Data Exploration

December 2022

1 MOOC

This public dataset consists of actions done by studentson a MOOC online course. This dataset consists of 7,047 users interacting with 98 items (videos, answers, etc.) resulting in over 411,749 interactions.

1.1 Script to the Pre-processing

```
cd utils
python preprocess_data.py –data mooc –bipartite
```

1.2 Script to run TGN

```
python train_self_supervised.py –use_memory –prefix tgn-attn –n_runs 10 -d mooc
```

2 LastFM

This public dataset has one month of wholistens-to-which song information. We selected all 1000 users and the 1000 most listened songs resulting in 1,293,103 interactions.

2.1 Script to the Pre-processing

```
cd utils
python preprocess_data.py –data lastfm –bipartite
```

2.2 Script to run TGN

```
python train_self_supervised.py –use_memory –prefix tgn-attn –n_runs 10 -d lastfm
```

3 Autonomous systems AS-733

Autonomous systems graph (Leskovec et al. (2005)) is a communication network from the Border Gateway Protocol logs. The graph has 733 daily snapshots from Nov 1997 to Jan 2000. The graph grows from 103 to 6,474 nodes and from 243 to 13,233 edges.

3.1 Pre-processing

This dataset is a daily graph. The file name contains the date of the events. And each file contains FromNodeId (user_id) and ToNodeId (item_id), so we extract them and put them into a table.

3.2 Script to the Pre-processing

```
python explore_data.py
cd utils
python preprocess_data.py –data autosys
```

3.3 Script to run TGN

```
python train_self_supervised.py –use_memory –prefix tgn-attn –n_runs 10 -d autosys
```

4 MovieLens-10M

The MovieLens dataset (Harper and Konstan (2016)) is a dynamic user-tag interactions dataset. It consists of 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. The dataset shows the tagging behavior of users on the movies they rated.

4.1 Pre-processing

In this dataset, each line of the file represents one tag applied to one movie by one user, and has the following format: {UserID, MovieID, Tag, Timestamp}. Among these four items, UserID is the user_id that we need, and MovieID is the item_id. We extract them and put them into a table, then sort the table by ascending order according to the timestamp.

4.2 Script to the Pre-processing

```
python explore_movie.py
cd utils
python preprocess_data.py –data movie –bipartite
```

4.3 Script to run TGN

```
python train_self_supervised.py –use_memory –prefix tgn-attn –n_runs 10 -d movie
```

5 FB-Forum

This dataset comes from a Facebook-like online community of students at the University of California. This is a bipartite graph where the nodes represent students and groups while the edges represent students' broadcast messages on the groups.

5.1 Pre-processing

In this dataset, the first column is the user_id, the second column is the item_id, the third column is the timestamp. We extract these three items and put them into a table, then sort the table by ascending order according to the timestamp.

5.2 Script to the Pre-processing

```
python explore_fb.py
cd utils
python preprocess_data.py –data fb –bipartite
```

5.3 Script to run TGN

```
python train_self_supervised.py –use_memory –prefix tgn-attn –n_runs 10 -d fb
```

6 Enron-Email

Enron email dataset (Klimt and Yang (2004)) is the network of email exchanges among the employees of Enron. This data was originally released by the Federal Energy Regulatory Commission as part of their investigation. It has been widely used in the community (Nguyen et al. (2018b); De Winter et al. (2018); Chen et al. (2018a); Sankar et al. (2018))

6.1 Pre-processing

This dataset is shown in an email format. Each email contains various items including Message-ID, Date, From, To and so on. We extract the date from 'Date' and transfer it into timestamp. Then we transfer the sender and receiver into ID respectively and extract them as user_id and item_id. The data is also sorted by ascending order according to the timestamp.

6.2 Script to the Pre-processing

```
python explore_enron.py
cd utils
python preprocess_data.py –data enron
```

6.3 Script to run TGN

```
python train_self_supervised.py –use_memory –prefix tgn-attn –n_runs 10 -d enron
```

7 Data Info

Data	Users	Items	Interactions	Download Link
MOOC	7047	97	411749	http://snap.stanford.edu/jodie/mooc.csv
LastFM	980	1000	1293103	http://snap.stanford.edu/jodie/lastfm.csv
AS-733	4078	500	2282501	http://snap.stanford.edu/data/as-733.html
MovieLens-10M	71567	10681	95580	https://grouplens.org/datasets/movielens/10m/
FB-Forum	899	522	33720	https://networkrepository.com/fb-forum.php
Enron-Email	18993	25274	517401	https://www.kaggle.com/datasets/wcukierski/enron-email-dataset

8 Performance

Data	Test AP (Transductive)	Test AP (Inductive)
MOOC	0.9055	0.8953
LastFM	0.7849	0.7255
AS-733	0.9913	0.9829
MovieLens-10M	0.8108	0.7419
FB-Forum	0.7915	0.6563
Enron-Email	0.9690	0.9176