INTRODUCTION TO WEB SCIENCES: Assignment 5

Manoj Chandra Kompalli

3 March 2016

Contents

1 Question 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.

1.1 Understanding

It was a real tough time to understand what was expected from me to show. I had understood that I should show the difference between actual split and predicted split and prove they are almost same. There are a few algorithms that can predict the split of Karate problem. Here a Karate group is split into two due to some issues. Some people join one group and some join the other. Rest of them do not join any group. We want to predict using the algorithm, who joins which group.

In a graph, vertices represent the people and edges represent the contact. People having strong contact with their leader ideally choose that group. Mathematically, this can be proved by a few algorithms. Through a bit of research, I found out that Girvan-Newman Algorithm for edge betweenness is one of the most accurate around. Betweenness centrality is an indicator of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node. A node with high betweenness centrality has a large influence on the transfer of items through the network, under the assumption that item transfer follows the shortest paths. According to this algorithm, we have to find the edge with maximum betweenness and delete it. We have to iterate the process until we get two groups.

1.2 Implementation

- I had to download the karate graphML .
- Install the igraph library for python
- Install pycairo for igraph.
- Load the graph
- Find edge betweeness of graph
- Find the edge with maximum edge betweenness
- Delete the edge
- Repeat the process until you make two clusters

1.3 Code Listing

1.3.1 karateGroup.py

```
from igraph import *
#loads a graphML file
graph =load('karate.GraphML')
#labels a graph with name
graph.vs["label"] = graph.vs["name"]
#assigning a layout to the graph
mylayout = graph.layout("kk")
#Array of colors
color_dict = {1: "blue", 2: "pink"}
#plotting the original graph before seperation
plot(graph, 'graphkk.pdf', layout = mylayout, vertex_color = [color_dict[faction] for
    faction  in graph.vs["Faction"]])
print 'Removed Edges \n '
#looping until you get two clusters
while len(graph.clusters()) < 2 :
  #edge betweenness
    ebs = graph.edge_betweenness()
    max_eb = max(ebs)
    #finding the index of edge which has maximum edge betweenness
    indexEdgeMaxBtwness = max(xrange(len(ebs)), key = ebs.__getitem__)
    #deleting the edges with maximum edge betweenness
    graph.delete_edges(indexEdgeMaxBtwness)
    print indexEdgeMaxBtwness
#assigning another layout to the final graph
layout = graph.layout("kk")
#plotting the final graph
plot(graph, 'finalgraph.pdf' , layout = layout)
```

1.4 Input

1.4.1 karate.GraphML

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
         http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
<!-- Created by igraph -->
  <key id="name" for="graph" attr.name="name" attr.type="string"/>
  <key id="Citation" for="graph" attr.name="Citation" attr.type="string"/>
  <key id="Author" for="graph" attr.name="Author" attr.type="string"/>
  <key id="Faction" for="node" attr.name="Faction" attr.type="double"/>
  <key id="name" for="node" attr.name="name" attr.type="string"/>
  <key id="weight" for="edge" attr.name="weight" attr.type="double"/>
  <graph id="G" edgedefault="undirected">
    <data key="name">Zachary&apos;s karate club network</data>
    <data key="Citation">Wayne W. Zachary. An Information Flow Model for Conflict and
     Fission in Small Groups. Journal of Anthropological Research Vol. 33, No. 4
     452-473</data>
    <data key="Author">Wayne W. Zachary</data>
    <node id="n0">
      <data key="Faction">1</data>
      <data key="name">Mr Hi</data>
    </node>
    <node id="n1">
      <data key="Faction">1</data>
      <data key="name">Actor 2</data>
    </node>
    <node id="n2">
      <data key="Faction">1</data>
      <data key="name">Actor 3</data>
    </node>
    <node id="n3">
      <data key="Faction">1</data>
      <data key="name">Actor 4</data>
    </node>
    <node id="n4">
      <data key="Faction">1</data>
      <data key="name">Actor 5</data>
    </node>
    <node id="n5">
      <data key="Faction">1</data>
      <data key="name">Actor 6</data>
    </node>
    <node id="n6">
      <data key="Faction">1</data>
      <data key="name">Actor 7</data>s
    </node>
    <node id="n7">
      <data key="Faction">1</data>
      <data key="name">Actor 8</data>
    </node>
```

1.5 Summary and Output

Figure 1 is the initial graph generated from the karate.GraphML. The initial graph consists of all the vertices generated from karate graphML. The actual graph in Figure 2 shows actual separation of 34 members into two groups based on faction data. Blue belong to Mr. Hi and the pink belong to John. By comparing it with Figure 3. By careful observation, we can see that only the actor 3 who was predicted to be in Mr. John s group is actually in Mr. Hi's group. This proves that our prediction using the algorithm is near perfect. We can use this model to predict the number of groups in a real world social network.

1.5.1 Initial Graph

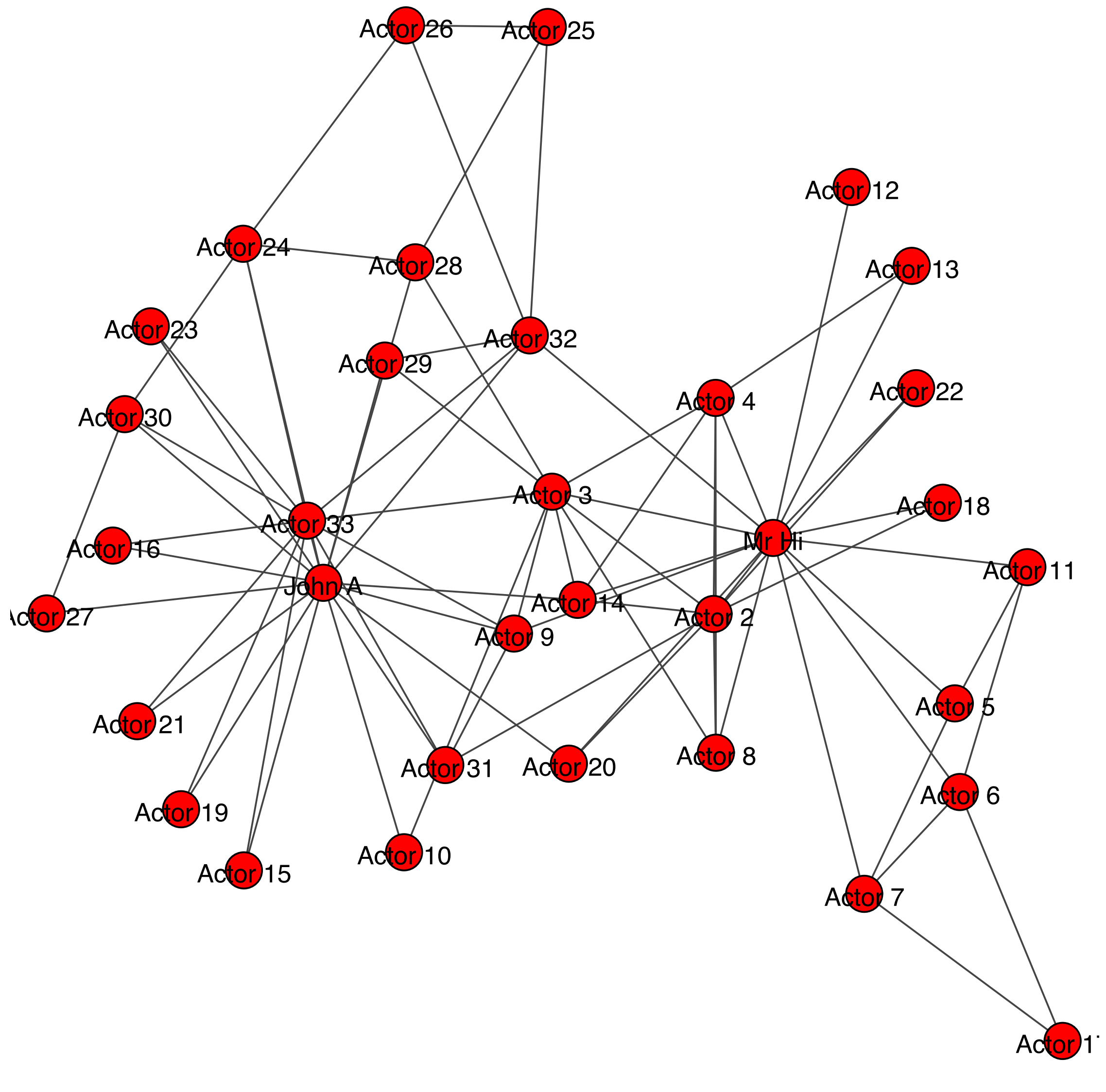


Figure 1: initial-graph

1.6 Processed graphs

1.6.1 Graph showing actual division based on Faction

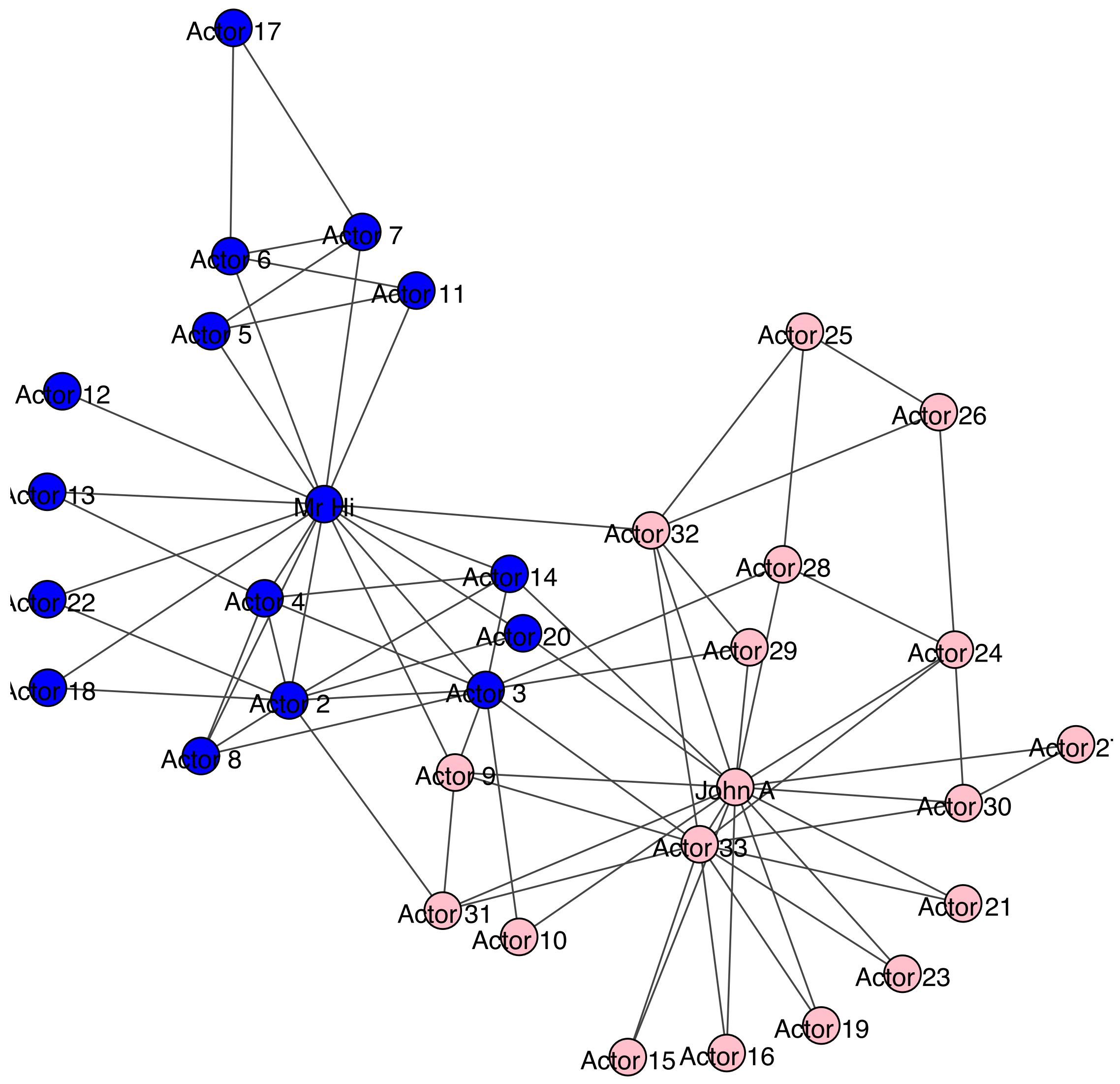


Figure 2: Actual Seperation of Groups

1.6.2 Graph showing actual division based on Faction

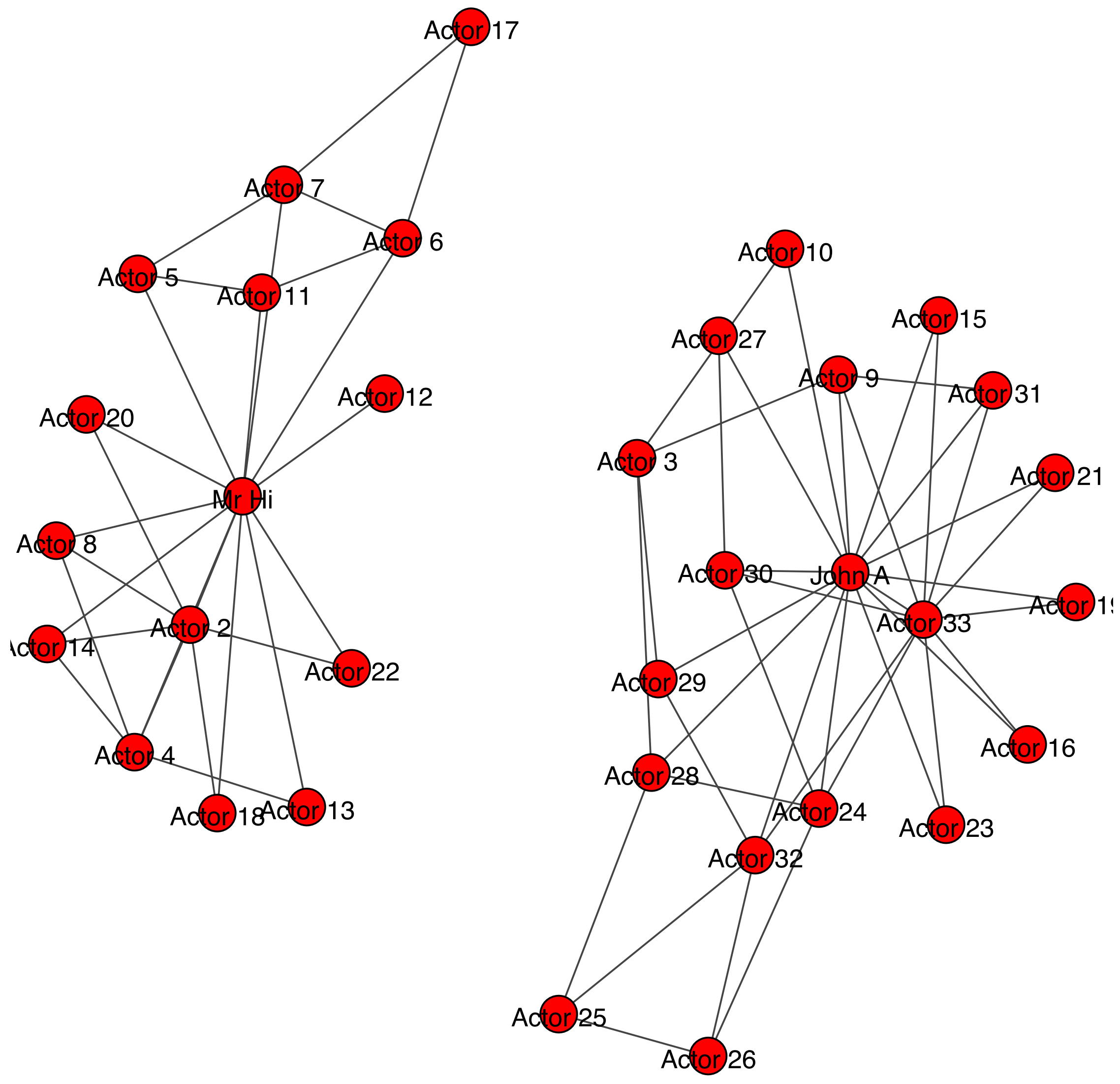
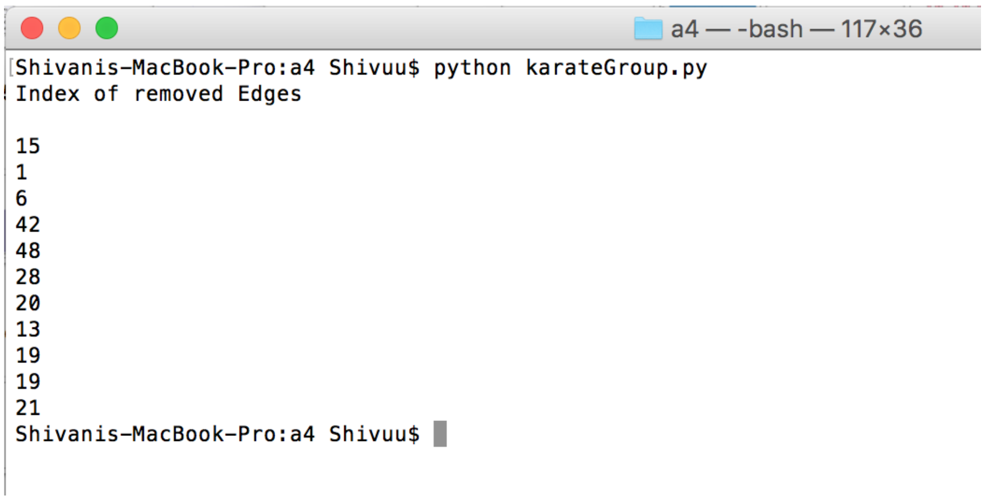


Figure 3: Predicted Seperation using Edge betweenness

1.7 Output on console showing removed edges



```
Shivanis-MacBook-Pro:a4 Shivuu$ python karateGroup.py
Index of removed Edges

15
1
6
42
48
28
20
13
19
19
21
Shivanis-MacBook-Pro:a4 Shivuu$
```

Figure 4: Removed edges

2 Question 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?

2.1 Understanding

I have decided to follow the same approach I was using earlier to find the two groups. Now, I will have to repeat the loop and delete the edges until I get three, four, five clusters respectively.

2.2 Code Listing

2.2.1 karateAll.py

```
from igraph import *
graph =load('karate.GraphML')
graph.vs["label"] = graph.vs["name"]
mylayout = graph.layout("kk")
plot(graph, 'graph1.pdf', layout = mylayout)
print 'Index of removed Edges for forming 3 clusters  \n '
while len(graph.clusters()) < 3 :
    edgeBtw = graph.edge_betweenness()
    maxEdgeBtw = max(edgeBtw)
    indexEdgeMaxBtwness = max(xrange(len(edgeBtw)), key = edgeBtw.__getitem__)
    print indexEdgeMaxBtwness
    graph.delete_edges(indexEdgeMaxBtwness)

layout = graph.layout("kk")
plot(graph, 'finalgraph3.pdf' , layout = layout)
print 'Index of removed Edges for forming 4 clusters  \n '
while len(graph.clusters()) < 4:
    edgeBtw = graph.edge_betweenness()
    maxEdgeBtw = max(edgeBtw)
    indexEdgeMaxBtwness = max(xrange(len(edgeBtw)), key = edgeBtw.__getitem__)
    print indexEdgeMaxBtwness
    graph.delete_edges(indexEdgeMaxBtwness)

layout = graph.layout("kk")
plot(graph, 'finalgraph4.pdf' , layout = layout)
print 'Index of removed Edges for forming 5 clusters  \n '
while len(graph.clusters()) < 5:
    edgeBtw = graph.edge_betweenness()
    maxEdgeBtw = max(edgeBtw)
    indexEdgeMaxBtwness = max(xrange(len(edgeBtw)), key = edgeBtw.__getitem__)
    print indexEdgeMaxBtwness
    graph.delete_edges(indexEdgeMaxBtwness)
layout = graph.layout("kk")
plot(graph, 'finalgraph5.pdf' , layout = layout)
```

2.3 Input

2.3.1 karate.GraphML

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
         http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
<!-- Created by igraph -->
  <key id="name" for="graph" attr.name="name" attr.type="string"/>
  <key id="Citation" for="graph" attr.name="Citation" attr.type="string"/>
  <key id="Author" for="graph" attr.name="Author" attr.type="string"/>
  <key id="Faction" for="node" attr.name="Faction" attr.type="double"/>
  <key id="name" for="node" attr.name="name" attr.type="string"/>
  <key id="weight" for="edge" attr.name="weight" attr.type="double"/>
  <graph id="G" edgedefault="undirected">
    <data key="name">Zachary&apos;s karate club network</data>
    <data key="Citation">Wayne W. Zachary. An Information Flow Model for Conflict and
    Fission in Small Groups. Journal of Anthropological Research Vol. 33, No. 4
    452-473</data>
    <data key="Author">Wayne W. Zachary</data>
    <node id="n0">
      <data key="Faction">1</data>
      <data key="name">Mr Hi</data>
    </node>
    <node id="n1">
      <data key="Faction">1</data>
      <data key="name">Actor 2</data>
    </node>
    <node id="n2">
      <data key="Faction">1</data>
      <data key="name">Actor 3</data>
    </node>
    <node id="n3">
      <data key="Faction">1</data>
      <data key="name">Actor 4</data>
    </node>
    <node id="n4">
      <data key="Faction">1</data>
      <data key="name">Actor 5</data>
    </node>
    <node id="n5">
      <data key="Faction">1</data>
      <data key="name">Actor 6</data>
    </node>
    <node id="n6">
      <data key="Faction">1</data>
      <data key="name">Actor 7</data>s
    </node>
    <node id="n7">
      <data key="Faction">1</data>
      <data key="name">Actor 8</data>
    </node>
```

2.4 Output Files

2.4.1 Intial-graph

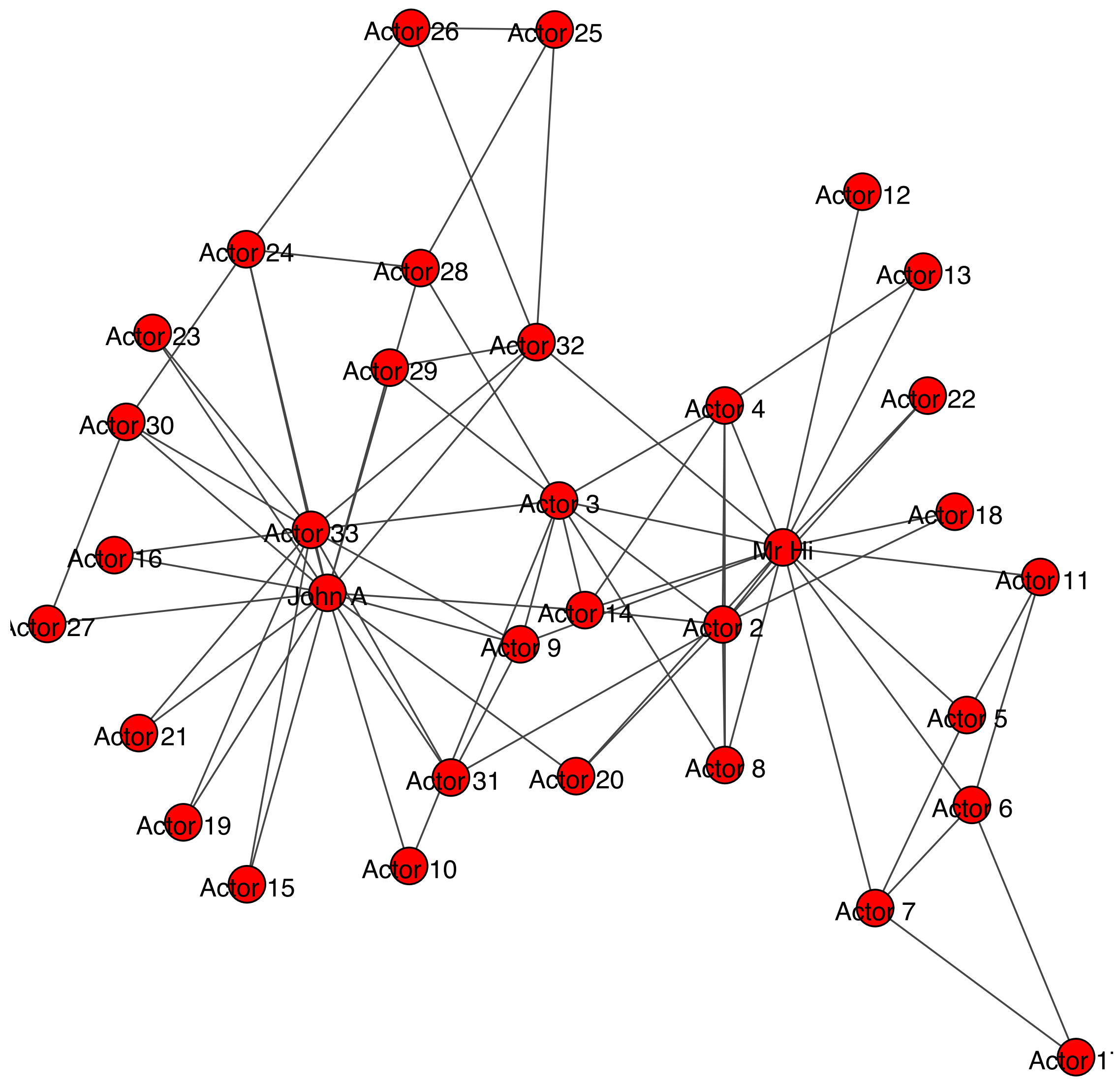


Figure 5: Initial-graph

2.4.2 Graph after the split

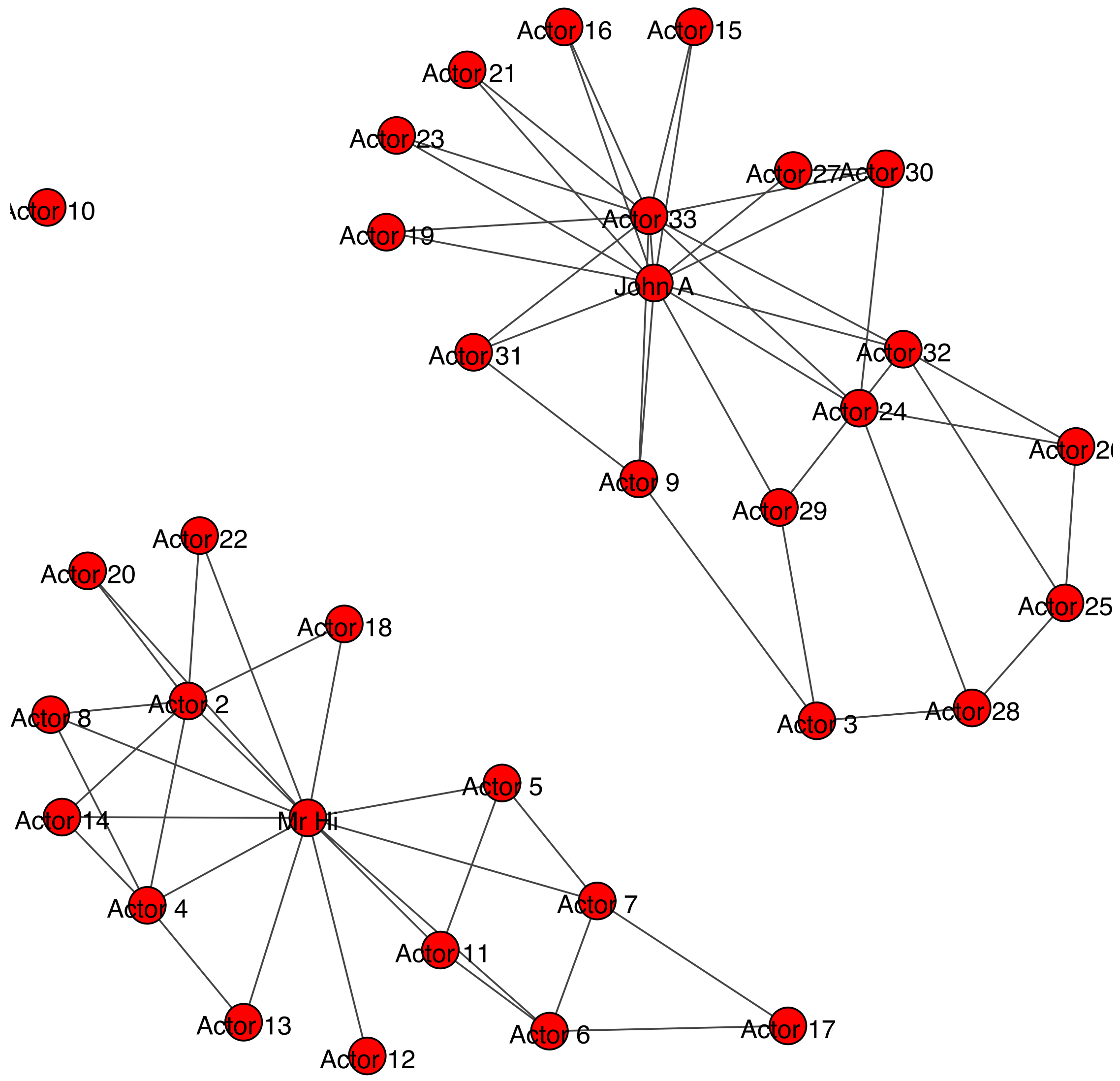


Figure 6: Initial graph split into 3 groups

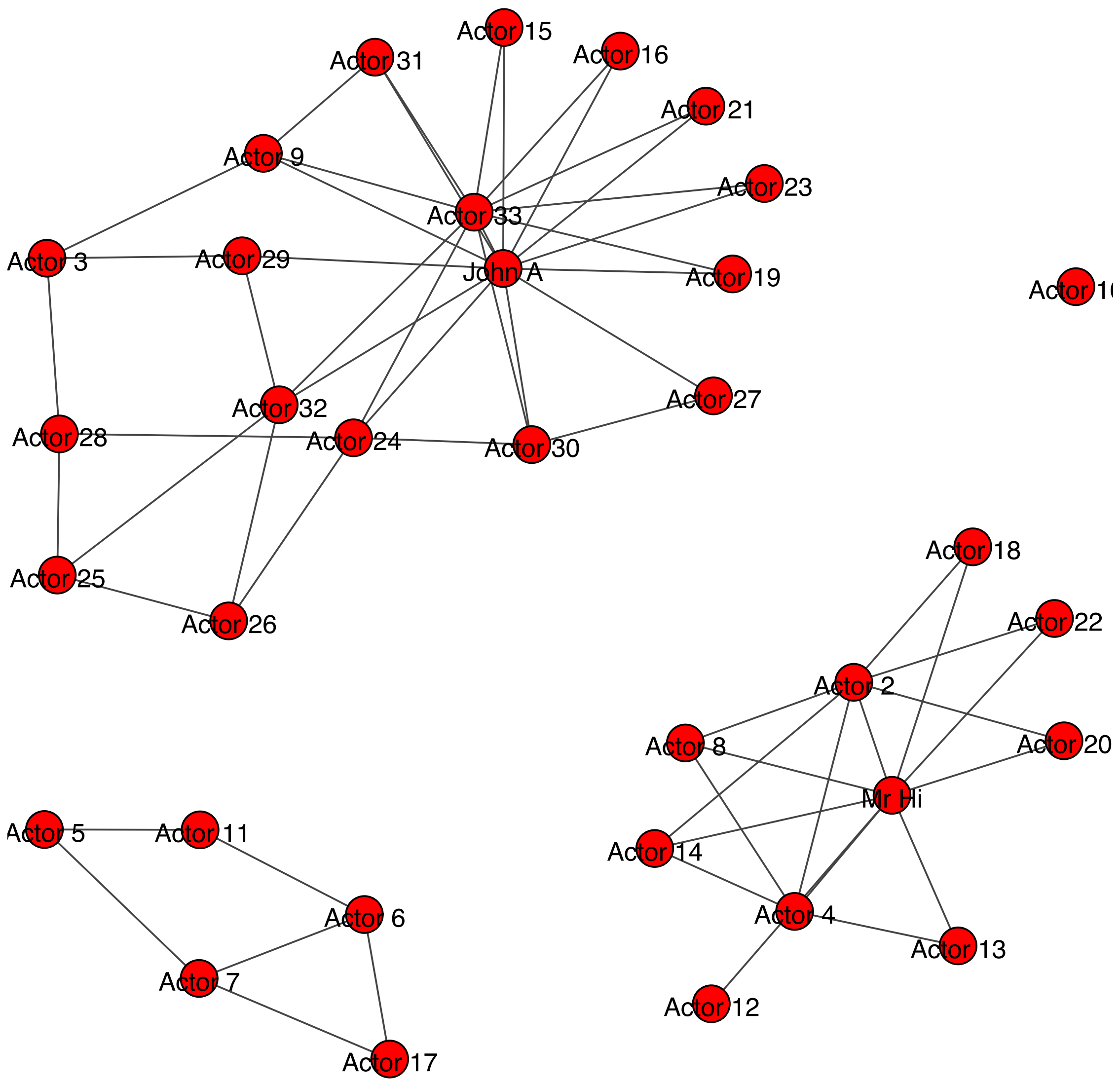
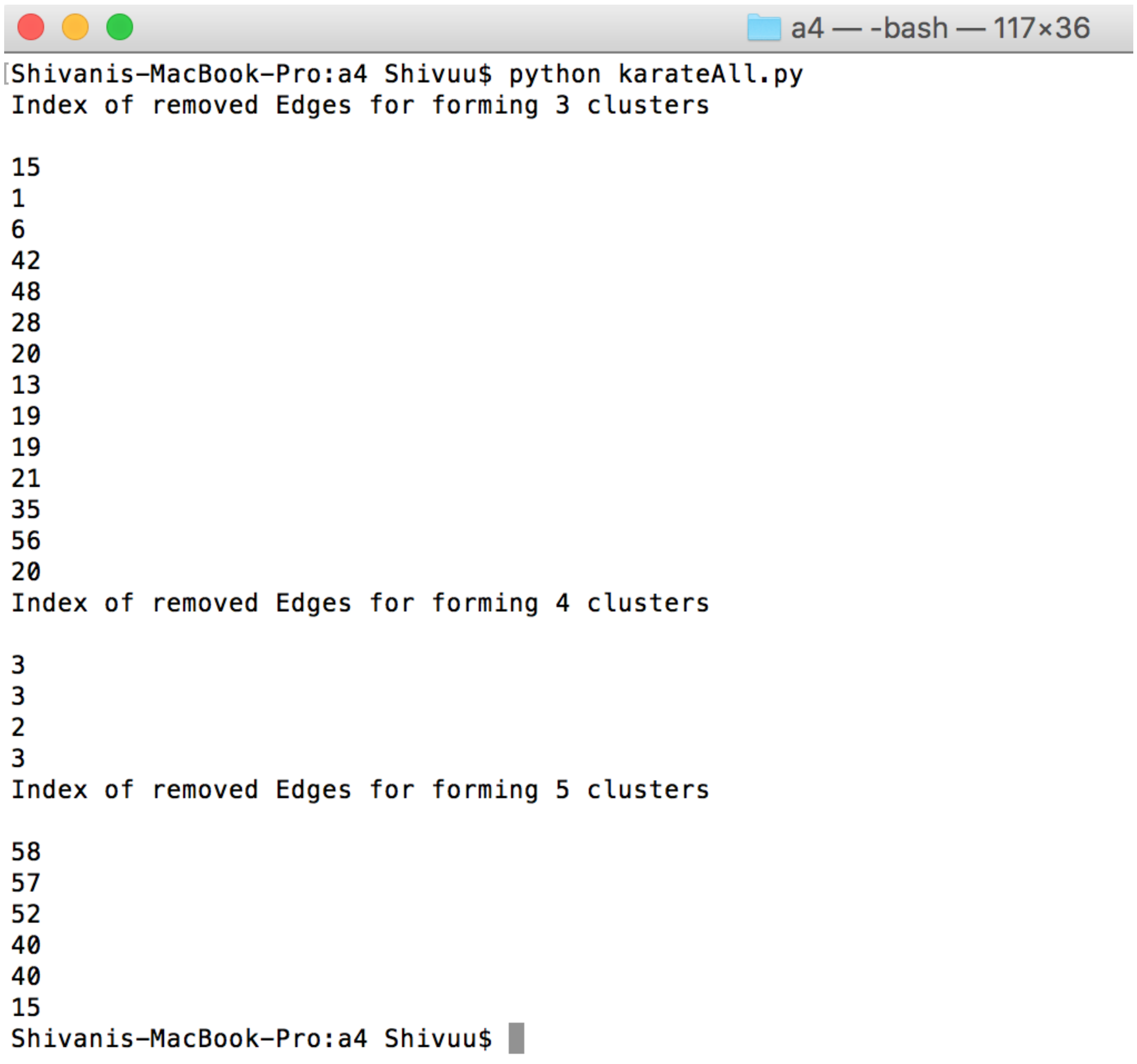


Figure 7: Initial graph split into 4 groups



```
Shivanis-MacBook-Pro:a4 Shivuu$ python karateAll.py
Index of removed Edges for forming 3 clusters

15
1
6
42
48
28
20
13
19
19
21
35
56
20
Index of removed Edges for forming 4 clusters

3
3
2
3
Index of removed Edges for forming 5 clusters

58
57
52
40
40
15
Shivanis-MacBook-Pro:a4 Shivuu$
```

Figure 8: Initial graph split into 4 groups

References

[1] igraph api. `https://www.cs.rhul.ac.uk/home/tamas/development/igraph/tutorial/tutorial.html`.

[2] installation. `http://cairographics.org/pycairo`.

[3] Lecture slides. `http://www-personal.umich.edu/~ladamic/courses/networks/si614w06/ppt/lecture18.ppt`.

[4] Paper-an information flow model for conflict and fission in small groups. `http://aris.ss.uci.edu/~lin/76.pdf`.

[]