Extended Related Works of "Fast Counting of Triangles in Large Real Networks without counting:

Algorithms and Laws"

Amanda Wulf
University of Texas at San Antonio
1 UTSA Circle
San Antonio, Texas 78240
amandakwulf@gmail.com

ABSTRACT

This paper serves as an extended background explanation, motivation, and related works section for [16].

1. INTRODUCTION

This paper's goal is to provide an extended background and related works for [16]. [16] is a paper detailing a method of triangle estimation in real-world graphs using a mathematical relationship between the number of eigenvalues of real-world graphs and their triangle count. This paper introduces the basic ideas of graph theory, the motivations behind this work, the previous algorithms developed for triangle counting, and insights into why this particular triangle counting algorithm works.

2. GRAPH THEORY BACKGROUND

A graph G is defined as a way of encoding pairwise relationships among a set of objects. It consists of a collection of nodes V and a collection of edges E, each of which joins two nodes. Each edge v is an ordered pair (u, v). An edge e is said to leave node u and enter node v. Nodes are also called vertices [8].

Graphs can be either directed (meaning an edge can only be traversed in a certain direction) or undirected [8]. In this paper, the graphs we consider are undirected.

A self-edge (or loop) is a node that has an edge connecting it to itself [5].

Two nodes are adjacent if they have an edge connecting them. A fully connected graph (also known as a complete graph) is a graph in which all nodes are pairwise adjacent [5].

A triangle is defined to be a set of three fully connected nodes in an undirected graph without self-edges [16].

The degree of a node is defined to be the number of edges that connect to it [5].

A closed walk of a graph is a finite, loop-free, non-empty alternating sequence of nodes and edges such that each edge is connected to each subsequent node, and the end node is the same as the start node [5].

3. MOTIVATION

According to Watts and Strogatz [17], small-world graphs are the norm for real-world data. A small-world graph is defined as a tightly-clustered graph with small path lengths [17]. It is easy to see that a tightly-clustered graph with small path lengths will contain many triangles. The number of triagles are thus important for understanding the data in a graph because they give us an idea of the cliquishness of a graph.

The standard metrics for the cliquishness of a graph are the clustering coefficient and the transitivity ratio, which have a high correlation to the number of triangles in a graph. The clustering coefficient of a node is the fraction of possible edges it could have to other nodes divided by the number of edges it actually has [17]. One of the ways of calculating the global clustering coefficient is actually by calculating the number of triangles, multiplying that number by three, and dividing by the number of connected triples of vertices [11]. The transitivity of a graph (which is the transitivity ratio restricted to undirected graphs) is another term for the global clustering coefficient [12].

4. TRIANGLE COUNTING AND LISTING WORKS

4.1 Exact counting and listing algorithms

Schank and Wagner present an overview of triangle algorithms. They divide triangle algorithms into either counting or listing algorithms, where counting algorithms simply produce the number of triangles, while listing algorithms produce a list of every triangle present in the graph [13].

They present several exact counting methods in their paper. The simplest traverses the adjacency matrix using the matrix multiplication solution to the shortest paths problem to get a count of the triangles, with a runtime of $O(n^3)$ [13]. A slightly better exact counting algorithm, which they

call *node-iterator*, traverses all nodes and counts every set of three connected nodes and obtains a runtime of $O(nd_{max}^2)$.

The fastest currently-known exact triangle counting algorithm is known as AYZ for the authors Alon, Yuster, and Zwick, which uses fast matrix multiplication to traverse the nodes and count all triangles and has a runtime of $O(m^{3/2})$ [1]. Thus, the fastest exact triangle counting takes exponential time, which is why [16] presents a fast estimation algorithm.

4.2 Streaming algorithms

A streaming algorithm is an algorithm that handles data that arrives in a data stream. Bar-Yossef et. al. have developed a streaming algorithm to approximate the number of triangles in a large graph. This algorithm has the advantage of being space efficient and only requiring a single pass over the data [2].

4.3 Semi-streaming algorithms

Becchetti, Boldi, and Castillo present an algorithm to estimate the local number of triangles in a graph (for each node, compute the number of triangles that the node participates in). They chose to use a semi-streaming algorithm for this because a streaming algorithm would constrain the memory usage too much to be useful. The semi-streaming algorithm limits the passes over the data to be at most $O(\log N)$ [3].

4.4 Algorithms that minimize disk I/O

Triangle listing algorithms suffer from huge datasets and the fact that in-memory algorithms are unable to handle looking at each node without significant slowdowns from disk I/O. Chu and Cheng have solved this problem using an algorithm that is designed to work on neighboring vertices as a whole in order to minimize disk I/O accesses [4].

4.5 Algorithms that use MapReduce and Hadoop

Exact triangle counting also has the issue of huge graphs and many disk I/O accesses. Suri and Vassilvitskii have developed an algorithm to calculate the number of triangles using MapReduce on a computing cluster in order to mitigate this issue [15].

The ability to massively parallelize triangle counting is important due to the size of the graphs being mined, and the algorithm presented in [16] is able to be run on a MapReduce system as well.

5. BACKGROUND WORKS

This section will introduce the topics needed to understand the theorems and algorithms [16] presents.

5.1 Adjacency matrices

An adjacency matrix of a graph which has n nodes is a matrix $A = (a_{ij})_{nxn}$ which is defined by $a_{ij} = 1$ if $v_i v_j \in E$ and 0 otherwise [5]. This gives us a way to do linear algebra operations on graphs.

5.2 Eigenvalues

For a matrix A, given the expression $Ax = \lambda x$, where x is a vector and λ is a real or complex number, x is said to be the eigenvector of A and λ is said to be its eigenvalue. [9]

5.3 Estimating eigenvalues

The Lanczos method is an iterative method of estimating the eigenvalues in a matrix. It provides a good, fast estimate for large graphs with sparse eigenvalues [6].

5.4 Power laws

A power law is a J-shaped, highly skewed distribution function of some empirical data with a long tail [14].

A degree power law is a graph with a power law for graph degrees. In other words, it has a few nodes with a high degree and many nodes with a low degree [10].

If a graph has a degree power law, it also has an eigenvalue power law [10]. Degree power laws are common in real networks [16], and so by extension eigenvalue power laws are also common. The fact that these graphs have eigenvalue power laws makes the eigenvalues fast to calculate because there are only a few important eigenvalues, and many unimportant (small) eigenvalues, so the Lanczos method can be used to find the most important eigenvalues and thus find a close approximation quickly [16].

5.5 Number of closed walks in a graph

In [7], Harary and Schwenk prove that the number of closed walks of length n in a graph is the sum of the nth powers of the graph's eigenvalues. By extension, the number of closed walks of length 3 is the sum of cubes of the graph's eigenvalues, and a closed walk of length 3 is a triangle.

6. METHOD

The main idea of [16] is the fact that the total number of triangles in a graph is proportional to the sum of cubes of its adjacency matrix eigenvalues. The formula developed by [16] is the following:

$$\Delta(G) = \frac{1}{6}\sum_{i=1}^{n}\lambda_i^3$$

The proposed algorithm EigenTriangle which calculates the number of triangles involves iteratively using the Lanczos method to calculate the eigenvalues, cubing and summing them as we calculate them.

In addition, [16] presents another formula and algorithm which are extensions of EigenTriangle that count the number of triangles Δ_i that node i participates in, calling these EigenTriangleLocal. The formula for this is the following:

$$\Delta_i = \frac{\sum_j \lambda_j^3 u_{i,j}^2}{2}$$

Similarly, the algorithm for this works by using the Lanczos method to compute the cubes of the eigenvalues of the adjacency matrix.

This method provides a mean speedup of 250x compared to the aforementioned *node-iterator* algorithm presented by [13].

In addition to the algorithms developed, [16] also uncovers a power law between a graph's average degree and the

number of triangles present in the graph. [16] calls this the DEGREE-TRIANGLE power law.

[16] also notes that the idea of using eigenvalues to calculate the number of triangles in a graph also works for Kronecker graphs and Erdos-Renyi graphs.

7. CONCLUSION

This has been an explanation and related works for [16]. This paper has explained the basic concepts, motivation, background and related works so that [16] is easier to understand.

8. REFERENCES

[1] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17, 1997.

[2] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 623–632. Society for Industrial and Applied Mathematics, 2002.

[3] L. Becchetti, P. Boldi, and C. Castillo. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of ACM KDD*, 2008.

[4] S. Chu and J. Cheng. Triangle listing in massive networks. *ACM Trans. Knowl. Discov. Data*, 2012.

[5] R. Diestel. *Graph Theory*. Springer, New York, second edition, 2000.

[6] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, third edition, 1996.

[7] F. Harary and A. J. Schwenk. The spectral approach to determining the number of walks in a graph. *Pacific Journal of Mathematics*, 80(2), 1979.

[8] J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson, 2005.

[9] L. Lovasz. Eigenvalues of graphs. Technical report, Department of Computer Science, Eotvos Lorand University, Budapest, Hungary, November 2007.

[10] M. Mihail and C. Papadimitriou. On the eigenvalue power law. 2002.

[11] M. Newman, D. Watts, and S. Strogatz. Random graph models of social networks. *PNAS*, 99, 2002.

[12] T. Schank and D. Wagner. Approximating cluster coefficient and transitivity. *Journal of Graph Algorithms and Applications*, 9(2), 2005.

[13] T. Schank and D. Wagner. Finding, counting, and listing all triangles in large graphs, an experimental study. In *Proceedings of the 4th international conference on Experimental and Efficient Algorithms*, May 2005.

[14] H. A. Simon. On a class of skew distribution functions. 1955.

[15] S. Suri and S. Vassilvitskii. Counting triangles and the curse of the last reducer. International World Wide Web Conference Committee, 2011.

[16] C. E. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *2008 IEEE International Conference on Data Mining*, pages 608–617, 2008.

[17] D. J. Watts and S. H. Strogatz. Collection of 'small-world' networks. *Nature*, 1998.