

Internship Project Report

DETECTING AND ANALYZING CORE VERTICES IN SOCIAL NETWORK'S COMMUNITIES

SUBMITTED BY

Anu yadav
1114005

Under the guidance of
David A. Bader
Professor and Chair

School of Computational Science and Engineering
College of Computing
Georgia Institute of Technology, Atlanta, GA 30332 USA

INDEX

1. Summary	4
2. Introduction and Basic Concepts	5
3. Detailed Problem Statement	7
4. Related Works	8
5. Research Work Description	10
6. Solution	12
1. Algorithms for Community Detection	12
2. Step Wise Implementation in Gephi	15
7. Experimental Data Set	18
8. Plots	19
9. Observations	22
10. Future Works	22
11. References	23



July 10, 2014

Letter of Certification

This is to certify that **Anu Yadav** an undergraduate student at Indian Institute of Technology Roorkee performed innovative research as part of the Computing Research Undergraduate Intern Summer Experience (CRUISE) program, in the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, USA, from 5 May 2014 through 11 July 2014.

The 10-week long CRUISE program encourages undergraduate students to consider doctoral studies in Computational Science and Engineering. CRUISE interns are matched with faculty mentors and assist with on-going research projects in Computational Science and Engineering at Georgia Tech. CRUISE students attend weekly education and training seminars. On July 8, 2014, each student presented a high quality research talk on their summer research outcomes. I attended the 2014 CRUISE Symposium and certify that the research work performed has **exceeded my expectations**.

My professional qualifications are as follows. I earned my Ph.D. in Electrical Engineering from the University of Maryland, College Park, and following, was awarded a National Science Foundation (NSF) Postdoctoral Research Associateship in Experimental Computer Science. I was a professor of Electrical and Computer Engineering at University of New Mexico for 8 years, and have been at Georgia Tech since 2005. I am a Full Professor and Chair of the School of Computational Science and Engineering, and Executive Director of High Performance Computing. I am an active member of several international professional organizations, including the Institute of Electrical and Electronic Engineering (IEEE) as an IEEE Fellow, AAAS Fellow, Association for Computing Machinery (ACM), as well as Sigma Xi, the Research Honor Society. My research interests are in areas of high performance computing and computational applications.

Sincerely,

David A. Bader
Professor and Chair

Summary

Work Profile

I worked as student intern as a part of Computing Research Undergraduate Intern Summer Experience (CRUISE) program, in the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, USA. CRUISE is a program to give research experience to students to encourage them for further studies. On-going research topics were given to interns.

Institution Profile

The Georgia Institute of Technology (commonly referred to as Georgia Tech, Tech, or GT) is a public research university in Atlanta, Georgia, in the United States. Georgia Tech ranks #6 in the world for engineering. The Georgia Institute of Technology is one of the USA's leading public research universities, and has been an innovation engine since its founding in 1885. It is focused on solving some of the toughest problems facing Georgia, the nation, and the world. This focus transforms industries – and lives – through groundbreaking research underway in research centers and laboratories.

High -level problem statement

Take an evolving "community decomposition" (clustering) of a graph, identify "core" vertices, and analyze them. The changes denote important actions (possibly anomalies, depending on "typical").

Solution

It is basically an exploratory research. In a graph, a community decomposition is a clustering of vertices roughly optimizing a metric like modularity[7]. As edges are added, removed, and modified over time, the community structure can change. Algorithms assign arbitrary labels to communities; they have no meaning on their own. A community's identity better is determined by a set of core members, or vertices identified as important by a separate metric.

Currently, there is no good intuition on how to choose good core vertices. I experimented with different possibilities, not worrying about computing performance yet. I analyzed snapshots of the graph's community decomposition. For each snapshot, metric, and way to select core vertices, I produce a set of sets of core vertices.

Experiments / Results & Discussion /Application / Achievements

Different metrics were experimented like global PageRank, global betweenness centrality[6] and modularity. Simple statistical techniques like mean, median and standard deviation were used to compute core vertices.

Threshold of 1, 2.5 were set to distinguish between core and non-core members of the communities. The number of core vertices varies approximately linearly with the size of community. We can set different threshold for different communities. Similar results were obtained on testing with four different social network graphs. As PageRank and betweenness centrality are two different graph properties and expected to behave differently but they tend to exhibit relation in case of core vertices.

It has applications in social networks to locate the common interest groups like people interested in FIFA, cricket etc. It can even be used to determine the important parts of human brains which control the normal functionality. Even It can be used by brands to locate users who mentioned their products with high score on twitter. It uses detailed graph properties rather than simply counting and slicing.

This work can be extended to see the change of core members with respect to time. For each set of core vertices in the previous snapshot, we can check if they share the same community label in the new snapshot. It can also be seen how many core vertices switch communities.

Introduction and Basic Concepts

The study of social networks has thrown up more than a few surprises over the years. It's easy to imagine that because the links that form between various individuals in a society are not governed by any overarching rules, they must have a random structure. So the discovery in the 1980s that social networks are very different came as something of a surprise. In a social network, most nodes are not linked to each other but can easily be reached by a small number of steps. This is the so-called small worlds network.

Communities are often defined in terms of the partition of the set of vertices, that is each node is put into one and only one community. This might happen in a social network where each vertex represents a person, and the communities represent the different groups of friends: one community for family, another community for co-workers, one for friends in the same sports club, and so on.

Often, networks have certain attributes that can be calculated to analyze the properties & characteristics of the network. These network properties often define network models and can be used to analyze how certain models contrast to each other. Few network properties which are useful are discussed below.

Centrality measures

Information about the relative importance of nodes and edges in a graph can be obtained through centrality measures, widely used in disciplines like sociology. Centrality measures are essential when a network analysis has to answer questions such as: "Which nodes in the network should be targeted to ensure that a message or information spreads to all or most nodes in the network?" or conversely, "Which nodes should be targeted to curtail the spread of a disease?". Formally established measures of centrality are degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, and katz centrality. The objective of network analysis generally determines the type of centrality measures to be used.

Degree centrality of a node in a network is the number of links (vertices) incident on the node.

Closeness centrality determines how "close" a node is to other nodes in a network by measuring the sum of the shortest distances (geodesic paths) between that node and all other nodes in the network.

Betweenness centrality determines the relative importance of a node by measuring the amount of traffic flowing through that node to other nodes in the network. This is done by measuring the fraction of paths connecting all pairs of nodes and containing the node of interest.

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where denominator is total number of shortest paths from node s to node t and numerator is the number of those paths that pass through v.

PageRank

PageRank works by randomly picking "nodes" or websites and then with a certain probability, "randomly jumping" to other nodes. By randomly jumping to these other nodes, it helps PageRank completely traverse the network as some webpages exist on the periphery and would not as readily be assessed.

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

where p_1, p_2, \dots, p_N are the pages under consideration, $M(p_i)$ is the set of pages that link to p_i , $L(p_j)$ is the number of outbound links on page p_j and N is the total number of pages.

Community structure and modularity

The main idea behind Modularity [7] is that the links within a community is higher than the expected links in that community. Thus, we can use a function named Q to denote the difference between real number of links and the expected links:

$$Q = (\text{number of edges within communities}) - (\text{expected number of such edges})$$

Suppose our network contains n vertices. For a particular division of the network into two groups let $s_i = 1$ if vertex i belongs to group 1 and $s_i = -1$ if it belongs to group 2. And let the number of edges between vertices i and j be A_{ij} , which will normally be 0 or 1, although larger values are possible in networks where multiple edges are allowed. (The quantities A_{ij} are the elements of the so-called adjacency matrix.) At the same time, the expected number of edges between vertices i and j if edges are placed at random is $k_i k_j / 2m$, where k_i and k_j are the degrees of the vertices and $m = \frac{1}{2} \sum k_i$ is the total number of edges in the network. Thus the modularity Q is given by the sum of $A_{ij} - k_i k_j / 2m$ over all pairs of vertices i, j that fall in the same group.

Observing that the quantity $\frac{1}{2}(s_i s_j + 1)$ is 1 if i and j are in the same group and 0 otherwise, we can then express the modularity as

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j, \quad [1]$$

By a community structure of such a graph, we mean a partition of the set of nodes into a number of groups, called communities, such that all nodes belonging to any one of these groups satisfy a certain property of relative cohesiveness. Note that one may consider partitions, which are not necessarily strict, i.e., one may allow the case of overlapping communities, when there exist graph nodes belonging to more than one groups (communities) of the partition.



Detailed Problem Statement

High-Level Problem Statement

Take an evolving "community decomposition" (clustering) of a graph, identify "core" vertices, and analyze them. The changes denote important actions (possibly anomalies, depending on "typical").

Description

In a graph, a community decomposition is a clustering of vertices roughly optimizing a metric like modularity. As edges are added, removed, and modified over time, the community structure can change. Algorithms assign arbitrary labels to communities; they have no meaning on their own. A community's identity better is determined by a set of core members, or vertices identified as important by a separate metric.

Currently, there is no good intuition on how to choose good core vertices. So the first task is to experiment with different possibilities, not worrying about computing performance yet. Some streaming graph data (likely artificially generated) is taken and analysis of snapshots of the graph's community decomposition. For each snapshot, metric, and way to select core vertice. A set of sets of core vertices is produced.

Different metrics may be interesting:

- global PageRank,
- global betweenness centrality,
- the ratio of the in-community local clustering coefficient to the vertex's whole-graph clustering coefficient (number of neighboring triangles within the community over the total number of neighboring triangles), and possibly others.
- modularity

There are different possible ways to select the core vertices:

- Compute the community's mean and standard deviation for the metric and select vertices at least 2.5 standard deviations above the mean
 - $bc_norm > 2.5$
 - $pr_norm > 2.5$
- Tried with threshold of 1 as well
 - $bc_norm > 1$
 - $pr_norm > 1$
- Combined two metric and then analyzed with different threshold.
 - $(bc_norm)^2 + (pr_norm)^2 > 1$
 - $bc_norm^2 + (pr_norm)^2 > 2.5^2$

Related Works

The problem of community detection is a long standing research appeared in various forms in several disciplines including sociology and computer science. The first analysis of community structure dates back to 1955 and the work carried out by Weiss and Jacobson [1] in which they searched for work groups within a government agency. However, research on communities actually started even earlier than this work. In 1927, Stuart Rice tried to look for clusters of people in small political bodies, based on the similarity of their voting patterns [2]. Traditional techniques to find communities in social networks are hierarchical and partitional clustering, where vertices are joined into groups according to their mutual similarity.

Several works have been done in the literature which can be categorized into two main groups: optimization methods and methods with no optimization, which search for some predetermined structures. From these methods one can refer to the works done by Girvan and Newman in 2002 and 2004 introducing two important concepts “edge betweenness”[6] and “modularity” [7], the work of Brandes and Erlebach which coins the term “conductance”[8] and the work done by Palla et al. [9]. In [6], Girvan and Newman proposed anew algorithm to identify edges lying between communities which by their successive removal, the isolation of the communities happens. The inter-community edges are detected according to the values of a centrality measure, the edge betweenness that expresses the importance of the role of the edges in processes where signals are transmitted across the graph following paths of minimal length. That work triggered a big activity in the field where many new methods have been proposed in recent years . In particular, physicists entered the game, bringing in their tools and techniques such as spin models, optimization, percolation, random walks, synchronization and etc., which rapidly became the main ingredients of new original algorithms. The field has also taken advantages of concepts and methods from computer science, nonlinear dynamics, sociology and discrete mathematics.

Community detection approaches to social media networks have been conducted for services such as Facebook , Twitter and Wikipedia . In comparison, less attention has been paid to the similar problem of collaboration in open source development communities. We first examined Newman and Girvan[4]’s approach to discovering communities in a social network. Although we found their methodology sound, their model lacked sophistication and left out important considerations about the quality of interaction between users [4]. The work by Brandes et al. [3] presented a powerful model of collaborative structures within Wikipedia and is the motivation for our research on the GitHub community [3]. The emphasize the use of a collaboration score between users to quantify the extent of user relationships, weighting positive and negative interactions. Finally, the paper by Jin et al. [5] attempted to combine the work of a weighted user interaction model with a community detection algorithm. Although their algorithm requires careful parameter tuning, their modeling approach used weighted edges in determining both modularity and betweenness scores.

A closely related concept to (α, β) -community is that of degree core [10–13]. Given degree d , a degree core of a graph G is a maximal connected subgraph of G in which all vertices have degree at least d . Equivalently, it is one of the connected components of the subgraph of G formed by repeatedly deleting all vertices of degree less than d . Every d -core is a $(d-1, d+1)$ -community, but there are many (α, β) -communities that are not degree cores. The concept of (α, β) - community can capture some structural properties of large social networks that other methods (such as degree core) method cannot discover. Degree cores tend to identify subsets of high-degree vertices as communities, while the concept of (α, β) -community highlights more the contrast of intra- and inter-connectivity. This is a natural type of community that we are interested in. I don’t have to be a star to belong to some community, but I should

belong to this community if I have (many) more connections inside this community than anybody outside this community does. A substantial amount of work has been devoted to the task of identifying and evaluating close-knit communities in large social networks, most of which is based on the premise that it is a matter of common experience that communities exist in these networks [14]. A community was often considered to be a subset of vertices that are densely connected internally but sparsely connected to the rest of the network [17–21]. For example, Newman constructed the measure of betweenness and modularity to partition a social network into disjoint communities [15, 16]. Andersen et al. [17] proposed a local graph partitioning algorithm based on personalized PageRank vectors. An information-theoretic framework was also established to obtain an optimal partition and to find communities at multiple levels [18, 19]. However, communities can overlap and may also have dense external connections. Mishra et al. [20] proposed the concept of (α, β) -community and algorithms to efficiently find such communities. Ahn et al. [21] provided a novel perspective for finding hierarchical community structure by categorizing links instead of vertices.

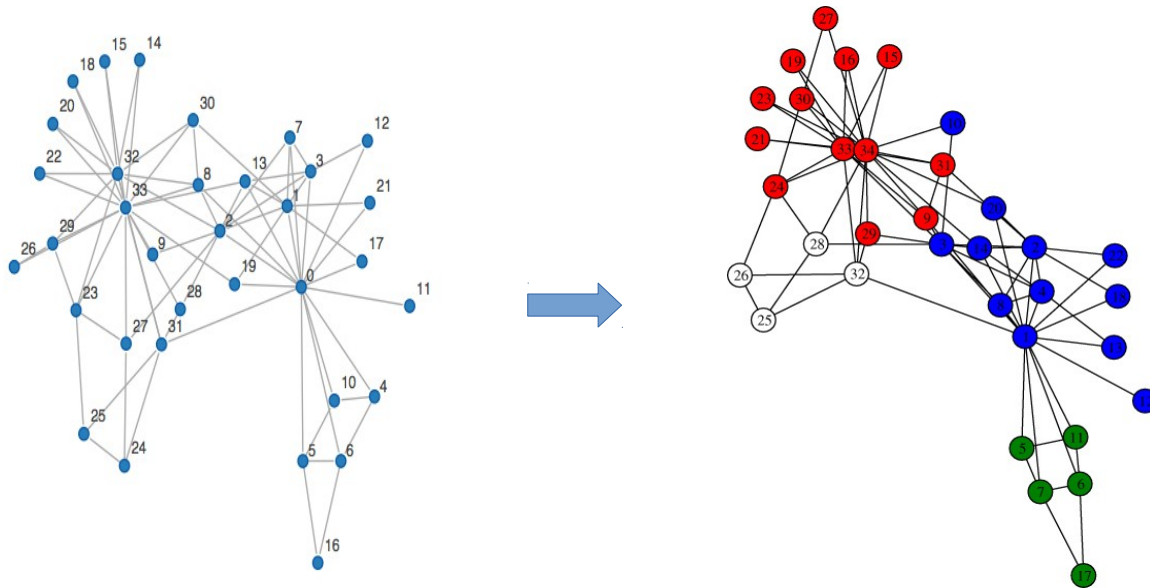
Research work description

Community detection constitutes a significant tool for the analysis of complex networks by enabling the study of mesoscopic structures that are often associated with organizational and functional characteristics of the underlying networks. Community detection has proven to be valuable in a series of domains, e.g. biology, social sciences, bibliometrics. However, despite the unprecedented scale, complexity and the dynamic nature of the networks derived from Social Media data, there has only been limited discussion of community detection in this context. More specifically, there is hardly any discussion on the performance characteristics of community detection methods as well as the exploitation of their results in the context of real-world web mining and information retrieval scenarios.

Problem

Given a social network graph. First task is to decompose the graph into communities. Take an evolving "community decomposition" (clustering)[4] of a graph, identify "core" vertices, and analyze in them. The changes denote important actions (possibly anomalies, depending on "typical").

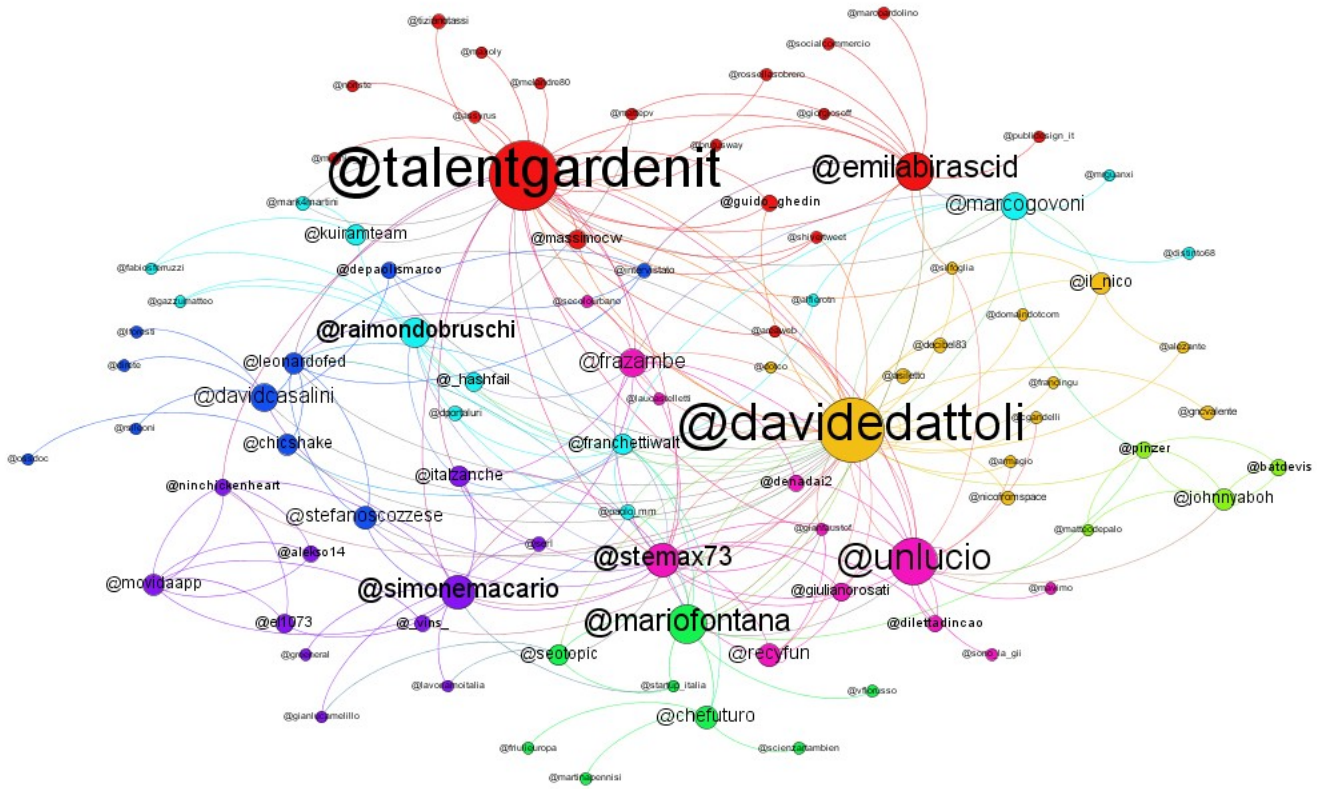
We are trying to capture that vertices are more similar within one community than between communities. A community locally maximizes some metric like Modularity, conductance etc.



Zachary's karate club: social network of friendships between 34 members of a karate club at a US university in the 1970s. W. W. Zachary, An information flow model for conflict and fission in small groups, *Journal of Anthropological Research* 33, 452- 473 (1977).

In graphical terms, from a given graph, we have to detect communities. Different communities have different colors in order to distinguish them.

Core Vertices: These are special vertices which bind the communities. These vertices are of special nature because they can represent a community. For example: a person belonging to some group in facebook has more friends among the given group can be considered as core members, a person within a group like some football fan club having highest number of followers on twitters can be considered as core. There can be more than one core vertex in a community.



The bigger nodes are core vertices and smaller ones are non-core members. Each community is represented in different color. It is not necessary that core vertex should appear in the center of the graph. Core vertices are selected based on various centrality measures like PageRank, betweenness centrality, degree centrality etc.

Solution

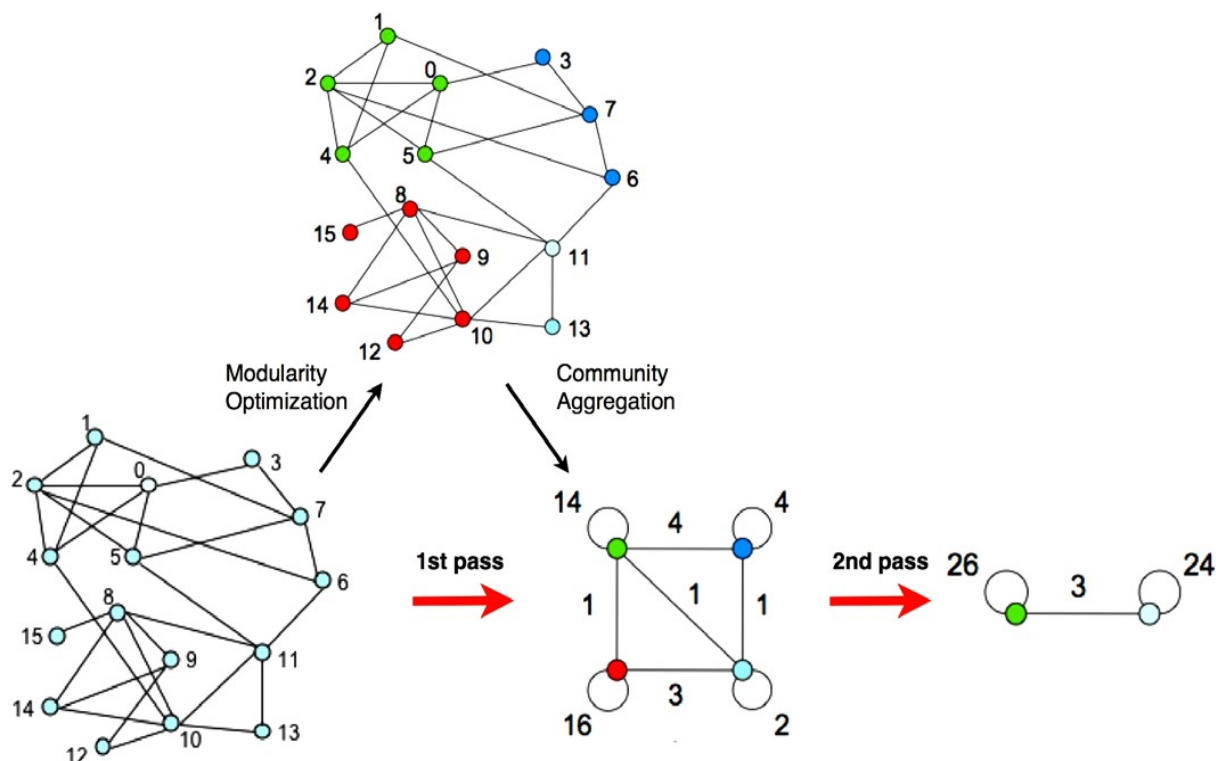
Gephi[23] is an open-source network analysis and visualization software package written in Java on the NetBeans platform, initially developed by students of the University of Technology of Compiègne (UTC) in France. It is used to detect communities in a graph. It implements Louvain community detection algorithm to detect communities.

Algorithms for Detecting Communities

1. The Louvain method for community detection in large networks

The Louvain method[8] is a simple, efficient and easy-to-implement method for identifying communities in large networks. The method has been used with success for networks of many different type (see references below) and for sizes up to 100 million nodes and billions of links. The analysis of a typical network of 2 million nodes takes 2 minutes on a standard PC. The method unveils hierarchies of communities and allows to zoom within communities to discover sub-communities, sub-sub-communities, etc. It is today one of the most widely used method for detecting communities in large networks.

The method is a greedy optimization method that attempts to optimize the "modularity" of a partition of the network (modularity is defined here). The optimization is performed in two steps. First, the method looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes belonging to the same community and builds a new network whose nodes are the communities. These steps are repeated iteratively until a maximum of modularity is attained and a hierarchy of communities is produced. Although the exact computational complexity of the method is not known, the method seems to run in time $O(n \log n)$ with most of the computational effort spent on the optimization at the first level. Exact modularity[7] optimization is known to be NP-hard.



2. Girvan and Newman Algorithm

The Girvan–Newman algorithm[14] detects communities by progressively removing edges from the original network. The connected components of the remaining network are the communities. Instead of trying to construct a measure that tells us which edges are the most central to communities, the Girvan–Newman algorithm focuses on edges that are most likely "between" communities.

The algorithm's steps for community detection are summarized below

- The betweenness of all existing edges in the network is calculated first.
- The edge with the highest betweenness is removed.
- The betweenness of all edges affected by the removal is recalculated.
- Steps 2 and 3 are repeated until no edges remain.

The fact that the only betweennesses[7] being recalculated are only the ones which are affected by the removal, may lessen the running time of the process' simulation in computers. However, the betweenness centrality must be recalculated with each step, or severe errors occur. The reason is that the network adapts itself to the new conditions set after the edge removal. For instance, if two communities are connected by more than one edge, then there is no guarantee that all of these edges will have high betweenness. According to the method, we know that at least one of them will have, but nothing more than that is known. By recalculating betweennesses after the removal of each edge, it is ensured that at least one of the remaining edges between two communities will always have a high value.

The betweenness of a vertex v in a graph $G:=(V,E)$ with V vertices is computed as follows:

- For each pair of vertices (s,t) , compute the shortest paths between them.
- For each pair of vertices (s,t) , determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).
- Sum this fraction over all pairs of vertices (s,t) .

More compactly the betweenness can be represented as:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v . The betweenness may be normalised by dividing through the number of pairs of vertices not including v , which for directed graphs is $(n-1)(n-2)$ and for undirected graphs is $(n-1)(n-2)/2$. For example, in an undirected star graph, the center vertex (which is contained in every possible shortest path) would have a betweenness of $(n-1)(n-2)/2$ (1, if normalised) while the leaves (which are contained in no shortest paths) would have a betweenness of 0.

From a calculation aspect, both betweenness and closeness centralities of all vertices in a graph involve calculating the shortest paths between all pairs of vertices on a graph, which requires $O(V^3)$ time with the Floyd–Warshall algorithm. However, on sparse graphs, Johnson's algorithm may be more efficient, taking $O(V^2(\log V) + VE)$ time. In the case of unweighted graphs the calculations can be done with Brandes'

algorithm which takes $O(V E)$ time. Normally, these algorithms assume that graphs are undirected and connected with the allowance of loops and multiple edges. When specifically dealing with network graphs, often graphs are without loops or multiple edges to maintain simple relationships (where edges represent connections between two people or vertices). In this case, using Brandes' algorithm will divide final centrality scores by 2 to account for each shortest path being counted twice.

After getting communities, the next task is to detect core vertices from them. There are some simple statistical techniques to select core vertices. In order to proceed further, a term needs to be defined.

normalized value of any metric (or z- score) = (value of metric – mean over the community)/
standard deviation over the community

$$z = \frac{(x - \mu)}{\sigma}$$

μ is the mean of the observed data;

σ is the standard deviation of the observed data.

We are taking normalized value because its easy to plot their curve instead of their absolute value. Since it is normalized by the standard deviation, it can be used for comparison between standard scores on different data.

In statistics, the standard score, also known as z – score, is defined as the the (signed) number of standard deviations an observation or datum is above the mean (expected value). Thus, a positive standard score represents that the actual value is above the mean, while a negative standard score represents that the actual value is below the mean. It is a dimensionless quantity obtained by subtracting mean by actual value (raw score) and then dividing the difference by the standard deviation.

bc_norm = normalized value of betweenness centrality

pr_norm = normalized value of PageRank

The following are simple methods to detect core vertices.

Look at each metric individually:

– bc_norm > 1 , pr_norm > 1

– bc_norm > 2.5 , pr_norm > 2.5

Look at these metrics together:

– $(bc_norm)^2 + (pr_norm)^2 > 1$

– $(bc_norm)^2 + (pr_norm)^2 > (2.5)^2$

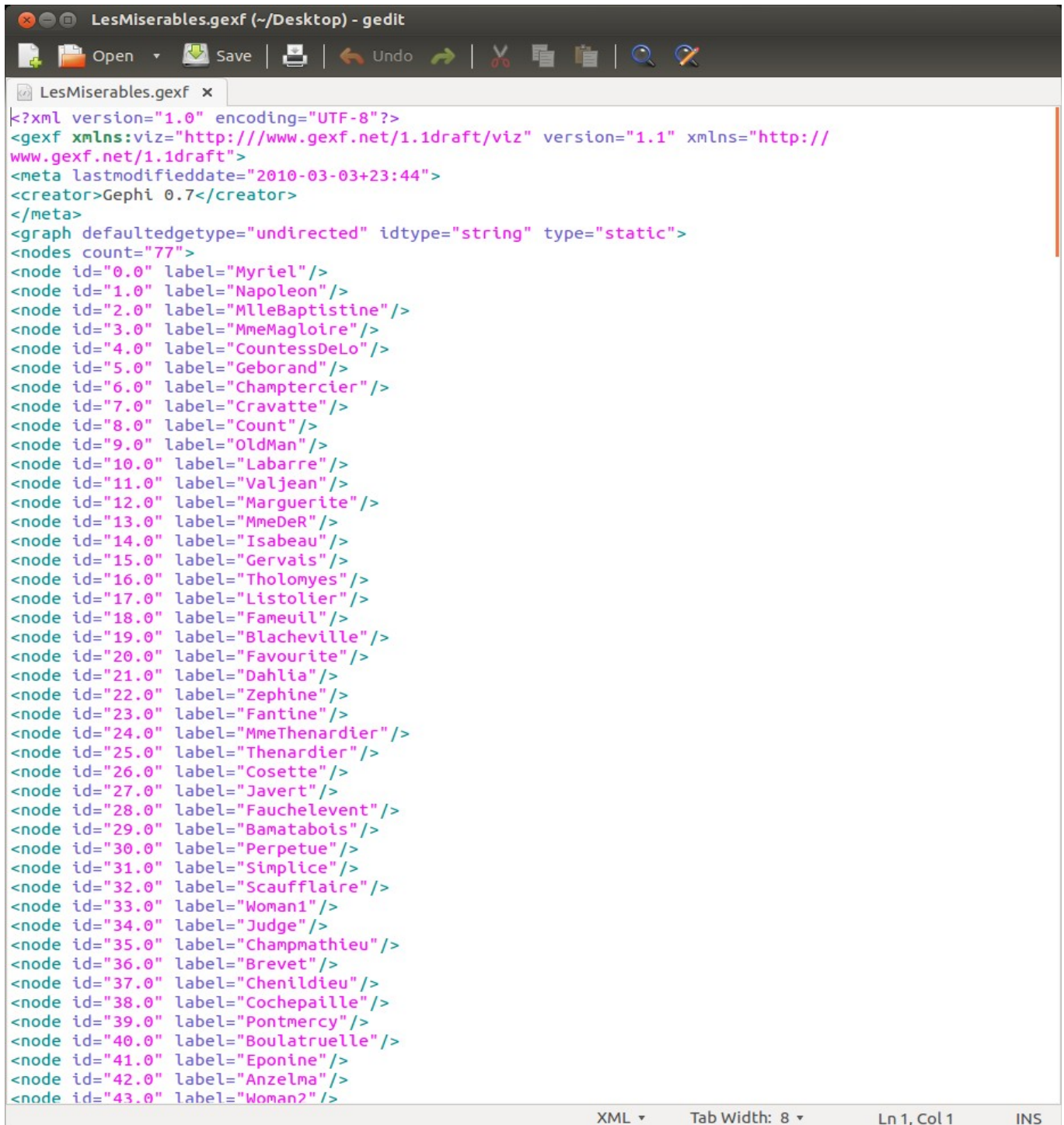
The vertices which satisfies the above cases are core vertices. Every time we get different number of core vertices. Different plots are plotted between page rank and betweenness centrality for the four largest communities of a social network graph. Even the sorted value of metrics are also plotted to observe the sudden change in behaviour. With this method we get the core vertices in each community.

In order to see their change with respect to time, streaming data is used. In case of dynamic graphs, we can see how core vertices changes with addition of new vertices.

Step wise implementation in gephi[22]

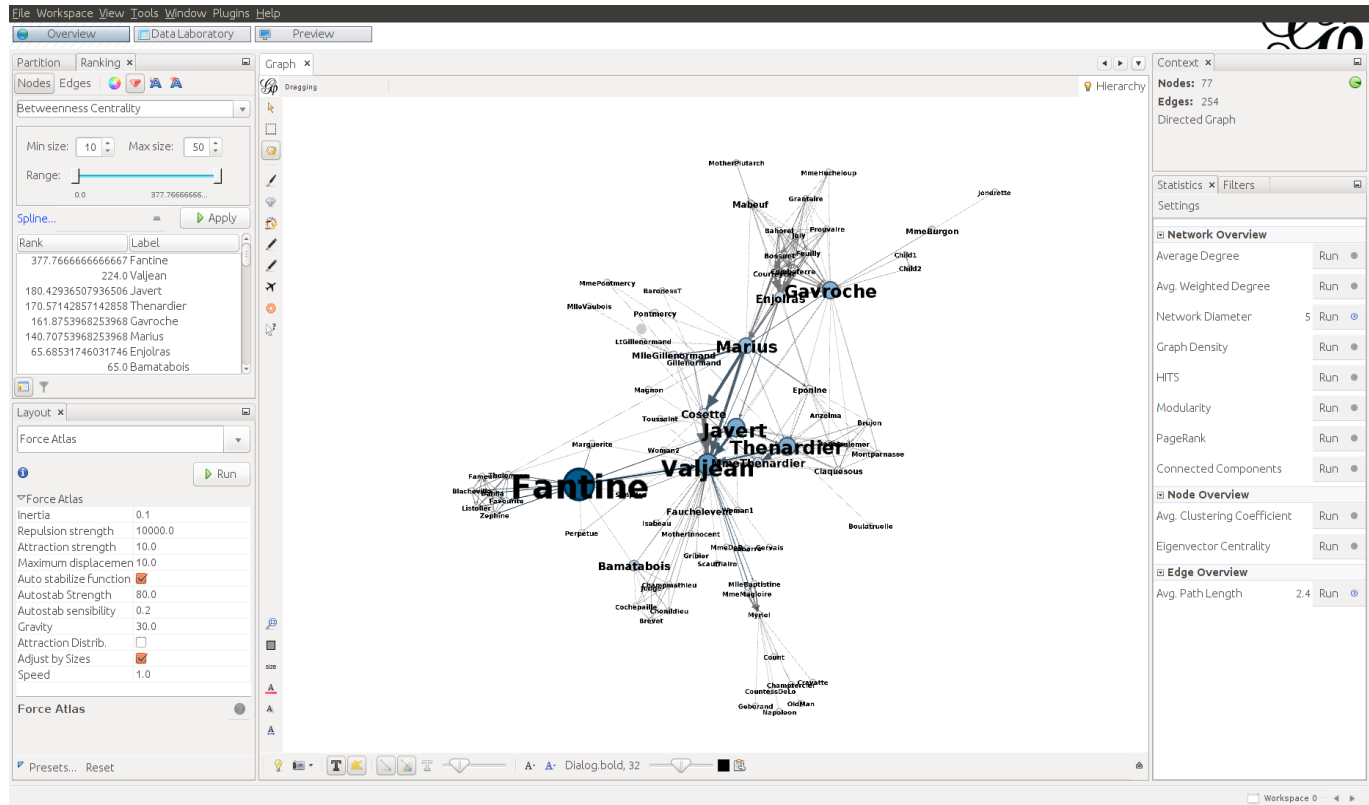
In this analysis, the character relationships in the classic epic novel Les Miserables are explored. Looking at the graph, it can be seen that there are 77 distinct vertices and 254 edges. This means that there are 77 different characters in Les Miserables and that there are 254 connections between those 77 characters. Though some of the characters may be clustered, there is only one connected component. So the reader knows that any character can follow an acquaintance path to any other character in the novel.

The dataset that was used to generate these diagrams is in XML format:

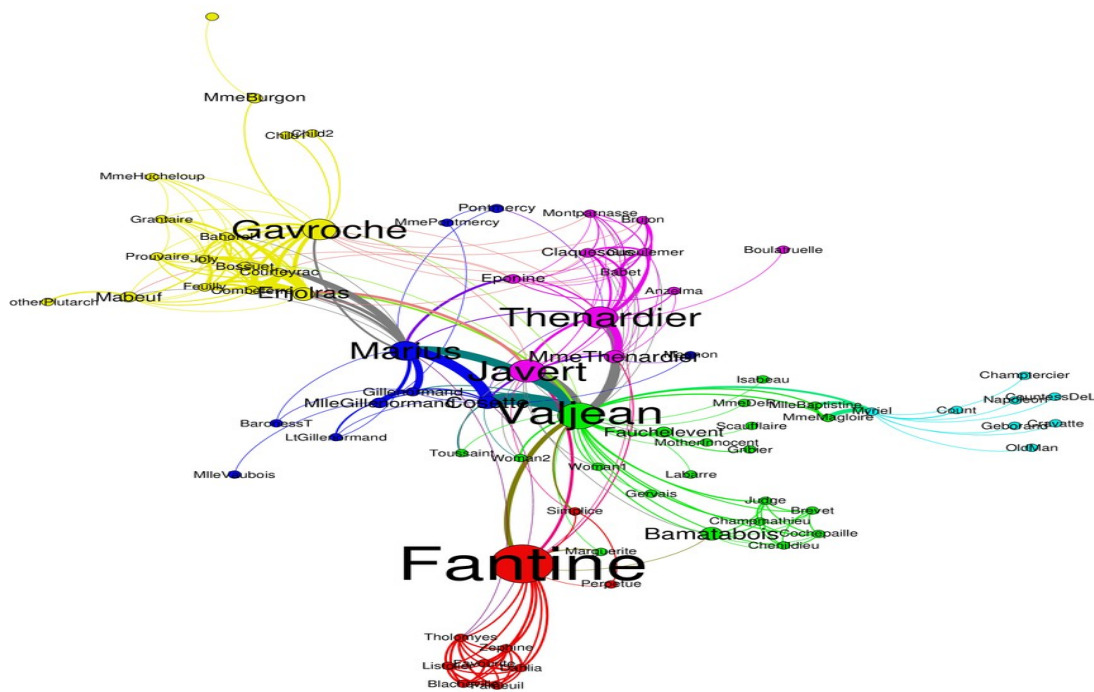


```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns:viz="http://www.gexf.net/1.1draft/viz" version="1.1" xmlns="http://
www.gexf.net/1.1draft">
  <meta lastmodifieddate="2010-03-03+23:44">
    <creator>Gephi 0.7</creator>
  </meta>
  <graph defaultedgetype="undirected" idtype="string" type="static">
    <nodes count="77">
      <node id="0.0" label="Myriel"/>
      <node id="1.0" label="Napoleon"/>
      <node id="2.0" label="MlleBaptistine"/>
      <node id="3.0" label="MmeMagloire"/>
      <node id="4.0" label="CountessDeLo"/>
      <node id="5.0" label="Geborand"/>
      <node id="6.0" label="Champtercier"/>
      <node id="7.0" label="Cravatte"/>
      <node id="8.0" label="Count"/>
      <node id="9.0" label="OldMan"/>
      <node id="10.0" label="Labarre"/>
      <node id="11.0" label="Valjean"/>
      <node id="12.0" label="Marguerite"/>
      <node id="13.0" label="MmeDeR"/>
      <node id="14.0" label="Isabeau"/>
      <node id="15.0" label="Gervais"/>
      <node id="16.0" label="Tholomyes"/>
      <node id="17.0" label="Listolier"/>
      <node id="18.0" label="Fameuil"/>
      <node id="19.0" label="Blacheville"/>
      <node id="20.0" label="Favourite"/>
      <node id="21.0" label="Dahlia"/>
      <node id="22.0" label="Zephine"/>
      <node id="23.0" label="Fantine"/>
      <node id="24.0" label="MmeThenardier"/>
      <node id="25.0" label="Thenardier"/>
      <node id="26.0" label="Cosette"/>
      <node id="27.0" label="Javert"/>
      <node id="28.0" label="Fauchelevent"/>
      <node id="29.0" label="Bamatambois"/>
      <node id="30.0" label="Perpetue"/>
      <node id="31.0" label="Simplice"/>
      <node id="32.0" label="Scaufflaire"/>
      <node id="33.0" label="Woman1"/>
      <node id="34.0" label="Judge"/>
      <node id="35.0" label="Champmathieu"/>
      <node id="36.0" label="Brevet"/>
      <node id="37.0" label="Chenildieu"/>
      <node id="38.0" label="Cochepaille"/>
      <node id="39.0" label="Pontmercy"/>
      <node id="40.0" label="Boulatruelle"/>
      <node id="41.0" label="Eponine"/>
      <node id="42.0" label="Anzelma"/>
      <node id="43.0" label="Woman2"/>
    </nodes>
  </graph>
</gexf>
```

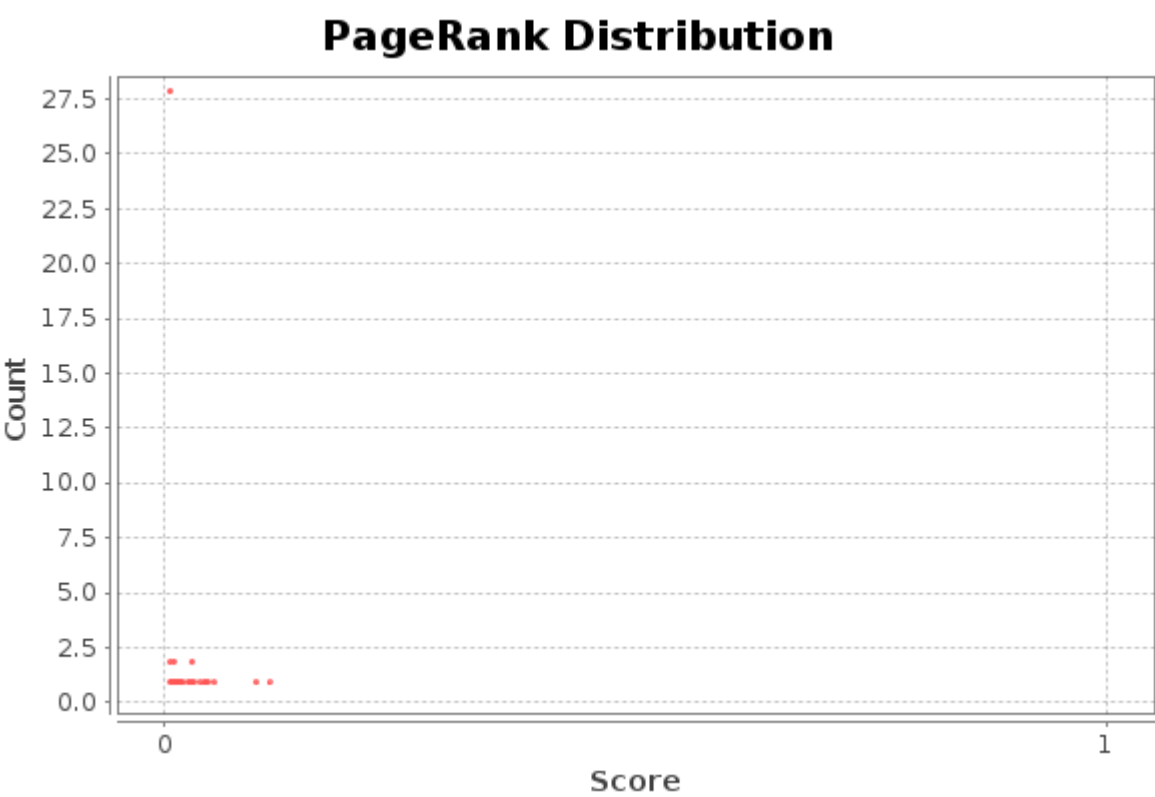
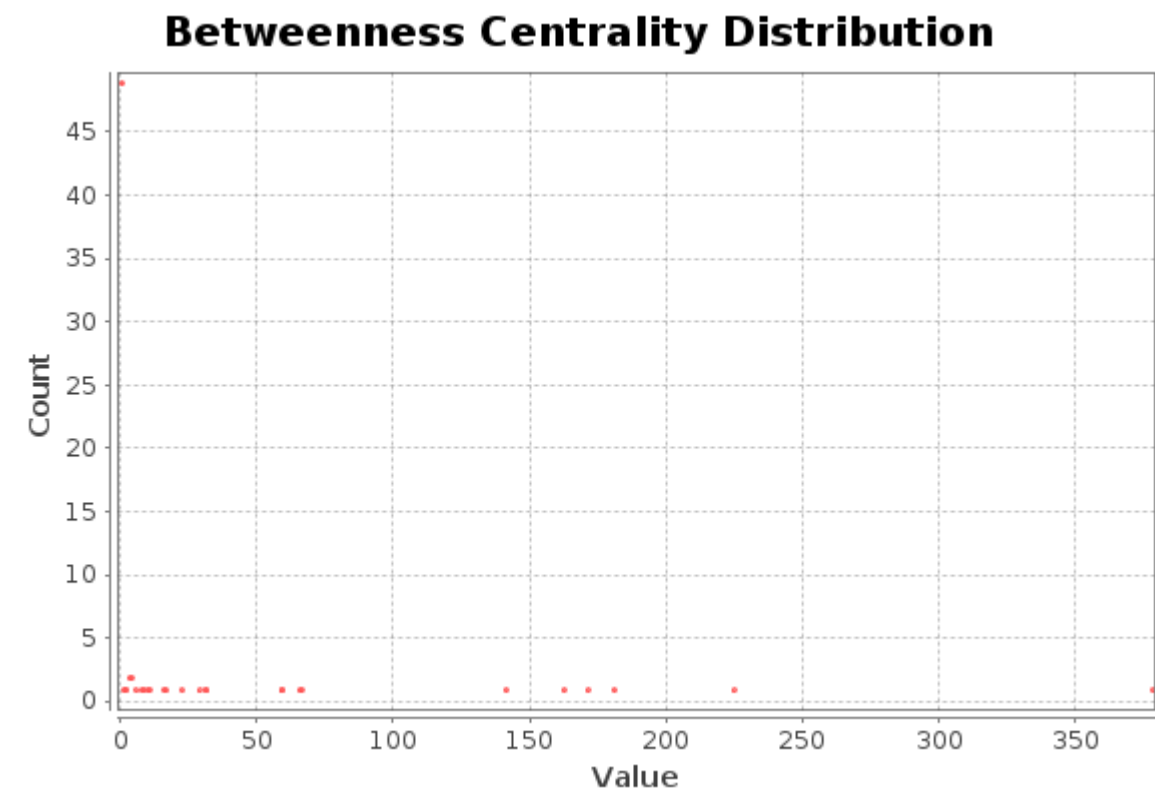
The complete GUI looks like this



The graph consists of circles, called nodes, and lines connecting these nodes, called edges. Each represents a character that appears in the novel. Each line represents an association between characters. The size of the circles and names of the characters vary proportionally with the number of connections that a character has. As you can see here, Jean Valjean, the main character, has the greatest number of connections. However, just because a character has the most connections doesn't mean they are the most influential, and an alternative measure, called betweenness centrality, is a measure of a node's importance. Below, we can see that Fantine has the highest betweenness centrality:



The two curves given below are generated by gephi.



Experimental Data Set[23]

The following are the sample data set used to detect communities and then analysis of their core vertices.

1. **Astrophysics collaborations** : weighted network of coauthorships between scientists posting preprints on the Astrophysics E-Print Archive between Jan 1, 1995 and December 31, 1999.

M. E. J. Newman, Proc. Natl. Acad. Sci. USA 98, 404-409 (2001).

vertices : 16706

edges : 121251

communities : 574

effective communities : 20 (communities which have more than 30 vertices)

2. **Condensed matter collaborations 1999** : weighted network of coauthorships between scientists posting preprints on the Condensed Matter E-Print Archive between Jan 1, 1995 and December 31, 1999.

M. E. J. Newman, The structure of scientific collaboration networks, Proc. Natl. Acad. Sci. USA 98, 404-409 (2001).

vertices : 16726

edges : 47594

communities : 1140

effective communities : 28 (communities which have more than 30 vertices)

3. **Power grid** : An undirected, unweighted network representing the topology of the Western States Power Grid of the United States. Data compiled by D. Watts and S. Strogatz and made available on the web here. D. J. Watts and S. H. Strogatz, Nature 393, 440-442 (1998).

vertices : 4941

edges : 6594

communities : 7

effective communities : 7 (communities which have more than 30 vertices)

4. **E-mail network URV** : List of edges of the network of e-mail interchanges between members of the Univeristy Rovira i Virgili (Tarragona). R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt and A. Arenas, Physical Review E , vol. 68, 065103(R), (2003).

vertices : 1133

edges : 5451

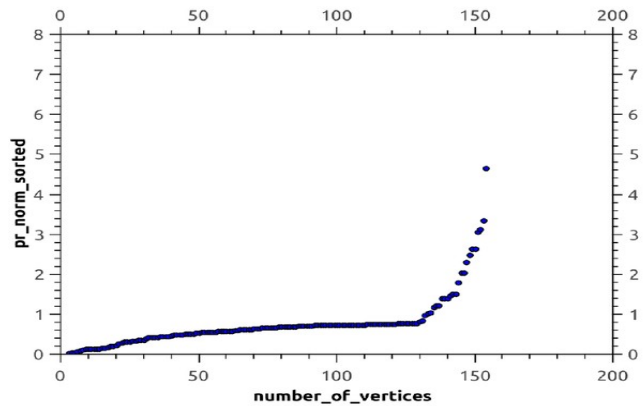
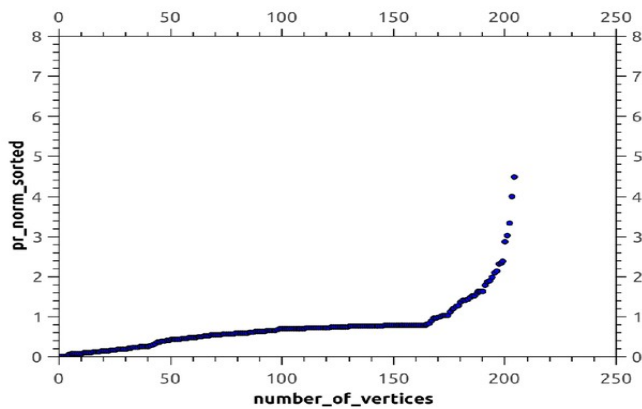
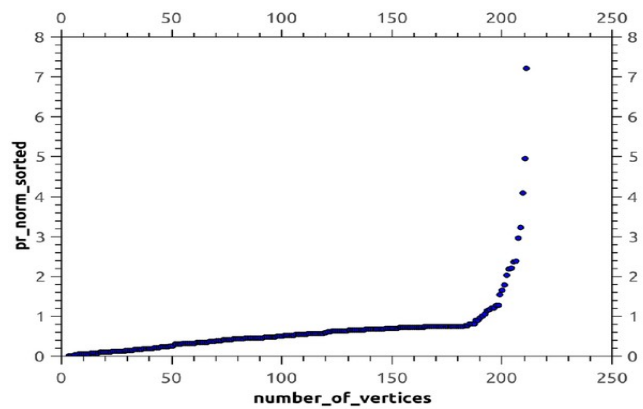
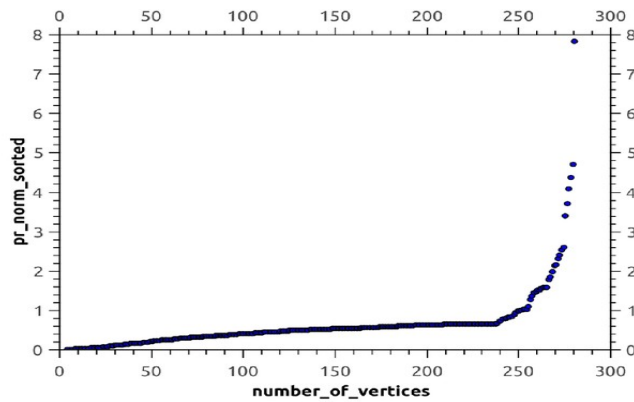
communities : 29

effective communities : 29 (communities which have more than 30 vertices)

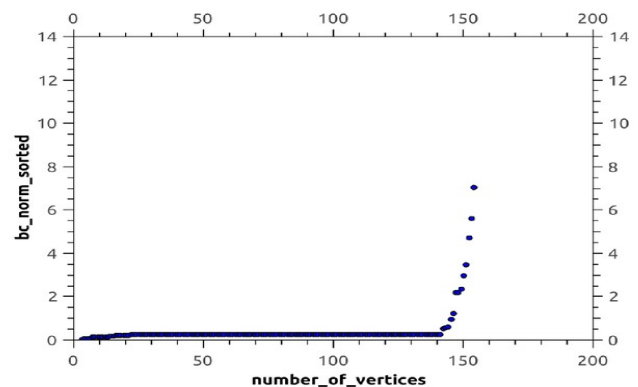
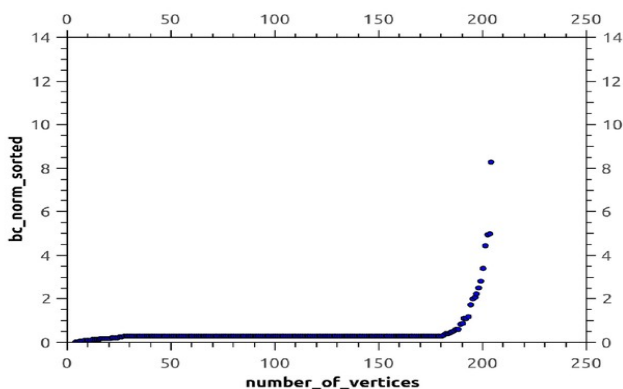
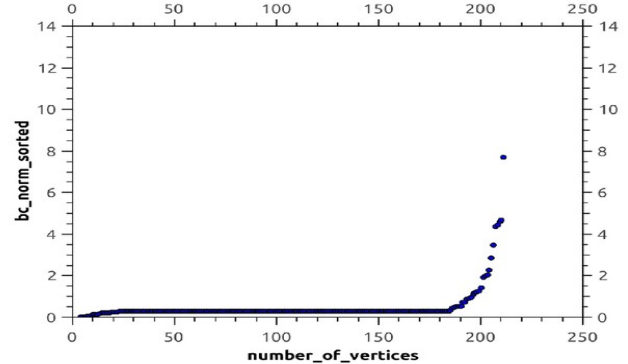
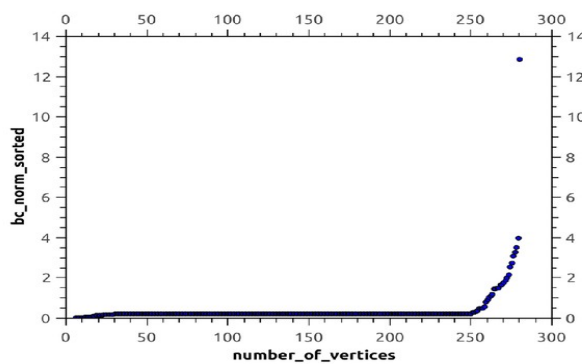
Plots

1. Email Graph

- Sorted and Normalized values of PageRank of the four largest communities of the graph.

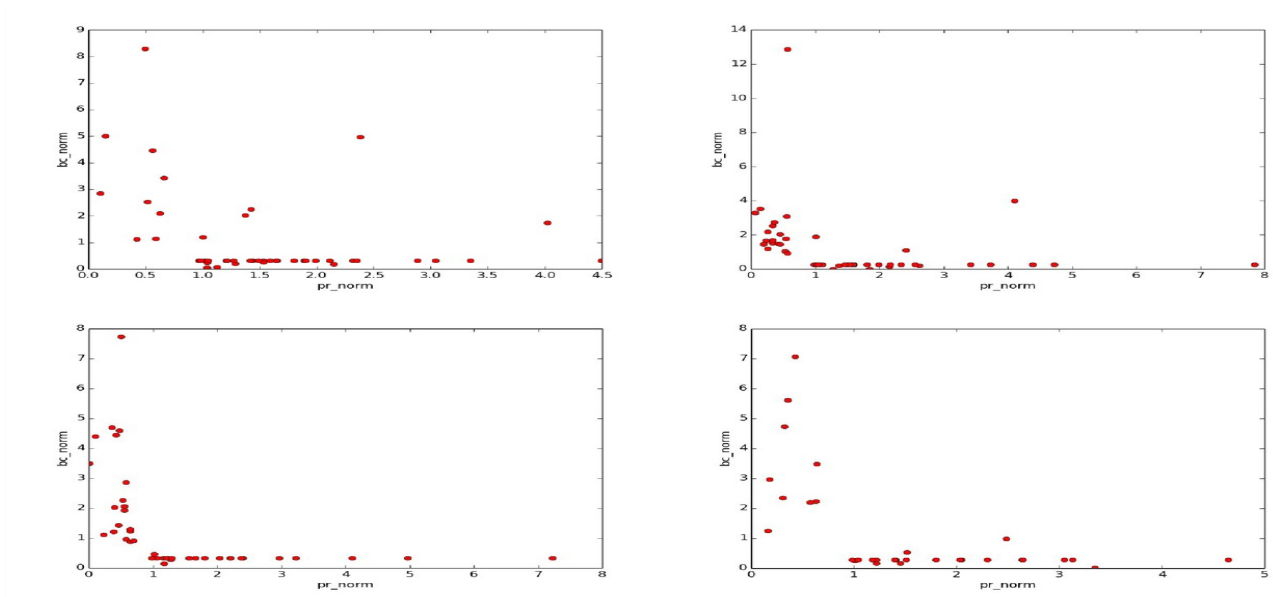


- Sorted and normalized values of betweenness centrality of the four largest communities of the graph.

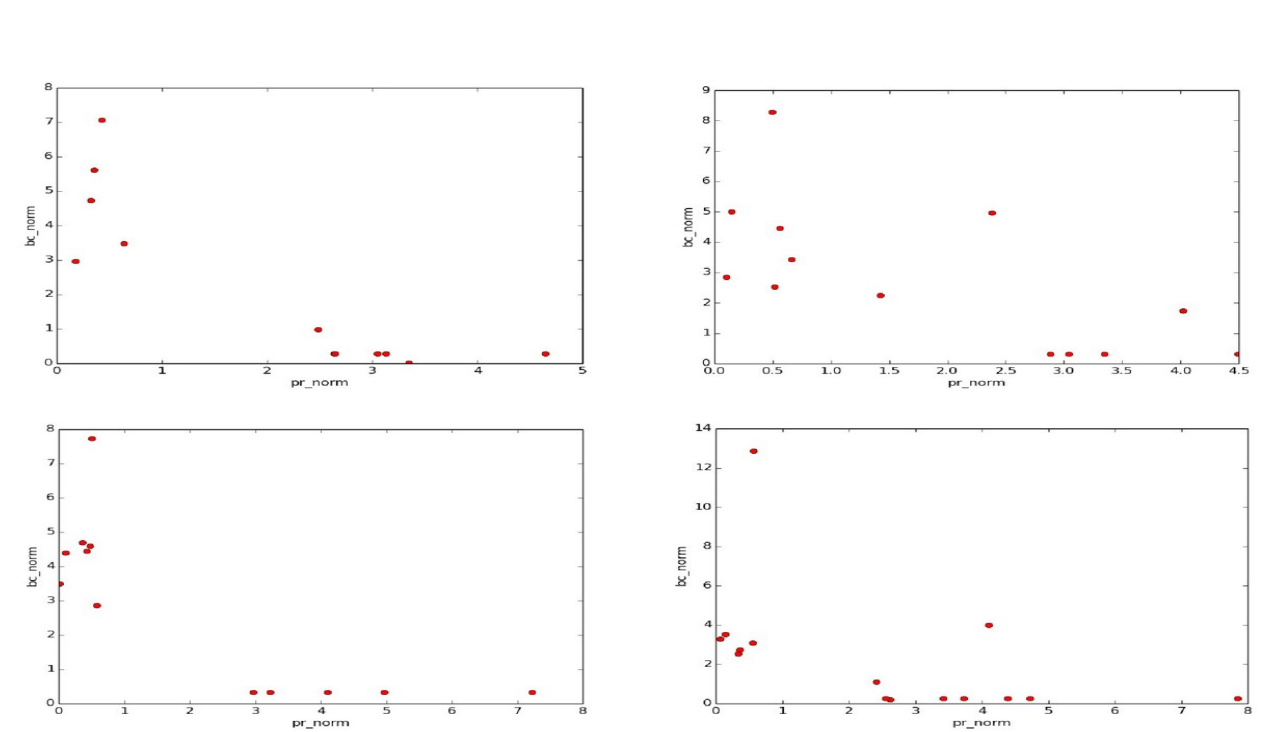


For selecting core vertices a line can be drawn parallel to x- axis with a threshold of 1, 2.5. The vertices which lie above this line are core vertices. Both PageRank and betweenness centrality behaved in the same manner as can be seen from the diagrams.

- The diagram shows the vertices satisfying the equation $(bc_norm)^2 + (pr_norm)^2 > 1$

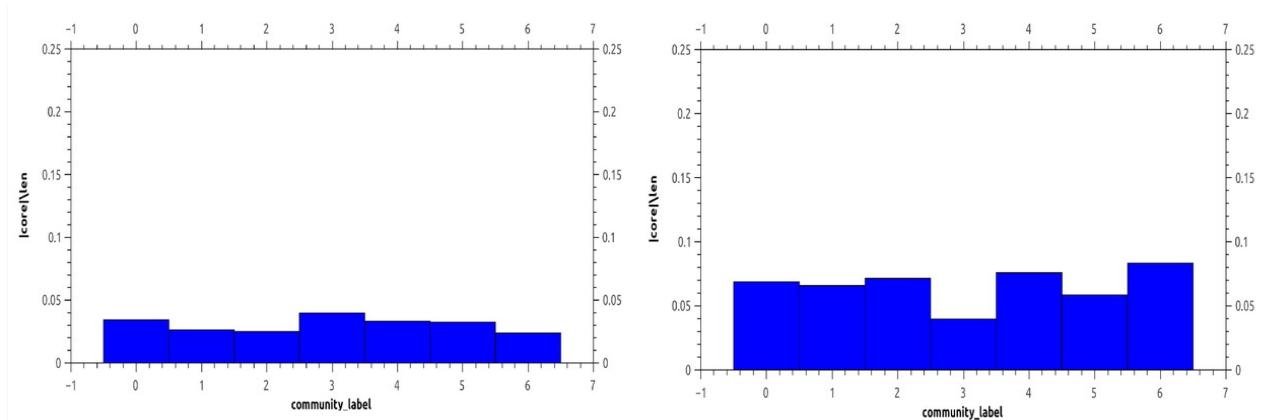


- The diagram shows the vertices satisfying the equation $(bc_norm)^2 + (pr_norm)^2 > 2.5^2$

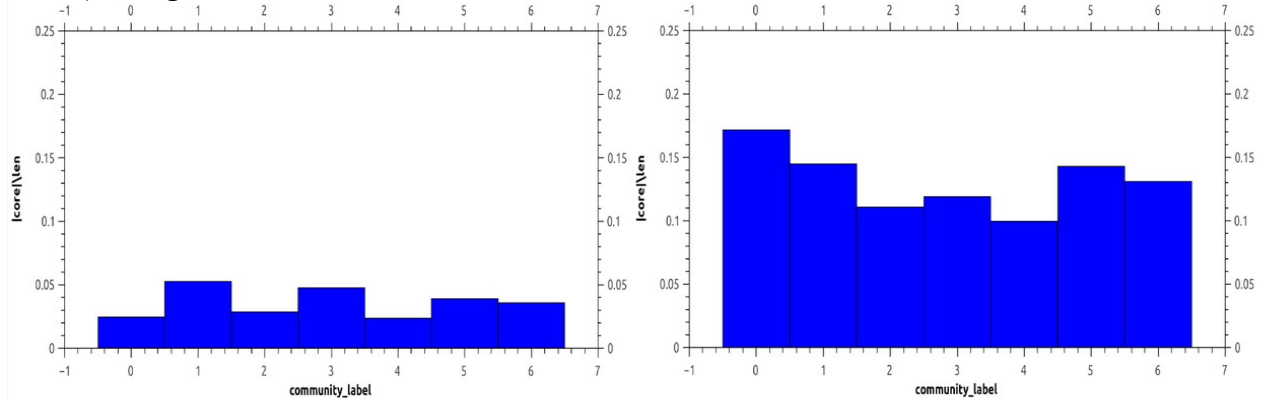


- A Histogram showing fraction of core vertices in each community of email graph.

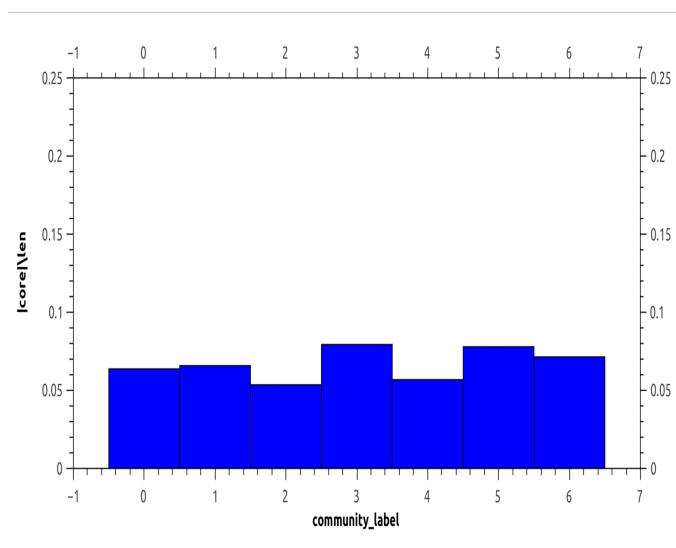
a) Using BC as metric



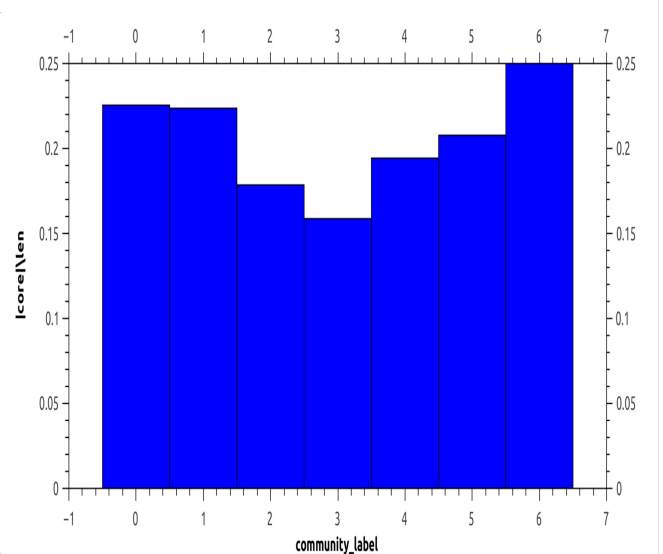
b) Using PR as metric



c) $(bc_norm)^2 + (pr_norm)^2 > 6.25$



d) $(bc_norm)^2 + (pr_norm)^2 > 1$



Observations

The threshold value for normalized value of metric can be different for different communities, though they belong to same graph. The threshold value 1 has more core vertices whereas using 2.5 has less core vertices. 1 and 2.5 are not standard values but can be changed according to the number of vertices communities has.

It has been observed that as we increase the threshold the number of core members decreases in a community. But the fraction of core vertices remain same on changing threshold in one of the community of email graph having label 3. This is a very interesting observation. It suggests that this community has same core vertices for two different threshold. Core vertices have very high z-score and can be analyzed for further study. They are of special nature.

The number of core vertices in each community varies linearly with the size of the community. The community having large amount of vertices tend to have more special (core) vertices. It can be explained from example that consider two groups of people. One has 50 members and other has 1000 members. In order to represent the group obviously more people are required in case of 1000 membered group as compared to 50 people group. As the network size increases more special nodes are required to bind all members.

PageRank and betweenness centrality behaved in similar way. It can be seen from the two curves between sorted and normalized value of the metric v/s the number of vertices. In case of betweenness centrality curve most of the vertices have same value but then there is sudden rise in curve. It behaved in exponential way. The vertices above the knee are core vertices. In case of PageRank, firstly normalized and sorted value of metric increases linearly with number of vertices but then rises suddenly for some vertices and behaved exponentially. The vertices above the knee are core vertices.

Same results are obtained for other social network graphs as well. It suggests that communities tend to behave in similar way irrespective of their graph from which they are derived. Communities have their independent identity.

Future Works

- Now after having communities and their core members, we can see that how core vertices changes with respect to time. Some members which were initially non-core transformed to core members at different time stamps.
- For each set of core vertices in the previous snapshot, we can check if they share the same community label in the new snapshot.
- We can see how many core vertices switch communities.
- We can see how many vertices share more than one community labels in case of overlapping communities and how they changes with respect to time.

References

- [1] Weiss R S, Jacobson E. *Am. Sociol. Rev.* 20, 1955, 661.
- [2] M. Rosvall and C.T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118– 1123, 2008.
- [4] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review*, vol. E 69, no. 026113, 2004.
- [5] J. Jin, L. Pan, C. Wang, and J. Xie, “A center-based community detection method in weighted networks,” in *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, ser. ICTAI ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 513–518. [Online]. Available: <http://dx.doi.org/10.1109/ICTAI.2011.83>
- [6] Newman M E J. *SIAM Rev.*, 2003,45(2),167.
- [8] Fortunato S. Community detection in graphs. *ArXiv:0906.0612*, 2009.
- [7] Newman M E J, Girvan M. *Phys. Rev. E* 69 (2), 2004, 026113.
- [8] Brandes U, Erlebach T. *Network analysis :methodological foundations*. Springer Verlag, Berlin, 2005.
- [9] Palla G, Derányi I, Farkas I, Vicsek T. Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. *Nature* 435, 2005, 814-818.
- [10] J. I. Alvarez-Hamelin, A. Barrat, L. Dall’Asta, and A. Vespignani. k-core decomposition: a tool for the visualization of large scale networks. *CoRR*, cs.NI/0504107, 2005.
- [11] V. Batagelj and A. Mrvar. Generalized cores. *Journal of the ACM*, 5, 2002.
- [12] J. Healy, J. Janssen, E. E. Milios, and W. Aiello. Characterization of graphs using degree cores. In *Proc. 3rd Workshop on Algorithms and Models for the Web Graph (WAW)*, 2006.
- [13] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. In *Proc. 17th Int’l World Wide Web Conf. (WWW)*, 2008.
- [14] M. E. J. Newman. Detecting community structure in networks. *The European Physical J. B*, 38:321–330, 2004.
- [15] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(066133), 2004.
- [16] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(036104), 2006.
- [17] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Proc. 47th IEEE Symp. Found. Comp. Sci. (FOCS)*, 2006.
- [18] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proc. Natl. Acad. Sci. USA*, 104(18):7327–7331, 2007.
- [19] S. Papadimitriou, J. Sun, C. Faloutsos, and P. S. Yu. Hierarchical, parameter-free community discovery. In *Proc. 19th European Conf. Mach. Learn. (ECML)*, 2008.
- [20] N. Mishra, R. Schreiber, I. Stanton, and R. E. Tarjan. Finding strongly-knit clusters in social networks. *Internet Mathematics*, 5(1–2):155–174, 2009.
- [21] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.
- [22] <http://www.gephi.org>
- [23] <http://www.cc.gatech.edu/dimacs10/archive/clustering.shtml>