Schaferct-reproducibility

Readme for reproducibility submission of paper ID [mmsys24ods-p76]

A) Source code info

Repository: https://github.com/n13eho/Schaferct

Programming Language: Python

Additional Programming Language info: Python 3.10

Packages Needed: numpy, onnxruntime, pandas, pyrallis, scikit_learn, torch, tqdm, matplotlib

B) Datasets info

Repository: use the following scripts to download training dataset and evaluation dataset

training dataset: https://github.com/microsoft/RL4BandwidthEstimationChallenge/blob/main/download-testbed-dataset.sh

evaluation dataset: https://github.com/microsoft/RL4BandwidthEstimationChallenge/blob/main/download-emulated-dataset.sh

C) Hardware info

Ubuntu 20.04.6 LTS with a 12GB GPU (we use a NVIDIA GeForce RTX 3080 Ti)

D) Experimentation info

1. Clone repo and install packets

   ```
   git clone https://github.com/n13eho/Schaferct.git
   cd Schaferct
   pip install -r requirements.txt
   ```

2. Download datasets (links mentioned before)

3. Remake training datasets

   You can use the pickle we have made during our training at `./traning_dataset_pickle/v8.pickle`, or remake a new training dataset by:

   a. Rearrange the datasets by different type of behavior policy. Before run this script (`reorganize_by_policy_id.py`), you need to modify the two path of your downloaded datasets.

   ```
   mkdir ALLdatasets
   mkdir ALLdatasets/train
   mkdir ALLdatasets/evaluate
   cd ./code
   python reorganize_by_policy_id.py
   ```

   b. Make new pickle dataset. You can modify the `K` in `make_training_dataset_pickle.py` to put more or less sessions into training dataset

   ```
   python make_training_dataset_pickle.py
   ```

   The dataset-making process takes about 1.5 hours.

4. Train model

   *The code we use is modified from CORL

   You can modify the variables below:

   a. `pickle_path`: path to training dataset, you can use your own dataset

   b. `ENUM`: how many sessions in each policy type of evaluation dataset to evaluate model every `eval_freq` steps

   c. `USE_WANDB`: trun on the wandb or not. You can turn it on to monitor the training process by mse, error_rate, q_score, loss, etc.

   Then you can run:

   ```
   python v14_iql.py
   ```

   The training process takes about 4 hours.

5. Evaluate models

   You can use the offical evaluation scripts to evaluate your model, and here we offer two other srcipts to help evaluation, please modify path and names of variables before runing.

   The detail instructions are in the comments of the scripts.

   a. To run a small evaluation on a small dataset: (download the 24 sessions and modify their path first)

```
python detail_evaluate_on_24_sessions.py
```

b. To evaluate the metrics (mse, errorate) over all evaluation dataset:

```
python evaluate_all.py
```

The whole evaluate process takes about 2 hours.

!!Once again, remember to modify/adjust/rename the path/name of variables mentioned above or in the code’s comments.