Proposal to Implement Community Detection Algorithms in NetworkX

Ben Edwards

April 5, 2011

1 Synopsis

NetworkX is a powerful network analysis toolkit for the python programming language. While NetworkX has a very extensive algorithmic base, it has few community detection algorithms. This project would seek to identify the most widely used community detection algorithms and implement them for the NetworkX package.

2 Project Details

2.1 Identification of Algorithms

The identification of communities within complex networks is a popular topic of research. This has lead to the proliferation of algorithms to detect communities in all manner of complex networks (see Fortunato "Community detection in graphs" *Phys. Rep.* 486, 75-174, 2010 and Freeman "Finding social groups" in *Dynamic Social Network Modeling and Analysis*, 2003 for a review). Some of these algorithms have seen more use than others. The first portion of these projects would be to identify which algorithms should be implemented *first* in NetworkX. This would be highly community dependent and would require feedback from the NetworkX user community as well as a thorough literature search to determine which algorithms are most frequently used.

2.2 Implementation of Algorithms

Once the important algorithms have been identified, work can commence on implementing the each algorithm. In addition to a simple (and correct) implementation of the algorithm, clear documentation with reference to the literature which gives concise descriptions of the algorithms, and unit tests which exercise key features of the algorithms will be include. Additionally, I will ask the NetworkX community for help in testing algorithms as there development progresses.

2.3 Mentors

Here is a list of potential mentors and their GSoC mentor IDs:

- Aric Hagberg(ahagberg)
- Loïc Séguin-charbonneau (loicseguin)
- Dan Schult(dschult)
- Chris Ellison (hei7boht)

3 Benefits for NetworkX

NetworkX has one of the most extensive libraries of algorithms, and with the addition of community detection algorithm, it would make NetworkX an easy choice of software for anyone interested in studying complex networks.

4 Success Criteria

Have no less than 10 community detection algorithms, along with documentations and tests, implemented in NetworkX.

5 Project Timeline

Pre-Coding Work with NetworkX community to prioritize known algorithms into ‘essential’, ‘needed’, and ‘wanted’. Identify no less than 10 ‘essential’ community detection algorithms.

Week 1-2 Collect original papers on ‘essential’ algorithms, as well as the data sets and original software on which the algorithms were tested.

Week 3-6 Broken into two possible time lines

Ambitious Goal Have all high priority 'essential' algorithms implemented, and documented. This will include tests of the algorithms to ensure that NetworkX produces results that correspond to those presented in the original publication of the algorithm

Reasonable Goal Have all 'essential' algorithms implemented. Have the majority of documentation and tests for all algorithms completed

Week 7-10 Broken into two possible time lines

Ambitions Goal Collect original papers, data, and software for 'needed' algorithms and implement as many as seems reasonable to complete in remaining time.

Reasonable Goal Have all 'essential' algorithms implemented with documentation and tests.

Week 11-12 Complete documentation and bug fixes for all implemented algorithms, and patch appropriately into NetworkX.

6 Biography

6.1 Personal History

I received by bachelor's degree in Mathematics and Computer Engineering in 2006 from the South Dakota School of Mines and Technology. I am currently seeking a PhD in Computer Science from the University of New Mexico under Stephanie Forrest. My current research focuses on evaluating Internet growth using agent based models. I am also interested in the structure of social networks and how demographic processes create small world social networks, as well as graph models embedded in metric spaces.

6.2 Python

I use Python extensively in my research, as it allows for quick manipulation and analysis of data in a variety of formats. I have produced a number of useful libraries available called python_lib to provide additional functionality to matplotlib, parallel computation, and statistical computing. I have also collected several functions written by others (some of which I have modified) that have proven to be very useful.

6.3 NetworkX

I have used Python and NetworkX extensively in my research. I became involved in the NetworkX project in the Summer of 2010, and in the past year have made several contributions which are in the current codebase: Tickets #356, #323, #357, #375, and #388. I also have contributed code in several pending tickets and discussions (Tickets #378, #359, #345, #390, #387, #371, #396, #533, #360, #395, and #355). In particular I worked on Ticket #423 which attempts to address some of the problems with matplotlib drawing. Additionally, a mailing list discussion attempts to provide an openGL drawing method to NetworkX. This makes me very familiar with the codebase, and able to quickly develop new functionality for NetworkX.

6.4 Contact

Ben Edwards bedwards <at> cs <dot> unm <dot> edu Farris Engineering Center 355d University of New Mexico Albuquerque, NM 87131