
DaisyRec-v2.0 Documentation

Release 0.1

DaisyRec Contributors

Jul 05, 2023

PARAMETER SETTINGS

1	Basic Settings	3
1.1	-problem_type	3
1.2	-optimization_metric	3
1.3	-hyperopt_trail	3
1.4	-hyperopt_pack	3
1.5	-algo_name	4
1.6	-dataset	4
1.7	-prepro	5
1.8	-val_method	5
1.9	-test_method	5
1.10	-val_size	5
1.11	-test_size	5
1.12	-topk	5
1.13	-fold_num	6
1.14	-cand_num	6
1.15	-sample_method	6
1.16	-sample_ratio	6
1.17	-num_ng	6
1.18	-positive_threshold	6
1.19	-loss_type	6
1.20	-gpu	7
2	Algorithm Specific Settings	9
2.1	-init_method	9
2.2	-optimizer	9
2.3	-early_stop	9
2.4	-tune_testset	10
2.5	-factors	10
2.6	-reg_1	10
2.7	-reg_2	10
2.8	-dropout	10
2.9	-lr	10
2.10	-epochs	10
2.11	-batch_size	10
2.12	-num_layers	11
2.13	-alpha	11
2.14	-elastic	11
2.15	-pop_n	11
2.16	-maxk	11
2.17	-node_dropout	11

2.18	<code>-mess_dropout</code>	11
2.19	<code>-kl_reg</code>	11

The description of all parameters is listed below.

BASIC SETTINGS

1.1 `-problem_type`

define a point-wise or pair-wise problem.

- **point-wise**: point-wise algorithm
- **pair-wise**: pair-wise algorithm

1.2 `-optimization_metric`

the metric to be optimized for hyper-parameter tuning via HyperOpt

- **ndcg**
- **precision**
- **recall**
- **hr**
- **map**
- **mrr**

1.3 `-hyperopt_trail`

the number of trails of HyperOpt

1.4 `-hyperopt_pack`

record the searching space of hyper-parameters for HyperOpt

1.5 `-algo_name`

the algorithm to be executed

- **mostpop**
- **itemknn**
- **puresvd**
- **slim**
- **mf**
- **fm**
- **neumf**
- **nfm**
- **ngcf**
- **multi-vae**

1.6 `-dataset`

the dataset to be evaluated

- **ml-100k**
- **ml-1m**
- **ml-10m**
- **ml-20m**
- **lastfm**
- **book-x**
- **amazon-cloth**
- **amazon-electronic**
- **amazon-book**
- **amazon-music**
- **epinions**
- **yelp**
- **citeulike**
- **netflix**

1.7 `-prepro`

the data pre-processing strategy

- **origin**: adopt the raw data
- **Fcore**: recursively filter users and items that have interactions no less than N, e.g., 5score
- **Ffilter**: only filter users and items that have interactions no less than N once, e.g., 5filter

1.8 `-val_method`

training and validation data splitting strategy

- **tsbr**: time-aware split-by-ratio
- **rsbr**: random-aware split-by-ratio
- **tloo**: time-aware leave-one-out
- **rlloo**: random-aware leave-one-out

1.9 `-test_method`

training and test data splitting strategy, which should be consistent with the settings for **val_method**

1.10 `-val_size`

ratio of validation set size in the range of (0,1), e.g., 0.1 means retaining 10% of training data as validation data

1.11 `-test_size`

ratio of test set size in the range of (0,1), e.g., 0.2 means retaining 20% of the whole data as test data

1.12 `-topk`

the length of recommendation list

1.13 `-fold_num`

the fold number of cross-validation

1.14 `-cand_num`

the number of candidate items used for ranking

1.15 `-sample_method`

negative sampling strategy

- **uniform**: uniformly sample negative items
- **low-pop**: sample popular items with low rank
- **high-pop**: sample popular items with high rank

1.16 `-sample_ratio`

control the ratio of popularity sampling for the hybrid sampling strategy in the range of (0,1), e.g., for the hybrid sampling strategy uniform+low-pop, `-sample_ratio=0.1` means 10% of the negative items are sampled via low-pop

1.17 `-num_ng`

the number of negative samples

1.18 `-positive_threshold`

the threshold for binarizing the ratings into positive samples (for example if the threshold = 4, it means the items with ratings no less than 4 will be treated as positive items)

1.19 `-loss_type`

type of loss function

- **CL**: cross-entropy loss for point-wise problem
- **SL**: square error loss for point-wise problem
- **BPR**: BPR loss for pair-wise problem
- **HL**: hinge loss for pair-wise problem
- **TL**: top-1 Loss for pair-wise problem

1.20 `-gpu`

the ID of GPU card

ALGORITHM SPECIFIC SETTINGS

2.1 `--init_method`

parameter initializers

- **default**: initialize parameters according to the original paper
- **normal**: initialize parameters with normal distribution
- **uniform**: initialize parameters with uniform distribution
- **xavier_normal**: initialize parameters with xavier_normal distribution
- **xavier_uniform**: initialize parameters with xavier_uniform distribution

2.2 `--optimizer`

optimization method for training the algorithms

- **default** (optimizer in the original paper)
- **sgd**
- **adam**
- **adagrad**

2.3 `--early_stop`

whether to activate the early-stop mechanism

- **true**
- **false**

2.4 `-tune_testset`

whether to directly tune on testset, and the default value is false

- `true`
- `false`

2.5 `-factors`

the dimension of latent factors (embeddings)

2.6 `-reg_1`

the coefficient of L1 regularization

2.7 `-reg_2`

the coefficient of L2 regularization

2.8 `-dropout`

dropout rate

2.9 `-lr`

learning rate

2.10 `-epochs`

training epochs

2.11 `-batch_size`

batch size for training

2.12 `--num_layers`

number of layers for MLP

2.13 `--alpha`

constant to multiply the penalty terms for SLIM

2.14 `--elastic`

the ElasticNet mixing parameter for SLIM in the range of (0,1)

2.15 `--pop_n`

the preliminary selected top-n popular candidate items to reduce the time complexity for MostPop

2.16 `--maxk`

the number of neighbors to take into account for ItemKNN

2.17 `--node_dropout`

node dropout ratio for NGCF

2.18 `--mess_dropout`

message dropout ratio for NGCF

2.19 `--kl_reg`

the coefficient of KL regularization for Multi-VAE