AFRL-AFOSR-JP-TR-2018-0070

Real-time Anomaly Detection in High-Speed Time-evolving Graphs

U Kang
SEOUL NATIONAL UNIVERSITY

09/22/2018
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOA
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
22-09-2018	Final	23 Sep 2016 to 22 Sep 2018

4. TITLE AND SUBTITLE
Real-time Anomaly Detection in High-Speed Time-evolving Graphs

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA2386-16-1-4044

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
U Kang

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SEOUL NATIONAL UNIVERSITY
SNUR&DB FOUNDATION RESEARCH PARK CENTER
SEOUL, 151742 KR

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR IOA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-JP-TR-2018-0070

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The PI was successful in this research grant. The goal of this project was to 1) develop memory-efficient and accurate local triangle counting method in a multigraph stream using fixed/varying sampling rates, and 2) detect anomalies using triangle information. They created and tested two local triangle counting methods MASCOT and FURL. Experimental results demonstrate that FURL provides the best accuracy compared to the state-of-the-art algorithm in a memory- efficient way. The PI has 1 peer reviewed papery published and 1 currently in review as a direct result of this grant award.

15. SUBJECT TERMS
Online triangle counting, Real time anomaly detection, Diagnosis, Time-evolving graph stream

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			SINGLETON, BRIANA
Unclassified	Unclassified	Unclassified	SAR		19b. TELEPHONE NUMBER *(Include area code)* 315-227-7007

Final Report for AOARD Grant FA2386-16-1-4044

"**Real-time Anomaly Detection in High-Speed Time-evolving Graphs**"

16 September, 2018

Name of Principal Investigators (PI and Co-PIs): U Kang
- e-mail address : ukang@snu.ac.kr
- Institution : Seoul National University
- Mailing Address : Dept. of CSE, Seoul National University, 1 Gwanak-ro Gwanak-gu Seoul 08826, Republic of Korea
- Phone : +82-2-880-7254
- Fax : +82-2-886-7589

Period of Performance: 09/23/2016 – 09/22/2018

Abstract:

How can we detect anomalies in real time graph streams? Real time anomaly detection in graph streams is an important task with high impact applications including cyber security, terrorist detection, fake user detection, etc. In this work, we develop methods to detect anomalies in real time graph streams using local triangle information. For the purpose, we propose local triangle counting algorithms for multigraph streams where same edges appear multiple times. We develop two algorithms MASCOT and FURL. MASCOT is a baseline method which estimate the number of triangles memory-efficiently and FURL is an improved method which uses a fixed memory size. Through extensive experiments, MASCOT and FURL show high accuracy even with small memory usages. Moreover, we detect anomalies in real-world graphs using our methods.

Introduction: Anomaly detection is an extremely important problem with various applications including cyber security, terrorist detection, fake user detection in social networks, etc. Recently, real time anomaly detection in graphs is becoming more important due to the following two reasons. First, graphs are used to model many interesting and important interactions in real world, including computer networks, covert terrorist networks, phone call networks, biological networks, etc. Second, recent graph data are generated continuously at a very high speed; thus, storing all the graph data in disk based systems and performing off-line analysis later is not a feasible option. Furthermore, many applications of anomaly detection require real time analysis.

For real time anomaly detection, we develop local triangle counting algorithms for multigraph streams, and detect anomalies in real-world graphs using local triangle counts. Although there have been several works for local triangle counting, it is still challenging to handle a massive graph due to the complexity of the problem—superlinear time on the graph size is inevitable. Also, recent real world graph streams contain duplicate edges, i.e. they are multigraph streams. Thus, designing a streaming algorithm is required for efficient online analysis of a huge graph and handling duplicate edges.

The goal of this project is to 1) develop memory-efficient and accurate local triangle counting method in a multigraph stream using fixed/varying sampling rates, and 2) detect anomalies using triangle information. We propose two local triangle counting methods MASCOT and FURL. MASCOT has four advantages: 1) uses memory space efficiently, 2) processes data in real time, 3) provides good accuracy, and 4) handles duplicate edges for multigraph. However, MASCOT has a limitation: its memory usage is proportional to the number of edges since it uses a fixed sampling rate and it causes out-of-memory error when the number of edges is very large. FURL solves this problem using only a fixed size memory. FURL uses a varying sampling rate which provides uniform sampling on all the edges regardless of the number of edges. Additionally, FURL improves the accuracy of estimation reducing variance.

There are two ways to count triangles in multigraph: binary and weighted counting. Binary counting is local triangle counting on the corresponding simple graph of a streamed multigraph without explicit

graph conversion. Weighted counting is local triangle counting which considers repeated occurrences of an edge as its weight and counts each triangle as the product of its three edge weights. Thus, we develop 2 versions of MASCOT and FURL. We use MULTIBMASCOT and FURLB for binary counting, and MULTIWMASCOT and FURLW for weighted counting. Moreover, we discover interesting anomalies using triangle information.

Experiment:

The experiments are performed to answer the following questions.

- How accurate are MultiBMASCOT and MultiWMASCOT?
- How accurate are FURLB and FURLW?
- What are the discoveries from the proposed methods?

We compare our proposed methods with competitors. We use the average of measurements obtained by 10 independent runs since our algorithms are randomized. We use real-world graphs which are listed in the following table.

Name	Nodes	Edges	Description
Enron	86,978	1,134,990	Enron email network
Actor	382,219	33,115,812	Actor collaboration in movies
Baidu	415,641	3,284,387	"related to" links in Baidu Encyclopedia
WikiGER	506,174	4,555,759	Communication network of the German Wikipedia
DBLP	1,314,050	18,986,618	Co-author network in DBLP
ItWiki	1,703,605	86,548,398	Hyperlinks in Italian Wikipedia
ChinWiki	1,930,275	9,359,108	Hyperlinks in Chinese Wikipedia
Bitcoin	6,297,539	28,143,065	Bitcoin transaction network
PhoneCall	26,578,926	480,652,650	Phone call history

Results and Discussion:

- Performance of MultiBMASCOT and MultiWMASCOT

We evaluate MultiBMASCOT and MultiWMASCOT. We calculate Pearson correlation coefficient to evaluate our methods.

Figure 1 shows Pearson correlation coefficient of MultiBMASCOT and MultiWMASCOT over the ratio of the number of sampled edges. For all graphs, MultiBMASCOT and MultiWMASCOT show high correlations even with small memory usages.

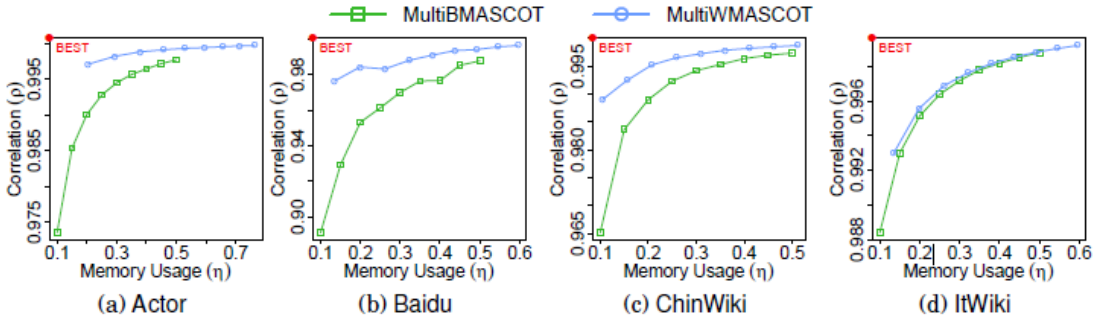

Figure 1. Pearson correlation coefficient over the ratio of sampled edges for MultiBMASCOT and MultiWMASCOT.

Figure 2 shows the mean of absolute relative error of MultiBMASCOT over the ratio of sampled edges. MultiBMASCOT, which works directly on the multigraph stream, performs as good as MASCOT-C, a local triangle counting method which requires converting a multigraph stream to the corresponding simple graph.

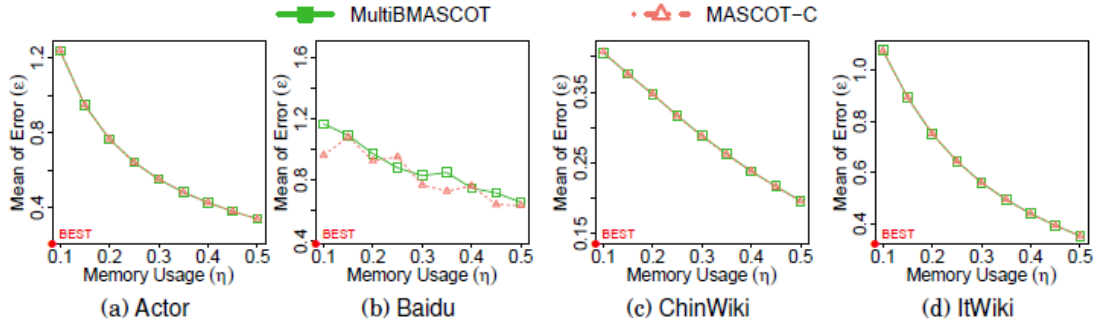

Figure 2. Mean of absolute relative error over the ratio of the number of sampled edges for MultiBMASCOT, compared with MASCOT-C on the corresponding simple graph stream.

- Performance of FURL$_B$ and FURL$_W$

We compare FURL with state-of-the-art methods. FURL-0 is an extended version of MASCOT which uses only a fixed memory, and FURL reduces variance of FURL-0 to improve the accuracy.

Figure 3 shows the comparison between FURL$_B$, FURL-0$_B$, and two competing methods PartitionCT and MultiMascot$_B$ for binary counting in terms of error over the memory usage. MultiMascot$_B$ denotes MultiBMASCOT, and PartitionCT is the state-of-the-art method for binary local triangle counting. Note that our proposed method FURL$_B$ outperforms MultiMascot$_B$ and PartitionCT. The MRE (mean of relative error) of FURL$_B$ is 1.12x ~ 2.57x and 1.11x ~ 2.55x smaller than that of MultiMascot$_B$ and PartitionCT, respectively. Also, the MRE of FURL$_B$ is 1.10x ~ 1.23x smaller than that of FURL-0$_B$. In terms of memory efficiency, as in the case of weighted counting, FURL$_B$ requires the smallest memory usage for a given error level.

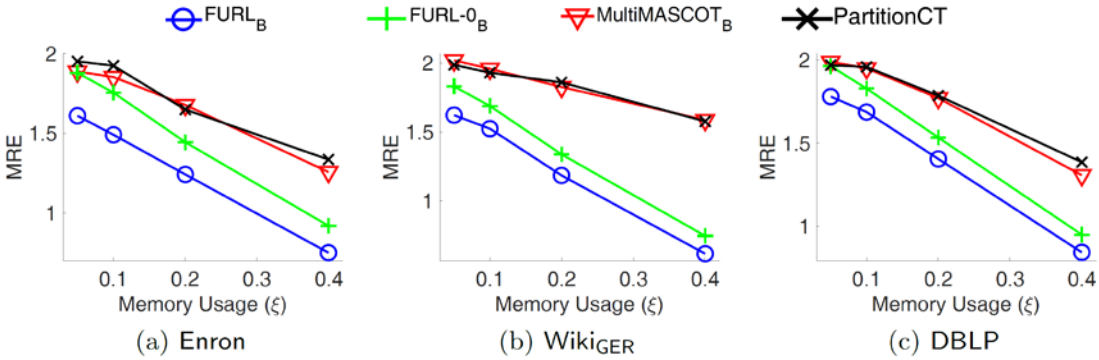

Figure 3. Mean of relative error (MRE) vs. memory usage of FURL$_B$, FURL-0$_B$ and two competing methods MultiMascotB and PartitionCT for binary local triangle counting in a multigraph stream.

Figure 4 shows the comparison between FURL$_W$, FURL-0$_W$, TRIEST and MultiMascot$_W$ for weighted counting in MRE over the memory usage. MultiMascot$_W$ denotes MultiWMASCOT, and TRIEST is the state-of-the-art method for weighted local triangle counting. Note that our proposed method FURL$_W$ outperforms the competing methods. The MRE of FURL$_W$ is 1.23x ~ 4.33x and 1.15x ~ 4.19x smaller than that of MultiMascot$_W$ and TRIEST, respectively. Also, the MRE of FURL$_W$ is 1.09x ~ 1.45x smaller than that of FURL-0$_W$. In terms of memory efficiency, FURL$_W$ requires the smallest memory usage for a given error level.

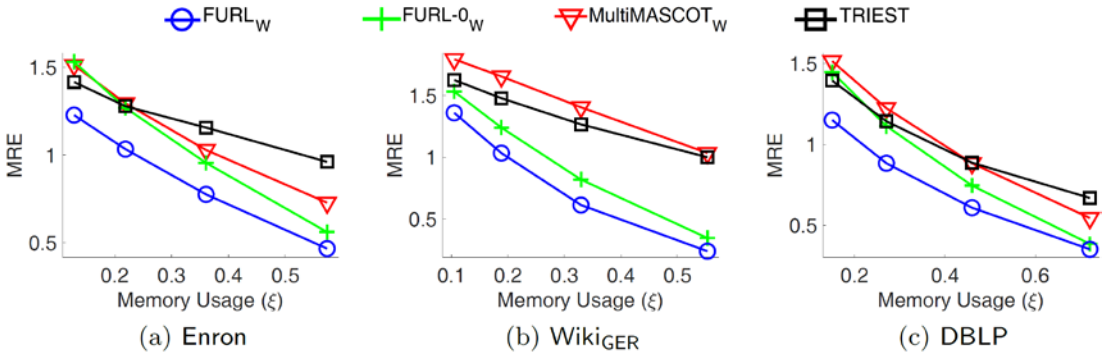

Figure 4. Mean of relative error (MRE) vs. memory usage of FURLw and FURL-0w compared to competing methods MultiMascotw and TRIEST for weighted local triangle counting.

- Discovery

We show that our proposed algorithms detect anomalous nodes in real world multigraph streams.

a) TELEMARKETERS

The PhoneCall dataset contains 'who calls whom' information. In PhoneCall, there is an anomalous person who sends numerous calls to many people who do not know each other well. Figure 5 shows such person in a red circle. We use MultiBMASCOT to estimate the number of triangles; the anomalous person shows a significantly low ratio of the number of triangles compared to degree. Such a pattern is likely to come from telemarketing, or from voice phishing in which a swindler contacts a number of arbitrary people.

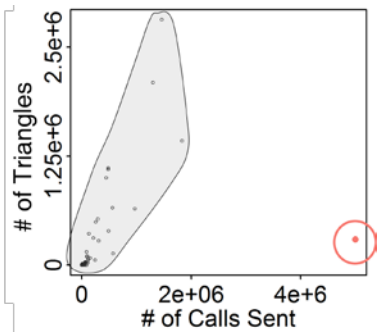

Figure 5. Degree over the number of triangles in PhoneCall. We use MultiBMASCOT to estimate the number of triangles.

b) INTENSIVE COLLABORATION

In DBLP, there are groups of authors who write many papers with each other. Figure 6 shows two anomalous groups of nodes, colored red and green, discovered in DBLP. We find that each group corresponds to co-authors writing many papers. We estimate the number of binary and weighted triangles using MultiBMASCOT and MultiWMASCOT for each node respectively. For instance, we observe the red group whose four members appear with extremely high weighted local triangle counts on the y-axis, which is caused by intensive collaboration within themselves. Each of the four authors in the red group has three internal triangles that contribute 38%, 27%, 19%, and 50% to their total weighted triangle counts but correspond to only 0.5%, 0.6%, 0.8%, and 0.7% of their total unweighted triangles, respectively. A similar pattern is also observed in the green group. A large portion of the weighted triangle counts of the three authors comes from the one internal triangle whose contribution is 50%, 51%, and 43% of the total, respectively.

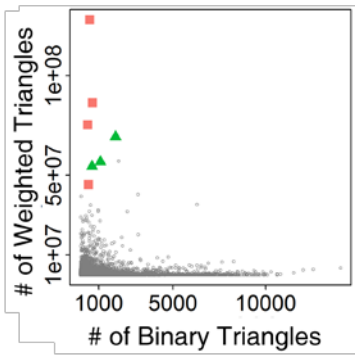

Figure 6. The number of binary triangles over the number of weighted triangles in DBLP. We use MultiBMASCOT and MultiWMASCOT to estimate the number of binary and weighted triangles respectively.

c) ANOMALY DETECTION ON BITCOIN NETWORK

We investigate anomalous nodes discovered by FURL in a real world Bitcoin network where each node is an account and an edge denotes at least one transaction between accounts. In Bitcoin, there are two anomalous accounts (marked as purple and blue points in Figure 7) with large degrees and few triangles: i.e., they make a lot of transactions to their neighbors, but the neighbors make few transactions among them. We investigate the two accounts at https://blockchain.info/tags that informs each account's corresponding website if there is any. It turns out the purple point belongs to a gambling site, and the blue point belongs to a Bitcoin mining pool. In both egonets, neighbors are sparsely connected to each other since any two accounts in them are not likely to know each other by nature. In the purple point, the account for the gambling site makes transactions to diverse people that do not interact with each other frequently. In the blue point, the account for the Bitcoin mining pool also makes transactions to people that do not make transactions with each other. Note that a Bitcoin mining pool is used by Bitcoin miners to pool their computing power and distribute the reward according to the amount they contributed; diverse identities of Bitcoin miners lead to diverse neighborhoods.

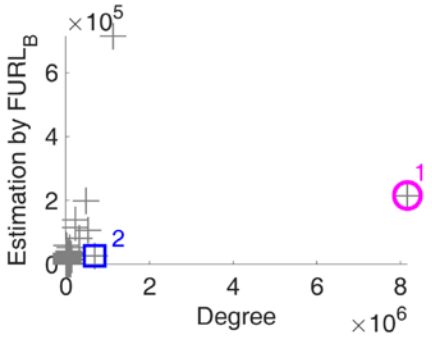

Figure 7. Degree vs. the number of local triangles estimated by FURLB in Bitcoin.

List of Publications and Significant Collaborations that resulted from your AOARD supported project: In standard format showing authors, title, journal, issue, pages, and date, for each category list the following:

a) papers published in peer-reviewed journals,

- Yongsub Lim, Minsoo Jung, and U Kang. “Memory-efficient and Accurate Sampling for Counting Local Triangles in Graph Streams: From Simple to Multigraphs” in ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 12, issue 1, Feburuary 2018.

b) papers published in peer-reviewed conference proceedings,

c) papers published in non-peer-reviewed journals and conference proceedings,

d) conference presentations without papers,

e) manuscripts submitted but not yet published, and

- Minsoo Jung, Yongsub Lim, Sunmin Lee, and U Kang. “Furl: Fixed-memory and Uncertainty Reducing Local Triangle Counting for Multigraph Streams” in Data Mining and Knowledge

Discovery.

f) provide a list any interactions with industry or with Air Force Research Laboratory scientists or significant collaborations that resulted from this work.

Attachments: Publications a), b) and c) listed above if possible.

DD882: As a separate document, please complete and sign the inventions disclosure form.

Important Note: If the work has been adequately described in refereed publications, submit an abstract as described above and refer the reader to your above List of Publications for details. If a full report needs to be written, then submission of a final report that is very similar to a full length journal article will be sufficient in most cases. This document may be as long or as short as needed to give a fair account of the work performed during the period of performance. There will be variations depending on the scope of the work. As such, there is no length or formatting constraints for the final report. Keep in mind the amount of funding you received relative to the amount of effort you put into the report. For example, do not submit a $300k report for $50k worth of funding; likewise, do not submit a $50k report for $300k worth of funding. Include as many charts and figures as required to explain the work.