Assignment 5

Ryan Dill
CS 432
March 15, 2018

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 partioning 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?

For this question I used the code given and demonstrated by user Anmika Chhabra on her Youtube channel, https://www.youtube.com/watch?v=5P7ZCk3GM2o

A snippet of the algorithm used is as follows.

```
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 = "graph%(id)02d.png" % {"id": count}
        pl.figure()
        nx.draw_networkx(G, pos)
        pl.savefig(output)
        pl.close()
        while(l < 5):
                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(nx.number_of_nodes(G))
                output = "graph%(id)02d.png" % {"id": count}
                pl.figure()
                nx.draw_networkx(G)
                pl.savefig(output)
                pl.close()

        return c
```

First step was drawing the original Karate Club Graph

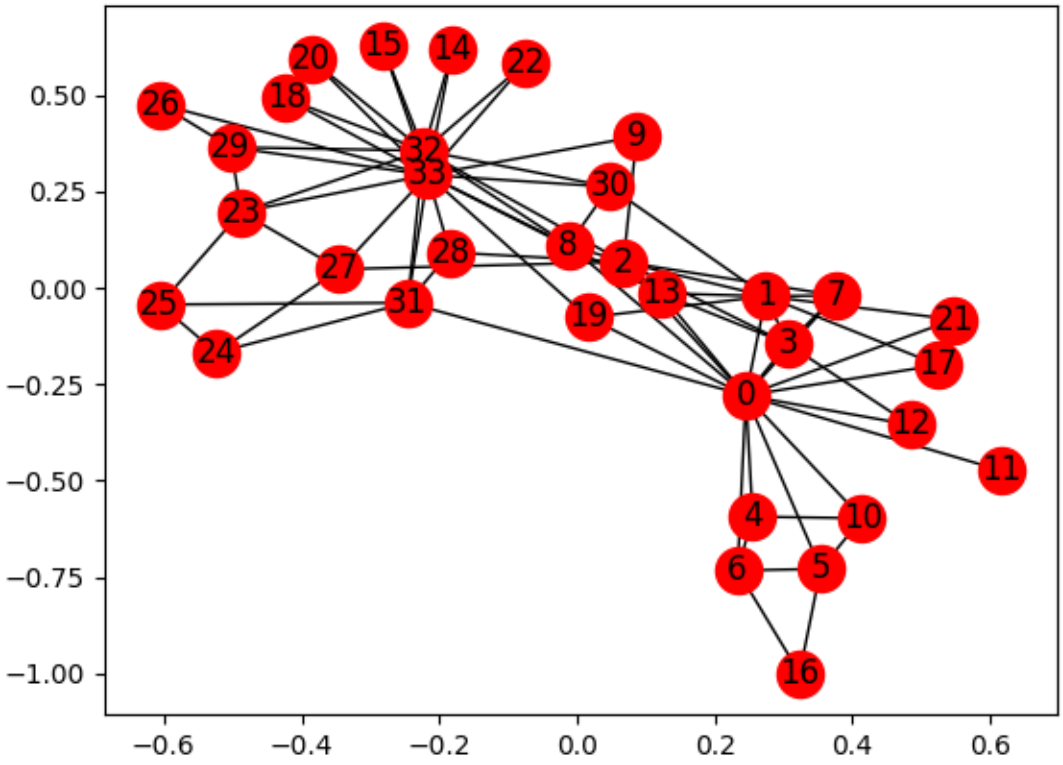

Figure 1

I then iterated through the Girvan Newman algorithm, recalculating the betweenness after every iteration until two separate communities were identified.

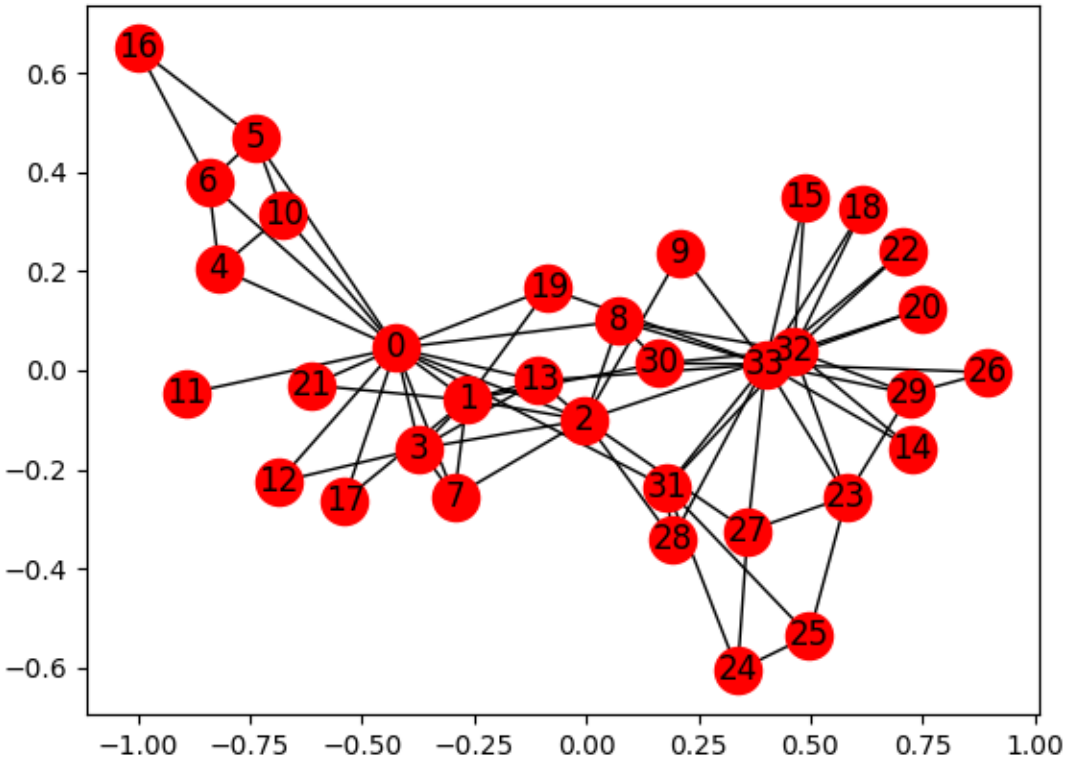
16
5
6
10
4
0
11
21
1
13
2
3
12
17
7
19
8
9
30
33
32
15
18
22
20
29
26
14
31
23
27
28
25
24
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

Figure 2

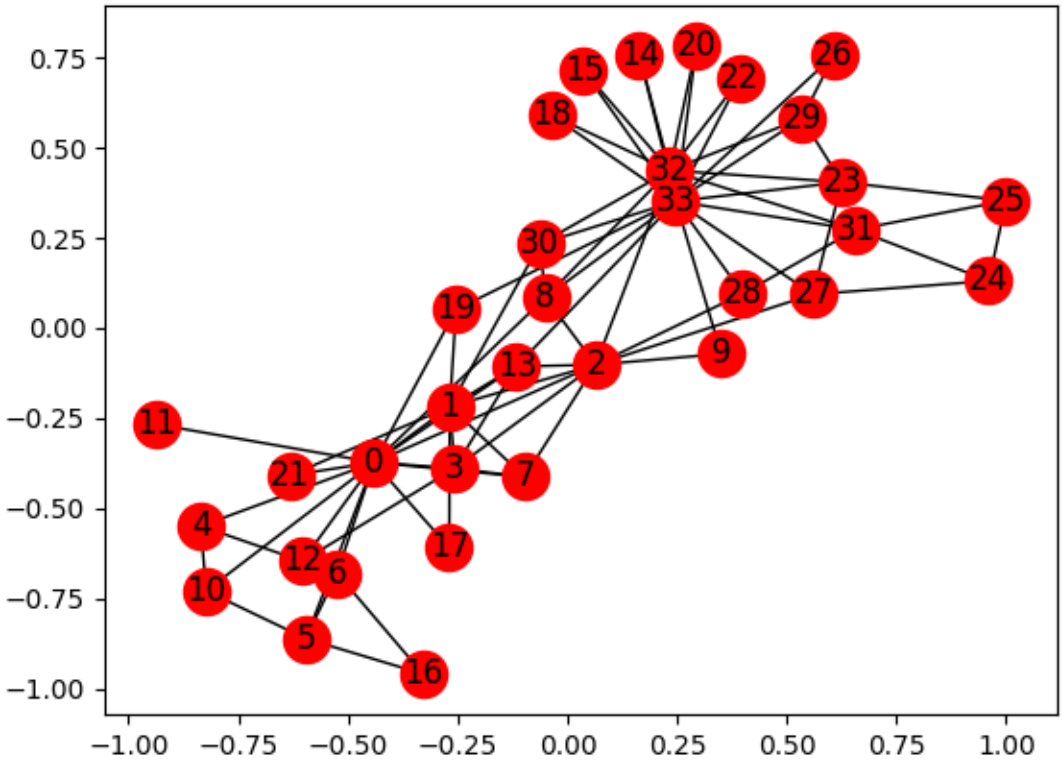
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

Figure 3

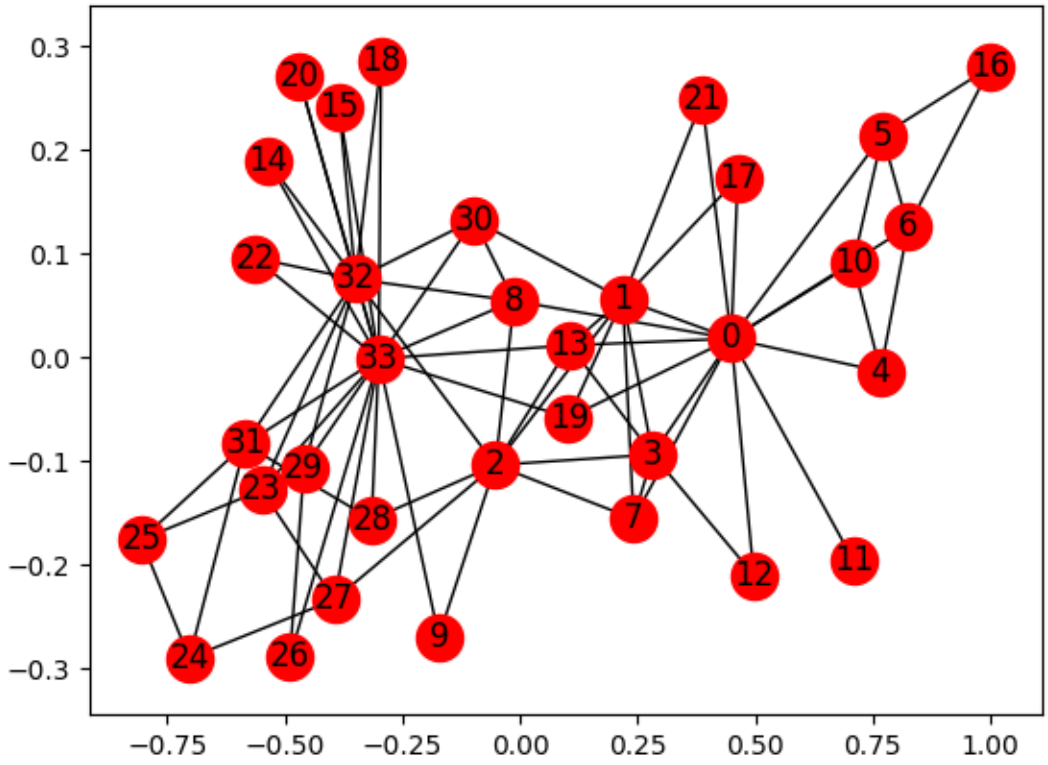
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

Figure 4

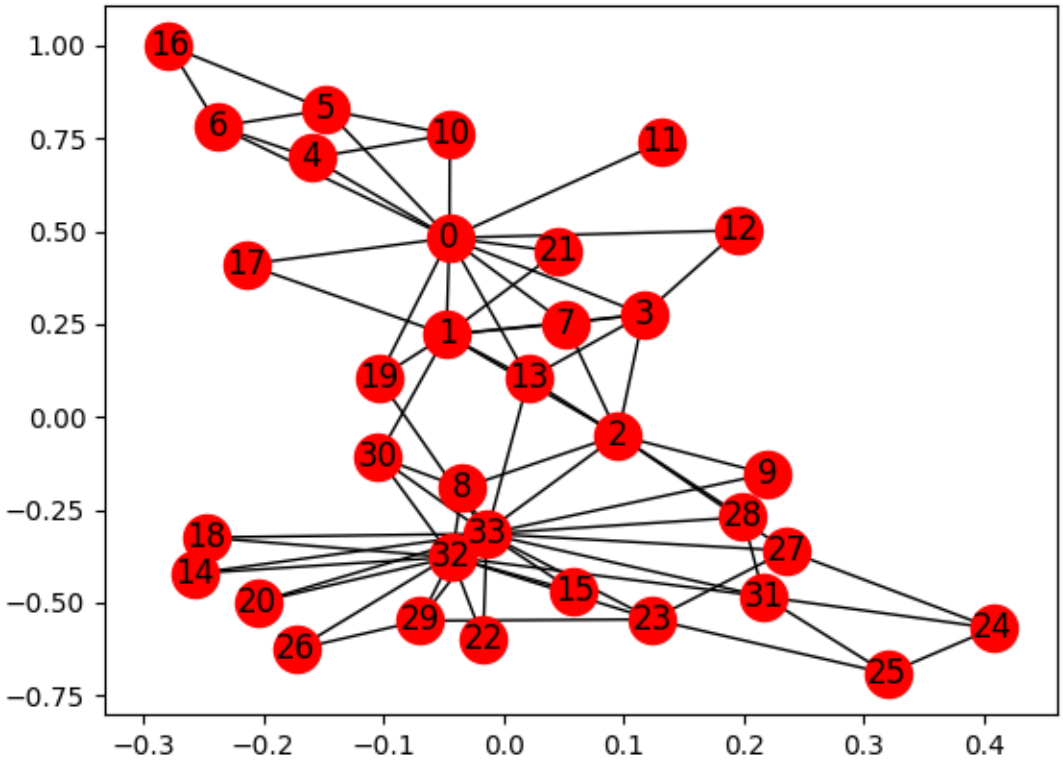
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
16
6
5
10
11
4
0
12
17
21
3
7
1
19
13
2
30
9
8
28
18
33
27
32
14
15
31
20
29
23
24
26
22
25

Figure 5

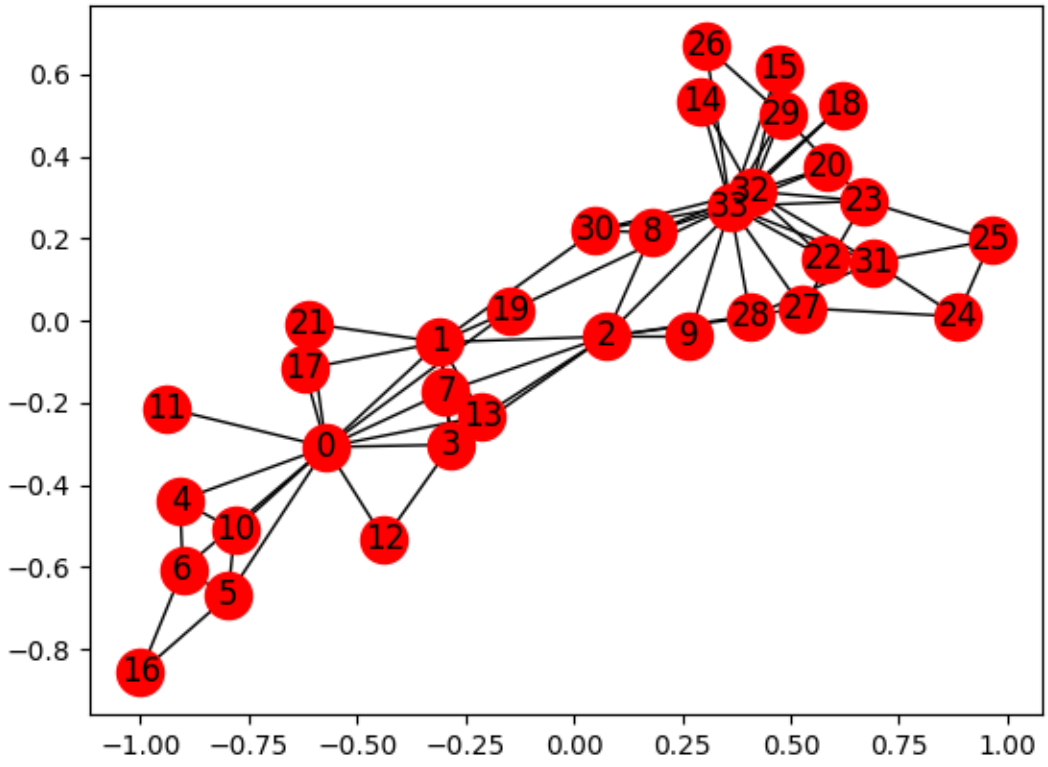
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
26
15
14
29
18
20
32
33
23
30
8
25
22
31
19
21
1
2
9
28
27
24
17
7
13
11
0
3
4
10
12
6
5
16

Figure 6

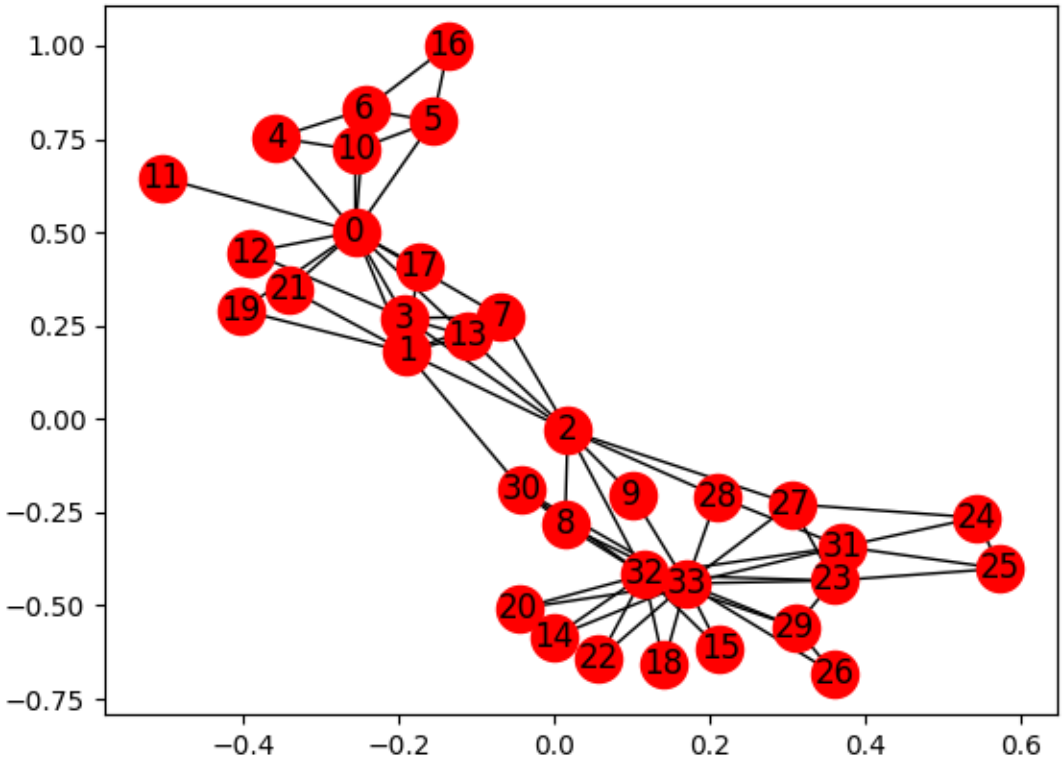
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.4
−0.2
0.0
0.2
0.4
0.6
16
6
5
4
10
11
0
12
17
21
19
3
7
13
1
2
30
9
28
27
24
8
31
32
33
23
25
20
14
22
18
15
29
26

Figure 7

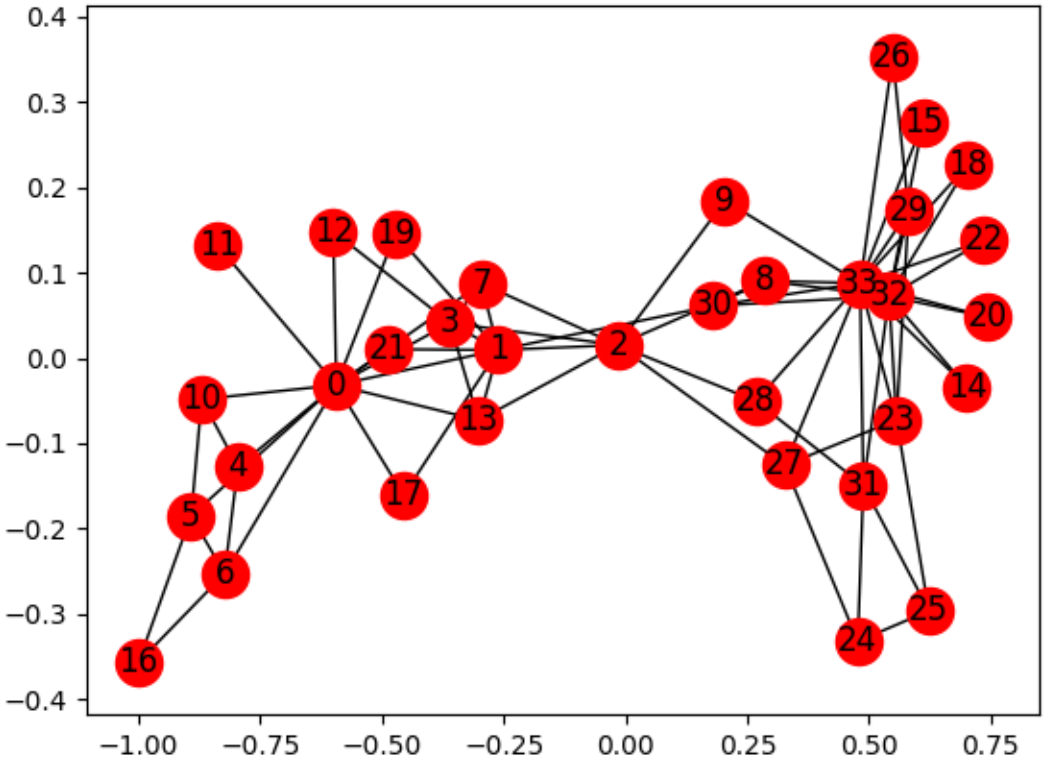
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
26
15
18
9
29
22
11
12
19
7
8
33
32
20
30
3
21
1
2
0
14
10
28
13
23
4
27
17
31
5
6
25
16
24

Figure 8

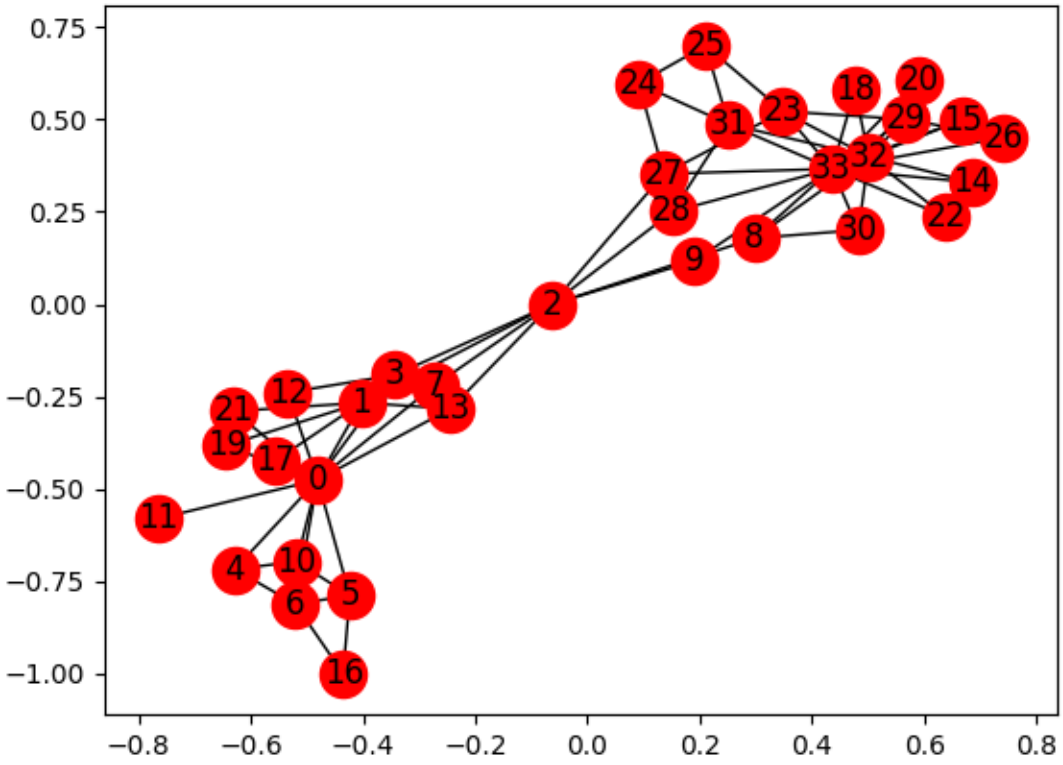
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
25
24
18
20
31
23
29
15
26
32
33
27
14
28
22
8
30
9
2
3
7
12
1
13
21
19
17
0
11
4
10
5
6
16

Figure 9

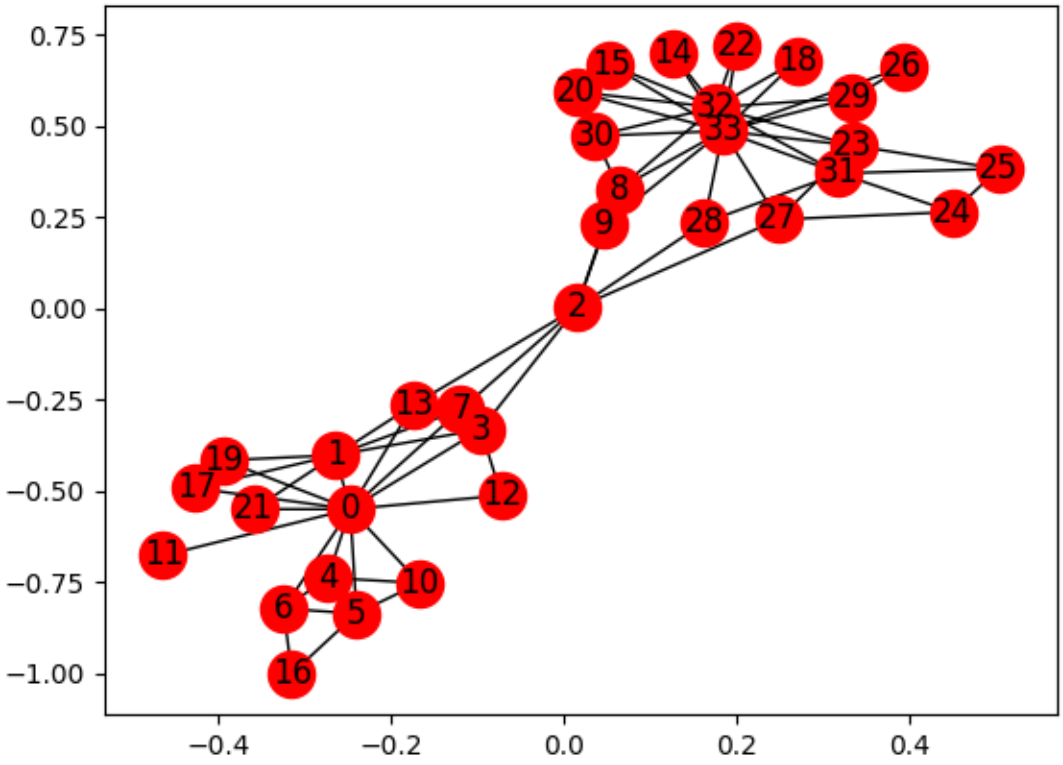
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.4
−0.2
0.0
0.2
0.4
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

Figure 10

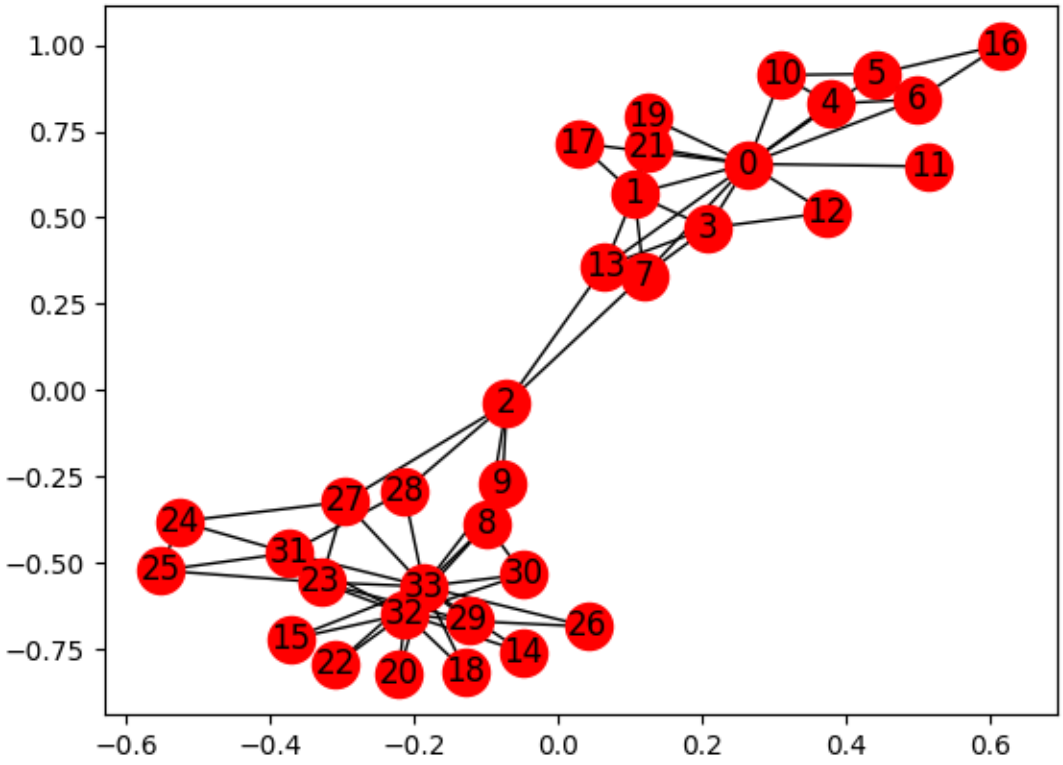
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
16
10
5
6
4
19
17
21
0
11
1
12
3
13
7
2
9
28
27
24
8
31
25
23
33
30
32
29
26
15
22
20
18
14

Figure 11

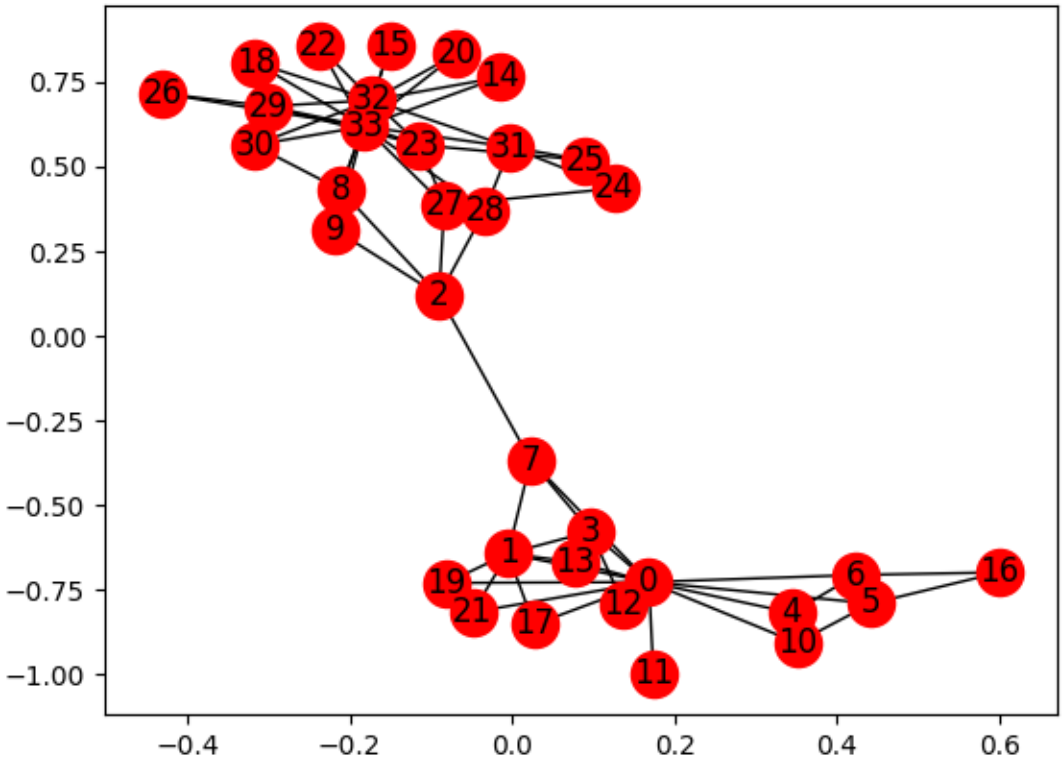
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.4
−0.2
0.0
0.2
0.4
0.6
22
15
20
18
14
26
32
29
33
30
23
31
25
24
8
27
28
9
2
7
3
1
13
19
0
6
16
21
12
4
5
17
10
11

Figure 12

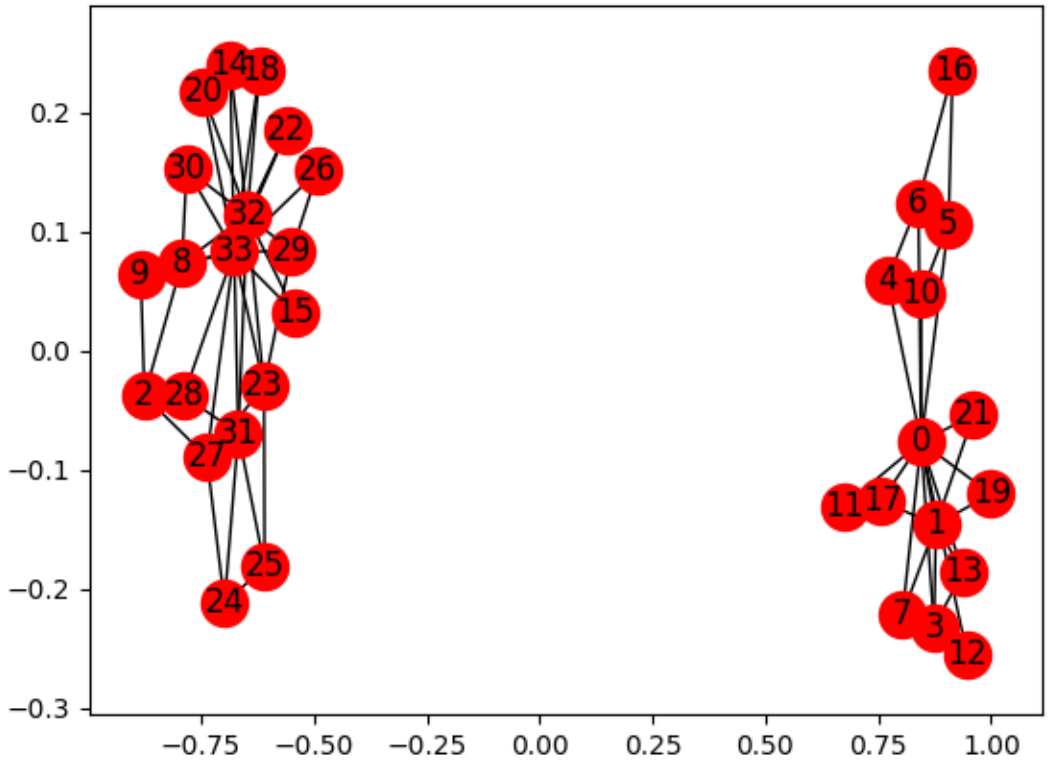

Figure 13

Using the data provided in class, the number of nodes and their split is identical to the provided data. The nodes are divided into two separate groups:

[32, 33, 2, 8, 9, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]

[0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 16, 17, 19, 21]

This shows that the data aligns with what was expected.