RecBole (伯乐)

"世有伯乐，然后有千里马。千里马常有，而伯乐不常有。"——韩愈《马说》

pypi v1.0.1 | Anaconda.org 1.0.1 | License MIT | arXiv RecBole

HomePage | Docs | Datasets | Paper | Blogs | 中文版

RecBole is developed based on Python and PyTorch for reproducing and developing recommendation algorithms in a unified, comprehensive and efficient framework for research purpose. Our library includes 78 recommendation algorithms, covering four major categories:

- General Recommendation
- Sequential Recommendation

- Context-aware Recommendation
- Knowledge-based Recommendation

We design a unified and flexible data file format, and provide the support for 28 benchmark recommendation datasets. A user can apply the provided script to process the original data copy, or simply download the processed datasets by our team.

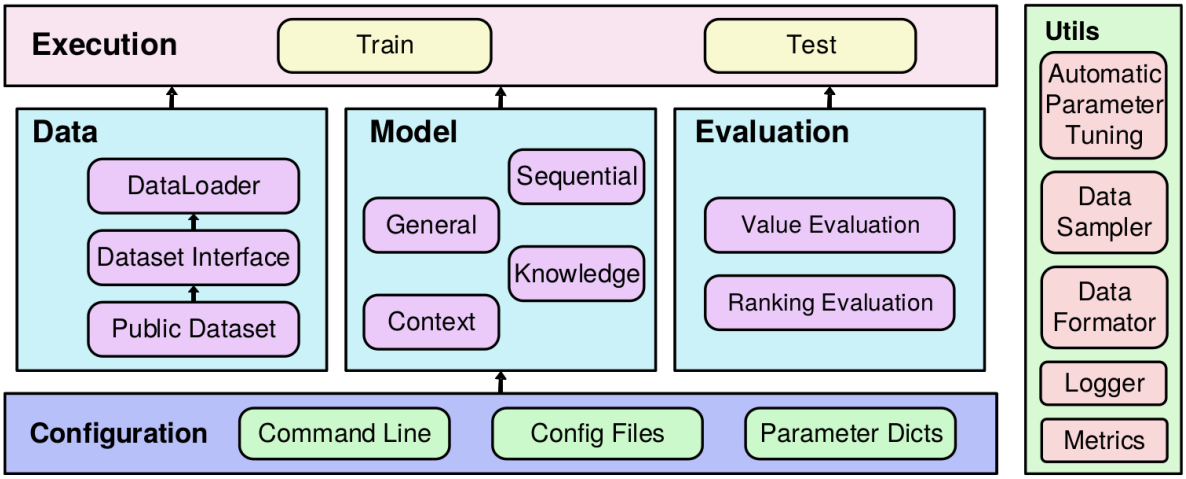

Figure: RecBole Overall Architecture

In order to support the study of recent advances in recommender systems, we construct an extended recommendation library RecBole2.0 consisting of 8 packages for up-to-date topics and architectures (e.g., debased, fairness and GNNs).

Feature

- **General and extensible data structure.** We design general and extensible data structures to unify the formatting and usage of various recommendation datasets.
- **Comprehensive benchmark models and datasets.** We implement 78 commonly used recommendation algorithms, and provide the formatted copies of 28 recommendation datasets.
- **Efficient GPU-accelerated execution.** We optimize the efficiency of our library with a number of improved techniques oriented to the GPU environment.
- **Extensive and standard evaluation protocols.** We support a series of widely adopted evaluation protocols or settings for testing and comparing recommendation algorithms.

RecBole News

NEW **06/28/2022**: We release **RecBole2.0** with **8 packages** consisting of **65 newly implement models**.

02/25/2022: We release RecBole v1.0.1.

09/17/2021: We release RecBole v1.0.0.

03/22/2021: We release RecBole v0.2.1.

01/15/2021: We release RecBole v0.2.0.

12/10/2020: 我们发布了RecBole小白入门系列中文博客（持续更新中）。

12/06/2020: We release RecBole v0.1.2.

11/29/2020: We constructed preliminary experiments to test the time and memory cost on three different-sized datasets and provided the test result for reference.

11/03/2020: We release the first version of RecBole **v0.1.1**.

Installation

RecBole works with the following operating systems:

- Linux
- Windows 10
- macOS X

RecBole requires Python version 3.7 or later.

RecBole requires torch version 1.7.0 or later. If you want to use RecBole with GPU, please ensure that CUDA or cudatoolkit version is 9.2 or later. This requires NVIDIA driver version >= 396.26 (for Linux) or >= 397.44 (for Windows10).

Install from conda

```
conda install -c aibox recbole
```

Install from pip

```
pip install recbole
```

Install from source

```
git clone https://github.com/RUCAIBox/RecBole.git && cd RecBole
pip install -e . --verbose
```

Quick-Start

With the source code, you can use the provided script for initial usage of our library:

```
python run_recbole.py
```

This script will run the BPR model on the ml-100k dataset.

Typically, this example takes less than one minute. We will obtain some output like:

```
INFO ml-100k
The number of users: 944
Average actions of users: 106.04453870625663
The number of items: 1683
Average actions of items: 59.45303210463734
The number of inters: 100000
The sparsity of the dataset: 93.70575143257098%

INFO Evaluation Settings:
Group by user_id
Ordering: {'strategy': 'shuffle'}
Splitting: {'strategy': 'by_ratio', 'ratios': [0.8, 0.1, 0.1]}
Negative Sampling: {'strategy': 'full', 'distribution': 'uniform'}

INFO BPRMF(
    (user_embedding): Embedding(944, 64)
    (item_embedding): Embedding(1683, 64)
    (loss): BPRLoss()
)
Trainable parameters: 168128

INFO epoch 0 training [time: 0.27s, train loss: 27.7231]
INFO epoch 0 evaluating [time: 0.12s, valid_score: 0.021900]
INFO valid result:
recall@10: 0.0073  mrr@10: 0.0219  ndcg@10: 0.0093  hit@10: 0.0795
precision@10: 0.0088

...

INFO epoch 63 training [time: 0.19s, train loss: 4.7660]
INFO epoch 63 evaluating [time: 0.08s, valid_score: 0.394500]
INFO valid result:
recall@10: 0.2156  mrr@10: 0.3945  ndcg@10: 0.2332  hit@10: 0.7593
precision@10: 0.1591

INFO Finished training, best eval result in epoch 52
INFO Loading model structure and parameters from saved/***.pth
INFO best valid result:
recall@10: 0.2169  mrr@10: 0.4005  ndcg@10: 0.235  hit@10: 0.7582
```

```
precision@10: 0.1598
INFO test result:
recall@10: 0.2368  mrr@10: 0.4519  ndcg@10: 0.2768  hit@10: 0.7614
precision@10: 0.1901
```

If you want to change the parameters, such as `learning_rate`, `embedding_size`, just set the additional command parameters as you need:

```
python run_recbole.py --learning_rate=0.0001 --embedding_size=128
```

If you want to change the models, just run the script by setting additional command parameters:

```
python run_recbole.py --model=[model_name]
```

Auto-tuning Hyperparameter

Open `RecBole/hyper.test` and set several hyperparameters to auto-searching in parameter list. The following has two ways to search best hyperparameter:

- **loguniform**: indicates that the parameters obey the uniform distribution, randomly taking values from e^{-8} to e^{0}.
- **choice**: indicates that the parameter takes discrete values from the setting list.

Here is an example for `hyper.test`:

```
learning_rate loguniform -8, 0
embedding_size choice [64, 96 , 128]
train_batch_size choice [512, 1024, 2048]
mlp_hidden_size choice ['[64, 64, 64]','[128, 128]']
```

Set training command parameters as you need to run:

```
python run_hyper.py --model=[model_name] --dataset=[data_name] --
config_files=xxxx.yaml --params_file=hyper.test
e.g.
python run_hyper.py --model=BPR --dataset=ml-100k --config_files=test.yaml --
params_file=hyper.test
```

Note that `--config_files=test.yaml` is optional, if you don't have any customize config settings, this parameter can be empty.

This processing maybe take a long time to output best hyperparameter and result:

```
running parameters:
{'embedding_size': 64, 'learning_rate': 0.005947474154838498,
'mlp_hidden_size': '[64,64,64]', 'train_batch_size': 512}
  0%|
| 0/18 [00:00<?, ?trial/s, best loss=?]
```

More information about parameter tuning can be found in our docs.

Time and Memory Costs

We constructed preliminary experiments to test the time and memory cost on three different-sized datasets (small, medium and large). For detailed information, you can click the following links.

- General recommendation models
- Sequential recommendation models
- Context-aware recommendation models
- Knowledge-based recommendation models

NOTE: Our test results only gave the approximate time and memory cost of our implementations in the RecBole library (based on our machine server). Any feedback or suggestions about the implementations and test are welcome. We will keep improving our implementations, and update these test results.

RecBole Major Releases

Releases	Date
v1.0.0	09/17/2021
v0.2.0	01/15/2021
v0.1.1	11/03/2020

Contributing

Please let us know if you encounter a bug or have any suggestions by filing an issue.

We welcome all contributions from bug fixes to new features and extensions.

We expect all contributions discussed in the issue tracker and going through PRs.

We thank the insightful suggestions from @tszumowski, @rowedenny, @deklanw et.al.

We thank the nice contributions through PRs from @rowedenny, @deklanw et.al.

Cite

If you find RecBole useful for your research or development, please cite the following papers: RecBole and RecBole2.0

```
@inproceedings{recbole1.0,
  title={Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms},
  author={Zhao, Wayne Xin and Mu, Shanlei and Hou, Yupeng and Lin, Zihan and Chen, Yushuo and Pan, Xingyu and Li, Kaiyuan and Lu, Yujie and Wang, Hui and Tian, Changxin and others},
  booktitle={Proceedings of the 30th ACM International Conference on Information \& Knowledge Management},
  year={2021}
}
```

```
@article{recbole2.0,
  title={RecBole 2.0: Towards a More Up-to-Date Recommendation Library},
  author={Zhao, Wayne Xin and Hou, Yupeng and Pan, Xingyu and Yang, Chen and Zhang, Zeyu and Lin, Zihan and Zhang, Jingsen and Bian, Shuqing and Tang, Jiakai and Sun, Wenqi and others},
  journal={arXiv preprint arXiv:2206.07351},
  year={2022}
}
```

The Team

RecBole is developed by RUC, BUPT, ECNU, and maintained by RUC.

Here is the list of our lead developers in each development phase. They are the souls of RecBole and have made outstanding contributions.

Time	Version	Lead Developers
June 2020 ~ Nov. 2020	v0.1.1	Shanlei Mu (@ShanleiMu), Yupeng Hou (@hyp1231), Zihan Lin (@linzihan-backforward), Kaiyuan Li (@tsotfsk)

Time	Version	Lead Developers
Nov. 2020 ~ now	v0.1.2 ~ v1.0.0	Yushuo Chen (@chenyushuo), Xingyu Pan (@2017pxy)

License

RecBole uses MIT License.