InvestorCliques (796315) – EU-Project Milestone 1.2 Network Simplification Algorithm*

Le Viet Hung†

Department of Computing Sciences,

Tampere University, Tampere Finland

* I thank Mr. Kestutis Baltakys and Prof Juho Kanniainen for comments and support. A joint paper on the same topic will be published later.

† http://www.investorcliques.eu.; hung.le@tuni.fi

SUMMARY

Several complex systems have been studied and modeled as weighted graphs or networks. Food webs, eBay network, social networks, communication networks, the World Wide Web are good examples. The structure of such graphs provides important information on the features of the investigated systems. However, the complexity of the system is generally reflected in the densely connected structure. Especially the investor stock trading correlation network in our context is a complete graph. In the absence of any filtering procedure, all links among elements are present. Therefore, a general need to develop a source code that are able to filter out the noise and extract the core information of the investigated system. In addition, simplified graphs can be used to improve the visualization of a network and extract its main structure, or as a pre-processing step for other data mining algorithms. The aim of this milestone is to implement the existing network simplification techniques for the investor network:

- Maximum and Minimum Spanning Tree [1]
- Planar Maximally Filtered Graph[2]
- Minimum Loss Connectivity[3]

DATA

We first use the input correlation network from our lab's previous work [4]. In addition, we estimate network using [5]. The data used in this study is the central register of shareholdings for Finnish stocks from Finnish central depository, provided by Euroclear Finland. Our sample data consists of the marketplace transactions of 100 Finnish stocks consisting of investor's transactions around dot-com bubble from 1 January 1998 to 1 January 2002. A more detailed description of the data set is provided in Refs [4, 6–9].

IMPLEMENTATION

Maximum and Minimum Spanning Tree

We use Kruskals algorithm [1] implemented in networkx for finding a graph's spanning tree of minimum length. It sorts the links of a graph ascending order and then repeatedly adds links that connect separate nodes until the graph is connected. By negating the edge weights, the algorithm can be employed to find a maximum spanning tree. The advantage is the visualization. The disadvantage is that a large number of links are aggressively removed. In addition, in our context, the minimum spanning tree reflects distant trading strategies while the maximum one captures the herding pattern around bubbles. They are somehow conflict and not able to represent for the whole network.

Planar Maximally Filtered Graph (PMFG)

A planar graph is a graph that can be embedded in the plane. It can be drawn on the plane in such a way that its edges intersect only at their endpoints and no edges cross each. The name of the method spells out its implementation. However, the existing implementation (https://gmarti.gitlab.io/networks/2018/06/03/pmfg-algorithm.html) weight all links equally. We first remove less significant links which either have high p-value or weight in range [-0.1, 0.1], then apply the implementation. The detailed technical issues can be found at references.

The advantage of PMFG is that it will keep more links of the input networks while still having a meaningful visualization. Let n be the number of nodes, then the MST has n-1 links whereas the PMFG has 3*(n - 2) links. Furthermore, the MST is always contained in the PMFG.

Minimum Loss Connectivity

The main purpose is to filter out less important links and keep only the main representative links. Links are removed aggressively until network disconnected or number of removed exceed the input threshold. There are several version of this algorithm: naive approach, path simplification and the combine approach. In this milestone, we implemented the naive approach for the investor network. The path simplification and combine approach are compatible for network whose link weights are positive while the Pearson correlation of investor trading behavior can be negative. The link list is first sorted in an ascending order. Then, less important links on the top are removed iteratively. The algorithm stops when the number of edges removed reaches the input threshold or leads to disconnected components.

Furthermore, we test and compare all the three methods using the input correlation network of investor stock trading around dot-com bubble.

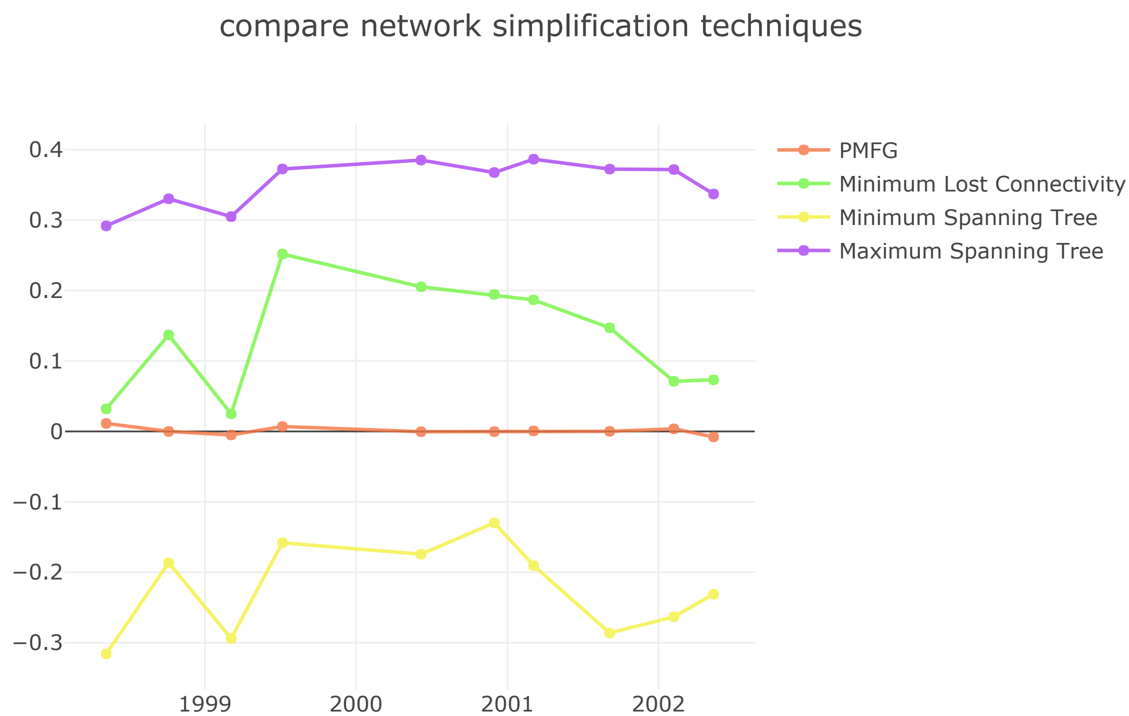

FIG. 1. The time evolution of the average link weight of the simplified networks. The original networks are constructed for within six-month time windows.

AVAILABILITY

Source code Python can be found at: http://www.investorcliques.eu/category/programming/

REFERENCE

https://networkx.github.io/documentation/networkx-1.10/reference/algorithms.mst.html

https://gmarti.gitlab.io/networks/2018/06/03/pmfg-algorithm.html

https://github.com/hagberg/planarity

http://jgaa.info/accepted/2004/BoyerMyrvold2004.8.3.pdf

https://networkx.github.io/documentation/stable/index.html

[1] J. B. Kruskal, Proceedings of the American Mathematical society **7**, 48 (1956).

[2] M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna, Proceedings of the National Academy of Sciences **102**, 10421 (2005).

[3] F. Zhou, S. Malher, and H. Toivonen, in *Data Mining (ICDM), 2010 IEEE 10th International Conference on* (IEEE, 2010) pp. 659–668.

[4] S. Ranganathan, M. Kivelä, and J. Kanniainen, PloS one **13**, e0198807 (2018).

[5] M. Tumminello, S. Micciche, F. Lillo, J. Piilo, and R. N. Mantegna, PloS one **6**, e17994 (2011).

[6] M. Grinblatt and M. Keloharju, Journal of financial economics **55**, 43 (2000).

[7] M. Tumminello, F. Lillo, J. Piilo, and R. N. Mantegna, New Journal of Physics **14**, 013041 (2012).

[8] K. Baltakys, J. Kanniainen, and F. Emmert-Streib, Scientific reports **8**, 8198 (2018).

[9] F. Musciotto, L. Marotta, J. Piilo, and R. N. Mantegna, Palgrave Communications **4**, 92 (2018).