Documentation

Binghui Wang, Zhonghao Liao

December 13, 2017

1 Terms of Use

The code is provided for research purposes only and without any warranty. Any commercial use is prohibited. The package includes the code that implements the GANG algorithm, and auxillary functions that generate the result. GANG is a structural-based fraudulent-user-detection method in directed online social networks. When using it in your research work, you should cite our github homepage XX

For any question, please contact Binghui Wang (binghuiw@iastate.edu) or Zhonghao Liao (hzliao@iastate.edu).

2 GANG Algorithm

2.1 Input

-graphfile GRAPHFILE

GRAPHFILE stores the edges and weights of a directed social graph. Basically, you need to have a file to store the bidirectional graph and a file to store the outgoing graph generated from your directed social graph, and name them as XX_bidreciontal.txt and XX_outgoing.txt, respectively.

The format of GRAPHFILE is as follows

```
0 1 0.8
0 2 0.6
...
1 0 0.8
...
2 0 0.6
...
```

It means that node 0 and node 1 are connected with edge weight 0.8, etc. Note that each edge in the GRAPHFILE appears twice, e.g., 0 1 0.8 and 1 0 0.8, and nodes are consecutive integers starting from 0.

-priorfile PRIORFILE

PRIORFILE stores the prior probabilities of all nodes of being benign.

The format of PRIORFILE is as follows

```
0 0.9
1 0.5
2 0.1
3 0.8
...
```

It means that node 0 has a prior probability 0.9 of being benign; node 2 has a prior probability 0.1 of being benign (or to say, prior probability 0.9 of being Sybil), etc.

Note that these prior probabilities can be user-defined for labeled benign nodes or/and labled Sybil nodes or/and unlabeled nodes. Or, they can be also learnt via a machine learning classifier. For example, we can extract local node features to train a binary classifier, which produces the probability of being benign for each node. Then, such probabilities can be treated as nodes' priors.

-trainfile TRAINFILE

TRAINFILE consists of two lines, where the first line includes a set of labeled benign nodes, separated by space; and the second line includes a set of Sybil nodes.

The format of TRAINFILE is as follows

```
1 2 4 6 7 8 9 10 ...
0 3 5 ...
```

It means that "1 2 4 6 7 8 9 10 ..." are labeled benign nodes and "0 3 5 ..." are labeled Sybil nodes.

-mIter MAXITER

MAXITER sets the number of maximum iterations. By default, MAXITER=5.

-tp THETA_POS

THETA_POS sets the prior probability of being benign for labeled benign nodes. By default, THETA_POS=0.9.

-tn THETA_NEG

THETA_NEG sets the prior probability of being benign for labeled Sybil nodes. By default, THETA_NEG=0.1.

-tu THETA_UNL

THETA_UNL sets the prior probability of unlabeled nodes. By default, THETA_UNL=0.5.

-nt NUM_THREADS

NUM_THREADS is the number of threads. By default, NUM_THREADS=1.

-wg WEIGHTED_GRAPH

WEIGHTED_GRAPH indicates whether the considered graph is weighted (WEIGHTED_GRAPH=1) or not (WEIGHTED_GRAPH=0). If the graph is weighted, then the weights of all edges can be user defined and are stored in the third column of the GRAPHFILE.

Otherwise, if it is unweighted, then the parameter -wei WEIGHT can be used to set the SAME weight for all edges. By default, WEIGHTED_GRAPH=0 and WEIGHT=0.9.

2.2 Output

-postfile POSTFILE

PRIORFILE stores the final posterior probabilities of all nodes after running SybilBelief.

The format of POSTFILE is as follows

```
0 1.0
1 0.8
2 0.1
...
```

It means that node 0 has a posterior probability 1.0 of being benign; node 1 has a posterior probability 0.8 of being benign; node 2 has a posterior probability 0.1 of being benign (or to say, probability 0.9 of being Sybil), etc.

2.3 Usage

```
Compile: g++ GANG.cpp -pthread -O3 -o GANG
Execute: ./GANG -graphfile GRAPHFILE -trainfile TRAINFILE -postfile POSTFILE
[-priorfile PRIORFILE] [-nt NUM\_THREADS] [-mIter MAXITER] [-tp THETA\_POS]
 [-tn THETA\_NEG] [-tu THETA\_UNL] [-wg WEIGHTED\_GRAPH] [-wei WEIGHT]
```

3 Metric Calculation

The code "metric.cpp" produces several metrics, e.g., accuracy, precision, recall, etc., to evaluate the overall proframance of an algorithm.

3.1 Input

-testfile TESTFILE

TESTFILE consists of two lines, where the first line includes a set of labeled benign nodes, separated by space; and the second line includes a set of Sybil nodes.

-postfile POSTFILE

POSTFILE is the output file generated by the GANG algorithm

3.2 Output

The output includes the following metrics:

```
TPR (True positive rate)
FPR (False positive rate)
TNR (True negative rate)
ACC (Accuracy)
Precision (Precision)
FNR (False negative rate)
Recall
AUC (Area under the curve)
```

3.3 Usage

```
Compile: g++ metric.cpp -O3 -o metric
Execute: ./metric -testfile TESTFILE -postfile POSTFILE > METRICFILE
```

4 Top Interval Ranking

The code "victim-top-interval.py" produces the top interval ranking result of users in the testfile.

4.1 Input

TESTFILE: it consists of two lines, where the first line includes a set of labeled benign nodes, separated by space; and the second line includes a set of Sybil nodes.

POSTFILE: it is the output file generated by the GANG algorithm.

NodePerIntervel: it indicates the number of users in each interval.

Intervals: it indicates the number of intervals.

4.2 Output

Suppose NodePerIntervel=1,000 and Intervals=10. The output displays the following information

```
1000 XX
1000 XX
...
1000 XX
```

where it has 10 lines and XX is a number that is less than or equal to 1,000.

4.3 Usage

```
python victim-top-interval.py TESTFILE POSTFILE NodePerIntervel Intervals >TOPRANKFILE
```