SAND2014-4651P

Fast Algorithms for Evolving graphs via Assays, Sampling, & Theory
Phase II: Quarter 1 Report*

May 28, 2014

Technical POC	Administrative POC
Dr. Tamara G. Kolda	Mr. Daniel P. Fleming
Sandia National Laboratories	Sandia National Laboratories
7011 East Ave., MS 9159	1515 Eubank Blvd. SE, MS 1074
Livermore, CA 94550	Albuquerque, NM 87123
Phone: 925-294-4769	Phone: 505-845-7829
Fax: 925-294-2234	Fax: 505-284-8097
E-mail: tgkolda@sandia.gov	E-mail: dpflemi@sandia.gov

*Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Contents

1 Objectives and accomplishments for phase II, quarter 1 **3**
1.1 Wedge-sampling for counting triangles . . . 3
1.2 The scalable BTER network model . . . 4
1.3 Fast personalized PageRank computation . . . 4
1.4 Partitioning into triangle-dense subgraphs . . . 4
1.5 Using triangles to improve community detection . . . 5
1.6 Using K-cores to accelerate community finding . . . 5

2 Plans for phase II, quarter 2 **7**
2.1 Measurements and theory for time-evolving graphs . . . 7
2.2 Massive cyber networks . . . 7
2.3 Streaming with multiple time windows . . . 7
2.4 Counting 4-node patterns in graphs . . . 8
2.5 Power law is not possible for massive graphs . . . 8
2.6 FEASTPACK software . . . 8

3 Publications & References **9**

4 Other References **9**

5 Presentations **11**

6 Synergistic Activities **12**

Summary Slide **13**

1 Objectives and accomplishments for phase II, quarter 1

The overall goals of the FEAST project are as follows:

- To create techniques for "measurements" on real-world large-scale graphs,
- To design scalable network models that match real-world graph measurements, and
- To develop efficient algorithms that take advantage of structure in real-world graphs.

In this quarter, our completed milestones are focused on efficient algorithms that take advantage of structure. We also have some newly accepted journal papers on triangle counting via sampling and implementing the scalable BTER graph model.

1.1 Wedge-sampling for counting triangles

Graphs are used to model interactions in a variety of contexts, and there is a growing need to quickly assess the structure of such graphs. Some of the most useful graph metrics are based on *triangles*, such as those measuring social cohesion. Unfortunately, algorithms to compute triangle-based measures can be extremely expensive, even for moderately-sized graphs with only millions of edges. We have considered a sampling-based approach for counting triangles. Previous work has considered node and edge sampling; in contrast, we consider *wedge sampling* (i.e., 2-path sampling), which provides faster and more accurate approximations than competing techniques. Unlike node or edge sampling, wedge sampling also enables estimation of local clustering coefficients, degree-wise clustering coefficients, and uniform triangle sampling. More importantly, wedge-sampling methods come with provable and practical probabilistic error estimates for all computations. Our previous work on this topic has been award the 2013 SIAM International Conference on Data Mining (SDM) Best Paper Prize [11]. In this quarter, we have a follow-on journal version in *Statistical Analysis and Data Mining* [12] that includes more detailed numerical results and extends the work to include directed triangles. We provide extensive results that show our methods are both more accurate and faster than state-of-the-art alternatives.

We note that the sampling approach is an *algorithmic* approach that can be adapted to nearly any storage scenario such as fast databases or distributed computing environments. We have developed a MapReduce implementation of the wedge sampling approach in [2], which has been accepted for publication this quarter to the *SIAM Journal on Scientific Computing*. The goal of this paper is to show that the wedge-sampling algorithm is appropriate for massive graphs. We show results on publicly-available networks, the largest of which is 132M nodes and 4.7B edges, as well as artificially generated networks (using the Graph500 benchmark), the largest of which has 240M nodes and 8.5B edges. We can estimate the clustering coefficient by degree bin (e.g., we use exponential binning) and the number of triangles per bin, as well as the global clustering coefficient and total number of triangles, in an average of 0.33 seconds per million edges plus overhead (approximately 225 seconds total for our configuration). The technique can also be used to study triangle statistics such as the ratio of the highest and lowest degree, and we highlight differences between social and non-social networks. To the best of our knowledge, these are the largest triangle-based graph computations published to date.

In related work that was also funded as part of this project, a streaming version of the wedge sampling method is presented in [8], which received the best student paper prize at KDD'13.

1.2 The scalable BTER network model

Network data is ubiquitous and growing, yet we lack realistic generative network models that can be calibrated to match real-world data. The recently proposed Block Two-Level Erdős-Rényi (BTER) model [10] can be tuned to capture two fundamental properties: degree distribution and clustering coefficients. The latter is particularly important for reproducing graphs with community structure, such as social networks. Our paper on how to implement the BTER model has been accepted for publication in the *SIAM Journal on Scientific Computing* [1]. In this paper, we compare BTER to other scalable models and show that it gives a better fit to real data. In particular, BTER gives a much better fit than that Stochastic Kronecker Graph (SKG) model, also known as R-MAT [7, 9]. Both methods are easily parallelized and require the same amount of work per edge. BTER is much easier to fit to real-world data than SKG. We provide a scalable implementation of BTER that requires only $O(d_{\max})$ storage where $d_{\max}$ is the maximum number of neighbors for a single node. The generator is trivially parallelizable, and we show results for a Hadoop MapReduce implementation for a modeling a real-world web graph with over 4.6 billion edges. We propose that the BTER model can be used as a graph generator for benchmarking purposes and provide idealized degree distributions and clustering coefficient profiles that can be tuned for user specifications.

1.3 Fast personalized PageRank computation

We have completed task C.3. In collaboration with Ashish Goel and others at Stanford, the goal was to develop methods for faster personalized PageRank computations using sublinear time algorithms. Finding communities around a vertex often involve Personalized PageRank queries, but these are too expensive to compute from all vertices. We proposed to use sampling methods based on collision statistics to design faster algorithms. This is closely related to sublinear time algorithms for finding short paths between vertices.

The paper [4] has been accepted for publication at KDD'14. The paper proposes a new algorithm, FAST-PPR, for the significant PageRank problem: given input nodes s and t in a directed graph and a threshold δ, decide if the personalized PageRank from s to t is at least δ. Existing algorithms for this problem have a running-time of $\Omega(1/\delta)$; this makes them unsuitable for use in large social-networks for applications requiring values of $\delta = O(1/n)$. FAST-PPR is based on a bidirectional search and requires no preprocessing of the graph. It has a provable average running-time guarantee of $O(\sqrt{d/\delta})$, where d is the average in-degree of the graph. We complement this result with an $\Omega(1/\sqrt{\delta})$ lower bound for significant PageRank, showing that the dependence on δ cannot be improved. We perform a detailed empirical study on numerous massive graphs showing that FAST-PPR dramatically outperforms existing algorithms. For example, on the a Twitter graph with 1.5 billion edges, FAST-PPR has a 30 factor speedup over the state of the art. With some additional preprocessing, we show how to reduce the time ever further. An enhanced version of FAST-PPR using global PageRank runs at least twice as fast on all our candidate graphs.

1.4 Partitioning into triangle-dense subgraphs

We have addressed task D.1. The goal is to find community by using subgraphs. We had originally proposed 4-vertex patterns and will continue that line of investigation (see our plans for this coming quarter). Nevertheless, we have also considered the case of k-cliques for finding community

structure. As a side benefit, we can quickly estimate the number of k-cliques in a network.

In an unpublished manuscript [6] (being submitted for publication), we consider the following. A clique of size r is a complete graph on r vertices and represents the most cohesive subgroup of that size. Counting the number of small cliques is a fundamental operation in graph analysis. These numbers are an important part of graphlet analysis in bioinformatics, subgraph frequency counts in social network analysis, and are input to graph models (like exponential random graph models). The simplest non-trivial incarnation of this problem is triangle counting, with $r = 3$. Triangle counting has a long and rich history in data mining research, but there has been little work done for larger values of r. We provide one of the first scalable algorithms for counting r-cliques in social networks. Our algorithm is based on a recent theoretical result on triangle-preserving decompositions for social networks. We implement a novel decomposition procedure that partitions a graph into dense, triangle-rich subgraphs. We exploit these special properties by approximately counting small cliques in these subgraphs using sampling methods. Our algorithm scales to millions of edges and approximates r-clique counts for r up to 9. It runs orders of magnitude faster than enumeration schemes. For example, for 8-clique counts, enumeration takes more than a day, while our algorithm terminates in minutes. The output is accurate, with error within 10% for almost all instances, and largely within 5%.

1.5 Using triangles to improve community detection

We have completed Task G.1 which had the goal to develop community detection methods that exploit information in directed graphs such as the existence of reciprocal edges and 3-cycles. Such information can be used to determine groups of nodes that should not be separated; i.e., those nodes can be given higher weights for similarity so that they are in the same community.

We consider the problem in a paper that has been accepted for publication in The Second ASE International Conference on Big Data Science and Computing (BigDataScience, Stanford, CA, May 2731, 2014) [3]. There are a variety of metrics that can be used to specify the quality of a given community, and one common theme is that flows tend to stay within communities. Hence, we expect cycles to play an important role in community detection. In directed graphs, our new idea is based on the four types of directed triangles that contain cycles. To identify communities in directed networks, we develop an undirected edge-weighting scheme based on the type of the directed triangles in which edges are involved. To demonstrate the impact of our new weighting, we use the standard METIS graph partitioning tool to determine communities and show experimentally that the resulting communities result in fewer 3-cycles being cut. The magnitude of the effect varies between a 10 and 50% reduction, and we also find evidence that this weighting scheme improves a task where plausible ground-truth communities are known.

1.6 Using K-cores to accelerate community finding

We have completed Task G.2, which was focused using K-cores to accelerate community finding. The majority of nodes in a graph are of low degree and have little impact on the community structure. We proposed to improve the efficiency of the stochastic blockmodel fitting procedure by focusing on small, densely connected cores. We have (increasing) reason to believe that the hearts of communities are cliques or near-cliques. Using k-cores greatly reduces and simplifies the community detection problem.

Ultimately, in the preprint [5] (submitted for publication), we developed a general approach that can be used with any community finding method, including the stochastic blockmodel. The K-core of a graph is the largest subgraph within which each node has at least K connections. The key observation of this paper is that the K-core may be much smaller than the original graph while retaining its community structure. Building on this observation, we propose a framework that can accelerate community detection algorithms by first focusing on the K-core and then inferring community labels for the remaining nodes. Finally, label propagation and local modularity maximization algorithms are adopted, for their speed and quality, to optimize the community structure of the whole graph. Our experiments demonstrate that the proposed framework can reduce the running time by up to 80% while preserving the quality of the solutions. Theoretical investigations via the likelihood function based on stochastic blockmodels support our framework.

2 Plans for phase II, quarter 2

2.1 Measurements and theory for time-evolving graphs

Related to Tasks A.1–A.3, we will pursue a couple of different approaches.

First, we will develop methods for comparing graphs based on changes in structure. These techniques may be used to detect changes in networks but also to measure the difference between two networks, such as two brain images. The techniques will be scalable to very large networks. We hope to collaborate with JHU on this endeavor.

Second, we will consider the problem of prediction in graphs (including time-evolving graphs) by considering fast methods for computing similarity scores such as the Jaccard scores. The difficulty in these methods is that the naive approach requires $O(n^2)$ calculations for an n-vertex graph. However, we are developing techniques based on 4-cycle counting that can be used to prune the search space to only the most promising candidates for having a high similarity score.

2.2 Massive cyber networks

We have been progressing on Tasks B.1–B.2 and should have results in the next quarter. Here the goal is to develop, validate, and deploy graph-based anomaly detection techniques to identify both temporal and spatial outliers in time-evolving, massive cyber networks.

Our testbed is based on more than a year's worth of data collect on the cyber traffic at a government institution. A major part of the effort has been developing tools to interrogate the data. This is needed to validate our findings and improve the methodologies.

We will develop anomaly detection techniques based on probabilistic models. We will validate the methods with both real-world anomalies and, as needed, artificial anomalies that are representative of "expected" anomalous behavior (Arbitrary random behavior is generally considered too easy to detect).

We will also deploy the anomaly detection system to cyber analysts at Sandia. We will use their feedback to improve assays and understand impact on cyber-situational awareness.

2.3 Streaming with multiple time windows

We have been making progress on Task C.1. The idea here is to be able to track measurements on multiple time windows for streaming data. So, our goal is to develop and validate multi-time-scale streaming methods for time-evolving networks for measuring changes in reciprocity, transitivity, degree distributions, etc.

A streaming algorithm processes the entire stream of (say) edges seen so far. We want to develop streaming algorithms that can maintain information about various time-scales together. We propose to do this by adapting our sampling techniques to selectively sample within specific time windows: for example, sample random edges from the past hour, past 2 hours, past day, week, etc.

We have in-house gaming data, with a lot of domain knowledge (like when certain wars happened). We will use our algorithms to try to detect such disruptive events as they are happening. Extend to attributed graphs as well. These developments enable real-time change detection in situational

awareness contexts. For instance, an increase in triangles may indicate group formation, etc.

2.4 Counting 4-node patterns in graphs

As presented in the special projects meeting, we have been making significant progress in sampling methods for 4-vertex patterns. This work will be part of Task D.1. Four-node vertex patterns are extremely expensive to count. Using ideas that are loosely based on the triangle sampling approach, we have developed a sampling method for estimating the number of 4-vertex patterns.

Along the way, we have also developed an extremely efficient approach for exactly counting 4-vertex patterns. To the best of our knowledge, this also represents a scientific advance.

2.5 Power law is not possible for massive graphs

In Task E.1, we consider the problem of generating synthetic degree distributions for modeling purposes. It is common to use closed form expressions (like power laws) to describe the degree distribution of real data (like the internet graph). We have empirically observed that it is not possible to construct a realistic degree distribution for large graphs (e.g., more than a million vertices) using power laws. There appears to be a fundamental mathematical reason for this, based on the Erdos-Gallai theorem.

Specifically, the Erdos-Gallai theorem gives conditions that say when a given degree sequence is realizable. We will prove that, with high probability, the degree distributions drawn from a power law distribution are either not realizable (for power law ≤ 2) or have an unrealistically low average degree (for power law > 2).

The implications are significant for modeling, since we need to find new methods to express and construct degree distributions.

2.6 FEASTPACK software

In addition to the tasks listed above, we plan to continue to make software products available as open source code. We will accelerate the release of those methods deemed most useful for the upcoming July workshop.

3 Publications & References

— Journal Publications —

[1] T. G. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri, A Scalable Generative Graph Model with Community Structure, *SIAM Journal on Scientific Computing*, accepted 2014-03-28, preprint available at arXiv:1302.6636 [cs.SI]

[2] T. G. Kolda, A. Pinar, T. Plantenga, C. Seshadhri, and C. Task, Counting Triangles in Massive Graphs with MapReduce, *SIAM Journal on Scientific Computing*, accepted 2013-12-04, preprint available at arXiv:1301.5887 [cs.SI]

— Refereed Conference and Workshop Proceedings —

[3] C. Klymko, D. F. Gleich, and T. G. Kolda, Using Triangles to Improve Community Detection in Directed Networks, in *The Second ASE International Conference on Big Data Science and Computing, BigDataScience*, (Stanford, CA, May 27–31, 2014), accepted 2014-04-16, preprint available at arXiv:1404.5874

[4] P. Lofgren, S. Banerjee, A. Goel, and C. Seshadhri, FAST-PPR: Scaling Personalized PageRank Estimation for Large Graphs, in *KDD'14: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (New York, NY, Aug. 24–27, 2014), 2014, preprint available at arXiv:1404.3181

— Preprints —

[5] C. Peng, T. G. Kolda, and A. Pinar, Accelerating Community Detection by Using K-core Subgraphs, Mar. 2014, arXiv:1403.2226

[6] J. Wang, R. Gupta, T. Roughgarden, and C. Seshadhri, Counting Small Cliques in Social Networks via Triangle-preserving Decompositions, Feb. 2014

4 Other References

[7] D. Chakrabarti, Y. Zhan, and C. Faloutsos, R-MAT: A Recursive Model for Graph Mining, in *SDM04: Proceedings of the Fourth SIAM International Conference on Data Mining*, Apr. 2004, DOI: 10.1137/1.9781611972740.43

[8] M. Jha, C. Seshadhri, and A. Pinar, A space efficient streaming algorithm for triangle counting using the birthday paradox, in *KDD '13: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, Illinois, USA: ACM, 2013, pp. 589–597, DOI: 10.1145/2487575.2487678

[9] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, Kronecker graphs: An Approach to Modeling Networks, *Journal of Machine Learning Research* 11:985–1042, Feb. 2010, http://jmlr.csail.mit.edu/papers/v11/leskovec10a.html

[10] C. Seshadhri, T. G. Kolda, and A. Pinar, Community structure and scale-free collections of Erdös-Rényi graphs, *Physical Review E* 85(5):056109, May 2012, DOI: 10.1103/PhysRevE.85.056109

[11] C. Seshadhri, A. Pinar, and T. G. Kolda, Triadic Measures on Graphs: The Power of Wedge Sampling, in *SDM13: Proceedings of the 2013 SIAM International Conference on Data Mining*, (Austin, TX, May 2–4, 2013), 2013, pp. 10–18, DOI: 10.1137/1.9781611972832.2

[12] C. Seshadhri, A. Pinar, and T. G. Kolda, Wedge Sampling for Computing Clustering Coefficients and Triangle Counts on Large Graphs, *Statistical Analysis and Data Mining*, DOI: 10.1002/sam.11224

5 Presentations

P1. T. Plantenga, *Generating Large Graphs with Desired Community Structure*, SIAM Conference on Parallel Processing in Scientific Computing, Portland, OR, February 18–21, 2014

P2. A. Pinar, *Generating Large Graphs for Benchmarking*, SIAM Conference on Parallel Processing in Scientific Computing, Portland, OR, February 18–21, 2014

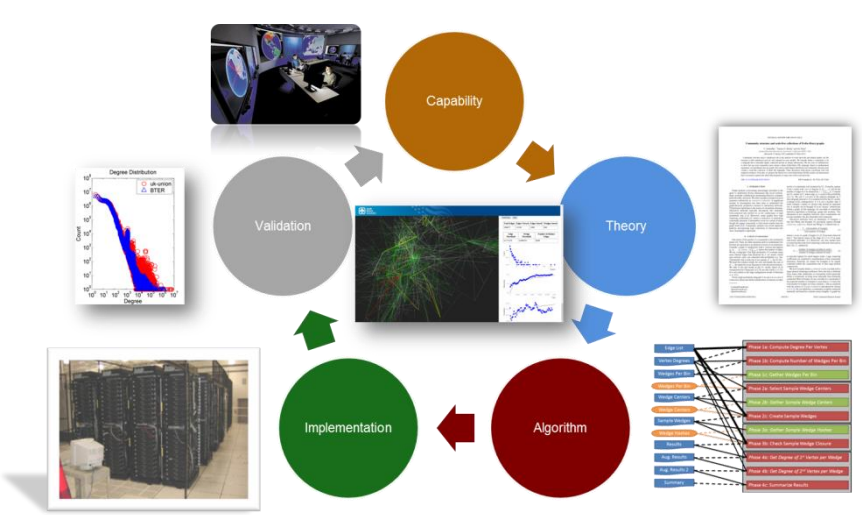
P3. T. Kolda, *Sandia Software for Networks from DARPA GRAPHS Program*, DARPA GRAPHS Special Projects Meeting, Arlington, VA, May 13-14 2014

6 Synergistic Activities

- Hosted visit by DTRA (Don Jones and Martin Hyatt) to Sandia on February 25, 2014
- Involvement with SIAM International Conference on Data Mining, Pittsburgh, PA, April 2014
  - Pinar co-organized Mining Networks and Graphs Workshop
  - Kolda on Senior Technical Program Committee
  - Pinar and Jha on Technical Program Committee
  - Kolda on Best Paper Prize Committee
- Kolda attends GRAPHS Special Projects Meeting, Arlington, VA, May 13–14, 2014

Fast algorithms for Evolving graphs via Assays, Sampling, & Theory (FEAST)

Sandia National Laboratories



Overarching Objectives

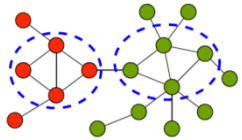
- To create techniques for measurements on real-world large-scale graphs
- To design scalable network models that match real-world graph measurements
- To develop efficient algorithms that take advantage of structure in real-world graphs

State-of-the-Art Innovations

- Sampling-based methods for triangle and 4-vertex pattern counting
- Accurate sublinear one-pass streaming methods for triangle counting
- Scalable generative graph model that produces specified degree distribution and triangle structure
- Improved community-finding methods

Phase II-Q1 Accomplishments

- Triangle estimation using MapReduce
- Scalable generative graph model: BTER
- Fast personalized PageRank
- Triangle-dense graph partitioning
- Directed triangles for community detection
- K-cores for accelerated community finding



POC: Tamara G. Kolda, tgkolda@sandia.gov