ASSIGNMENT 5

CS 432 Web Science

Mackenzie Kerchner

Mackenzie Kerchner

Contents

Mackenzie Kerchner

Problem 1

```
1.  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?

Clues:
1. Draw original Karate club graph (two connected components) after split
(Week 6 lecture, slide 98).
2. Run multiple iterations of graph partitioning algorithm (e.g., Girvan-
Newman Algorithm) on experimental Karate club graph until the graph splits
into two connected components.
3. Compare the connected components of the experimental graph (in 2.) with
the original connected components of the split Karate club graph (in 1.). Are
they similar?
```

Solution

To construct the graph the data is imported from networkx, which fortunately comes with a function karate_club_graph() that contains all the relevant information for this problem, including each member by number, the connections they have to other members, and group they went with after the split. A node color map is created to show member's final location in the actual split, and for each iteration a color map is made for the edges to show which connection is being cut. Because the girvan_newman() function returns a set of tuples instead of something that can be easily worked by networkx, the max_centrality function is used instead to find the strongest connection and cut it. I couldn't figure out how to stop iterating once two distinct communities are derived so I simply had it stop after 11 iterations, which produces the two groups. Comparing figures 2, the actual split from the slides provided by Professor Nwala, and 3, the constructed split, there are two errors in that persons 2 and 8 appear on the Officer side when they belong on the Mr. Hi side. Given that this graph is based solely on weighted connections between each member, it is surprising how accurate it is. Given some margin of error for sheer randomness, the model could be considered quite accurate for a live scenario involving a large number of people, and it could be said that it could have predicted the split with some degree of precision.

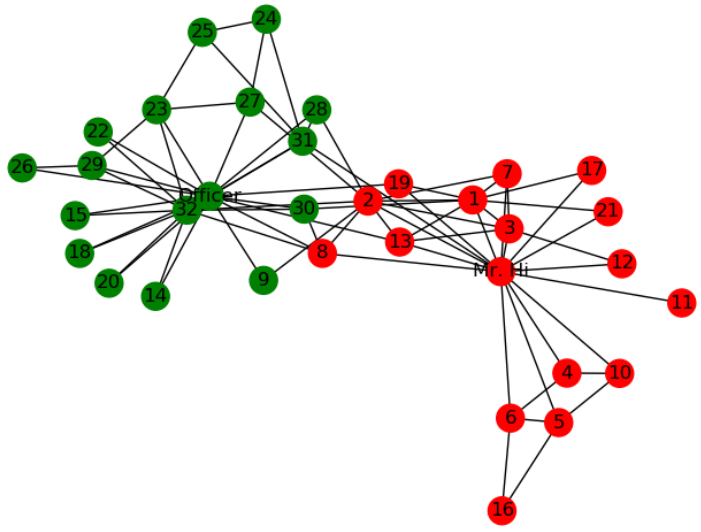

Figure 1: constructed graph of karate club pre-split

```
1 club: Mr. Hi , degrees: 9
2 club: Mr. Hi , degrees: 10
3 club: Mr. Hi , degrees: 6
4 club: Mr. Hi , degrees: 3
5 club: Mr. Hi , degrees: 4
6 club: Mr. Hi , degrees: 4
7 club: Mr. Hi , degrees: 4
8 club: Mr. Hi , degrees: 5
9 club: Officer , degrees: 2
10 club: Mr. Hi , degrees: 3
11 club: Mr. Hi , degrees: 1
12 club: Mr. Hi , degrees: 2
13 club: Mr. Hi , degrees: 5
14 club: Officer , degrees: 2
15 club: Officer , degrees: 2
16 club: Mr. Hi , degrees: 2
17 club: Mr. Hi , degrees: 2
18 club: Officer , degrees: 2
19 club: Mr. Hi , degrees: 3
20 club: Officer , degrees: 2
21 club: Mr. Hi , degrees: 2
22 club: Officer , degrees: 2
23 club: Officer , degrees: 5
24 club: Officer , degrees: 3
25 club: Officer , degrees: 3
26 club: Officer , degrees: 2
27 club: Officer , degrees: 4
28 club: Officer , degrees: 3
29 club: Officer , degrees: 4
30 club: Officer , degrees: 4
31 club: Officer , degrees: 6
32 club: Officer , degrees: 12
Officer club: Officer , degrees: 17
Mr. Hi club: Mr. Hi , degrees: 16
```

```
import networkx as nx
from networkx import edge_betweenness_centrality
import matplotlib.pyplot as plt

mapping = {0: 'Mr. Hi', 33: 'Officer'}  # gives mr Hi and officer explicit names in graph
G = nx.relabel_nodes(nx.karate_club_graph(), mapping)  # generates graph
color_map = []
for node in G:  # iterate through data and color each node for its club
    if G.node[node]['club'] == 'Mr. Hi':
        color_map.append('red')
    elif G.node[node]['club'] == 'Officer':
        color_map.append('green')
    else:
        print "error in determining club"

i = 0
while i < 11:
    edge_color_map = []  # declares edge color map, empty for each new iteration
    centrality = edge_betweenness_centrality(G)  # get list of edge strengths
    max_centrality = max(centrality, key=centrality.get)  # find strongest edge
    for edge in G.edges():  # color the edge that is about to be deleted
        if edge == max_centrality:
            edge_color_map.append('red')
        else:
            edge_color_map.append('black')
    nx.draw(G, node_color=color_map, edge_color=edge_color_map, with_labels=True)
    plt.show()  # draw graph with to-be removed edge in red

    G.remove_edge(max_centrality[0], max_centrality[1])  # remove max_centrality edge
    i += 1

print "final graph"
nx.draw(G, node_color=color_map, with_labels=True)
plt.show()
```

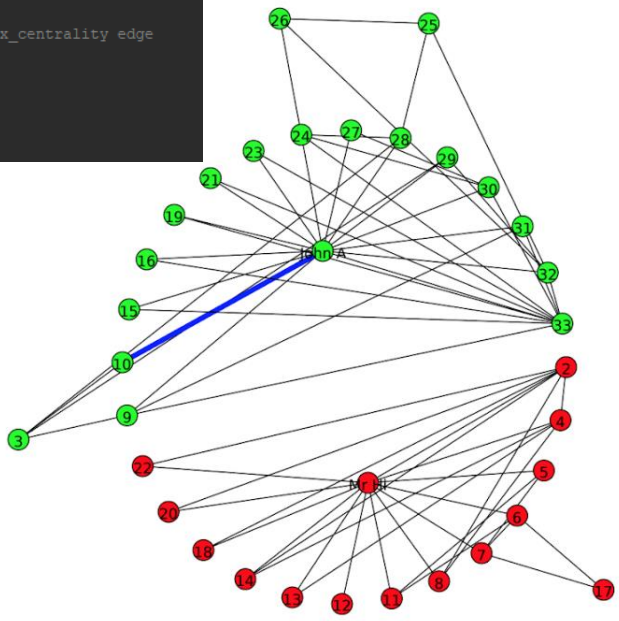

Figure 2: graph of actual split

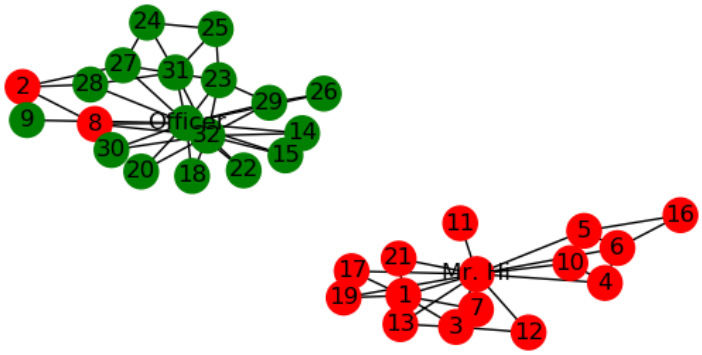

Figure 3: constructed graph of club post-split

Mackenzie Kerchner
ODU CS 432 Web science Spring 2019 Sunday, March 24, 2019

Problem 3

```
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?
```

Solution

The following figures are how the graph splits into 3,4, and 5 groups on the 15th, 19th, and 25th iteration respectively.

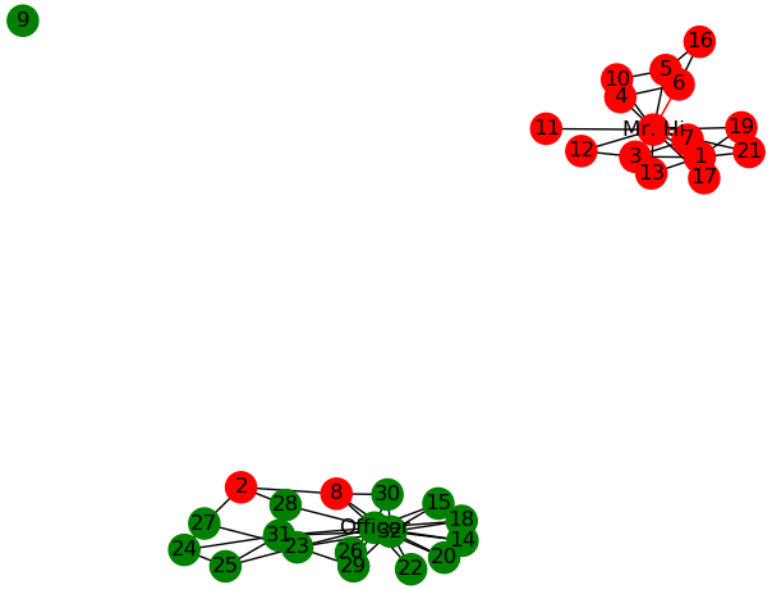

Figure 4:3 groups

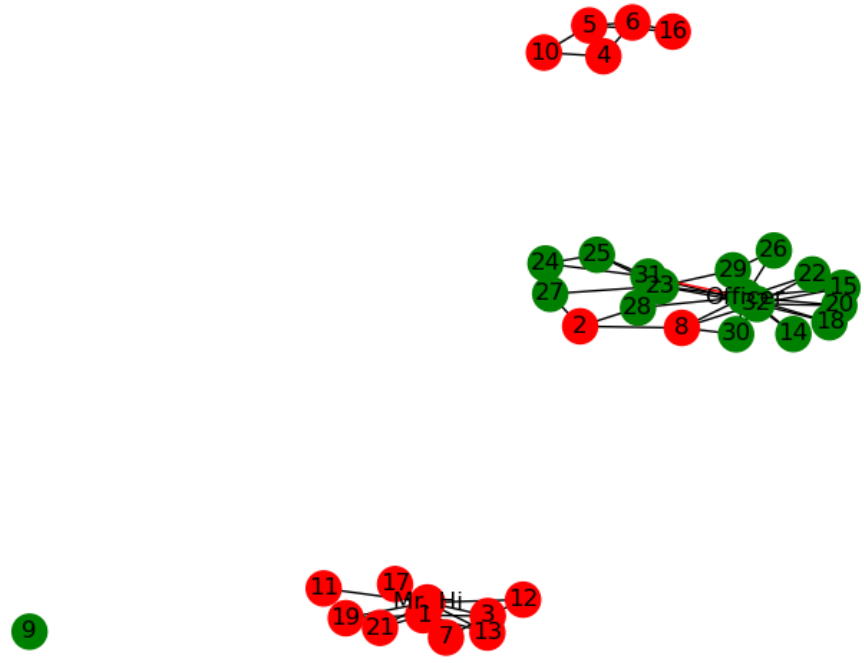

Figure 5: 4 groups

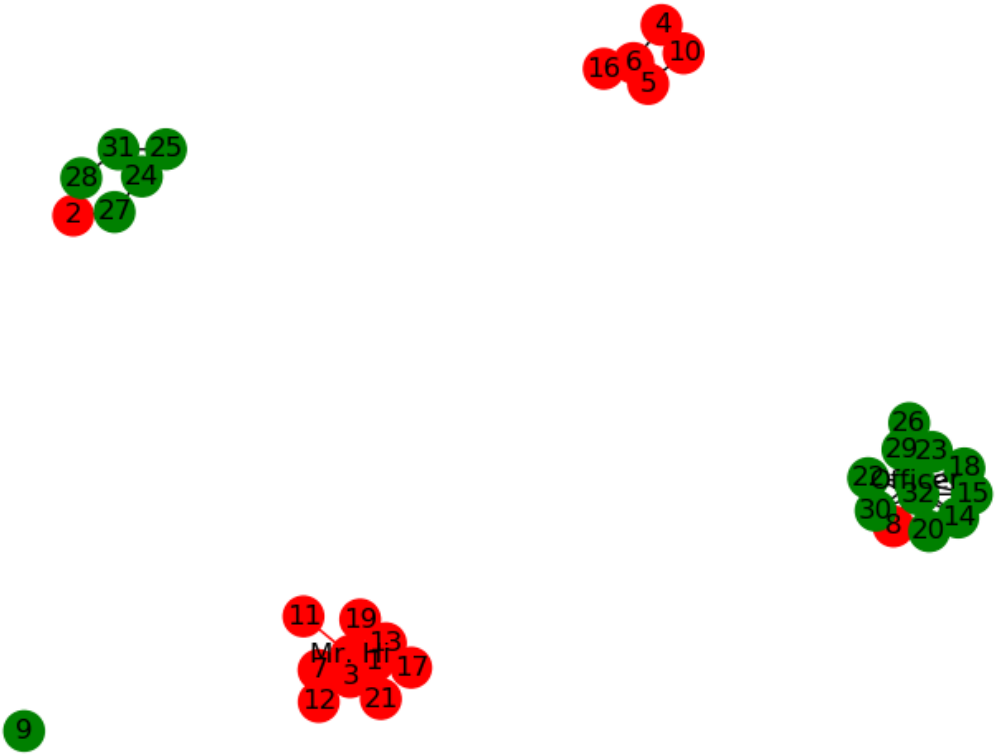

Figure 6: 5 groups