model name, and hyperparameters.

	A	B	C	D	E	F	G	
1	DATASET	Movielens						
2	MODEL	LCFN						
3	LR	0.002						
4	LAMDA	0.02						
5	LAYER	1						
6	EMB_DIM	64						
7	FREQUENCY_USER	300						
8	FREQUENCY_ITEM	200						
9	BATCH_SIZE	10000						
10	SAMPLE_RATE	1						
11	IF_PRETRAIN	0						
12	N_EPOCH	200						
13	TEST_VALIDATION	Validation						
14	TOP_K	2	5	10	20	50	100	
15	OPTIMIZATION	Adam						
16								

In the second and the third sheet, we record the  $F_1$ -score and NDCG@{2, 5, 10, 20, 50, 100} during the 200 iterations, respectively.

	A	B	C	D	E	F	G	H
1		2	5	10	20	50	100	
2	1	0.025491	0.037529	0.053227	0.065424	0.06836	0.063647	
3	2	0.022988	0.042395	0.056327	0.069336	0.070616	0.064615	
4	3	0.02905	0.048214	0.062033	0.070125	0.071661	0.06667	
5	4	0.027098	0.04641	0.060538	0.071113	0.071862	0.066837	
6	5	0.031212	0.049515	0.065905	0.077754	0.07928	0.073874	
7	6	0.031966	0.054503	0.072498	0.086336	0.08797	0.080106	
8	7	0.045252	0.069414	0.086018	0.098286	0.099943	0.08955	
9	8	0.044415	0.072093	0.090693	0.104253	0.104442	0.093117	
10	9	0.046861	0.079009	0.104209	0.115448	0.110983	0.097785	
11	10	0.052871	0.079945	0.098632	0.109409	0.110955	0.097611	
12	11	0.049676	0.08206	0.102947	0.116912	0.117828	0.102806	
13	12	0.047067	0.077682	0.100658	0.117632	0.119047	0.10567	
14	13	0.05586	0.089469	0.110189	0.12489	0.120897	0.105698	
15	14	0.058745	0.090191	0.112779	0.131653	0.129121	0.112494	
16	15	0.055181	0.086415	0.10719	0.123347	0.12354	0.107585	
17	16	0.053729	0.085096	0.1104	0.125996	0.12523	0.110235	
18	17	0.051863	0.085601	0.1093	0.127073	0.124483	0.108638	

When tuning models, for example, with respect to learning rate  $\eta$  and regularization coefficient  $\lambda$ , we run the model:

```

if __name__ == '__main__':
    for counter in range(10):
        for LR in [0.0002, 0.0005, 0.001, 0.002, 0.005]:
            for LAMDA in [0.002, 0.005, 0.01, 0.02, 0.05]:
                path_excel = ...
                para = ...
                para_name = ...
                ## print and save model hyperparameters
                print('GPU INDEX: ', GPU_INDEX)
                print_params(para_name, para)
                save_params(para_name, para, path_excel)
                ## train the model
                train_model(para, path_excel)

```

There will be  $10 \times 5 \times 5$  files.

data\_process.py can count the maximum and the average of top-5 metrics ( $F_1$ -score and NDCG) during 200 iterations, and calculate the average of 10 repeats. data\_process.py generates  $5 \times 5$  files in folder *data\_process*.

metric@2, 5, 10, 20, 50, 100

	epoch	2	5	10	20	50	100
1	epoch						
2	1	0.07601436	0.11485081	0.14147533	0.15470371	0.15074752	0.12812052
3	2	0.0780635	0.11718235	0.13913031	0.15421838	0.14784158	0.12558567
4	3	0.07706587	0.11543796	0.14164023	0.1528275	0.14739515	0.12579245
5	4	0.07780391	0.11480725	0.14054867	0.15392432	0.1484972	0.12767814
6	5	0.07538948	0.11623781	0.14166999	0.15758273	0.15081981	0.12691009
7	6	0.07736431	0.11644622	0.14214741	0.15480504	0.1504408	0.12661124
8	7	0.07678404	0.1154468	0.14026948	0.15302756	0.14623857	0.12575142
9	8	0.07883821	0.11778122	0.14427644	0.15392571	0.14917788	0.12681947
10	9	0.0753792	0.11684471	0.1401171	0.15505981	0.14950679	0.12665349
11	10	0.07859079	0.11556646	0.14295997	0.15383783	0.14917445	0.12647748
12	mean	0.07712937	0.11606016	0.14142349	0.15439126	0.14898397	0.12664
13	std	0.00117879	0.00095168	0.00142116	0.00126401	0.00143365	0.000776
14							

Records of 10 repeats

Average and std of 10 repeats

Maximum metrics

Average of top-5 metrics

data\_collection.py can count the metrics with varying hyperparameters from the files in folder *data\_process*. data\_collection.py generates a new files in the folder *data\_collection* and removes the files in the folder *data\_process*. There are 5 sheets in the newly generated file. The first sheet records the settings and hyperparameters. We can see which hyperparameters vary in this sheet.

	A	B	C	D	E	F
1		1	2	3	4	5
2	<b>DATASET</b>	Amazon				
3	<b>MODEL</b>	LCFN				
4	<b>LR</b>	0.001	0.002	0.0005	0.005	0.0002
5	<b>LAMDA</b>	0.02	0.002	0.05	0.01	0.005
6	<b>LAYER</b>	1				
7	<b>EMB_DIM</b>	64				
8	<b>EQUENCY_US</b>	100				
9	<b>EQUENCY_ITI</b>	50				
10	<b>BATCH_SIZE</b>	10000				
11	<b>SAMPLE_RATIO</b>	1				
12	<b>IF_PRETRAIN</b>	0				
13	<b>N_EPOCH</b>	200				
14	<b>ST_VALIDATION</b>	Validation				
15	<b>TOP_K</b>	[2, 5.0, 10.0, 20.0, 50.0, 100.0],				
16	<b>OPTIMIZATION</b>	Adam				
17						

Two variables

截图(Alt + A)

Parameters F1\_max F1\_top5 NDCG\_max NC ... (+) :

In the 2<sup>nd</sup>-5<sup>th</sup> sheets, we can see the performances with different parameters.

Record the second variable

	A	B	C	D	E	F
1	<b>LAMDA</b>	<b>0.0002</b>	<b>0.0005</b>	<b>0.001</b>	<b>0.002</b>	<b>0.005</b>
2	<b>0.002</b>	0.01118878	0.01587108	0.01579375	0.0155637	0.01536061
3	<b>0.005</b>	0.01136326	0.01541244	0.01628141	0.01628854	0.01574701
4	<b>0.01</b>	0.01075955	0.01560162	0.0167947	0.01663112	0.01696725
5	<b>0.02</b>	0.01041652	0.01502839	0.01605725	0.01638656	0.01625424
6	<b>0.05</b>	0.00970164	0.01263473	0.01472413	0.01488281	0.01557811
7						

The first variable (LR)

The second variable (LAMDA)

Parameters F1\_max F1\_top5 NDCG\_max NC ... (+) :

If there is only one variable:

Record the only variable

	A	B	C	D	E	F	G
1	<b>EMB_DIM</b>	<b>2</b>	<b>5</b>	<b>10</b>	<b>20</b>	<b>50</b>	<b>100</b>
2	<b>4</b>	0.00844664	0.01050925	0.00877103	0.00837161	0.00655393	0.00541106
3	<b>8</b>	0.00803461	0.01059264	0.00921428	0.00886108	0.00679444	0.00552517
4	<b>16</b>	0.00899602	0.01163743	0.01026933	0.00934713	0.00715538	0.00569933
5	<b>32</b>	0.01311633	0.01326839	0.01241064	0.01047799	0.00783093	0.00607293
6	<b>64</b>	0.02266172	0.02091059	0.01736126	0.01349501	0.00941275	0.00706493
7	<b>128</b>	0.03958247	0.03379229	0.02554971	0.01846453	0.01179932	0.00835102
8							

Metrics@2, 5, 10, 20, 50, 100

The variable

Parameters F1\_max F1\_top5 NDCG\_max NC ... (+) :

Please note that data\_collection.py only supports at most 2 varying hyperparameters. DATASET and MODEL are not hyperparameters, thus 2 varying hyperparameters with varying DATASET and MODEL are allowed.