Detecting Communities using Bibliographic Metrics

Hemant Balakrishnan
School of Computer Science
University of Central Florida
Orlando, Florida 32816-2362
hemant@cs.ucf.edu

Narsingh Deo
School of Computer Science
University of Central Florida
Orlando, Florida 32816-2362
deo@cs.ucf.edu

***Abstract*— We propose an efficient and novel approach for discovering communities in real-world random networks. *Communities* are formed by subsets of nodes in a graph, which are closely related. Extraction of these communities facilitates better understanding of such networks. Community related research has focused on two main problems: community discovery and community identification. *Community discovery* is the problem of extracting all the communities in a given network where as *community identification* is the problem of identifying the community to which a given set of nodes from the network belong. In this paper we first give a brief survey of the existing community-discovery algorithms and then propose a novel algorithm to discovering communities using bibliographic metrics. We also test the proposed algorithm on real-world networks and on computer-generated models with known community structures.**

***Index Terms*—Community discovery/identification, graph clustering.**

I. Introduction

Recent studies on real-world random networks have revealed several interesting and significant properties like degree distribution, average distance between pairs of nodes, and network transitivity. These properties differentiate real-world networks from the classical Erdos and Renyi random graphs. One such property is community structure. These networks are locally dense but globally sparse. Each of these locally dense regions may be viewed a community.

The term community has been defined in more than one way. Initially *cliques* and *near-cliques* were used to define communities with the idea that high connectivity corresponds to similarity between those nodes. Kleinberg while studying web graphs introduced the concept of "hubs" and "authorities". *Authorities* are web pages which are highly referenced and *hubs* are web pages that reference many authority pages. Later Gibson, Kleinberg and Raghavan define communities in web graph as a core of central, authoritative pages connected together by hub pages [1]. Kumar, et al. define communities as bipartite cores: a *bipartite core* in a graph G consists of two (not necessarily disjoint) sets of nodes L and R, such that every node in L is adjacent to every node in R [2]. Flake, Lawrence and Giles define them as a set of nodes C in a graph G that have more edges (undirected) to members of the community than to non-members [3]. This definition of community is very similar to the concept of defensive alliances in graphs introduced by Hedetniemi, Hedetniemi and Kristiansen [4]. A *defensive alliance* in a graph G is defined as a non-empty set of vertices $S \subseteq V$ such that for every vertex $v \in S$, $|N[v] \cap S| \geq |N[v] - S|$ where $N[v]$ represents the closed neighborhood of vertex v. Girvan and Newman define communities based on edge density: subsets of nodes within which edges are dense, but between which edges are sparse [5].

Two problems of interest are community discovery and community identification. From a graph theoretic perspective *community discovery* is the problem of classifying nodes of a graph G into subsets $C_i \subseteq V$, $0 \leq i < k$, such that nodes belonging to a subset C_i are all closely related whereas *community identification* is the problem of identifying the community C_i to which a set of nodes $S \subseteq V$ belong. In general community identification is considered an easier problem compared to community discovery, especially if the input graph is large.

Extracting communities in a graph has number of applications: in social and biological networks we could use communities to study interactions between groups of people or animals; in web graphs to automate the process of creating web directories like `http://directory.google.com`, focused crawling or as a tool for visualizing search results grouped by categories; in image segmentation to separate the background of an image from its foreground. The rest of the paper is organized as follows. In Section 2 we do a brief survey on some of the existing algorithms for discovering communities. In Section 3 we describe our new bibliometric algorithm and in Section 4 we provide some experimental results followed by conclusion and future work.

II. Existing algorithms

Most of the existing community-discovery algorithms employ hierarchical clustering techniques to extract communities. Every hierarchical clustering algorithms works in two phases, at first, a metric is defined to portray the similarity between two nodes. Then there are two possible ways of extracting communities using the defined metric i) agglomerative and ii) divisive. The *agglomerative* algorithms compute the similarity metric between every pair of nodes in the graph. They begin extracting the communities by initially considering each node in the graph to belong to an individual community and during the course of the algorithm combine nodes that are closely related to form bigger communities. *Divisive* algorithms, on the other hand, compute the similarity metric between adjacent nodes. Then begin extracting the communities by initially considering all the nodes in the graph to belong to a single community and during the course of the algorithm remove edges between pairs of nodes that are the least similar. This subdivides the graph into smaller but tighter communities. Hierarchical clustering is preferred to other methods because in addition to providing the different clusters in the network it also provides their hierarchy. The following sections provide insight into some existing algorithms to discover communities.

A. *Hierarchical Agglomerative Clustering*

Hierarchical Clustering algorithm [6] as the name implies uses the agglomerative approach. The algorithm first computes the number of node-independent or edge-independent paths between pairs of nodes. Two (or more) paths are *node-disjoint* if they don't share any node except maybe the initial and final nodes. Similarly, two (or more) paths are *edge-disjoint* if they don't share any edges. A large number of node-disjoint/edge-disjoint paths between a node pair represent better similarity. After computing the similarities between all pairs of nodes we start with *n* isolated nodes from the input graph and introduce edges between pairs of nodes, starting with the pair of highest similarity and progressing to the weakest. Introduction of these edges results in formation of components representing communities. The hierarchical clustering algorithm fails on some graphs. For example it fails on graphs with nodes of degree one. The value of edge-disjoint count and the node-disjoint count is very low for such nodes and as a result these nodes remain isolated from the network when communities are formed.

B. *Edge clustering-coefficient*

Radicchi, et al. [7] proposed a divisive algorithm that uses edge clustering co-efficient as a weight measure for the edges. *Edge clustering co-efficient* of an edge is given by:

$$\frac{T_{u,v}+1}{\min(d_u,d_v)},$$

where $T_{u,v}$ represents the number of triangles to which the edge *u, v* belongs to and d_u represents the degree of vertex *u*. The motivation behind this method is as follows: edges connecting nodes in different communities are included in few or no triangles, and tend to have small values of $T_{u,v}$. On the other hand edges connecting nodes in the same community would be a part of many triangles and would have a large value for $T_{u,v}$. Thus removing edges with low edge clustering-coefficient value would get rid of the inter-community edges and expose the underlying communities. Clearly the algorithm would fail on all triangle free graphs.

C. *Edge betweeness*

Girvan and Newman [5] suggested a divisive algorithm that uses inverse of edge betweenness as a weight measure of the edges. *Edge betweenness* of an edge is defined as the number of shortest paths between pairs of vertices that pass through the edge. The edge betweenness of inter-community edges would be high, as the shortest paths between nodes in the two different communities would have to pass through them. After computing the edge betweenness of all the edges in the graph, the edge with the lowest weight is removed and the edge betweeness of the remaining edges recomputed. This process is repeated there by exposing the underlying community structure. It is to be noted that all divisive algorithms would fail on certain class of graphs whose edges don't portray the community structure, for example bipartite graphs. Any bipartite graph consists of two sets of nodes, and no edges exist between nodes in the same set. However each of these sets would constitute a community. Removal of edges in any order using a divisive algorithm would not reveal the two communities.

D. *Flow based approach*

For the flow based algorithms the input graph is considered to be a flow network with the edges representing pipes with unit capacity. If one were to introduce a flow into this network via a *source node* then drain the network via a *sink node* then edges between the communities would act as a bottle neck controlling the amount of the flow. Using a *max flow-min cut* algorithm one can easily identify the edges forming a bottleneck. Removal of these edges would result in the bisection of the graph into two communities. One could further bisect these communities to obtain more communities in the same fashion. Flake, Lawrence and Giles came up with a modified flow based algorithm to identify communities in web graphs [3].

E. *Resistor network approach*

In the resistor network based approach introduced by Girvan and Newman in [8] the graph is considered as a resistor network with edges being substituted with a resistor of say unit resistance and then a fixed voltage is applied between a source node and sink node. Current would flow from the source to the sink via a number of paths and the paths with least resistance would carry the greatest fraction of the current. These paths can be identified by solving Kirchoff's equations. Removal of these edges would divide the graph into communities. A linear algorithm employing a modified resistor networks based technique for community discovery was given by Wu and Huberman in [9].

F. *Random Walks*

Another promising approach to discover communities is by using random walks [8]. This method would require two steps a) perform random walks of fixed length from different vertices in the graph, b) for all pairs of vertices count the number of occasions the same walk traversed both the vertices. A large count indicates better similarity between the vertex pair. Once the similarities between every vertex pair are computed one could use an agglomerative algorithm to discover the communities in the network.

Apart from the above mentioned algorithms spectral algorithms and graph partitioning, graph clustering algorithms can also be applied to perform community discovery/identification.

III. Bibliometric approach

The motivation for the current work is from bibliographic metrics which have been used to determine similarity between publications. There are two measures which have widely been used: bibliographic coupling and co-citation coupling. Given two documents, *bibliographic coupling* [10] is defined as number of publications that cite both the given documents and *co-citation coupling* [11] is defined as the number of publications that are cited by both the given documents. Combining the above two measures we obtain a unified metric that can be used to determine similarity between two nodes in a graph. The measure of similarity between two nodes *u* and *v* in a graph *G* is given by:

$$\frac{|N[u] \cap N[v]|}{\min(d_u, d_v) + 1},$$

where $N[u]$ refers to the closed neighborhood of node *u* and d_u refers to its degree. In simple terms we rate the similarity between two nodes in a network by the number of common neighbors they share. The more the number of common neighbors the better the similarity.

The following algorithm illustrates how to compute the bibliometric similarity between all pairs of vertices in the graph.

```
procedure: ComputeSimilarity($G(V,E)$)
for all $u \in V$
    for all $v \in V$
        if $(u, v) \in E$
            $d_{u} = d_u + 1$
        end if
        $N_{u,v} = 0$
    end for
end for
for all $u \in V$
    for all $v \in V$
        if $(u, v) \in E$
            $N_{u,v} = N_{u,v} + 2$
            for all $w \in V$
                if $(v,w) \in E$ and $w \neq u$
                    $N_{u,w} = N_{u,w} + 1$
                end if
            end for
        end if
    end for
end for
for all $u \in V$
    for all $v \in V$
        $S_{u,w} = \frac{N_{u,w}}{\min(d_u, d_v) + 1}$
    end for
end for
```

Given a graph *G* of order *n* we compute the measure of similarity between every pair of nodes in the graph using the above metric. Once the similarity between all pairs of vertices in the graph has been defined we start with *n* isolated nodes and introduce edges between pairs of nodes starting with the pair of highest similarity and progressing to the weakest.

One drawback of the agglomerative algorithms developed so far is that they classify pendent nodes as separate communities [5]. This is because the similarity metric used is some global property like number of paths or number of node independent paths between node pairs. As a result this value is low for edges connecting pendent nodes to the rest of the graph. This drawback could be overcome by using local measures of similarity like the one introduced above. And by using an agglomerative algorithm rather than a divisive one, we would be able to recognize communities in graphs like bipartite graphs where there are no edges between nodes of the same community.

IV. Experimental Results

In this section we test the performance of our algorithm on computer generated networks and real-world networks whose community structure is already known.

A. *Computer-generated networks:*

Graphs with known community structure were generated as described by Girvan and Newman in [5]. Each of the generated graphs consists of 128 nodes divided into 4 communities of equal size. Edges were placed uniformly at random, such that each node on average has z_{in} neighbors in the same community and z_{out} neighbors outside. The average degree of the graph is kept close to 16. Our algorithm was tested on these graphs and the fraction of nodes that were classified correctly was measured by varying the number of intercommunity edges per vertex from 0 to 16. The algorithm correctly classified up to 90% of the nodes in graphs with $z_{out} \leq 6$ and close to 70% of the nodes in graphs with $6 < z_{out} \leq 8$. For graphs with $z_{out} > 8$ each node on average has more neighbors outside the community than inside and the graphs no longer posses a well defined community structure. Our results are summarized in Figure 1.

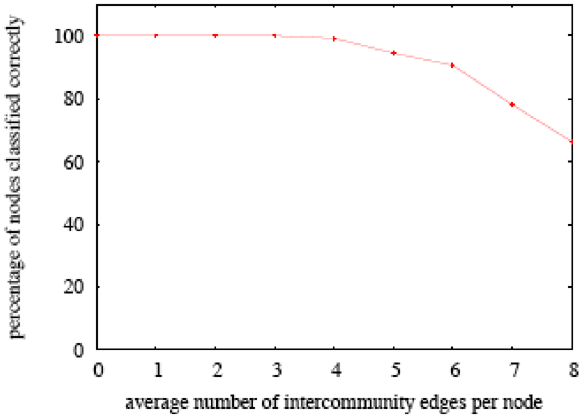

Figure 1: Performance on computer-generated models.

B. *The Zachary Karate Club network:*

The karate club network is a social network consisting of 34 nodes representing people from a karate club at an American university and edges representing friendships between them. This network was compiled by Zachary [12] who was studying the social interactions between the members of the club. During the course of the study a dispute between the administrator of the club and the instructor of the club resulted in the split of the club into two. The instructor opened another club with about half the members from the original club. The karate club network is shown in Figure 2, the square nodes indicate the instructors group and the round nodes indicate the administrators group.

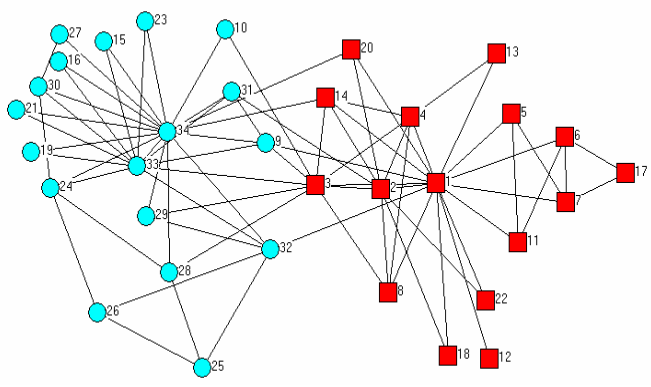

Figure 2: The Zachary Karate Club Network

We apply our bibliometric algorithm to the karate club network to identify the factions in the club. The dendrogram in Figure 3 shows the communities as discovered by our algorithm. All the nodes in the two groups were classified correctly except for the nodes, 10 and 28 which were not classified into any community.

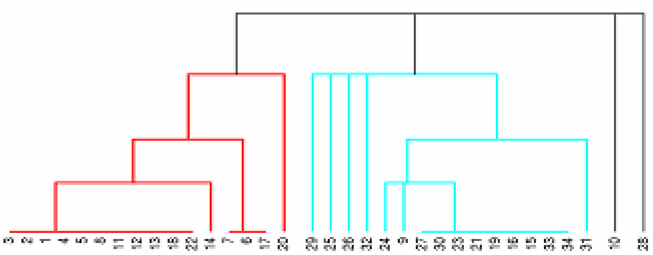

Figure 3: Communities identified in the Zachary Karate Club Network.

C. *The Football team network:*

The community structure of the football team network was first studied by Girvan and Newman in [5]. This network represents the regular schedule of the Division I college football games for the year 2000 and consists of 115 nodes. The nodes in the network represent teams and the edges represent matches between them. Out of the 115 teams 110 were divided into 11 conferences and the remaining 5 were not classified into any conference. Each team on average played seven intra-conference games and four inter-conference games during the season. More over the inter-conference games were not uniformly distributed with more games being played between teams that are geographically close to one another than with teams that are further apart. Applying our algorithm to this network we were able to extract the conference structure of the network with high precision. Our results are shown by means of a dendrogram in Figure 5. Labels in the dendrogram represent the name of the team followed by the conference number to which they belong. The teams that did not belong to any conference (represented by conference number 5 in Figure 5) ended up with the conferences with which they were closely associated. For certain teams the network structure did not portray the conference structure and these teams ended up being misclassified, which was anticipated. For example the Texas Christian team belonging to conference 4 played majority of their games with teams belonging to conference 11.

D. *The Santa Fe Institute collaboration network:*

Next we test out algorithm on a scientific collaboration network consisting of scientists from different disciplines at the Santa Fe Institute. The community structure of this network was studied by Girvan and Newman in [5]. The nodes of the network represent scientists from the Santa Fe Institute and an edge is drawn between two nodes if the corresponding scientists have coauthored at least one publication during the calendar year 1999 or 2000. On average each scientist coauthored articles with approximately five others. The actual network consists of 271 nodes, but here we study the largest component of the network consisting of 117 nodes as the community structure of the former was not available. Figure 4 shows the structure of the collaboration network with different node shapes indicating different disciplines of research. The entire network could be broken down into four major components and a few of these could be further divided.

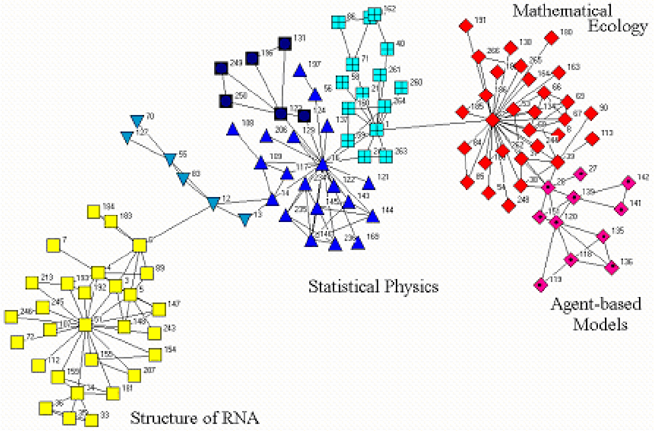

Figure 4: The largest component of the Santa Fe Institute collaboration network.

The nodes represented by squares represent the community of scientists working primarily on the structure of RNA. The nodes represented by triangles, inverted triangles, crossed and circled squares represent scientists working on different areas in statistical physics. The nodes in diamonds represent the scientists working on the mathematical models in ecology and the ones in dotted diamonds represents the group of scientists using agent-based models to study problems in economics and traffic flow. Application of our algorithm to this collaboration network identifies all the major communities in the network. Our results are shown by means of a dendrogram in Figure 6. The communities representing scientists using agent-based models and the ones working on mathematical ecology seem to be classified as a single community. Further divisions within the scientists working on statistical physics are also visible.

E. *Roget's thesaurus*

To put the algorithm to further testing we test our approach on the Roget's thesaurus network, which consists of 1022 nodes each representing one category in the 1879 edition of Peter Mark Roget's *Thesaurus of English Words and Phrases*, edited by John Lewis Roget. A directed edge is drawn from nodes *u* to node *v*, if Roget gave a reference to the category represented by node *v* among the words and phrases of category represented by node *u*.

Applying our algorithm to this lexical network resulted in division of the entire network into a number of small communities. Each community consisted of words that were closely related. Examples of these are listed in Table 1.

Table 1: Six communities identified in the Roget's thesaurus network.

uniformity agreement conformity concurrence cooperation concord peace Assent	velocity haste earliness instantaneity transientness present-time different-time time period…	heat thermometer furnace refrigeratory refrigeration cold calefaction investment covering…
untruth falsehood misteaching deception cunning misinterpretation ambush…	clergy churchdom belif thoelogy orthodoxy irreligion idolatory…	color ugliness ornamentation deterioration blemish beauty simplicity…

V. Conclusion and Future Work

In this paper we provide a brief survey on the various existing algorithms to identifying communities in a real-world random network. We also propose a novel algorithm for community discovery based on bibliographic metrics. The proposed algorithm addresses some of the drawbacks found in existing techniques. Algorithms employing local properties of the graph seem to produce better communities than the ones employing global properties. In near future we intend to test

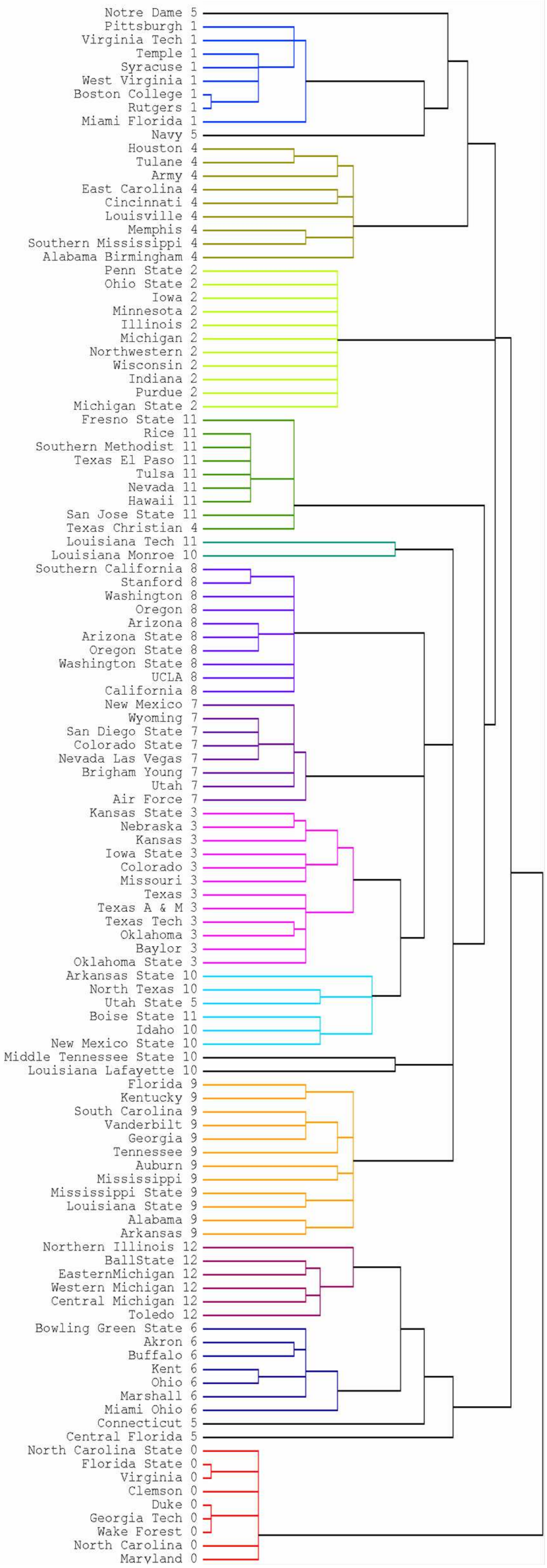

Figure 5: Communities identified in the College Football Network.

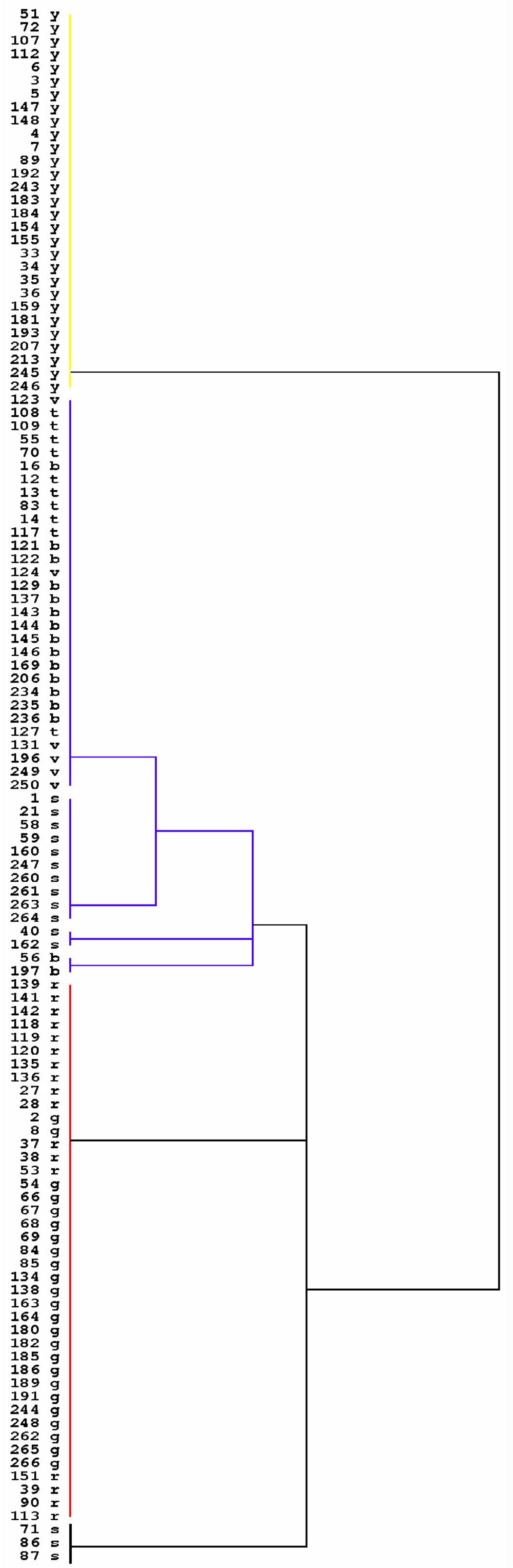

Figure 6: Communities identified in the largest component of the Santa Fe Institute collaboration network.

our algorithm on large, sparse, random networks like semantic networks, the Internet and the World Wide Web. We would also be employing the similarity measure introduced above to perform community identification

VI. Acknowledgment

We would like to thank Mark Newman for providing us with the data for the college football network and the Santa Fe Institute collaboration network.

References

[1] D. Gibson, J. M. Kleinberg, and P. Raghavan, "Inferring Web Communities from Link Topology," presented at 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA, 1998.

[2] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Extracting Large-Scale Knowledge Bases from the Web," presented at Proceedings of 25th International Conference on Very Large Data Bases, 1999.

[3] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient Identification of Web Communities," presented at Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.

[4] S. M. Hedetniemi, S. T. Hedetniemi, and P. Kristiansen, "Alliances in Graphs," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 48, pp. 157-177, 2004.

[5] M. Girvan and M. E. J. Newman, "Community Structure in Social and Biological Networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 7821-7826, 2002.

[6] J. Scott, *Social Network Analysis: A Handbook*: Sage Publications, 2000.

[7] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and Identifying Communities in Networks," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 2658-2663, 2004.

[8] M. E. J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical Review E*, vol. 69, pp. 026113-1, 2004.

[9] F. Wu and B. A. Huberman, "Finding Communities in Linear Time: a Physics Approach," *The European Physics Journal B*, vol. 38, pp. 331-338, 2004.

[10] M. Kessler, "Bibliographic coupling between scientific papers," *American Documentation*, vol. 14, pp. 10-25, 1963.

[11] H. Small, "Co-citation in the scientific litrature: A new measure of the relationship between two documents.," *Journal of American Society for Information Science*, vol. 24, pp. 265-269, 1973.

[12] W. W. Zachary, "An information flow model for conflict and fission in small groups.," *Journal of Anthropological Research*, vol. 33, pp. 452-473, 1977.