NOISY SUBGRAPH ISOMORPHISMS ON MULTIPLEX NETWORKS

Hui Jin, Xie He, Yanghui Wang, Hao Li, Andrea Bertozzi

SIAM Workshop on Network Science 2020
July 9–10 · Toronto

Summary

We focus on finding as many noisy subgraph isomorphisms within a noise tolerance as possible on large multiplex networks. Our main contribution is to propose novel heuristics based on the well-know A* search algorithm to estimate the number of missing edges of subgraph matches. We tested our method on one of the synthetic multiplex networks from the Modeling Adversarial Activity program of the Defense Advanced Research Projects Agency.

Note, the content of this abstract comes from a published paper in 2019 at the IEEE Big Data Conference in Los Angeles, California, in which the writer of the abstract (Xie He) is a co-author on and is intended to present during the SIAM NS 2020. This is not intended as a paper submission but as an abstract of a talk/poster.

Method

Subgraph isomorphism problems in general are NP-complete problems. For exact subgraph isomorphism we have the freedom to use in/out degree of nodes and other basic graph properties to rule out many nodes in the world graph, but this would be extremely different for noisy subgraph isomorphism: as important information such as node degrees are now inaccurate for each node. From the application stand point of view, one could consider noises such as node/edge insertions, deletions,and mismatches as noises. It would be very important to find a template in large noisy multiplex networks for many applications in sociological and biological science. Thus, we seek to solve this problem. We start by fixing two concepts: the definition of noisy subgraph isomorphism problem and the way to calculate/estimate missing edges and thus find the possible candidates of the template graph.

Models

We built two different noisy models by: 1) inserting edges in the template graph; 2) removing edges from the world graph. 1 and 2 describe these two model seperately. They provide us with the basic idea of how this problem might be solvable: by calculating the number of possible missing edges in the world graph, we assign a cost to a group of candidate world nodes to be the actual template. By setting a range of the cost, we are able to find the noisy subgraph isomorphism, denoted as NSI in the following paragraphs.

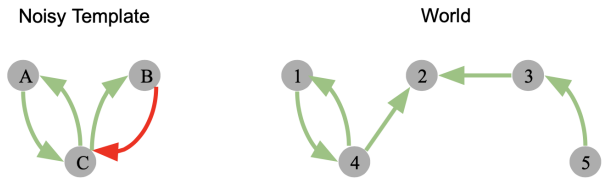

Figure 1: Template noise toy model where the added edge is edge (B, C), shown in red.

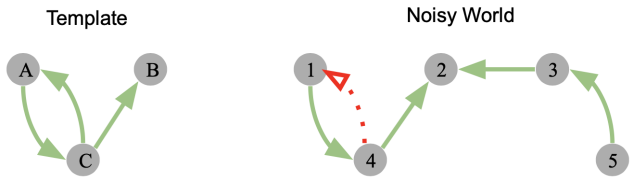

Figure 2: World Graph noise toy model with 20% noise; the removed edge is (4,1).

Algorithms

In the A* search algorithm, we need a heuristic function defined on each node in the search tree. The heuristic estimates the cost of the best path from the root node through the current node to a leaf node. In our problem setting, given a partial match, we need to estimate the cost of the best complete match extended from the current partial match. Here, estimating the cost is equivalent to estimating the number of missing edges.

In the previous work we done in [2] about exact subgraph isomorphism, we proposed a filtering algorithm to filter out nodes that are not possible to be the candidate of template graph. We borrowed this idea and developed two estimation to calculate the potential cost in the noisy

case. The Statistics Estimation uses in-degree and out-degree to estimate the number of missing edges of a node. The Topology Estimation estimates the number of missing edges for each pair of nodes in the template graph. In short, one is counting the missing edges itself, and the other is counting the missing edges in a node's neighborhood if it becomes one of the template node.

After we have the Statistics Estimation and the Topology Estimation, we use the A* search algorithm to find noisy subgraph isomorphisms. A partial match P is regarded as a state in the A* search algorithm. The f value of the state is evaluated by TopoEst(P). The detailed procedure is described in Algorithm 1. We start with the empty match where no nodes are matched. In each iteration of the while loop, we pick out a partial match with the lowest f value from the openList. If all template nodes are matched, we find one subgraph match with low cost. Otherwise we pick an unmatched node which minimizes the increment of TopoEst(P) and try to match it to any one of the unmatched world nodes. We generate lots of new partial matches in this procedure. Then we calculate the f values of these partial matches and add them to openList.

Algorithm 1: A* Search Algorithm

```
Put the empty match on the openList;
while openList is not empty do
    currentState = state with the lowest f value in
     openList;
    Remove currentState from openList;
    if all template nodes in currentState are
     matched then
        add currentState to solutionList;
    end
    Pick an unmatched template nodes u in
     currentState ;
    for each candidate v of u do
        newState.partialMatch =
         currentState.partialMatch + {u → v} ;
        newState.f =
         TopoEst(newState.partialMatch);
        add newState to openList;
    end
end
return solutionList
```

Results

We have applied our algorithm to the dataset developed by Pacific Northwest National Laboratory (PNNL) [1] as part of the DARPA-MAA program [3]. For the two different noise models we proposed, we can identify multiple NSIs. With 0.1 percent of missing edges in the world graph, we were able to find 22922 noisy subgraph isomorphsim for the tempalte graph. Given the size of the data set, which has 22996 nodes and 12381816 edges in the world graph, and 75 nodes and 1620 edges in the template graph, our result proved that our method is useful on large networks. Details of the results of the two different models are provided in 3 and 4

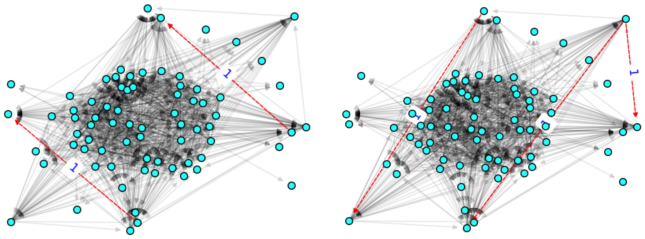

Figure 3: Two NSIs we find on PNNL V6 B0 with the template noise model. Left panel: We add 2 edges to the original template graph and present a NSI with two missing edges. Right panel: We add 3 edges to the original template graph and present a NSI with three missing edges. The missing edges are marked as red dashed lines with the numbers of missing edges.

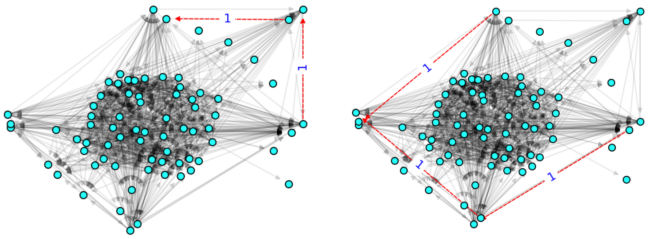

Figure 4: Two NSIs we find on PNNL V6 B0 with the world graph noise model. Left panel: We remove 0.1% edges from the original world graph and present a NSI with two missing edges. Right panel: We remove 0.2% edges from the original world graph and present a NSI with three missing edges. The missing edges are marked as red dashed lines with the numbers of missing edges.

Conclusion and Future Work

Overall, our algorithm is very successful in dealing with large multiplex networks. In the future, we could take other factors into consideration when we design the cost of subgraph matches. For example, additional edges and missing edges can both be considered as cost. We could modify our algorithm and try to find the match with the smallest number of edge differences.

Further, we could use parallel computing to further speed up our code. We can parallelize the Statistics Estimation and the Topology Estimation on each node and edge. Since we identify that these computations as bottlenecks, we expect significant speedup after parallelization.

ACKNOWLEDGMENT

This material is based on research sponsored by the Air Force Research Laboratory and DARPA under agreement number FA8750-18-2-0066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not with standing any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and DARPA or the U.S. Government. Data provided by the MAA groups at Pacific Northwest National Laboratory. We thank Thien Nguyen, Dominic Yang, Yurun Ge, Jacob Moorman, Thomas Tu, Qinyi Chen, Mason Porter for useful conversations.

References

[1] J. A. Cottam, S. Purohit, P. Mackey, and G. Chin. Multichannel large network simulation including adversarial activity. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3947–3950, Dec 2018.

[2] J. D. Moorman, Q. Chen, T. K. Tu, Z. M. Boyd, and A. L. Bertozzi. Filtering methods for subgraph matching on multiplex networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3980–3985. IEEE, 2018.

[3] B. Onyshkevych. Modeling adversarial activity (maa).