Modeling Sequences as Star Graphs to Address Over-smoothing in Self-attentive Sequential Recommendation

This is the implementation of our model MSSG.

Environments

- Python 3.9.13
- PyTorch (version: 1.10.2)

Please install PyTorch following the instructions in https://pytorch.org/.

Dataset

Please find the six processed datasets used in our experiments in the "data" folder.

Train and evaluate MSSG

Please refer to the following example on how to train MSSG on the Amazon-Beauty (Beauty) dataset. The evaluation will be conducted automatically. You are recommended to train MSSG using GPUs.

```
python main.py --data=Beauty --train_dir=Beauty --model=MSSG --num_epochs=201 --hidden_units=256 --maxlen=76 --num_blocks=3 --isTrain=0 --num_heads=16 --batch_size=256 --lr=1e-3 --attn_dropout_rate=0.0
```

`data` specifies the dataset used for training and evaluation.

`model` specifies the MSSG model to be used. Candidates are MSSG and MSSGU (MSSG-u in the paper).

`isTrain` is 1 for hyper-parameter tuning and 0 for evaluation. We will only save models when isTrain is 0.

`attn_dropout_rate` specifies the dropout rate on the attention weights. We set attn_dropout_rate as 0.0 on Beauty and Toys, and 0.5 for the other datasets.

Acknowledgement

The implementation leveraged the code in SASRec. Thanks for the great work!