CS 535: Assignment #5

Due on 11:59pm March 7, 2018

Alexander C. Nwala 16:20

David Sinclair

March 2, 2018

Contents

1 Problem 1

1.1 Question 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?

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

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?

Useful sources include:

* Original paper

http://aris.ss.uci.edu/ lin/76.pdf

* Week 6 Slides:

https://docs.google.com/presentation/d/1ihf6N8bHgzM5VLAyHkmF_i5JGUBVpCSdsvYpk8XgHwo/edit?usp=sharing
* Slides

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

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

* Code and data

https://networkx.github.io/documentation/networkx1.10/reference/generated/networkx.generators.social.karate_club_graph.htm

https://networkx.github.io/documentation/networkx1.9/examples/graph/karate_club.html

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

http://stackoverflow.com/questions/9471906/whatarethedifferencesbetweencommunitydetectionalgorithmsinigraph/9478989#94

http://stackoverflow.com/questions/5822265/arethereimplementationsofalgorithmsforcommunitydetectioningraphs

http://konect.unikoblenz.de/networks/ucidatazachary

http://vlado.fmf.unilj.si/pub/networks/data/ucinet/ucidata.htm#zachary

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

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

1.2 Answer 1

For this question I used the following program that I captured from two major sources.
@onlineYouTube, author = IIT ROPAR, title = Lecture 36 Community Detection Using Girvan Newman Algorithm, year = Oct 8, 2017, url = https://www.youtube.com/watch?v=c27VEbGdgCg,
@onlinePythonCode, author = millionsmile/gist:3676569, title = Karate club community detection by Girvan-Newman algorithm: gistfile1.py , year = 2012, url = https://gist.github.com/millionsmile/3676569,

After that I created the following code. I wanted to first create the original graph.

```
G = nx.karate_club_graph()
count = 0
pos = nx.spring_layout(G)
output = "karate%(id)02d.png" % {"id": count}
print('The file # is ', count,file=open('edgeremove.txt','a'))
pl.figure()
nx.draw_networkx(G, pos)
```

This created figure 1 which shows the original club.

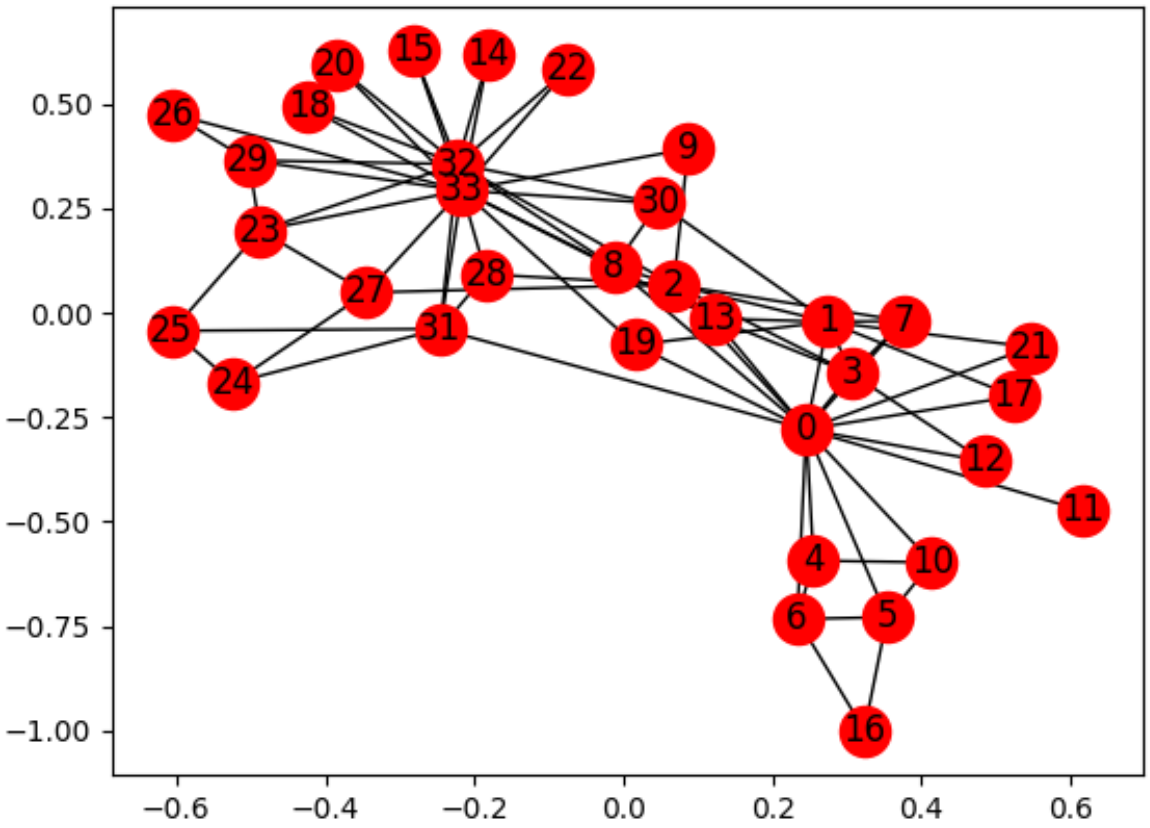


Figure 1: figure 0 Original Club

Know that I now I had my data correct. I began using the Girvan-Newman algorithm to remove the highest edge of betweenness. This is the code from the program that determined the highest edge of betweenness.

```
def edge_to_remove(G):
        dict1 = nx.edge_betweenness_centrality(G)
        list_of_tuples = dict1.items()
        list_of_tuples=sorted(list_of_tuples, key=lambda x:x[1], reverse=True)
        for x in list_of_tuples:
                return x[0]
```

Of note after getting the edge of betweenness values, it is important to sort the data from highest to lowest. If that is not done then the data will not randomly select what to remove based on the values inserted from the beginning to the end not taking into account weight or degree.

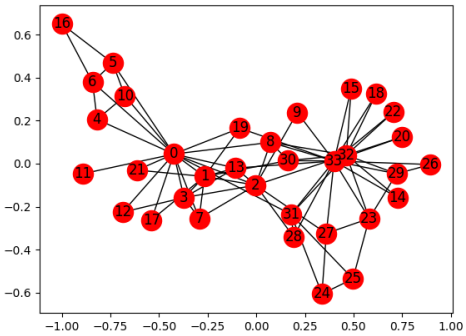
Then comes the data of what points to remove and what the graph will look like.

The rest of the program did this.

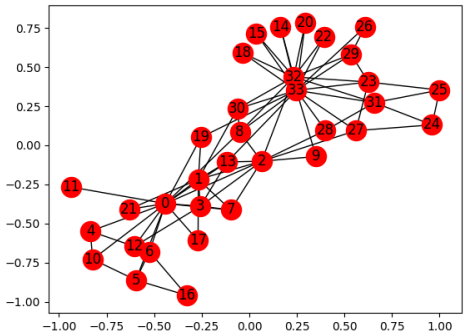
```
def girvan(G):
        c = nx.connected_component_subgraphs(G)
        l = len(list(c))
        pos = nx.spring_layout(G)
        count = 1
        print(count," The number connected compents are ",l)
        output = "karate%(id)02d.png" % {"id": count}
        print('The file # is ', count,' Edge that was removed: ',edge_to_remove(G),
            file=open('edgeremove.txt','a'))
        pl.figure()
        nx.draw_networkx(G, pos)
        pl.savefig(output)
        pl.close()
        while(l == 1):
                G.remove_edge(*edge_to_remove(G)) #((a,b)) --> (a,b)
                c = nx.connected_component_subgraphs(G)
                l = len(list(c))
                pos=nx.spring_layout(G)
                count +=1
                print(count, " The number connected compents are ",l)
                print('The file # is ', count,' Edge that was removed: ',
                    edge_to_remove(G),file=open('edgeremove.txt','a'))
                output = "karate%(id)02d.png" % {"id": count}
                pl.figure()
                nx.draw_networkx(G, pos)
                pl.savefig(output)
                pl.close()

        return c
```

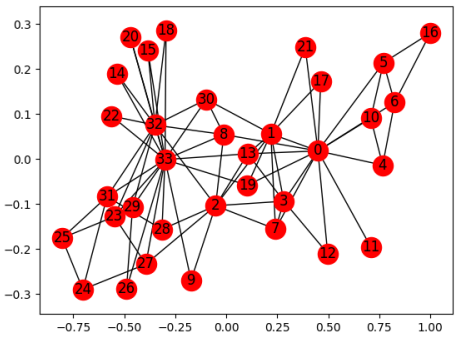
This created figure 2 which shows the progression by using the Girvan-Newman algorithm.



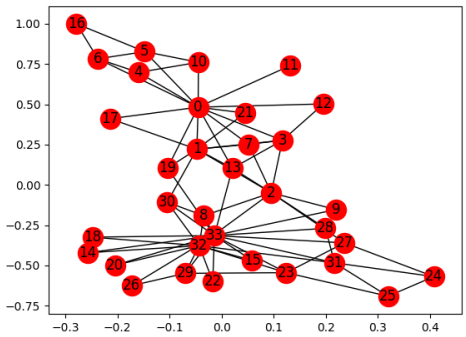
(a) fig 1 had edge (0, 31) removed



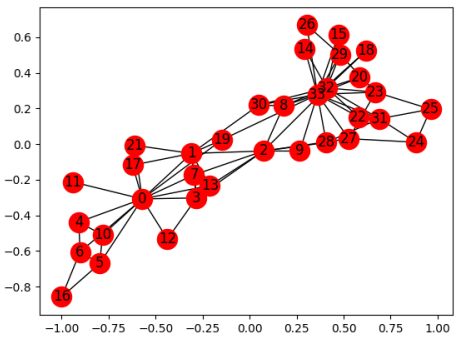
(b) fig 2 had edge (0, 2) removed



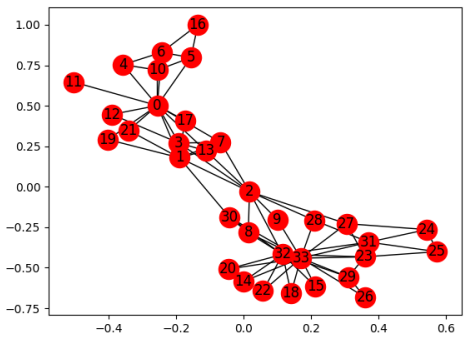
(c) fig 3 had edge (0, 8) removed



(d) fig 4 had edge (13, 33) removed



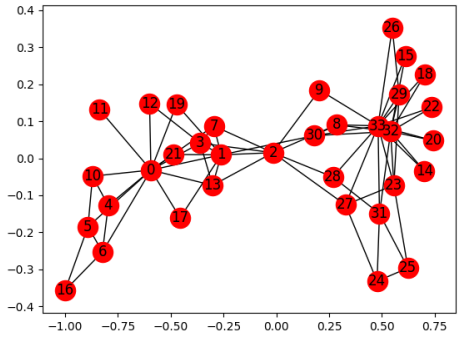
(e) fig 5 had edge (19, 33) removed



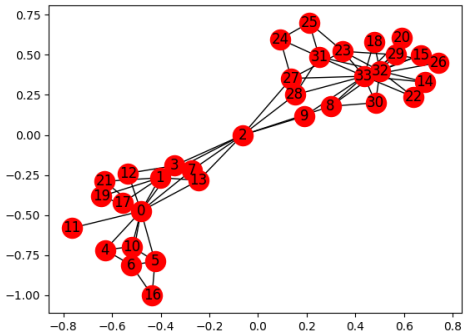
(f) fig 6 had edge (2, 32) removed

Figure 2: The karate club from 1977 moving to split.

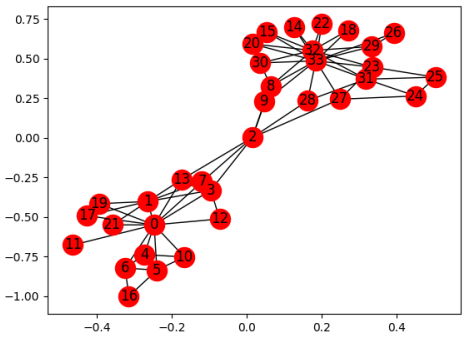
Figures 3 continued up to 12 files.



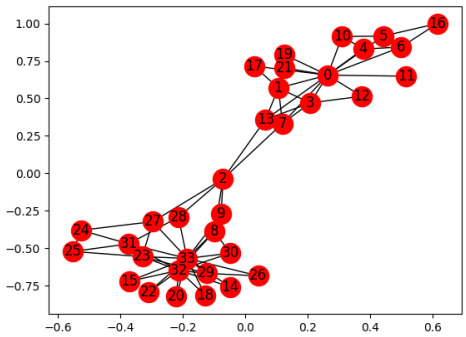
(a) fig 7 had edge (1, 30) removed



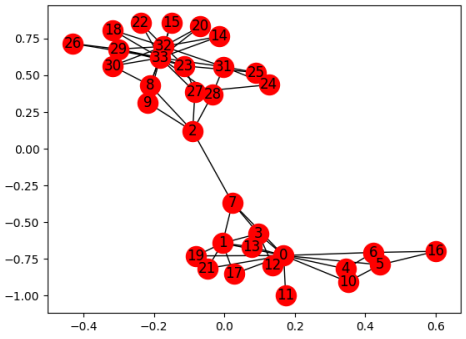
(b) fig 8 had edge (1, 2) removed



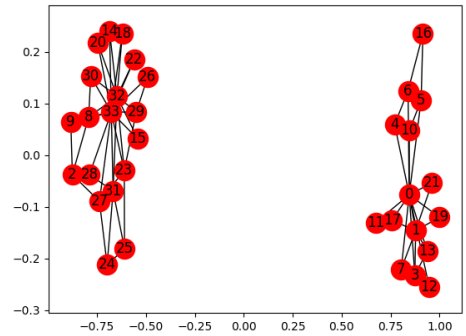
(c) fig 9 had edge (2, 3) removed



(d) fig 10 had edge (2, 13) removed



(e) fig 11 had edge (2, 7) removed

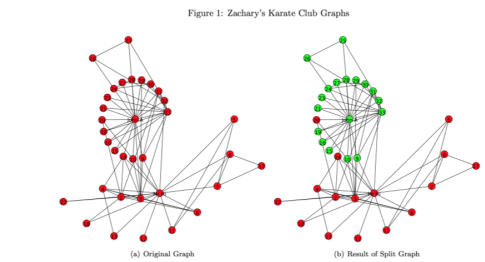


(f) fig 12 had edge (9, 33) removed

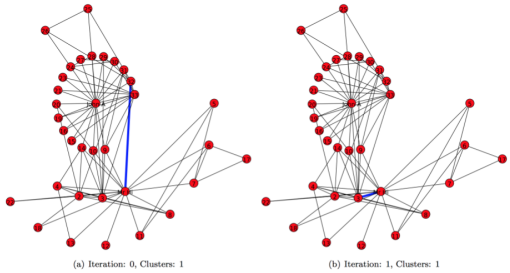
Figure 3: The karate club from 1977 moving to split (cont).

The above data will be called Edge Betweenness when I start comparing between the items I did and what was presented in class.

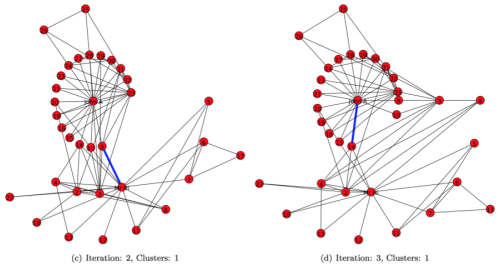
This is all the information that I collected. Now I will show you all the information from the power point slides that were presented in class. This is listed in figures slides 4



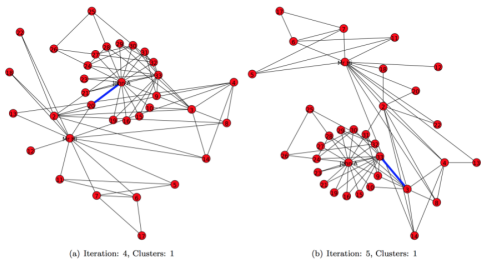
(a) slide 1 Original information slide 98



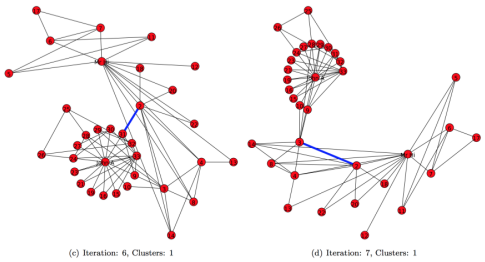
(b) slide 2 Original information slide 99



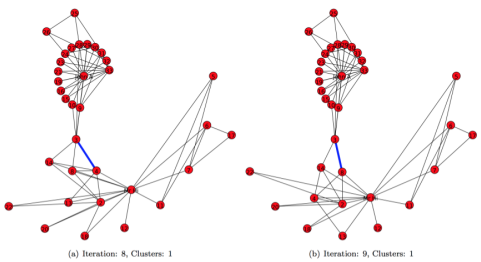
(c) slide 3 Original information slide 100



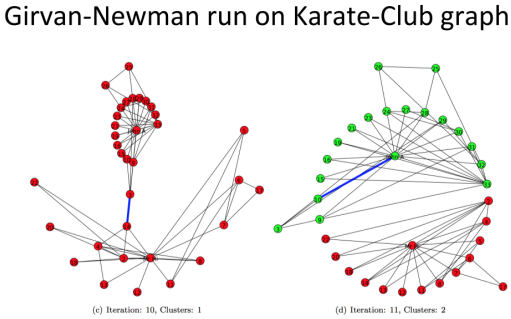
(d) slide 4 Original information slide 101



(e) slide 5 Original information slide 102



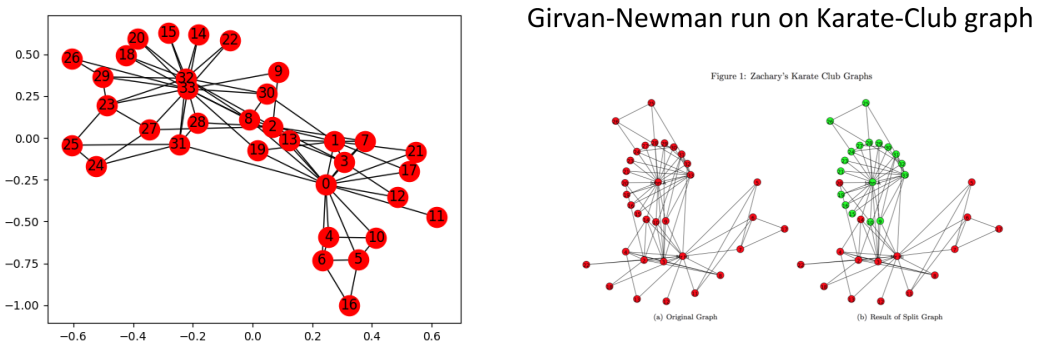
(f) slide 6 Original information slide 103



(g) slide 7 Original information slide 104

Figure 4: The karate club from 1977 class lecture information.

Comparing the first graphs they do not look quite the same but the connections are the same. This can be seen in figure 5



(a) Edge Betweenness original data (b) Slide information from class

Figure 5: The karate club from 1977 class information.

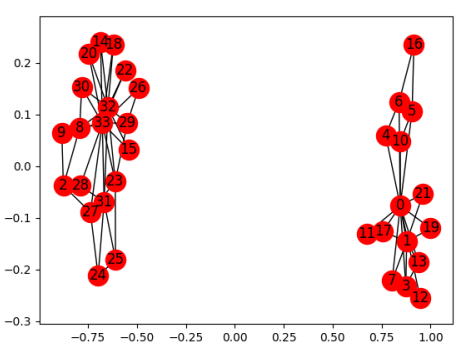
Now let us compare what edges were removed using the Girvan-Newman algorithm and the edge betweenness algorithm I used with pyton coding.

Table 1. Iterations compare edge removals

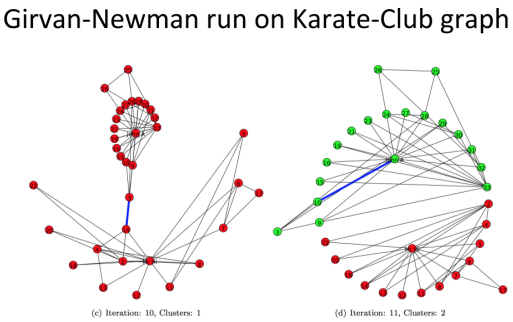
Iteration Number	Orignal Edge Removed	Edge Betweenness Removed
1	0 32	0 31
2	3 1	0 2
3	9 1	0 8
4	0 14	13 33
5	0 20	19 33
6	33 3	2 32
7	2 31	1 30
8	2 3	1 2
9	3 4	2 3
10	3 8	2 13
11	3 14	2 7
12		9 33

Since there is a chance that I may of used different numbers to represent people then what was used in the class slides. I decide to count the number of nodes between the two split apart groups. I also decide to compare the pictures of both split apart groups. Figure 6 shows the split of both my python edge betweenness and the class slide information.

In the Edge Betweenness split one group had 15 the other had 19. In the Class slide one group had 15 and the other had 19. So if the numbering is off comparing the date the number split of the nodes is correct.



(a) Edge Betweenness split apart



(b) Class slide information

Figure 6: The karate club from 1977 class information.

2 Problem 2

2.1 Question 2

(extra credit, 10 points)

2. Use D3.js's force-directed graph layout to draw the Karate Club Graph before split. Color the nodes according to the factions they belong to (John A or Mr. Hi). After a button is clicked, split the graph based on the original graph split. Include a link to the HTML/JavaScript files in your report and all necessary screenshots.

See: https://bl.ocks.org/mbostock/4062045

https://d3js.org/

2.2 Answer 2

3 Problem 3

3.1 Question 3

(extra credit, 3 points)

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?

3.2 Answer 3

The class if using the Edge Betweenness split If the class was to split into 3 groups using the Edge Betweenness split the figure 7. This took 15 iterations to great this group.

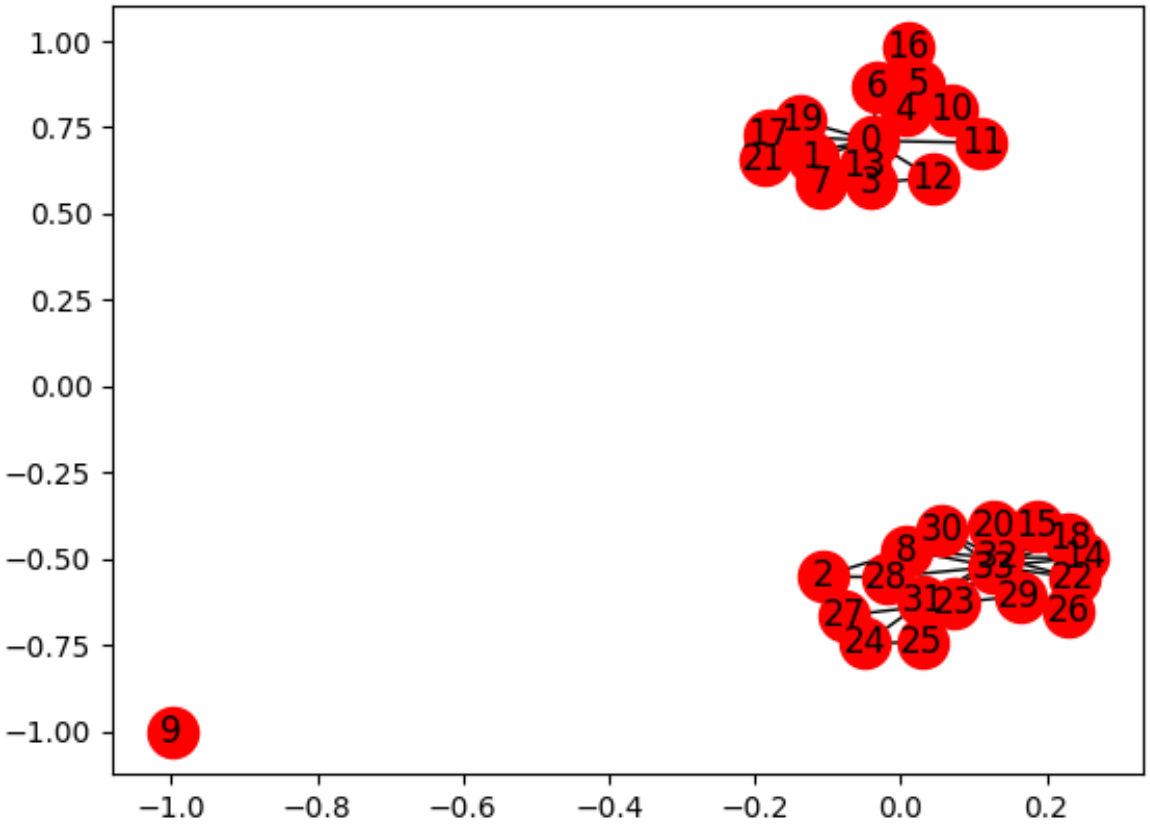


Figure 7: Karate club in 3 groups

The class if using the Edge Betweenness split If the class was to split into 4 groups using the Edge Betweenness split the figure 8. This took 19 iterations to great this group.

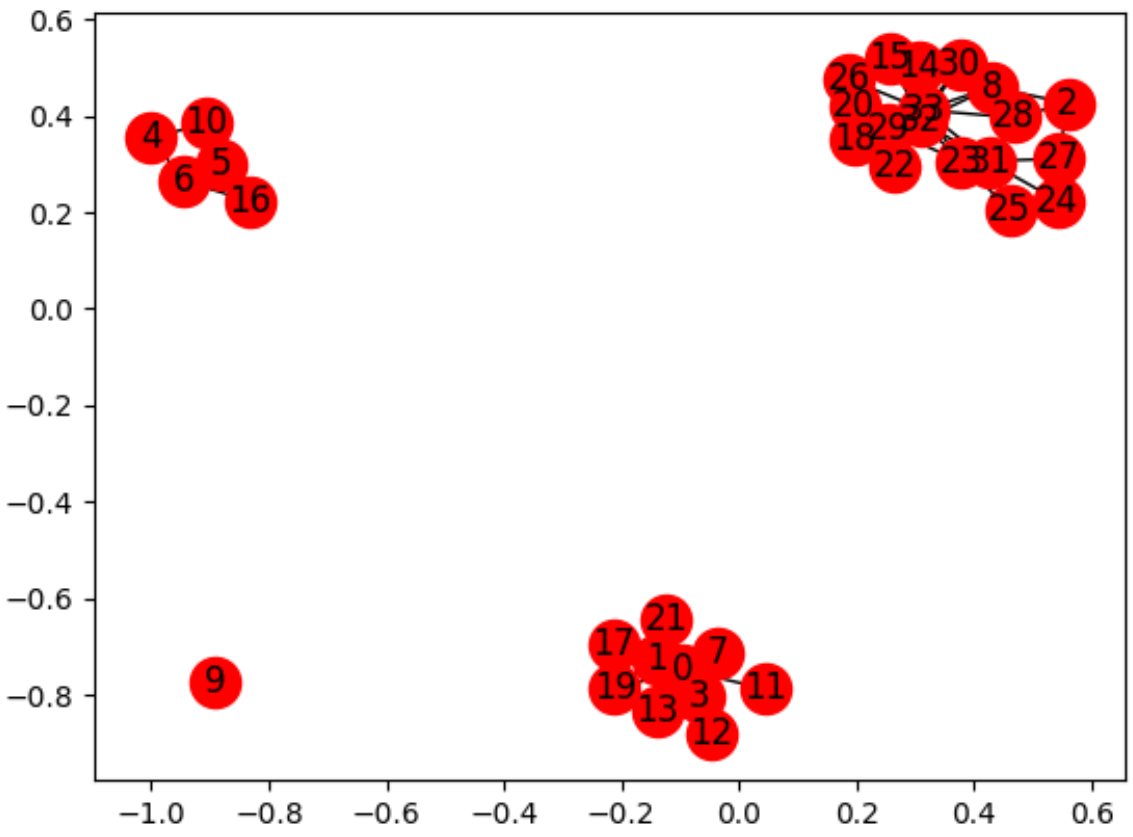


Figure 8: Karate club into 4 groups

The class if using the Edge Betweenness split If the class was to split into 5 groups using the Edge Betweenness split the figure 9. This took 25 iterations to great this group.

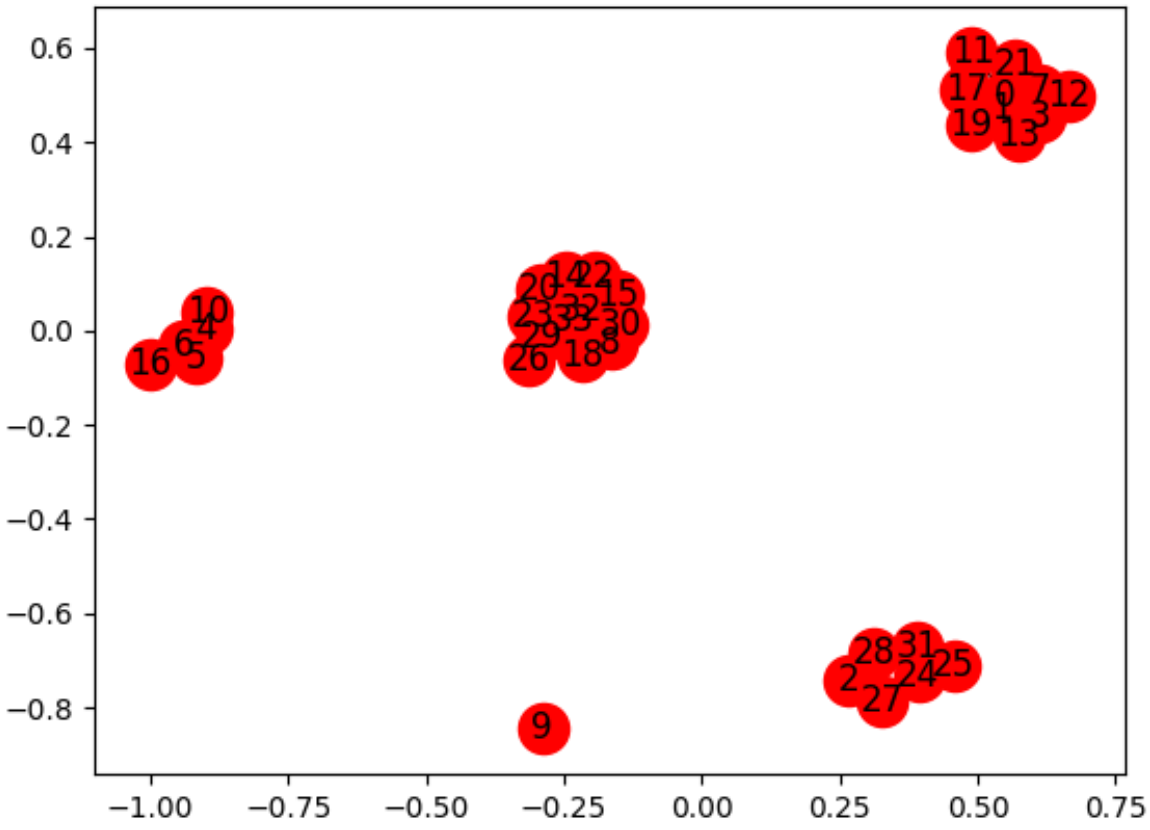


Figure 9: Karate club into 5 groups