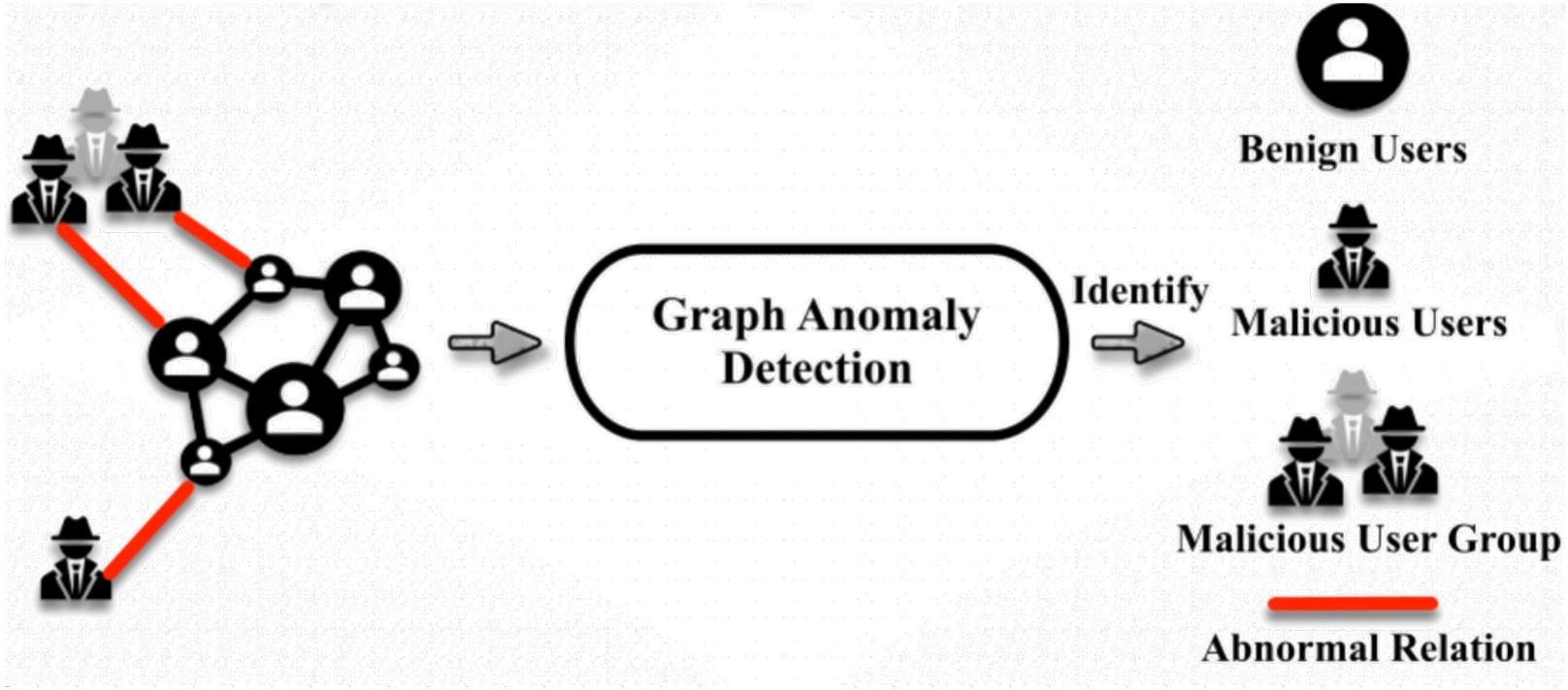
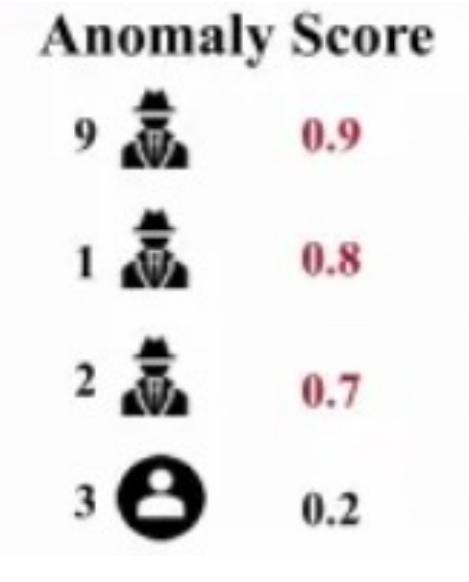
Data Science HW6:
Graph Anomaly Detection





[TA] 曹立武

Submission Deadline: 2023/6/20 23:59

Objectives

- In this homework, you need to implement any kind of Graph Neural Networks to find out the anomaly nodes.

- You will need to solve the problem based on the given graph with the specific node indexes.

Data Format for a homogeneous graph

```
Data(edge_index=[2, 42445086], feature=[39357, 10], label=[39357])
```

Edge Index: [2, num_of_edges]

- 2 means the edge that connect two nodes (node_A – node_B)
- There is no edge weight or edge attribute/feature in this simple graph.

Feature: [num_of_nodes, dim_of_node_feature]

- Each node means a transaction with10-dimension feature (We won't know the exact meaning for each dimensions)

```
tensor([ 11.0000, 576.0000,   3.0000,   0.0000, 486.0000,  10.0000,   9.0000,
Transaction    6.0000,   0.7273,   0.6364], dtype=torch.float64)
```

Label: [num_of_nodes]

- Anomaly: 1 , Normal: 0

Dataset files

- train_sub-graph_tensor.pt:
  - The sub-graph connect 15742 nodes (39357 nodes in total)

```
Data(edge_index=[2, 6784824], feature=[39357, 10], label=[15742])
```

- train_mask.npy:
  - Specify the index with “True/1” for 15742 nodes (39357 nodes in total)

```
[ True  True  True ... False False  True]
```

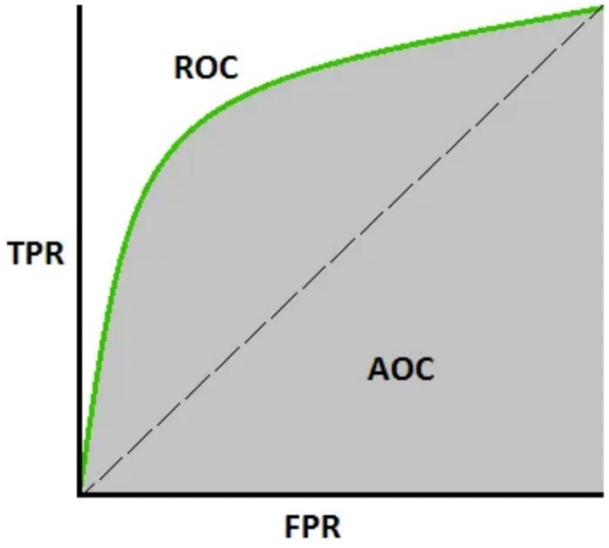
- test_sub-graph_tensor_noLabel.pt:
  - The sub-graph connect 15823 nodes (39357 nodes in total)

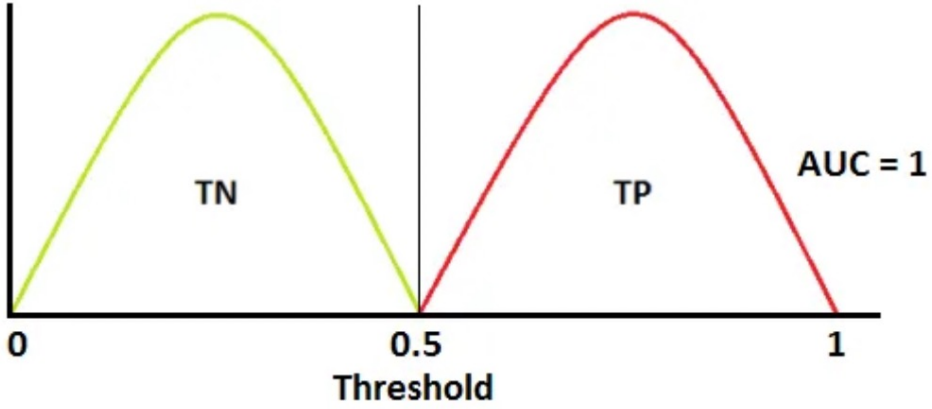
```
Data(edge_index=[2, 7000540], feature=[39357, 10])
```

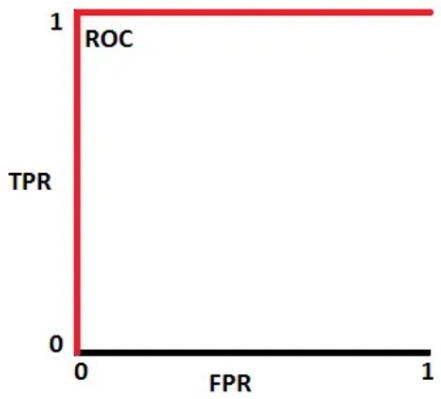
- test_mask.npy:
  - Specify the index with “True/1” for 15823 nodes (39357 nodes in total)

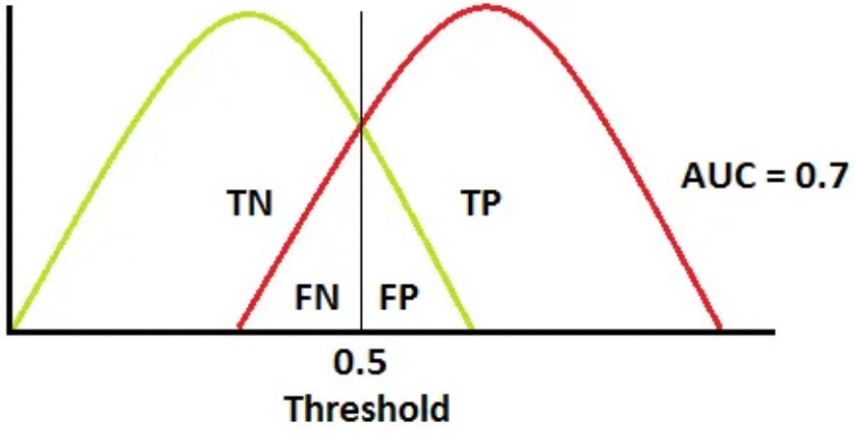
```
[False False False ...  True  True False]
```

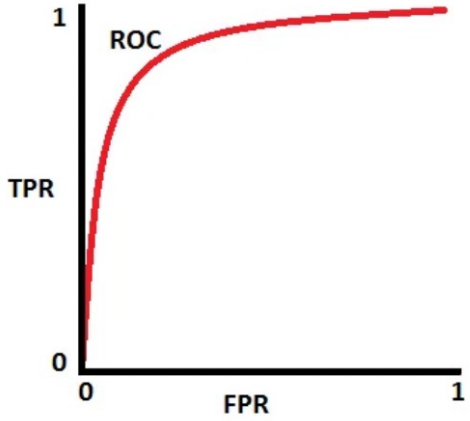
Evaluation Metric – AUC











Grading Policy

- Top 10%: 100 points
- Top 25%: 95 points
- Others: 90 points
- Below the baseline (shown in leaderboard): 0 point
- Public 50%, Private 50%

sample_submission.csv

node idx	node anomaly score
4	0.52018684
5	0.61603762
10	0.56093625
14	0.73761775
15	0.51340027
20	0.78291967
22	0.46858295
23	0.64463618

#	Team	Members	Score	Entries	Last
	baseline		0.83190		

Useful resources

- Pytorch_geometric
  - https://github.com/pyg-team/pytorch_geometric

- DGL (Deep Graph Library)
  - https://github.com/dmlc/dgl
  - Not allowed for this homework, but it also contains powerful Graph-based Deep Learning tools to use.

- Sklearn metrics for roc_auc_score

Several tips that you could try

- The feature that aggregate from Graph Neural Networks, such as GCN Convolution, should **concatenate with its own node feature**.
  - Feature: [Target_Node_feature, Aggregate_feature]

- **Take the anomaly probability for AUC score evaluation.** You can use Cross Entropy for your objective, but need to be careful to select the probability for the anomaly class, not the normal class.

Rules

- Use your student ID as the team name on Kaggle.
- A maximum of 5 submissions per day is allowed on Kaggle.
- **Do not** use additional accounts to get more submission quota.
- **Do not** plagiarise. Write your own codes.
- You can only use the datasets provided in this competition to learn your model.
- Do not attempt to recognize the datasets we used and hack the testing performance. You will not obtain scores for this homework if you violate this rule (we will re-implement your results).

Submissions

- Submit your results to Kaggle: https://www.kaggle.com/t/c1fb1aff1ef446c9948ad5dfc0d81c10

- Submit your zipped source code {student_id}.zip to E3. the zip file should contain a folder {student_id}:
  - {student_id}
    - {student_id}.sh: run this script should regenerate your final submission result.
    - requirements.txt: list the required libraries.
    - Other files

Homework Information

- Deadline: 2023/6/20 23:59
- Please post your question on the E3 forum
- [TA] 曹立武