Web Science: Assignment #5

Alexander Nwala

Puneeth Bikkasandra

Sunday, March 11, 2018

Contents

Problem 1 **3**

Problem 2 **12**

Problem 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. To Draw original Karate club graph (two connected components) after split (Week 6 lecture, slide 98).
2. To Run multiple iterations of graph partioning algorithm (e.g., Girvan-Newman Algorithm) on experimental Karate club graph until the graph splits into two connected components.
3. To 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:

1. **Original paper**

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

2. **Week 6 Slides:**

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

3. **Slides:**

   http://www-personal.umich.edu/ ladamic/courses/networks/si614w06/ppt/lecture18.ppt
   http://clair.si.umich.edu/si767/papers/Week03/Community/CommunityDetection.pptx

4. **Code and Data:**

   https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.generators.social.karate_club_graph.html

   https://networkx.github.io/documentation/networkx-1.9/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

Problem 1 continued on next page. . .

SOLUTION :
I have followed the pages and slides presented as part of the problem.

1. Install "Netwokx" library to generate the **karateClub** graph as below :

   ```
   pip install networkx
   ```

2. Import the **networkx** library and generate the **karateClub** graph

   ```
   import networkx as nx
   graph = nx.karate_club_graph()
   ```

3. Implement the **Girvan-Newman** algorithm as

   (a) while number of connected subgraphs $<$ specified number of clusters and the number of edges in graph $>$ 0:

   (b) calculate edge betweenness for every edge in the graph

   (c) remove edge(s) with highest betweenness

   (d) recalculate connected components

Listing 1: karateClubGraph.py

```

import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

graph = nx.karate_club_graph()
edgeCount = graph.number_of_edges()
maximumClusterThreshold = 2
count = 0
clusterCount = nx.algorithms.number_connected_components(graph)

# Drawing the initial karateclub graph
nx.draw(graph, with_labels=True)
plt.show()

while (edgeCount > 0 and clusterCount < maximumClusterThreshold):
   betweennessDict = nx.algorithms.betweenness.edge_betweenness(graph)
   maximumBetweennessValue = np.max(list(betweennessDict.values()))
   edgeValue = ’’

   for edge in betweennessDict:
      if betweennessDict[edge] == maximumBetweennessValue:
         edgeValue = edge

   print (’Removing Edges...’,str(edgeValue[0])+"--->"+str(edgeValue[1]))
   graph.remove_edge(edgeValue[0], edgeValue[1])

   # Drawing the sequential karateclub graph
   nx.draw(graph, with_labels=True)
   plt.show()
   count = count + 1
```

```
    clusterCount = nx.algorithms.number_connected_components(graph)

print('Total flows to separation:',count)
```

The above code, will separate the karateClub graph in to two components in a total of **11** iterations.

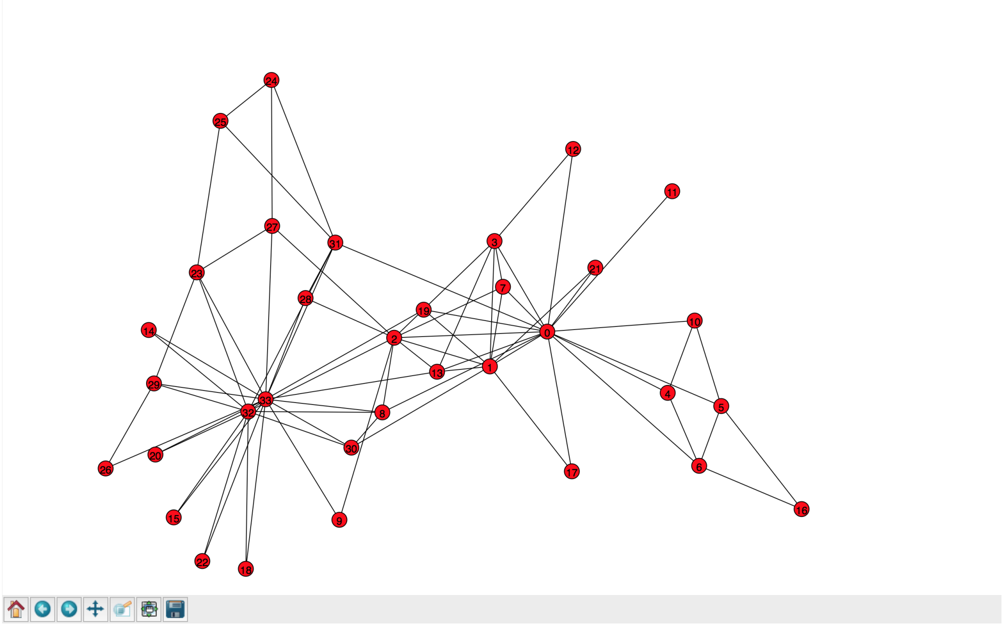

Figure 1: Original karate Club Graph

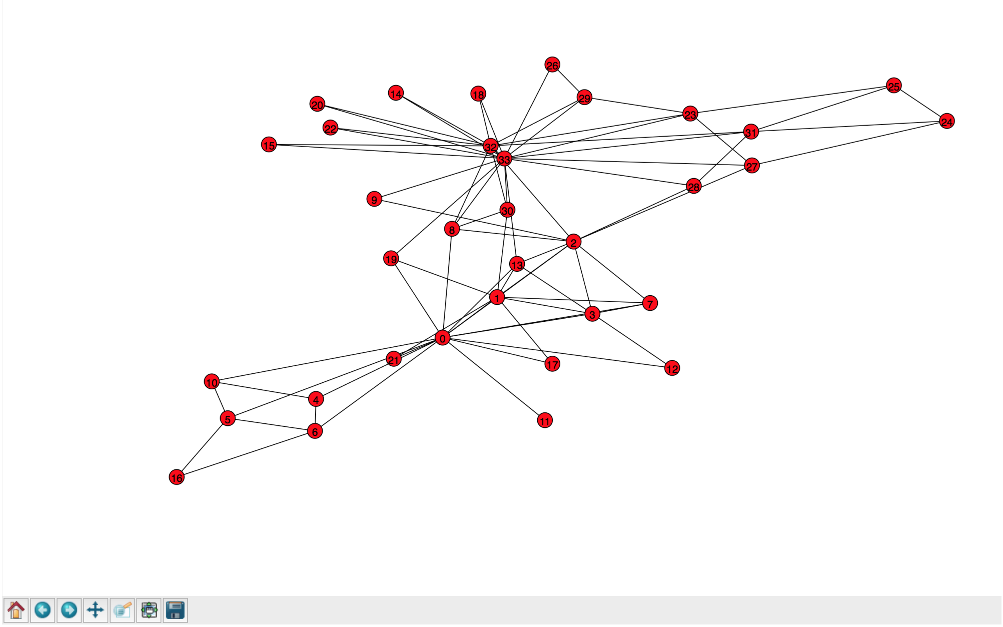

Figure 2: Iteration 1

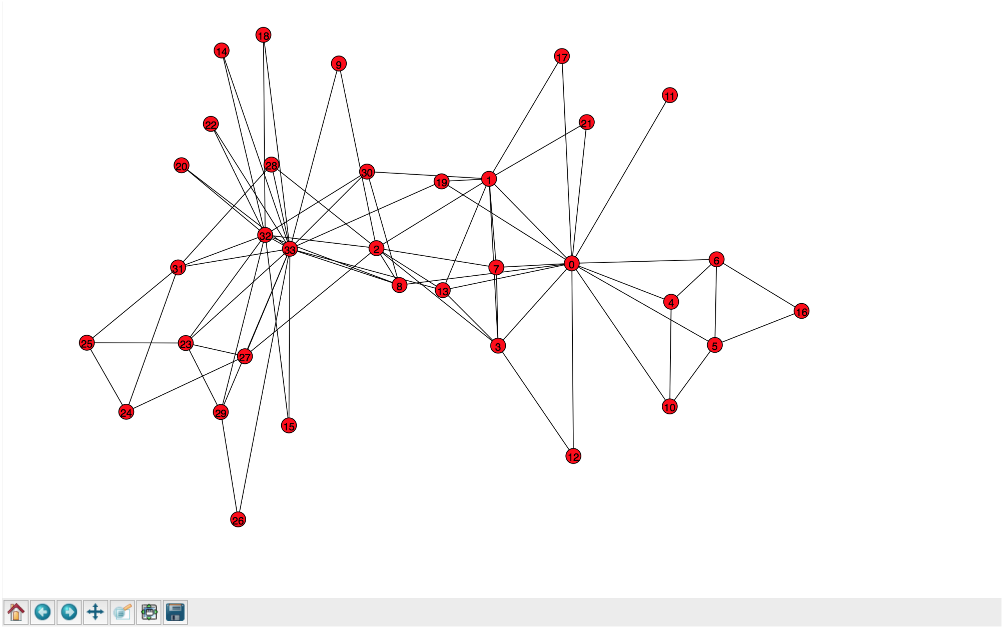

Figure 3: Iteration 2

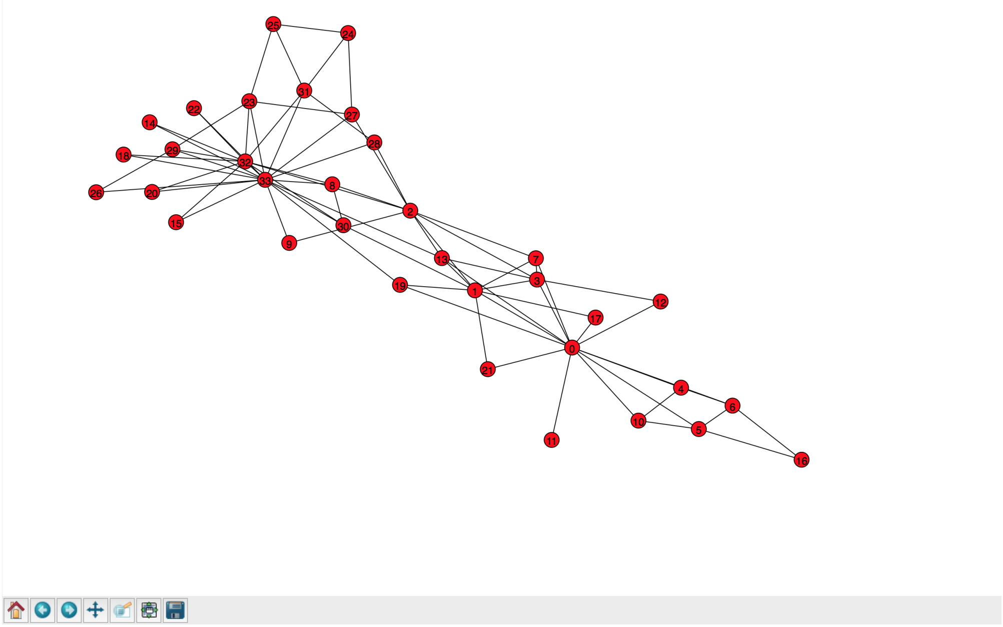

Figure 4: Iteration 3

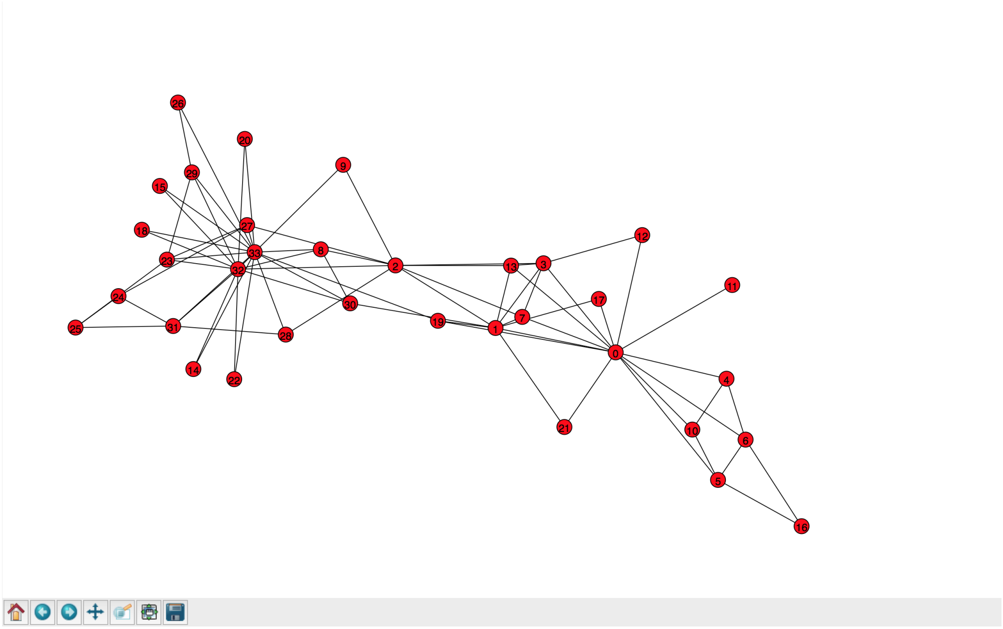

Figure 5: Iteration 4

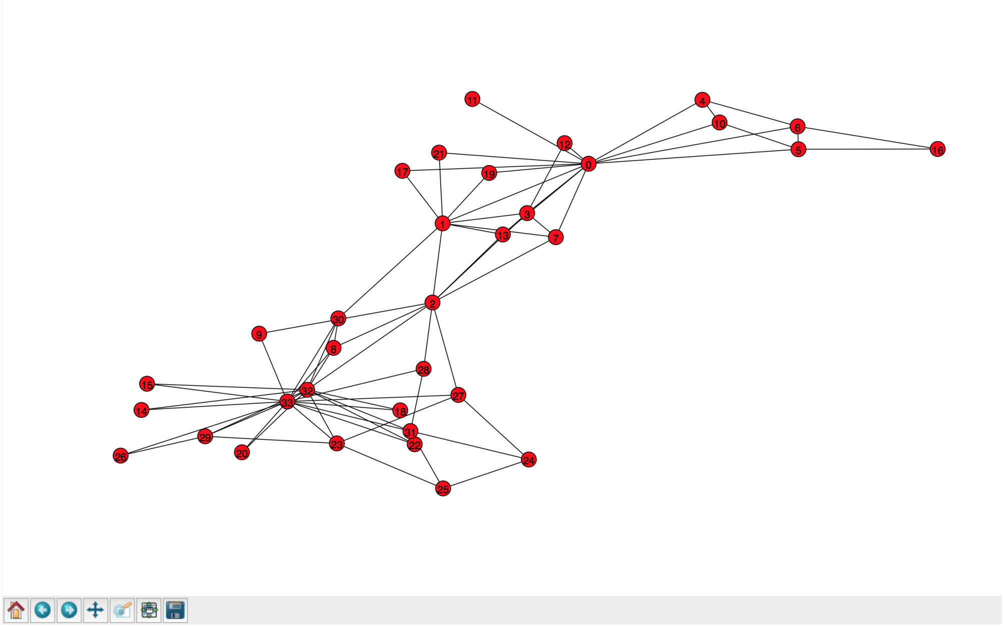

Figure 6: Iteration 5

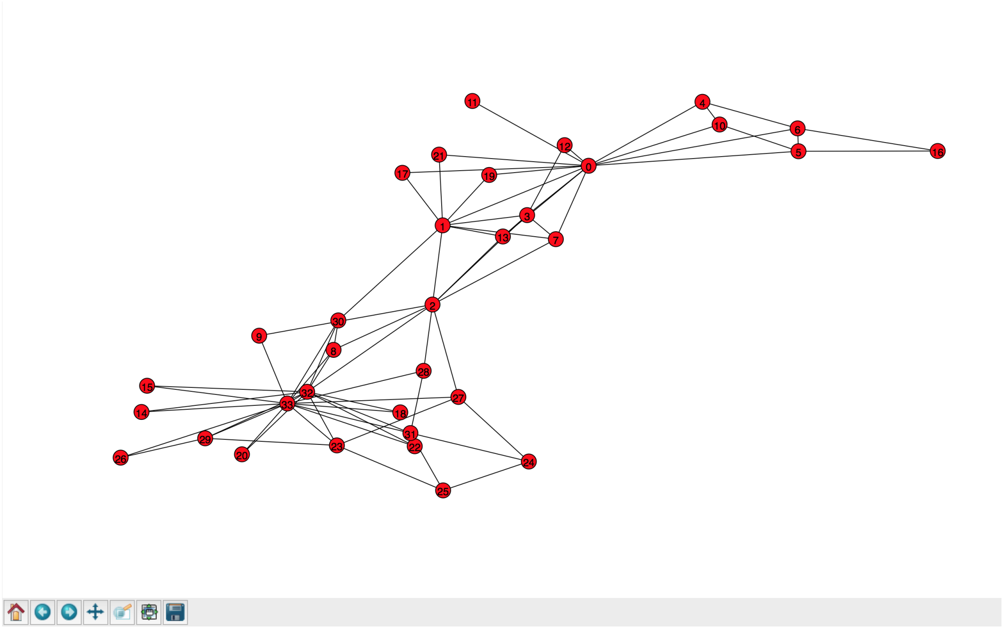

Figure 7: Iteration 5

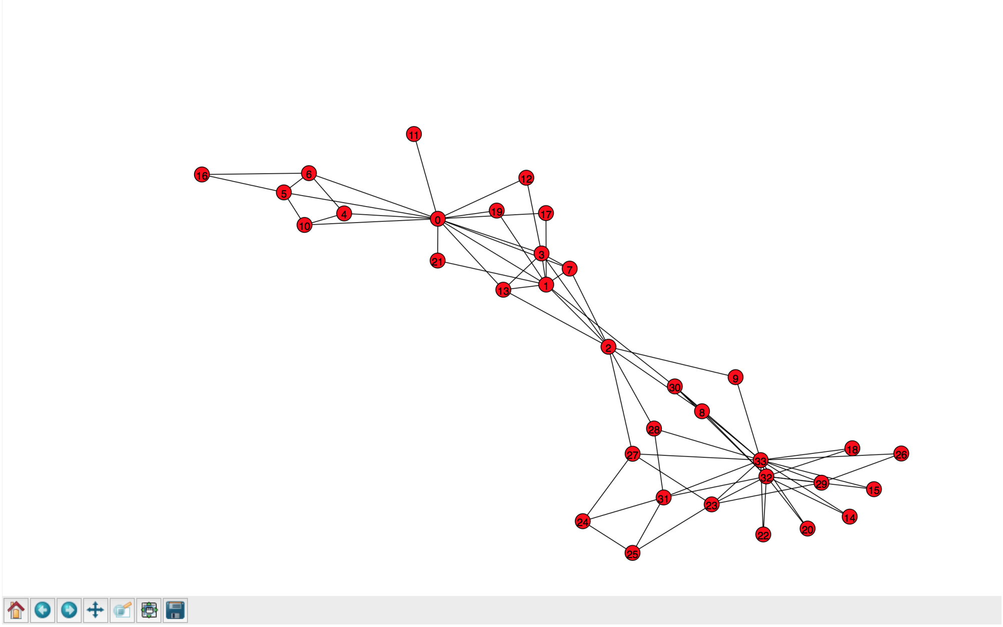

Figure 8: Iteration 6

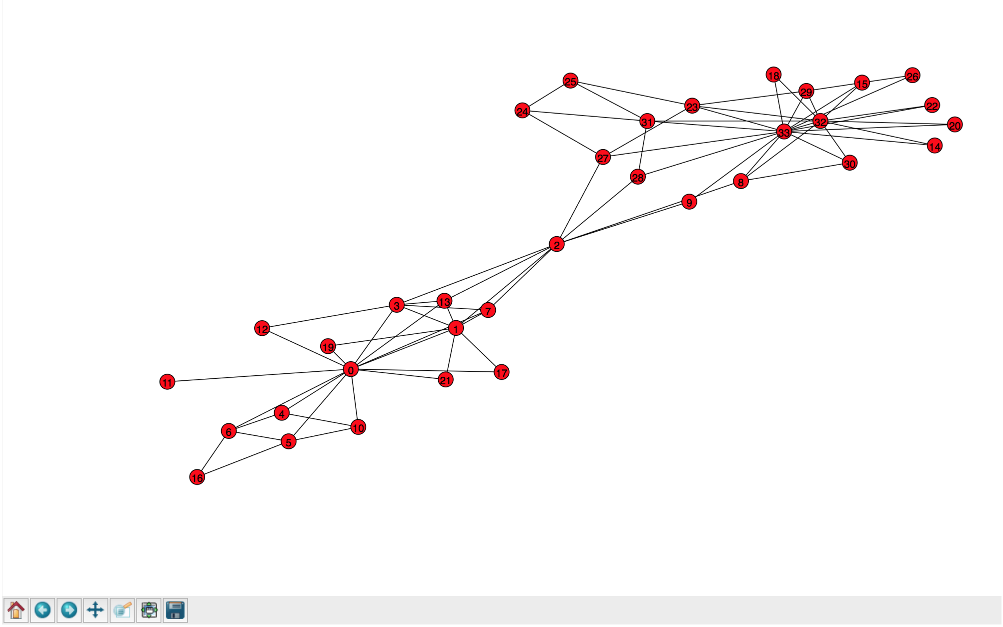

Figure 9: Iteration 7

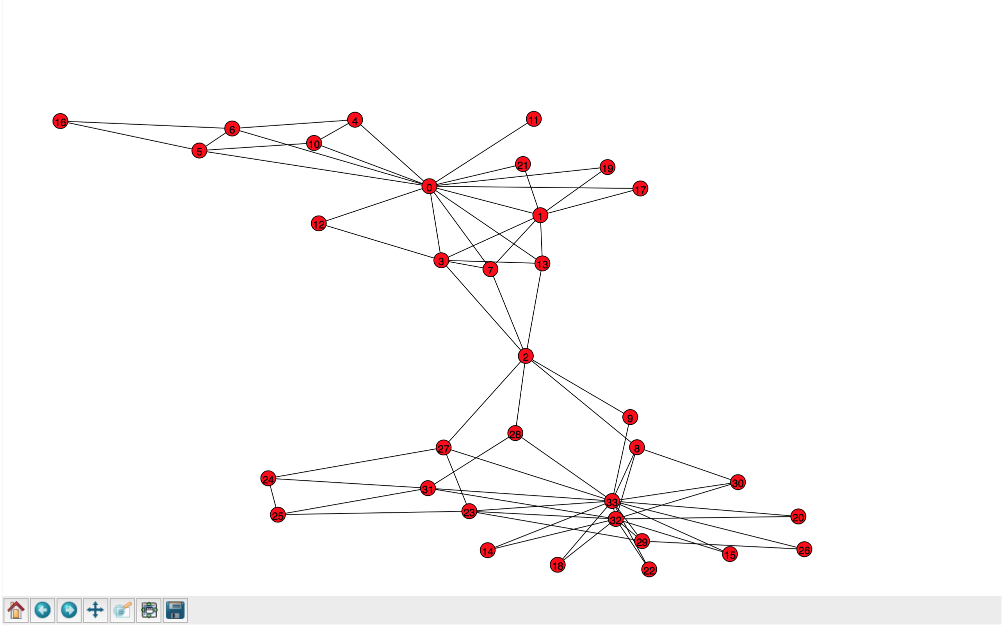

Figure 10: Iteration 8

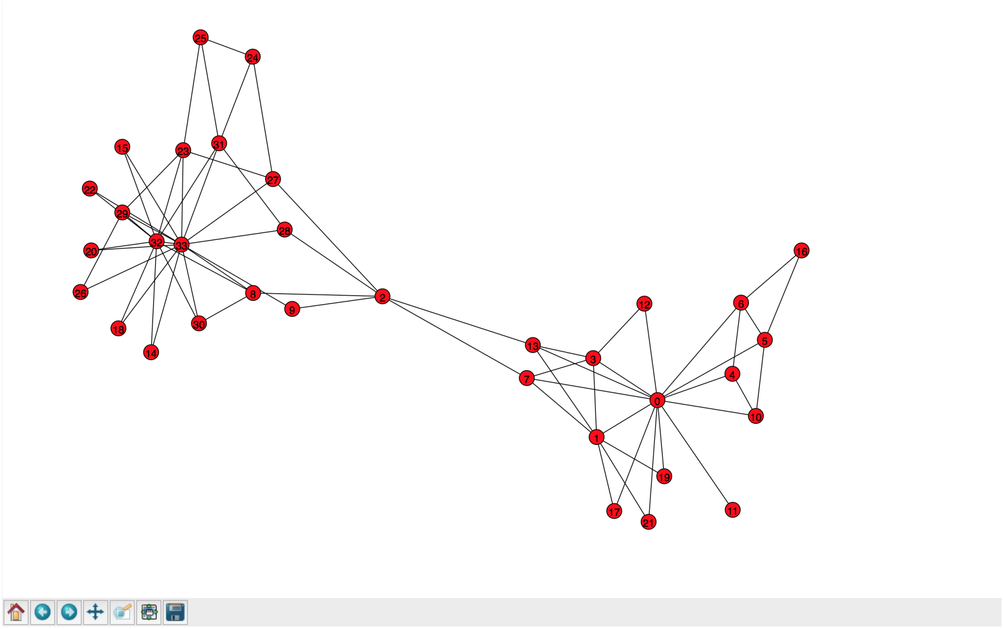

Figure 11: Iteration 9

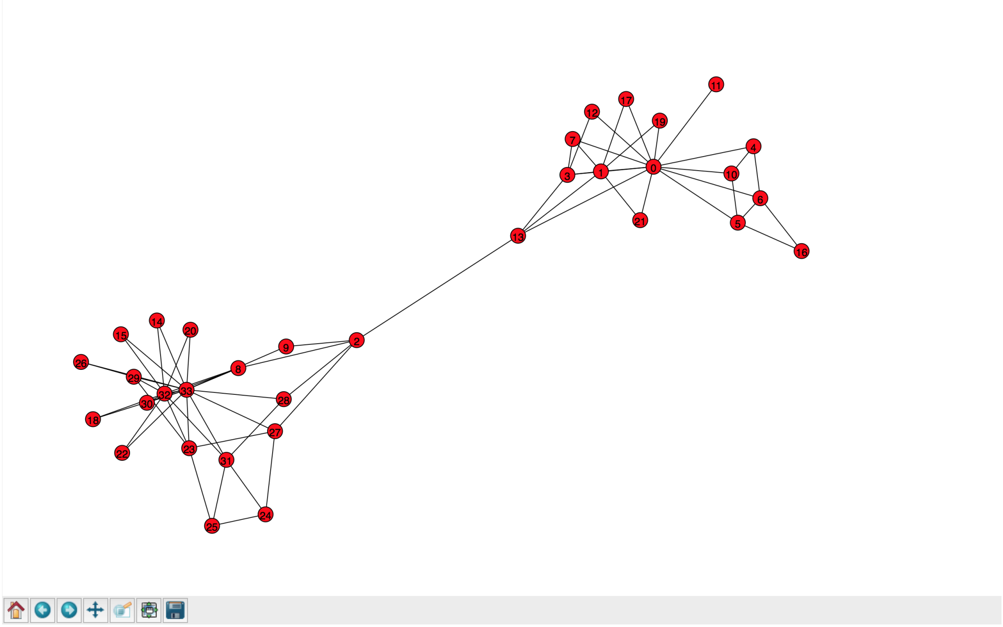

Figure 12: Iteration 10

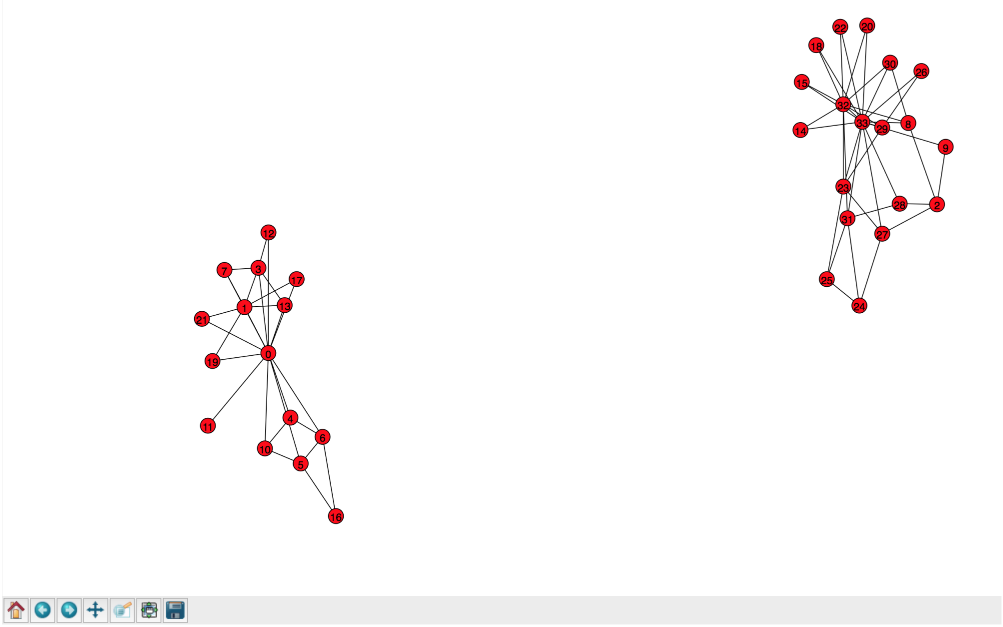

Figure 13: Final Segregation

Problem 2

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 solution for the problem can be obtained by changing the variable **maximumClusterThreshold** value to 3, 4 and 5.

Listing 2: karateClubGraph.py

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt


graph = nx.karate_club_graph()
edgeCount = graph.number_of_edges()
maximumClusterThreshold = 3
#maximumClusterThreshold = 4
#maximumClusterThreshold = 5
count = 0
clusterCount = nx.algorithms.number_connected_components(graph)

# Drawing the initial karateclub graph
nx.draw(graph, with_labels=True)
plt.show()

while (edgeCount > 0 and clusterCount < maximumClusterThreshold):
   betweennessDict = nx.algorithms.betweenness.edge_betweenness(graph)
   maximumBetweennessValue = np.max(list(betweennessDict.values()))
   edgeValue = ’’

   for edge in betweennessDict:
      if betweennessDict[edge] == maximumBetweennessValue:
         edgeValue = edge

   print (’Removing Edges...’,str(edgeValue[0])+"--->"+str(edgeValue[1]))
   graph.remove_edge(edgeValue[0], edgeValue[1])

   # Drawing the sequential karateclub graph
   nx.draw(graph, with_labels=True)
   plt.show()

   count = count + 1
   clusterCount = nx.algorithms.number_connected_components(graph)

print(’Total flows to separation:’,count)
```

The above code will generate 3,4 and 5 clusters with following iterations

1. Cluster 3 - 14 Iterations
2. Cluster 4 - 18 Iterations
3. Cluster 5 - 24 Iterations

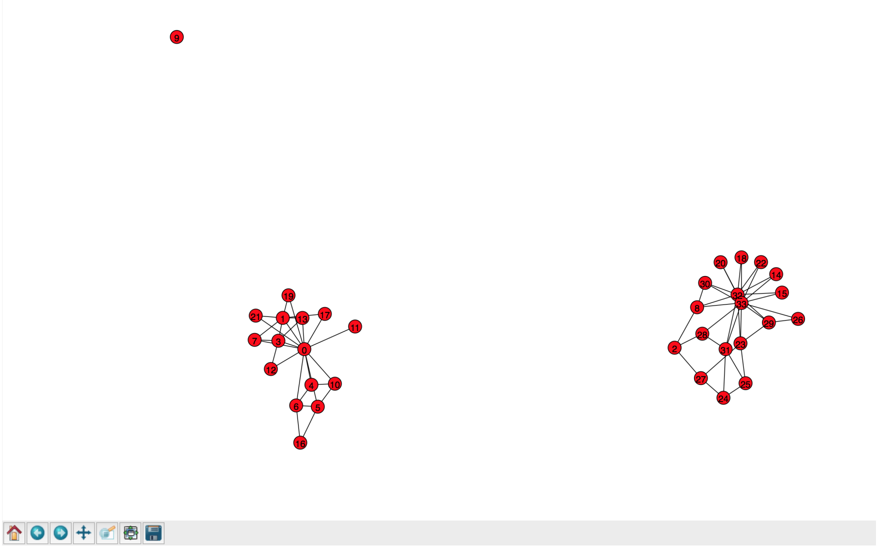

Figure 14: Cluster 3

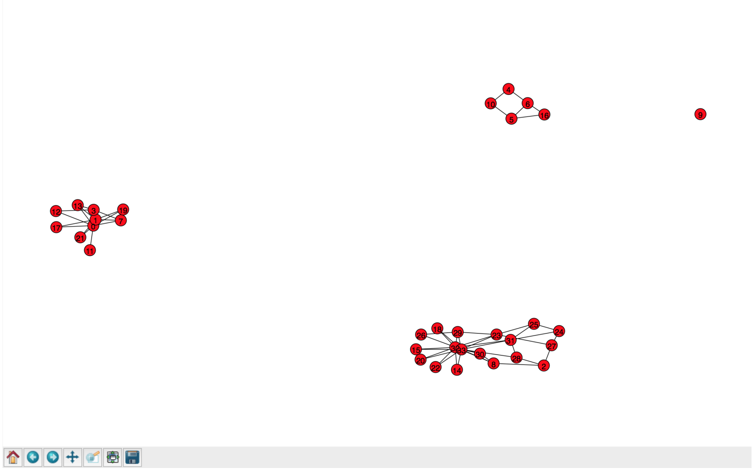

Figure 15: Cluster 4

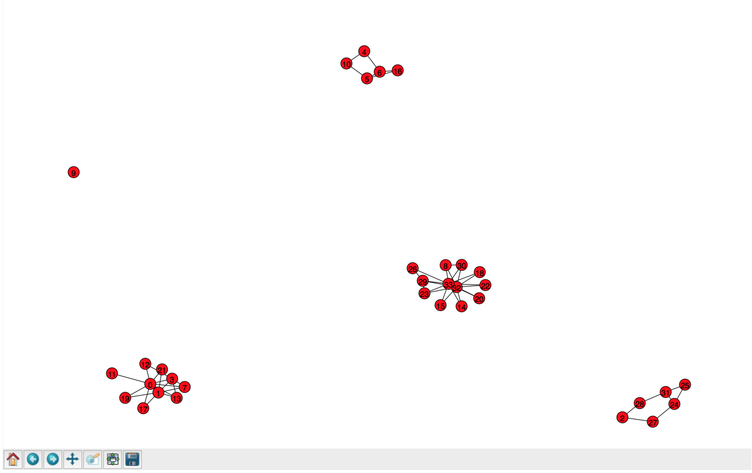

Figure 16: Cluster 5

Conclusion: I believe that graph obtained through **Girvan-Newman** algorithm fairly separates the original karate Club graph and thus can be proved that the result could be predicted through mathematical procedures. The final graph [Figure 12] is fairly similar to the karate club graph post split.

References

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

2. http://www.cl.cam.ac.uk/teaching/1617/MLRD/tasks/task12.html.

3. https://en.wikipedia.org/wiki/Girvan%E2%80%93Newman_algorithm.

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

5. https://networkx.github.io/documentation/networkx-1.10/reference/generated/ networkx.generators.social.karate_club_graph.html.

6. https://networkx.github.io/documentation/networkx-1.9/examples/graph/karate_club.html.