Project Milestone - CSE 590 Fall 2012

Anirban Mitra
anmitra@cs.stonybrook.edu
Stony Brook ID: 108672767

November 12, 2012

1 Introduction

The abundance of data has given rise to the art of mining data itself to get relevant and meaningful insights. A large class of this *BigData* can be modeled as graph and useful results can be inferred just from the structure of the graph. For example, recommendation system for products can be modelled as bipartite graph of users and products and we can recommend a user the products that many of his/her friends has bought in recent past. Similarly, problems like fraud detection, disease prevention, climate change etc are being attacked using such methods of graph analysis. The scale of *BigData* itself poses a engineering challenge. Here, distributed systems like *Hadoop* using *MapReduce* programming paradigm have become popular as solution.

But, the scale of *big data* is not the only challenge. On top of that, answering *NP-complete* or *NP-hard* problems about a given graph frequently come up. These questions are often part of larger practical graph problems that are interesting to us. A possible solution is to turn our attention to good approximate answers to such *hard problems*. Recently, *graph coordinate systems* like Orion [6] have been proposed, inspired from network coordinate system [2], to map nodes in graph to points in plane which can provide accurate and efficient solutions to such hard queries.

2 Motivation

With *graph coordinate systems* we can leverage a host of geometric algorithms developed for points in a plane problems for solving the corresponding *hard graph problems*. Thus we will be able to have good approximate polynomial time solutions *NP-complete* or *NP-hard* graph problems. The focus of this project will be to try to solve the *Community Detection* problems in a graph using the algorithms for the *Minimum N-Disk Problem*. Since we are now able to map graph nodes to points in a plane using *graph coordinate systems* like [2].

3 Definitions

3.1 Minimum N-Disk Problem

Given N points in a plane, what the minimum radius circle which contains all the N points.

If we apply the algorithm for this problem on the the *graph coordinates* of the complete graph then it should give us a good approximate diameter of the graph. But, in this project, we will be explore the solution of the following problem

3.2 Dense Community Problem

Given a community diameter d, which is the largest subgraph such that its diameter is at most d.

The algorithm presented in [1] can be adapted to solve the *Dense Community Problem*. The above problem contains the following as a subproblem

3.3 Dense Community Membership Problem

Given a diameter d and node v, which is the largest subgraph with its diameter at most d and containing v.

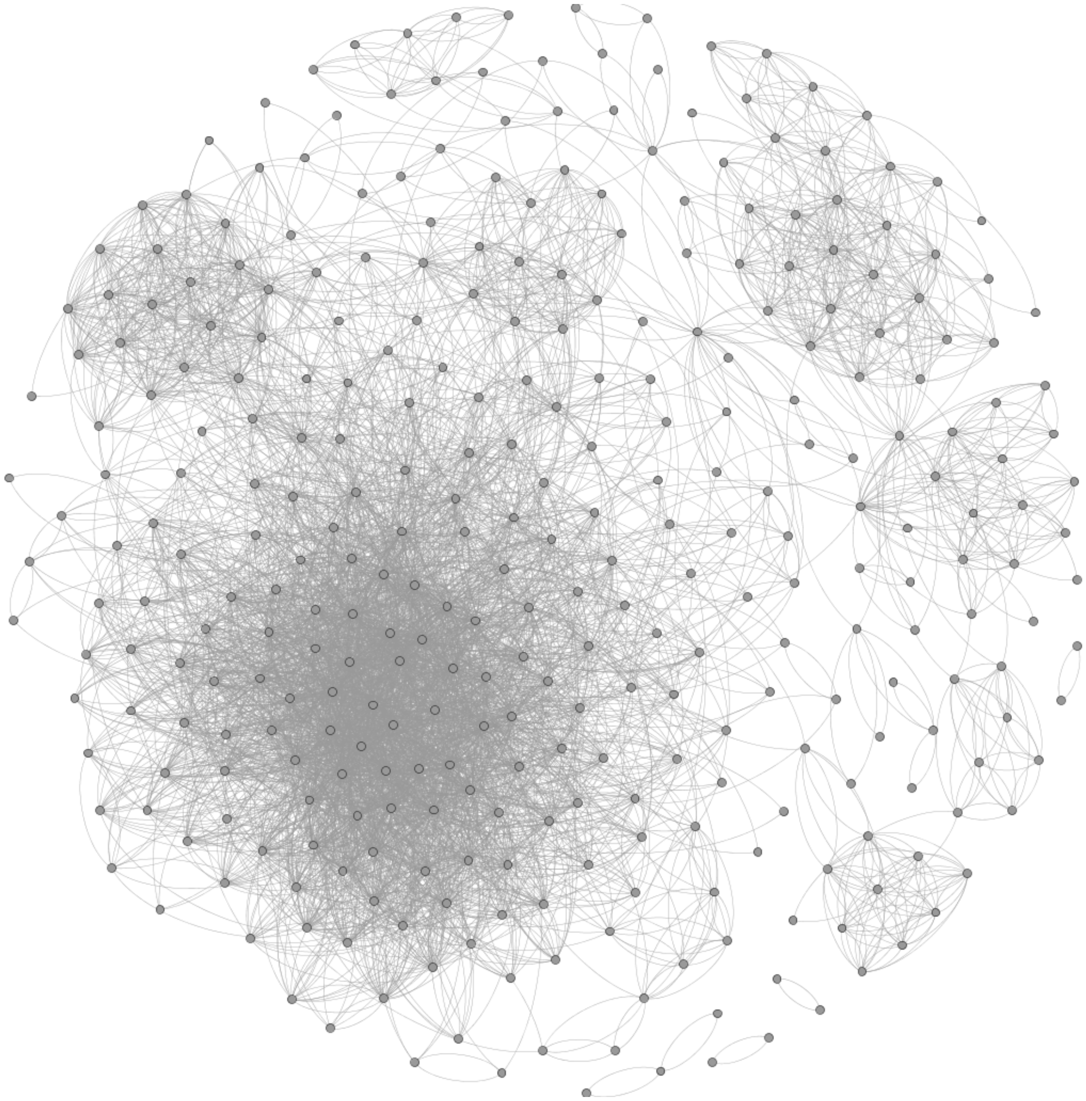
The algorithm will be adapted to *MapReduce* paradigm so that we can use *Hadoop* as a platform to apply this on very large graph coordinates with time complexity given a diameter *d*.

4 Dataset

The data of the ten Facebook networks provided from the *SNAP - Stanford Network Analysis Project* website [4] will be used for the evaluation here. Here are some interesting statistics about the data

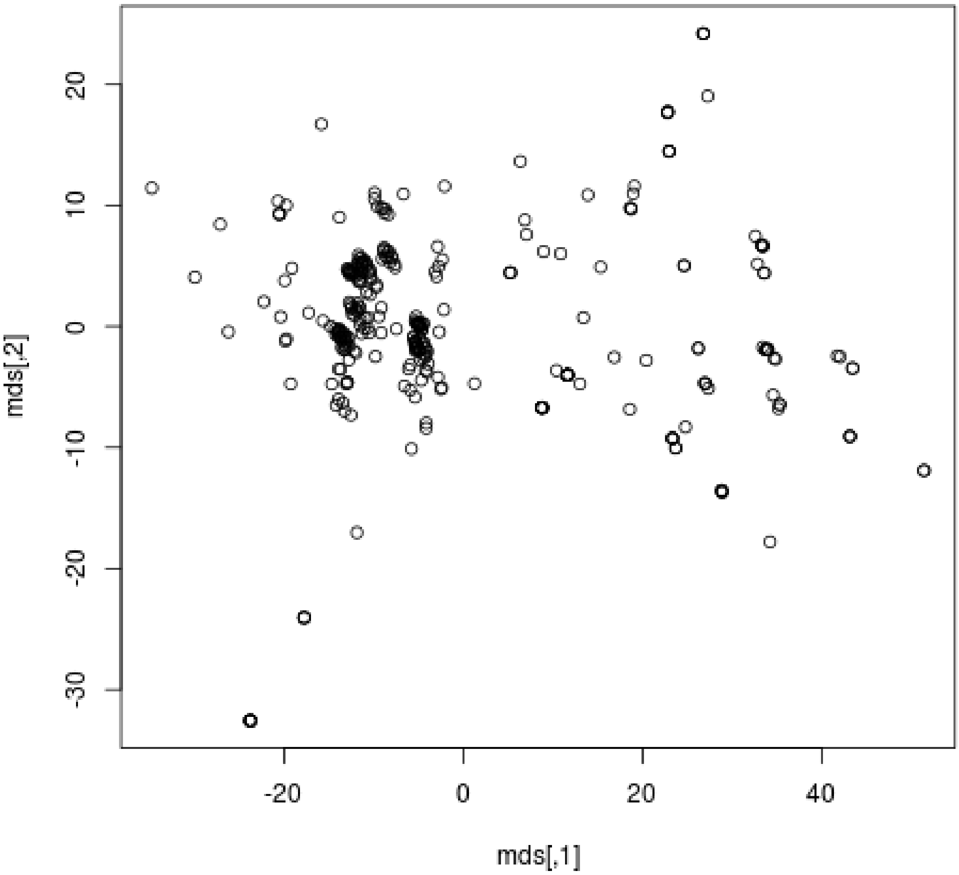
Nodes	4039
Edges	88234
Average Clustering Coefficient	0.6055
Diameter	8
90-percentile Diameter	4.7

Here is the visualization of one of the ten graphs present in the data using *Gephi* [3].



5 Results

Here is the 2D plot of graph data mapped on the plane points using *cmdscale* in *R* [5].



5.1 Runtime

Here we will evaluate the runtime of the proposed *Dense Community Problem* with respect to the *brute force* implementation. This comparison will be done on random data sets of different sizes. Obviously, this will give us good estimate of the runtime when we run this on *graph coordinates*.

5.2 All Pair Distances

Here we will compare the variation of the distances between any two nodes in the original graph and the *graph coordinates*. For a good approximation the variations of the all pairwise distances in the two instances should match with each other.

5.3 Diameter

If we use the solution for the *Dense Community Problem* and do a binary search on the on the diameter value d, we can easily find out the diameter of the graph. The intention here is to evaluate how well we are able to estimate the diameter of the graph using *graph coordinates* and the *Dense Community Problem* algorithm. This is a way to validate our approach to get a approximate solution of the diameter.

6 Code

The code that I wrote for the project can be found on *Github* at `https://github.com/nomind/DataMiningFall2012`.

7 References

References

[1] Alon Ziv Alon Efrat Micha Sharir. *Computing the Smallest k-Enclosing Circle and Related Problems.* 1999.

[2] Frans Kaashoek Robert Morris Frank Dabek Russ Cox. “Vivaldi: A Decentralized Network Coordinate System”. In: *SIGCOMM ’04 Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications* (2004).

[3] *Gephi - The Open Graph Viz Platform.* `https://gephi.org/`.

[4] Jure Leskovec. *SNAP - Stanford Network Analysis Project.* `http://snap.stanford.edu/data/egonets-Facebook.html`.

[5] *R - Classical (Metric) Multidimensional Scaling.* `http://stat.ethz.ch/R-manual/R-devel/library/stats/html/cmdscale.html`.

[6] Christo Wilson Haitao Zheng Ben Y. Zhao Xiaohan Zhao Alessandra Sala. “Orion: Shortest Path Estimation for Large Social Graphs”. In: *Proceedings of The 3rd Workshop on Online Social Networks (WOSN)* (2010).