Anomaly Analysis and Visualization through Compressed Graphs

Qi Liao*
Central Michigan University, USA

Lei Shi†
Chinese Academy of Sciences

Xiaohua Sun‡
Tongji University, China

Abstract

Large-scale networks ranging from enterprise networks to social networks have grown rapidly. However, understanding these networks has been falling behind due to the high dynamics and complexity of larger networks. We propose a novel method to analyze and visualize network anomaly using topology preserving compressed graphs, which scale to millions of nodes and effectively reduce the analytic complexity of big graph visualization.

1 Introduction

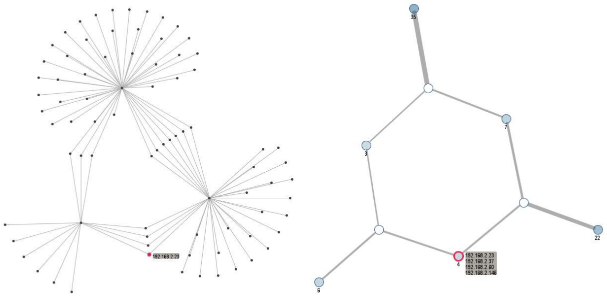
Network technologies have evolved rapidly, e.g., the emergence of Internet of Things (IoT) means an order of magnitude increase of interconnected devices such as smart phones, sensors, environmental meters, wearable devices, appliances and vehicles. one popular social network approaches one billion users. These large-scale networks can be naturally represented as graphs, to which we refer as the "big graphs". Understanding anomalies in these big graphs [1, 2] is crucial in many cases. For example, a cloud system administrator needs to keep track of the traffic distribution among servers and hosts for a better network/virtual-machine optimization. Network managers also need to monitor the latest traffic graphs to improve situation awareness, real-time troubleshooting and security-related investigation.

Anomaly analysis and visualization of large graphs remains challenging due to the non-linear increase of complexity and highly dynamic nature of such large networks. For example, mobile nodes can join and leave a network at any time and the network topologies are constantly changing. Even from a graph drawing point of view, visualizing a graph with more than approximately a hundred nodes faces two fundamental challenges. First, the classical force-directed methods in most cases fail to calculate an optimally aesthetic graph layout in real time. Second, even if graph layout can be computed, the visual clutters (mainly due to the edge crossings) created by the straight-line node-link representation prohibit the user from understanding the graph in details, which is important for analytical tasks.

To that end, we propose a novel graph visualization technique, i.e., compressed graphs, on which a overview+detail visual analytic tool was developed for analyzing anomalies in large-scale network graphs. The compressed graphs group the nodes with the similar neighbor sets into mega-nodes. Unlike clustering (or community detection), the proposed compressed graph visualization does not lose any topology information, has a much lower computational complexity, and scales to analyze graphs with millions of nodes.

2 Compressed Graph Visualization

We propose the topology-preserving compressed graph, which groups the graph nodes with similar neighbor sets together as mega-nodes and regenerates a compressed graph for the subsequent visualization and analysis. For example in Figure 1, the host



(a) Original graph

(b) Compressed graph

Figure 1: Concept illustration of topology-preserving compressed graph.

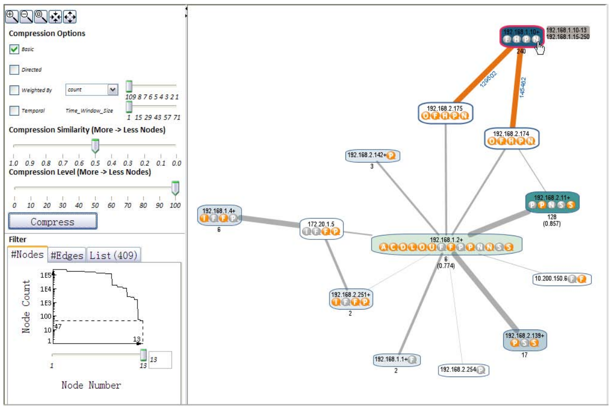

Compression Options
☑ Basic
□ Directed
□ Weighted By count
□ Temporal Time_Window_Size
Compression Similarity (More -> Less Nodes)
Compression Level (More -> Less Nodes)
Compress
Filter
#Nodes #Edges List (409)

Figure 2: The user interface for the compressed graph visualization

"192.168.2.23" connects to two hub nodes and is surrounded by three other hosts with the same connection pattern. As many similar subgraphs like this are embedded in a larger graph, the resulting redundancy of nodes and links can distract the user in understanding the graph information.

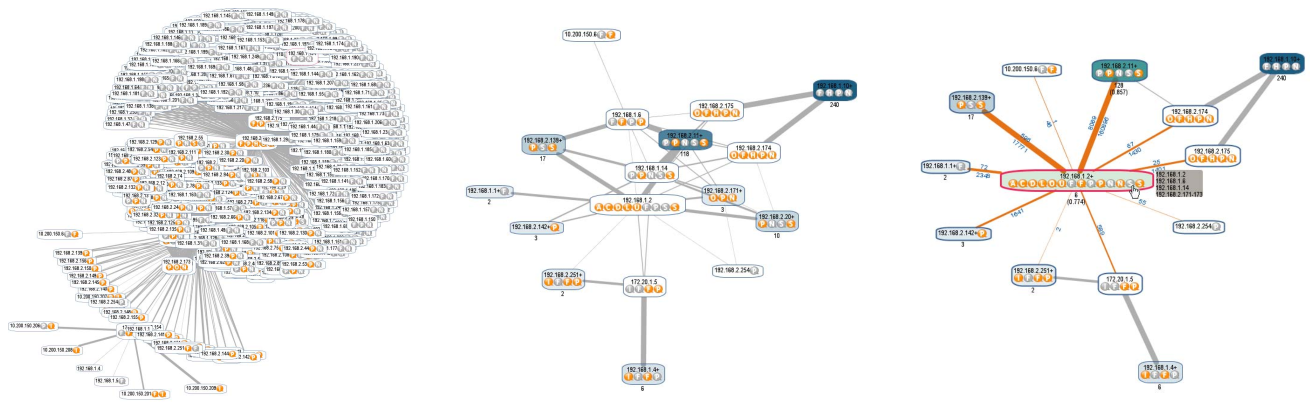
For formal definition, let $G = (V, E)$ be a directed, weighted and connected graph where $V = \{v_1, ..., v_n\}$ and $E = \{e_1, ..., e_m\}$ denote the node and link set. Let W be the graph adjacency matrix where $w_{ij} > 0$ indicates a link from v_i to v_j, with w_{ij} denoting the link weight. In each row of W, $R_i = \{w_{i1}, ..., w_{in}\}$ denotes the row vector for node v_i, representing its connection pattern. The compressed graph is denoted as $G^* = (V^*, E^*)$.

The basic algorithm takes the graph as a simple, undirected and unweighted one by setting $w_{ii} = 0$ and $w_{ij} = w_{ji} = 1$ for any $w_{ij} > 0$. On graph G, order its node list by the corresponding row vectors $R_i (i = 1, ..., n)$. For any collection of nodes with the same row vector (including the single outstanding node), aggregate them into a new mega-node $Gv_i = \{v_{i_1}, ..., v_{i_k}\}$. All Gv_i form the node set V^* for the compressed graph G^*. Also let $fv_i = v_{i_1}$ denote the first sub-node in Gv_i. The link set E^* in G^* are generated by simply replacing all fv_i with Gv_i in the original link set, and removing all the links not incident to any fv_i. In addition, we have extended the basic compression algorithm to support directed, weighted, and

[*] e-mail: qi.liao@cmich.edu

[†] e-mail: shil@ios.ac.cn

[‡] e-mail: xsun@tongji.edu.cn



(a) Original

(b) Compressed

(c) Compressed with similarity = 0.5

Figure 3: Overview traffic graphs of AFC corporate network with different settings.

Table 1: Performance evaluation on VAST Challenge dataset.

Data	nodes (before)	edges (before)	nodes (after)	edges (after)	rate (edges)	time (compress)	layout (before)	layout (after)
undirected sim=1	409	1613	17	50	96.9%	0.007	0.24	0.084
undirected, sim=0.8	409	1613	16	39	97.6%	0.012	0.242	0.088
undirected, sim=0.5	409	1613	13	23	98.6%	0.006	0.25	0.079
directed sim=1	409	1613	26	82	94.9%	0.005	0.245	0.084

Table 2: Performance evaluation on Honeypot dataset.

Data	nodes (before)	edges (before)	nodes (after)	edges (after)	rate (edges)	time (compress)	layout (before)	layout (after)
undirected	15380	16353	2	2	99.9%	0.123	10.179	0.079
undirected weighted #bin=10	15380	16353	5	8	99.9%	0.151	10.179	0.692
undirected	44668	45582	2	2	99.9%	0.401	35.09	0.08
undirected	1051595	1158150	2	2	99.9%	4.56	(500) 36.404	0.026
undirected dynamic #win=1	43602	47752	9	16	99.9%	1.27	33.504	0.1
undirected dynamic weighted #win=1 #bin=10	43602	47752	105	208	99.6%	1.102	33.504	0.946

dynamic graphs by generalizing the definition of adjacency matrix and the corresponding row vectors.

Unlike graph clustering, the proposed compressed graph can reduce the visual complexity without losing any graph information while preserving many critical features from the original graph, making it not only easy and but also accurate for human understanding and analysis. The right panel of Figure 2 gives an example of the compressed graph visualization after the basic compression algorithm. The mega-nodes are differentiated from the single-nodes by the node fill color, i.e., a larger group is filled with the more saturated color. Node labels of the mega-nodes are created by aggregating the labels of the sub-nodes in the original graph (+ sign) and become visible upon a mouse-over or click action. Various anomaly icons, each of them representing one specific type of anomaly, are used for easy analysis.

The visualization supports basic graph interactions, such as geometric zoom&pan, node drag&drop, node/link highlight&selection, as well as advanced node/link visual mappings. Beyond that, more controls over the compressed graph setting are accessible through a control panel as in the left of Figure 2. In this “Compression Options” control panel, multiple checkboxes allow user interaction for the basic, directed, weighted and dynamic compressed graphs.

3 Preliminary Results

We evaluate the performance in terms of the visual compression rate (by the number of edges), the compression time and the layout time before and after. The compression rate is defined by $\Gamma = 1 - \frac{|E^*|}{|E|}$. Notably, for the two data sets (VAST Challenge dataset and a publicly available Honeypot dataset) we tested, compressed graphs achieves more than 90% compression rates with the basic or fuzzy algorithm (Table 1). The deterministic compressed graphs can scale to a million of nodes and multi-millions of edges with a reasonable computation time. The fuzzy compressed graph with the optimized shingle implementation supports up to graphs with 10^5 nodes and returns results in just a few seconds (Table 2).

A comparison of topology-preserving compressed graphs with the original graphs is shown in Figure 3 over VAST 2011 Mini Challenge-II dataset, which contains traffic log from a multinational corporate network, i.e., a firewall log (similar to NetFlow data), an intrusion detection system (IDS) log, a system log, and a Nessus network vulnerability scan report. The graph in Figure 3a is a network connectivity graph with anomaly icons rendered on nodes. However, the view is cluttered and messy obfuscating the analysis due to the large number of nodes and edges. The graphs in Figure 3b and 3c shows a standard compressed graph and a fuzzy compressed graph with a similarity score of 0.5 derived from the original graph. With much less nodes and edges, it is easier for the investigator to analyze the anomalies on the transformed compressed graph, e.g., compromised machines 192.168.2.174/175 conducting port scan activities, denial of service (DoS) attacks on web server 172.20.1.5, etc. More details on the interactive analysis of the compressed graph visualization can be found in the video demo (`http://cps.cmich.edu/liao1q/video/LDAVCompressedGraphs.wmv`).

4 Conclusion

Large-scale graph analysis and visualization is an inherent component of big data era. Analyzing anomalies in these large networks is important but challenging due to the non-linear dynamics and complexity as graph size increases. The proposed topology-preserving compressed graphs show promising results in reducing visual complexity of large networks, and consequently provide a time-efficient solution for big graph anomaly analysis. Through on-site live demonstration and videos, we will show the compressed graphs have many potential applications (e.g., security and network management) in different scenarios of diversified networks.

References

[1] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J. Fekete, and D. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, September 2011.

[2] P. C. Wong, P. Mackey, K. A. Cook, R. M. Rohrer, H. Foote, and M. Whiting. A multi-level middle-out cross-zooming approach for large graph analytics. In *Proceedings IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 147–154, Salt Lake City, UT, October 12-13 2009.