A Toolkit for Large Scale Network Analysis

Shashikant Penumarthy, Ketan K. Mane & Katy Börner

{sprao, kmane, katy}@indiana.edu

ABSTRACT

This paper describes the architecture of a toolkit for large scale network analysis. The toolkit and the associated web-interface provide an extremely user-friendly manner to obtain network analysis results. The code library is programmed in Java with Perl-CGI for the front-end providing fast, efficient and scaleable system. The underlying classes and methods are written in a manner so as to facilitate the easy extension of the library. Users upload a network in one of the specified formats, the network is analyzed and the user receives an email with a pointer to the results when the analysis is completed

Keywords

Social Network Analysis, Cyberinfrastructure, Small World Network Properties, Scale Free Networks, Software Repository.

INTRODUCTION

Networks are natural structures that exist everywhere around us. In nature, we see predator-prey networks or protein interaction networks. Human beings form social networks. The internet in itself is a large scale network which is used frequently by everyone but there is very little known about its structure and evolution [1] One of the primary reasons for this is the lack of scalable network analysis tools that could be used to analyze such a large scale network. The code-library discussed in the paper computes most of the standard measures [2-4]conventionally used to describe such networks

ARCHITECTURE

The toolkit is divided into two main components: the client side and the server side. Once the data is uploaded from the local disk to the server end, analysis does not demand intermittent client intervention to compute each network property. Figure 1 shows the complete architecture of the toolkit. The computation is done entirely on the server side. The client uploads the file with the network data into the system and chooses what values should be computed. This eliminates the load from the client side. The server-side computers are fast machines and hence have better computational capability. Besides they are configured to work in accordance with the toolkit.

Features of the code library are listed below:

- User-friendly web interface for data upload[1].
- Ability to upload data sets with no particular size limit.
- Choice of computing several basic network properties.
- High-performance back-end processing.
- Ability to scale, not limited only by compute resources.
- Platform independent and highly portable code.
- Access to results via the web interface.
- Ability to download computational results.

The code is developed using J2SDK 1.4 with front-end processing in Perl-CGI. The Perl-CGI is primarily used to acquire information of the user interested properties. Based on the input parameters the necessary functions are called and the data is processed

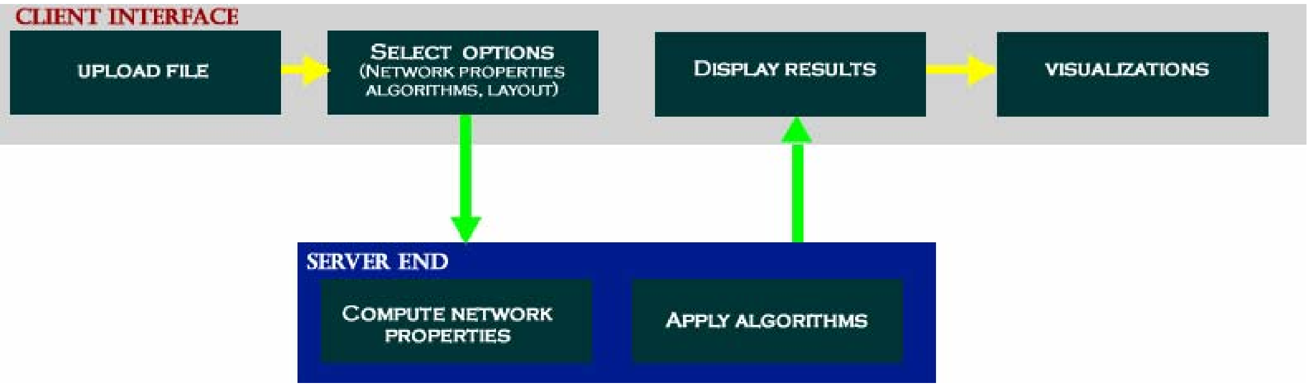

Figure 1: Architecture and the flow of computation through the envisioned system.

[1] The web-interface can be accessed at: http://ella.slis.indiana.edu/~sprao/na/

DATA FORMATS

The current version supports two formats:

I) Barabási Format:
The movie actor format as used by Albert-László Barabási[2]. A sample of the format is shown below:

```
 3 4
 4
 4 5225
 4 590 5679
 4 7186 3862 4615 6277
```

The first column is the primary node and the other columns are considered to be its neighbors. The user must provide information on whether the input network is directed or undirected through the web interface. Details about how the program interprets this file to form arcs and edges are on the network analysis website.

II) PAJEK[3] Format:
A sample of the format is shown below:

```
*vertices 3084
    1 "GILBERT GN"
    2 "NARIN F"
    3 "INHABER H"
    4 "DEBBEAVER D"
    5 "HUSTOPECKY J"
    6 "VLACHY J"
    …
*arcslist
 1    53   54   57   60
 105  106  149  225  487  506
 704  812 1119 1484 1496 1749
```

The first part of this file has information about the vertices and the latter part of the file contains information about the edges that exist between these vertices. Details about this format are on the Pajek[5] website.

CLIENT-SIDE WEB INTERFACE

The client side web interface allows the user to select the data format, the properties to be computed and algorithms to be applied to the network. For a novice, the network property definitions [6] are provided. Once the user submits the request, the processing is moved to the back-end and no more intervention by the user is required. When the computation finishes, an e-mail is sent out to the user at the e-mail address obtained during the upload process informing him/her that the results are ready to be viewed and downloaded.

New users must register when using this toolkit for the first time. The code is being released under the GNU LGPL. New users get a username and password which they can then use to login to the system and to access the analysis results. Please note that currently we do not archive the datasets that users upload. Hence, whenever the user uploads a dataset, the user's previous results are erased.

SERVER-SIDE CODE LIBRARY

The server side code consists of Java code which is invoked using the CGI script that is responsible for processing of client entered information at the front-end. The code library utilizes the following external libraries:

I) JUNG – Java Universal Network/Graph Framework.[7] This is a code library written in Java that enables the creation and manipulation of graphs.

II) CERN Colt – This is a high-performance scientific computation library written in Java that provides extremely fast methods for statistical and mathematical operations.

III) Xerces – This is a utility library that provides inter-convertibility between the internal graph representation and XML. This library is currently not being used. The aim is to eventually provide all graphs and sub-graphs obtained using this toolkit in GraphML.

The server side code is written in the form of Java interfaces and classes that perform the following functions:

- Specify the instantiation and deletion of objects.
- Specify the internal representation of networks as graphs.
- Provide standard prototypes for the methods for computation of network properties.
- Provide a seamless way to integrate the code library with the third-party libraries.
- Provide a scaleable and uniform manner for using and extending the functionality of the library.

The package hierarchy is given below:[4]

src/edu/iu/iv/analysis
src/edu/iu/iv/analysis/NetworkProperties
src/edu/iu/iv/analysis/NetworkAnalysis
src/edu/iu/iv/analysis/PFNet[5]

src/edu/iu/iv/clustering[6]
src/edu/iu/iv/clustering/BC
src/edu/iu/iv/clustering/Ward

src/edu/iu/iv/layout/[7]
src/edu/iu/iv/layout/KKLayout
src/edu/iu/iv/layout/FRLayout

[2] The Movie actor data set is available online at: http://www.nd.edu/~networks/database/index.html

[3] PAJEK is available online at: http://vlado.fmf.uni-lj.si/pub/networks/pajek/

[4] The JavaDoc for this code library is available at: http://ella.slis.indiana.edu/~sprao/na/javadoc/

[5] This is currently not available since it is impractical to apply PFNet to networks exceeding 300 nodes.

[6] This section has been implemented in the IV Repository.

[7] This is future work.

EXAMPLE

We are presenting the analysis of a ‘Small World’ network, a selection of citation networks from Garfield’s collection[8]. The analysis results can be accessed online in the following format (this result is for the ‘Small World’ network data set)

Number of nodes 1059
Number of Arcs 4913
Max in-degree 89
Max out-degree 232
Number of loops 0
Diameter 11
Number of isolated vertices 35
Size of largest weakly connected component 1024
Number of non-trivial weakly connected components 36
Clustering coefficient (Watts-Strogatz) 0.4443
Clustering coefficient (Newman) 0.0938
Scale free exponent
  In-degree -1.6879
  Out-degree -1.6879
In Degree Histogram   sprao_indegree_histogram.txt
Out Degree Histogram sprao_outdegree_histogram.txt

The result page has a table containing all the requested values as well as network links to download files. There are differences in the manner in which certain network measures are interpreted. We are working with researchers in this field to ensure that network analysis toolkit uses the standard way in which network measures are calculated and interpreted.

CHALLENGES AND OPPORTUNITIES

A major challenge in this project has been to get standard and widely accepted definitions for the properties computed using this code library. There seems to be a significant inconsistency in the manner in which these measures are defined and interpreted especially between the social networks community and the physics community. We have tried to provide all the accepted measures to enable comparison of these values. A second challenge has been to make the algorithms as scaleable as possible and this has been made possible by the code library that we are using to analyze the networks. The code is limited only by the amount of available compute resources.

Future work includes providing a large number of popular and useful algorithms to be applied to large-scale networks and a major visualization component.

ACKNOWLEDGMENTS

We thank Katy Börner for her constant support and encouragement, Filippo Menczer for making available to us a preprint of a chapter from the Vespignani’s book, Steve Borgatts for providing sample UCINet code and input and the developers of JUNG for making available their excellent library.

REFERENCES

1. Albert, R., Barabasi, A. L., *Statistical mechanics of complex networks.* cond-mat/0106096, 2002.
2. Newman, M.E.J.A., *A Study of Scientific Co-authorship networks.* . Journal Physics Review, 2000. **28**.
3. Newman, M.E.J., *The structure of scientific collaboration networks.* arXiv:cond-mat/0007214 v1, 2000. **12**.
4. Watts, D.J.a.S., S. H., *Collective dynamics of 'small-world' networks.* Nature, 1998. **393**: p. 440-442.
5. V. Batagelj, A.M., *Pajek - Program for Large Network Analysis.* Connections, 1998. **21**(2): p. 47-57.
6. Weisstein's, E., *World of Mathematics.* *http://mathworld.wolfram.com*.
7. JUNG, *Java Universal Network/Graph Framework.* http://jung.sourceforge.net.

[8] Garfield Collection for Networks is available online at http://vlado.fmf.unilj.si/pub/networks/ data/cite/