CS532 Web Science: Assignment 5

Finished on March 3, 2016

Dr. Michael L. Nelson

Naina Sai Tipparti
ntippart@cs.odu.edu

Contents

Listings

List of Figures

List of Tables

Problem 1

Question

We know the result of the Karate Club (Zachary, 1977) split. Prove or disprove that the result of split could have been predicted by the weighted graph of social interactions. How well does the mathematical model represent reality?

Generously document your answer with all supporting equations, code, graphs, arguments, etc.

Useful sources include:

Original paper

http://aris.ss.uci.edu/l̃in/76.pdf

Slides

http://www-personal.umich.edu/l̃adamic/courses/networks/si614w06/ppt/lecture18.ppt

http://clair.si.umich.edu/si767/papers/Week03/Community/CommunityDetection.pptx

Code and data

http://networkx.github.io/documentation/latest/examples/graph/karate_club.html

http://nbviewer.ipython.org/url/courses.cit.cornell.edu/info6010/resources/11notes.ipynb

http://stackoverflow.com/questions/9471906/what-are-the-differences-between
-community-detection-algorithms-in-igraph/9478989#9478989

http://stackoverflow.com/questions/5822265/are-there-implementations-of
-algorithms-for-community-detection-in-graphs

http://konect.uni-koblenz.de/networks/ucidata-zachary

http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm#zachary

https://snap.stanford.edu/snappy/doc/reference/CommunityGirvanNewman.html

http://igraph.org/python/doc/igraph-pysrc.html#Graph.community_edge_betweenness

Answer

As was illustrated in the original study of Zachary's karate club, [1], a prediction of the structure of the club if a separation were to occur can be made with a high degree of accuracy using weighted edges based on the perceived "strength" of each relationship it modeled. The prediction method outlined in the original paper was an implementation of the *maximum flow-minimum cut labeling procedure* [2]. The pickled [3] dataset of the existing karate club, with weights for each edge, was obtained from `http://nexus.igraph.org/api/dataset_info?id=1&format=html` and used to create the graph shown in Figure 1.

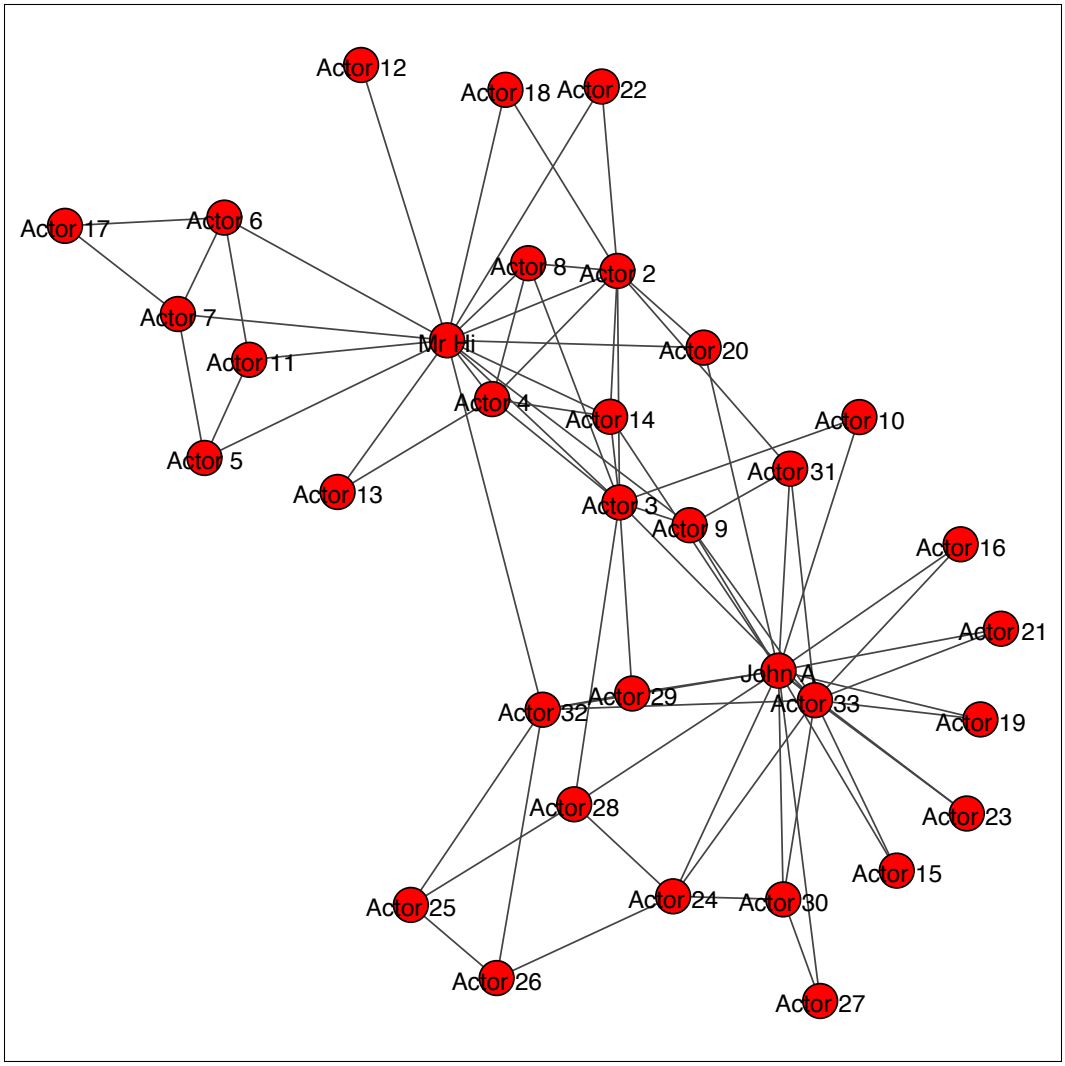

Figure 1: The Existing Graph

The actual structure of the two resulting clubs after the split are shown in Figure 2.

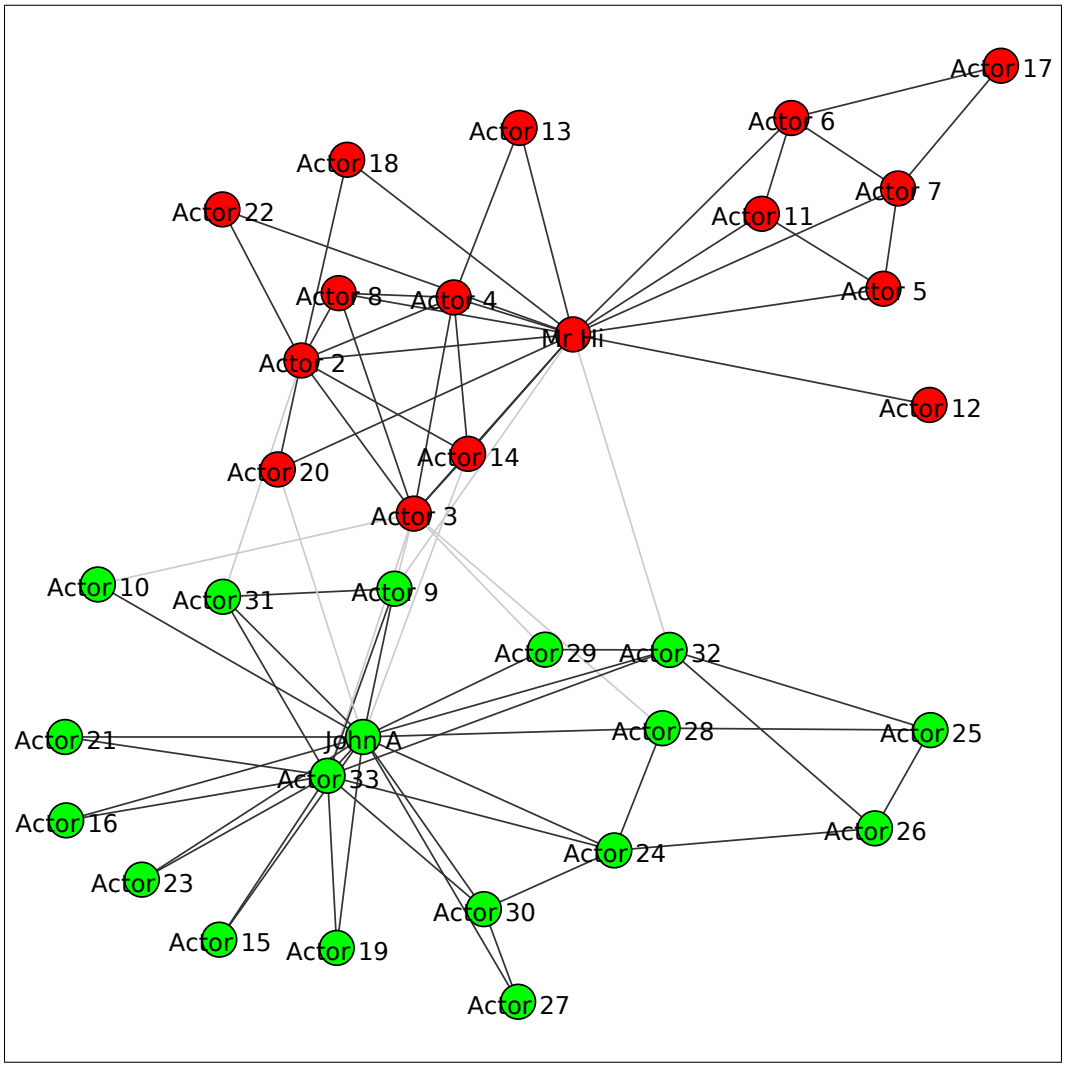

Figure 2: Actual Graph After Split

To predict a separation of the existing graph into two or more distinct community components a pair of community detection algorithms will be employed and the results will be compared to the actual results of the split from Zachary's original study [1]. Community Detection was chosen as a means for predicting the results of fission events because it is logical that a given community would less likely be split along strong inter-community edges than those weaker, community-spanning edges.

The first algorithm used was the Edge Betweenness algorithm, developed by Girvan and Newman [4]. This is a divisive algorithm that removes edges that have the highest betweenness measure because these tend to be community-spanning edges.

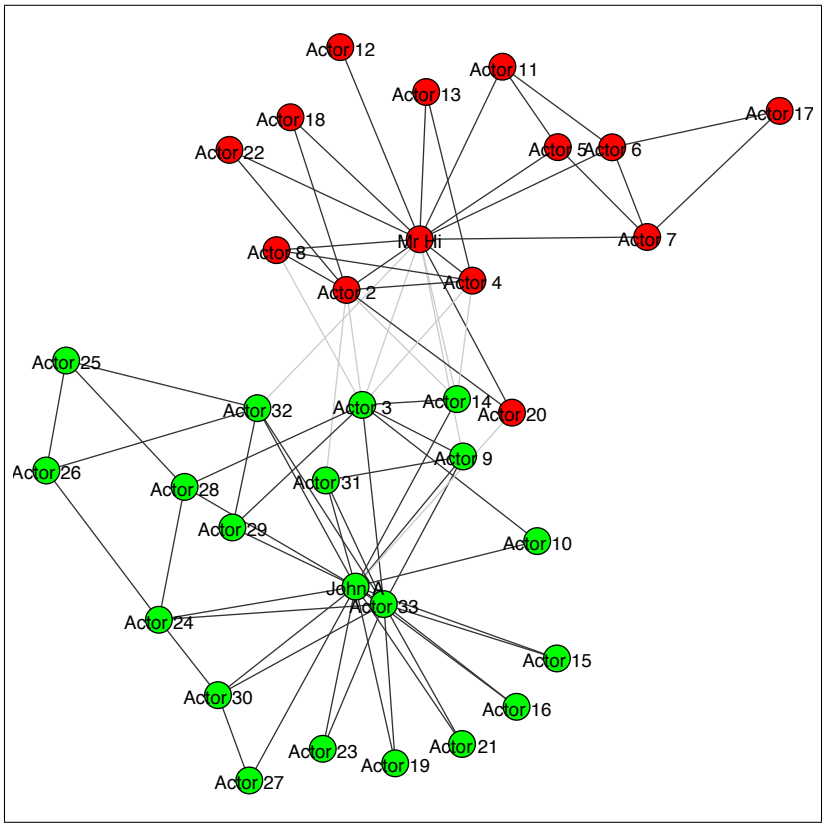

Figure 3: Prediction of Edge Betweenness Algorithm

```
Edge Betweenness method results:
Variant elements:
[3, 14]
94.12% accuracy
```

As you can see this method is fairly accurate, with over 94% of the prediction being correct.
My Girvan-Newman implementation has a $\frac{32}{34} = 94\%$ success rate, making it inferior in this case but still effective at predicting almost all of the group memberships. My implementation also predicted that individual 9 would stay with Mr. Hi, which is the one membership that Zachary missed.

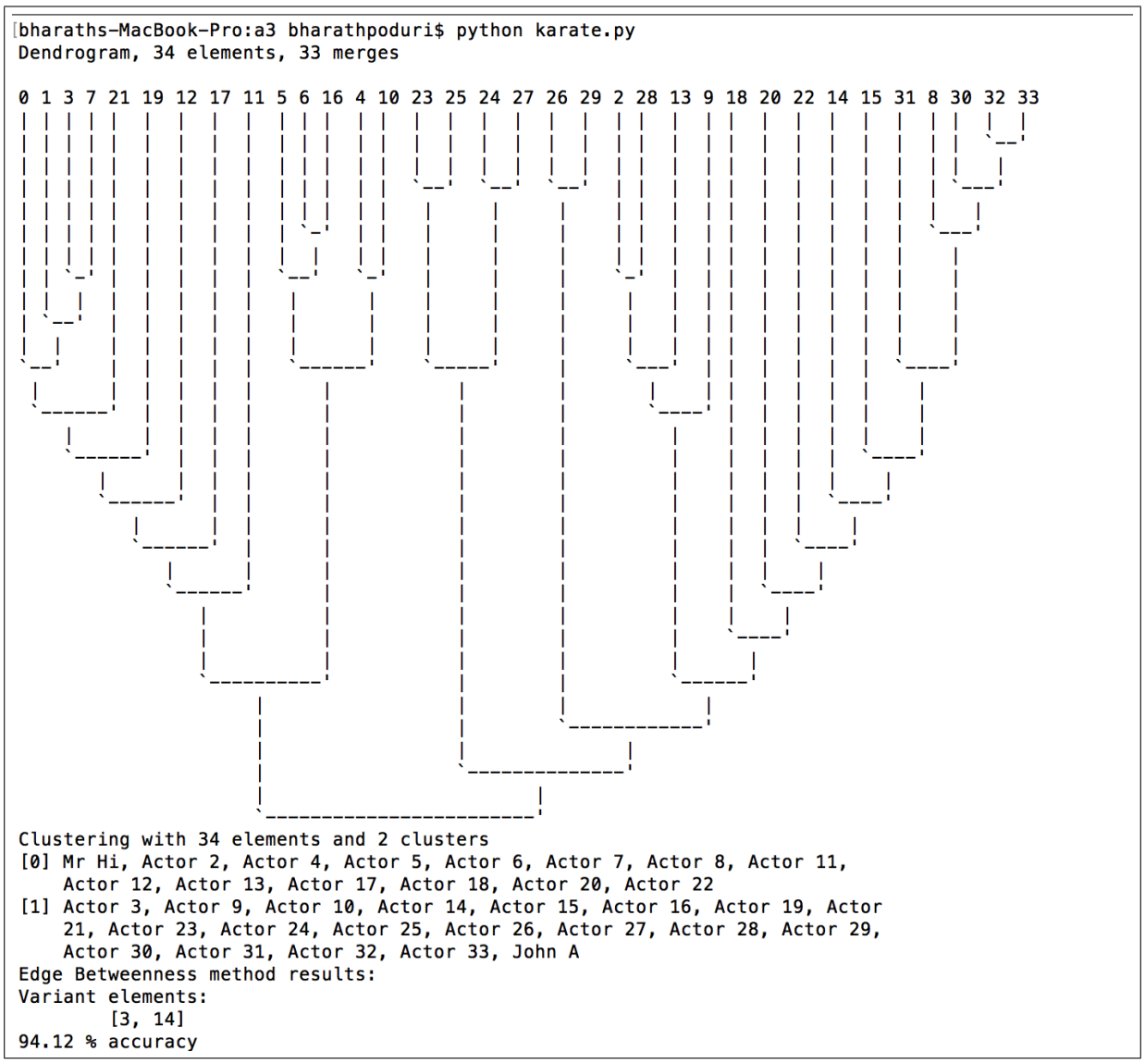

Figure 4: Dendrogram for Girvan-Newman

Table 1 shows the results compared with Zachary's original predictions and the actual data (*Officers'* is John A's faction). Column 5 shows whether my Girvan-Newman implementation resulted in a *Hit* (correctly calculated membership) or *Miss* (incorrectly calculated membership).

Individual	Actual Group Membership From Split	Zachary's Ford and Fulkerson Procedure Modeled Group Membership From Split	Girvan-Newman Modeled Group Membership From Split	Hit/Miss For Girvan-Newman
1	Mr. Hi	Mr. Hi	Mr. Hi	Hit
2	Mr. Hi	Mr. Hi	Mr. Hi	Hit
3	Mr. Hi	Mr. Hi	Mr. Hi	Hit
4	Mr. Hi	Mr. Hi	Mr. Hi	Hit
5	Mr. Hi	Mr. Hi	Mr. Hi	Hit
6	Mr. Hi	Mr. Hi	Mr. Hi	Hit
7	Mr. Hi	Mr. Hi	Mr. Hi	Hit
8	Mr. Hi	Mr. Hi	Mr. Hi	Hit
9	Mr. Hi	Officers'	Mr. Hi	Hit
10	Officers'	Officers'	Mr. Hi	Miss
11	Mr. Hi	Mr. Hi	Mr. Hi	Hit
12	Mr. Hi	Mr. Hi	Mr. Hi	Hit
13	Mr. Hi	Mr. Hi	Mr. Hi	Hit
14	Mr. Hi	Mr. Hi	Mr. Hi	Hit
15	Officers'	Officers'	Officers'	Hit
16	Officers'	Officers'	Officers'	Hit
17	Mr. Hi	Mr. Hi	Mr. Hi	Hit
18	Mr. Hi	Mr. Hi	Mr. Hi	Hit
19	Officers'	Officers'	Officers'	Hit
20	Mr. Hi	Mr. Hi	Mr. Hi	Hit
21	Officers'	Officers'	Officers'	Hit
22	Mr. Hi	Mr. Hi	Mr. Hi	Hit
23	Officers'	Officers'	Officers'	Hit
24	Officers'	Officers'	Officers'	Hit
25	Officers'	Officers'	Officers'	Hit
26	Officers'	Officers'	Officers'	Hit
27	Officers'	Officers'	Officers'	Hit
28	Officers'	Officers'	Officers'	Hit
29	Officers'	Officers'	Officers'	Hit
30	Officers'	Officers'	Officers'	Hit
31	Officers'	Officers'	Officers'	Hit
32	Officers'	Officers'	Mr. Hi	Miss
33	Officers'	Officers'	Officers'	Hit
34	Officers'	Officers'	Officers'	Hit

Table 1: Results of Split, as predicted by my Girvan-Newman Implementation and also compared to Zachary's predictions and the actual data

The second method used was the Leading Eigenvector algorithm developed by M. Newman [5]. This method uses a special matrix, called the modularity matrix, to determine which edges to remove.

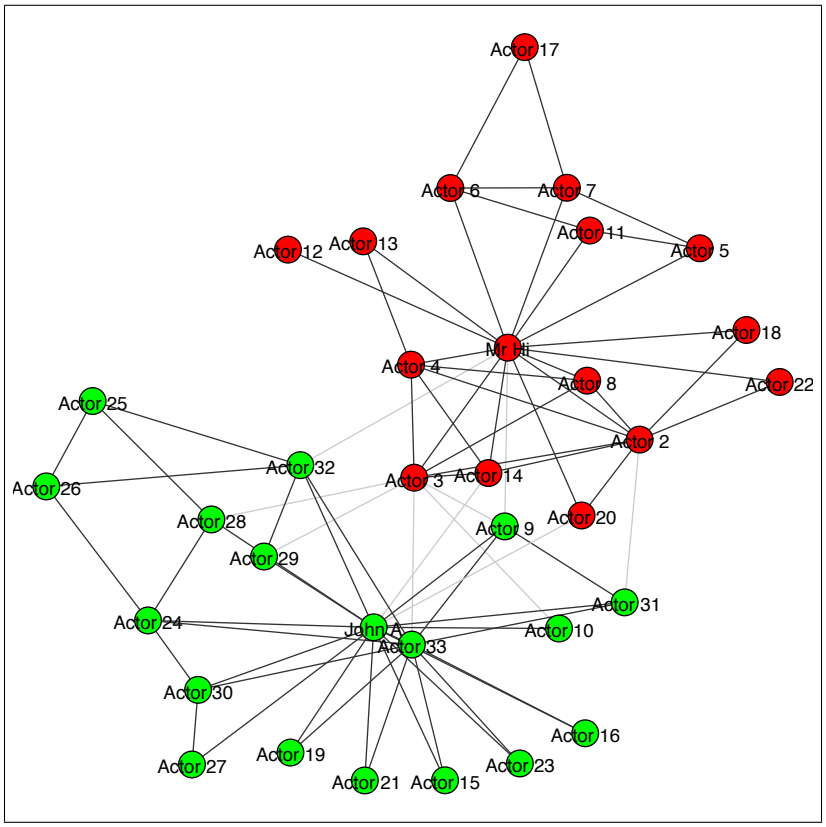

Figure 5: Prediction of Leading Eigenvector Algorithm

```
Leading Eigenvector method results:
Variant elements:
[]
100.0 % accuracy
```

This method proves 100% efficacy in its prediction.

```
Clustering with 34 elements and 2 clusters
[0] Mr Hi, Actor 2, Actor 3, Actor 4, Actor 5, Actor 6, Actor 7, Actor 8,
    Actor 11, Actor 12, Actor 13, Actor 14, Actor 17, Actor 18, Actor 20,
    Actor 22
[1] Actor 9, Actor 10, Actor 15, Actor 16, Actor 19, Actor 21, Actor 23, Actor
    24, Actor 25, Actor 26, Actor 27, Actor 28, Actor 29, Actor 30, Actor 31,
    Actor 32, Actor 33, John A
Leading Eigenvector method results:
Variant elements:
        []
100.0 % accuracy
```

Figure 6: Leading Eigenvector method from Listing 1

The python code to produce these graphs is shown in Listing 1.

```
#! /usr/bin/env python

import pickle
import igraph
from igraph import *
import cairo

FILENAME = 'karate.pickle'
# load the graph
data = pickle.loads(open(FILENAME).read())['karate']

# Create faction list of actual group membership after split,
# translated to values that would mirror a cut.membership list
factions_after_split = map(lambda x: int(x-1), data.vs['Faction'])

def compare(predicted):
    """
    Compares the predicted and actual numerical lists and
    returns a list of variant members and the accuracy of the prediction
    """
    res = []
    for idx, val in enumerate(factions_after_split):
        if predicted[idx] != val:
            # Translate index to match original dataset
            res.append(idx + 1)
    return res, 100 - round(float(len(res)) / float(len(factions_after_split)) * 100, 2)

def print_results(method, res, acc):
    print("{} method results: \nVariant elements:\n\t{}\n{} % accuracy".format(method, res,
        acc))

if __name__ == '__main__':
    # Plot existing graph
    layout = data.layout('fr')
    plot(data, "initial_karate_graph.pdf", layout=layout, vertex_label=data.vs['name'],
        margin=30)

    # Girvan-Newman Edge Betweenness method
    com_eb = data.community_edge_betweenness(
        clusters=2,
        directed=False,
        weights=data.es['weight'])
        print(com_eb)
    clust_eb = com_eb.as_clustering()
        print(clust_eb)
    res_eb, acc_eb = compare(clust_eb.membership)
    plot(clust_eb, "clust_eb.pdf", vertex_label=data.vs['name'], margin=25)
    print_results("Edge Betweenness", res_eb, acc_eb)

    # Newman Leading Eigenvector method
    clust_le = data.community_leading_eigenvector(clusters=2, weights=data.es['weight'])
        print(clust_le)
    res_le, acc_le = compare(clust_le.membership)
    plot(clust_le, "clust_le.pdf", vertex_label=data.vs['name'], margin=25)
    print_results("Leading Eigenvector", res_le, acc_le)

    # Plot  3..5 community predictions
    for i in xrange(3, 6):
        cluster = data.community_edge_betweenness(
            clusters=i,
            directed=False,
            weights=data.es['weight']).as_clustering()
                print (cluster)
        plot(cluster, "cluster" + str(i) + ".pdf", vertex_label=data.vs['name'], margin=25)
```

Listing 1: Finding Communities in Zachary's Karate Club

Problem 2

Question

We know the group split in two different groups. Suppose the disagreements in the group were more nuanced – what would the clubs look like if they split into groups of 3, 4, and 5?

Answer

The Edge Betweenness algorithm was run for target clusterings of three, four and five and the results are in Figures 8, 9, and 10.

```
Clustering with 34 elements and 3 clusters
[0] Mr Hi, Actor 2, Actor 4, Actor 5, Actor 6, Actor 7, Actor 8, Actor 11,
    Actor 12, Actor 13, Actor 17, Actor 18, Actor 20, Actor 22
[1] Actor 3, Actor 9, Actor 10, Actor 14, Actor 15, Actor 16, Actor 19, Actor
    21, Actor 23, Actor 27, Actor 29, Actor 30, Actor 31, Actor 32, Actor 33,
    John A
[2] Actor 24, Actor 25, Actor 26, Actor 28
Clustering with 34 elements and 4 clusters
[0] Mr Hi, Actor 2, Actor 4, Actor 5, Actor 6, Actor 7, Actor 8, Actor 11,
    Actor 12, Actor 13, Actor 17, Actor 18, Actor 20, Actor 22
[1] Actor 3, Actor 9, Actor 10, Actor 14, Actor 15, Actor 16, Actor 19, Actor
    21, Actor 23, Actor 29, Actor 31, Actor 32, Actor 33, John A
[2] Actor 24, Actor 25, Actor 26, Actor 28
[3] Actor 27, Actor 30
Clustering with 34 elements and 5 clusters
[0] Mr Hi, Actor 2, Actor 4, Actor 8, Actor 12, Actor 13, Actor 18, Actor 20,
    Actor 22
[1] Actor 3, Actor 9, Actor 10, Actor 14, Actor 15, Actor 16, Actor 19, Actor
    21, Actor 23, Actor 29, Actor 31, Actor 32, Actor 33, John A
[2] Actor 5, Actor 6, Actor 7, Actor 11, Actor 17
[3] Actor 24, Actor 25, Actor 26, Actor 28
[4] Actor 27, Actor 30
```

Figure 7: Groups Predicted by Girvan & Newman Betweenness Clustering from Listing 1

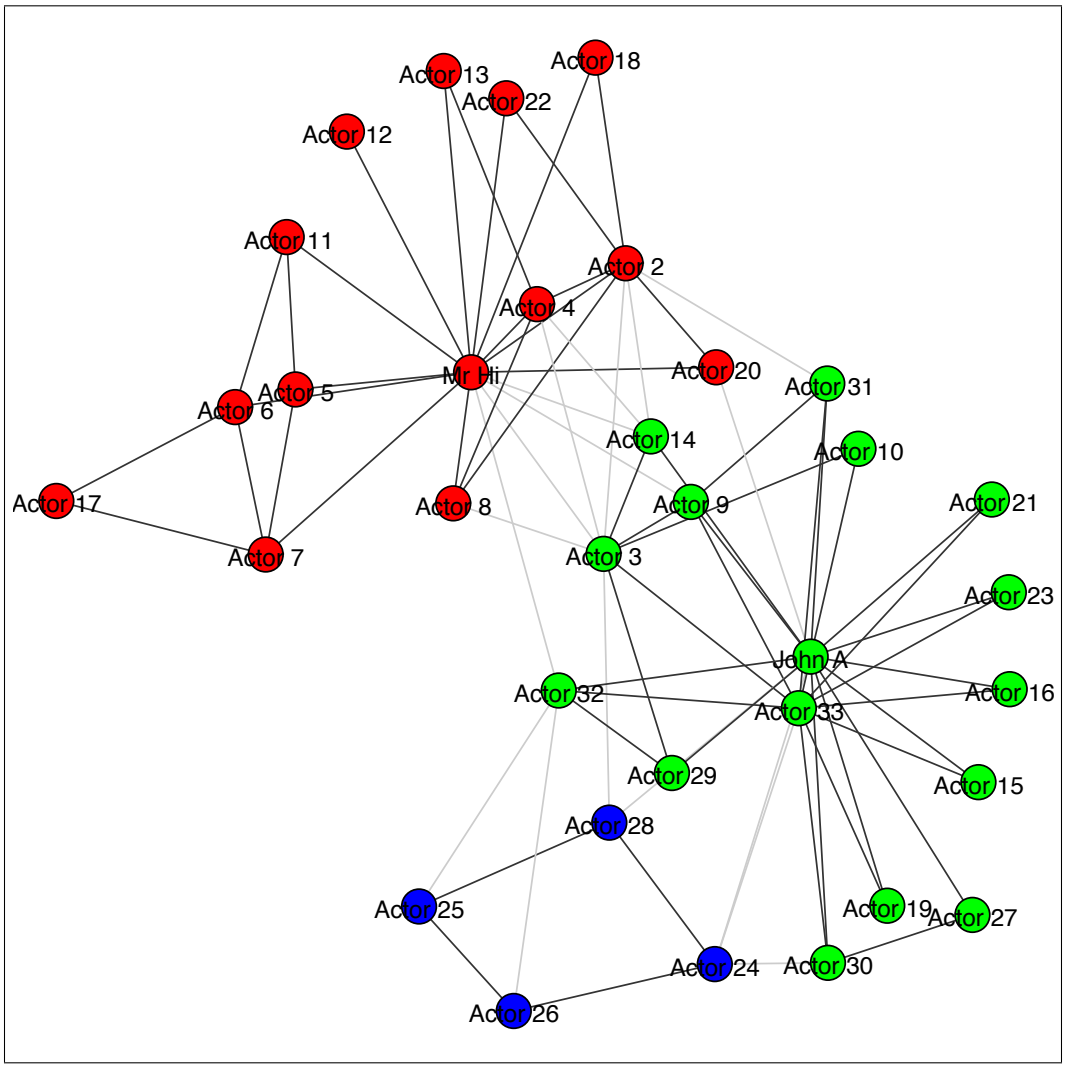

Figure 8: 3-Cluster Prediction

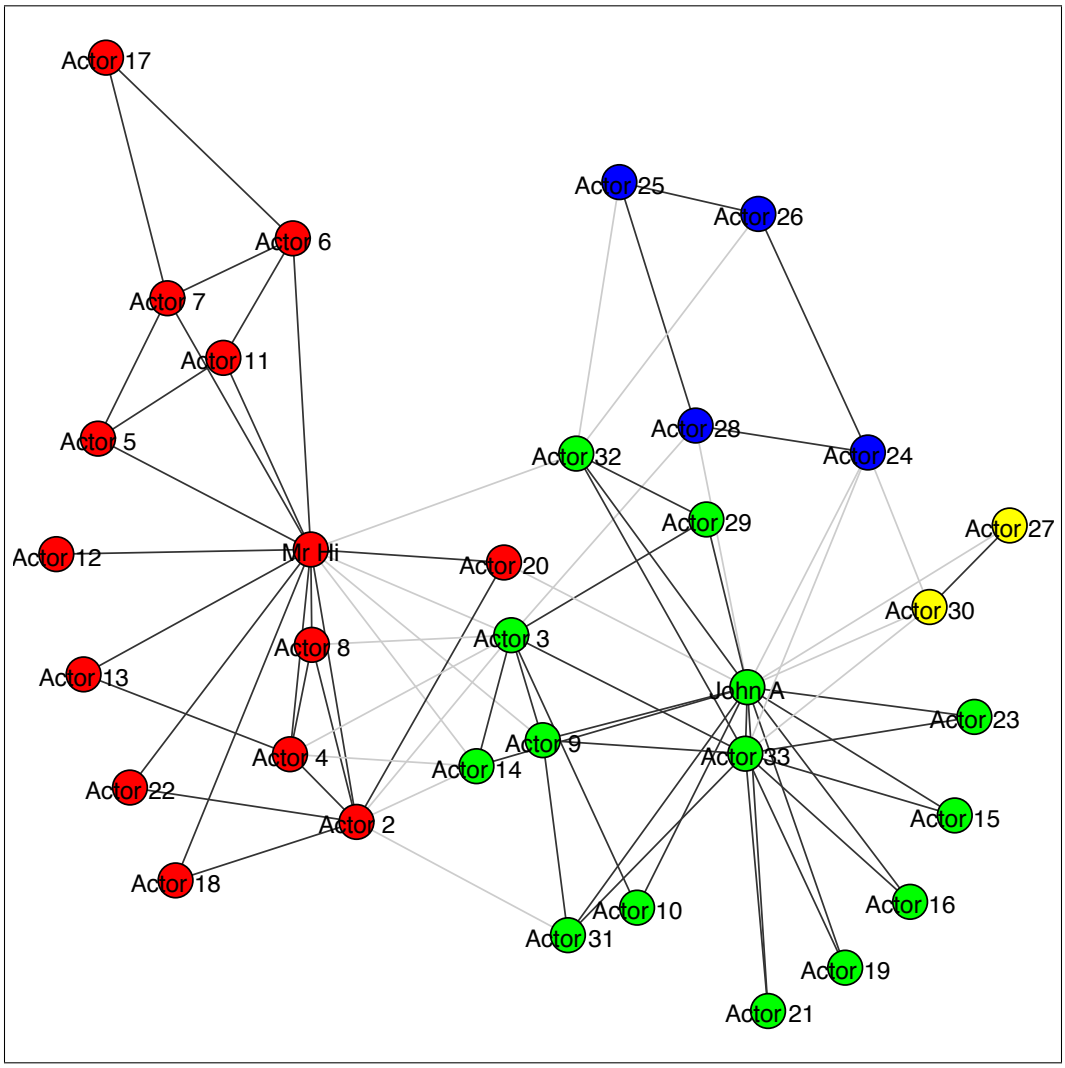

Figure 9: 4-Cluster Prediction

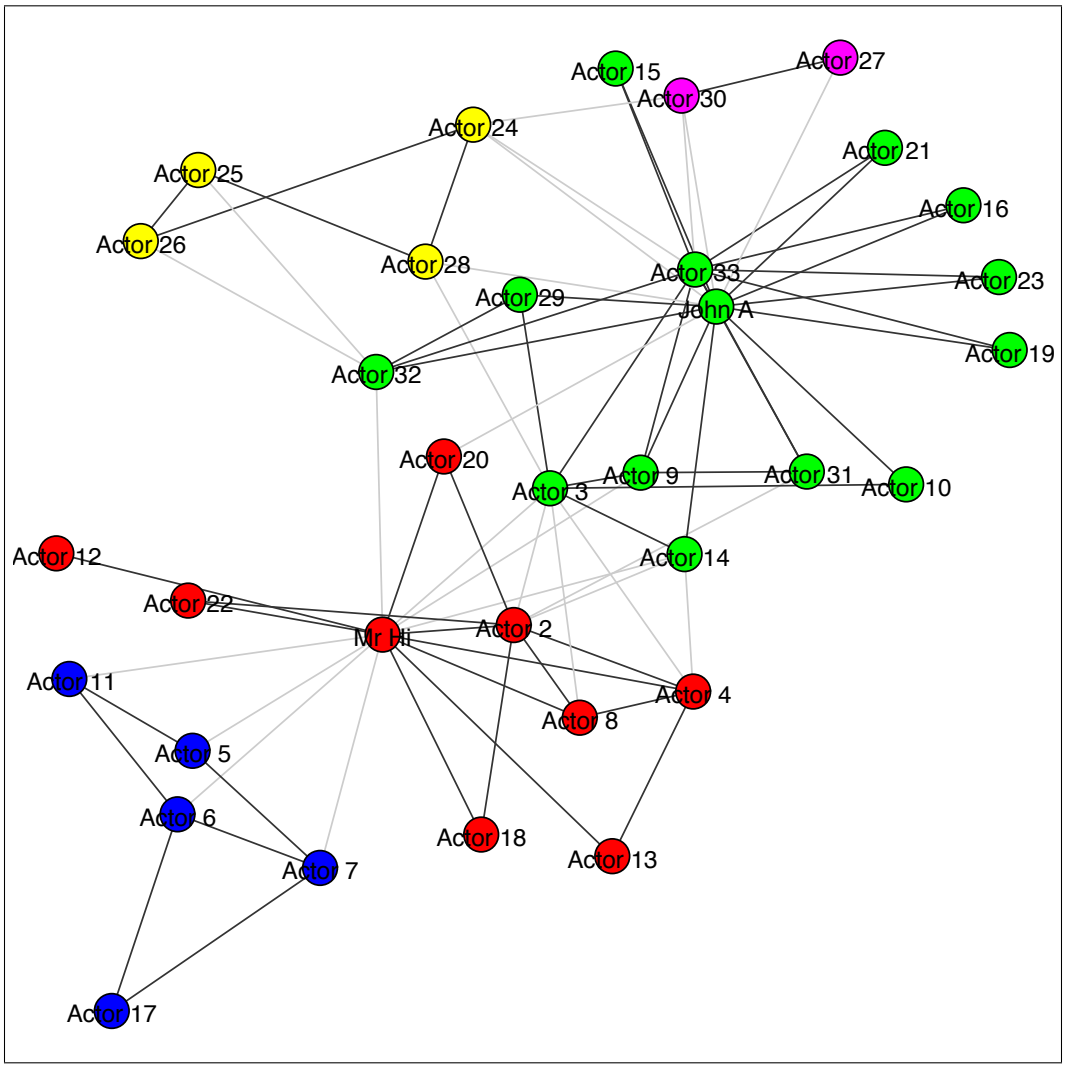

Figure 10: 5-Cluster Prediction

References

[1] Wayne W. Zachary. An information flow model for conflict and fission in small groups, 1977.

[2] Jr. L. R. Ford and D. R. Fulkerson. Flows in networks, 1962.

[3] The Python Software Foundation. Python pickle module. `https://docs.python.org/2/library/pickle.html`, February 2016.

[4] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in network. 69, 2004.

[5] M. E. J. Newman. Finding and evaluating community structure in networks using the eigenvectors of matrices. 74, 2006.