Introduction to Web Science/595: Assignment #6

Dr. Nelson

Francis Pruter

Thursday, 30 October, 2014

Contents

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?

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

SOLUTION
By using Girvan-Newman Community Edge Betweenness, there was a 94.1 percent accuracy in predicting the how the karate club would have split. Community Edge Betweenness used the weight provided by the python-graph man.
Girvan-Newman algorithm:

1. The betweenness of all existing edges in the network is calculated first.
2. The edge with the highest betweenness is removed.
3. The betweenness of all edges affected by the removal is recalculated.
4. Steps 2 and 3 are repeated until no edges remain.

Below is the python code I used to compare the actual split and the predicted split.

Listing 1: karate_club_igraph.py

```
#!/usr/bin/env python

"""
Data file from:
http://igraph.org/python/doc/tutorial/tutorial.html

Reference:
as_clustering():
    http://stackoverflow.com/questions/22046499/community-detection-with-igraph-in-
        python

"""

from igraph import *

karate = Graph.Read_GraphML("karate.GraphML")

layout=karate.layout('kk')
karate.vs["label"] = karate.vs["name"]


karate_actual_MrHi = []
karate_actual_John = []


#Community Edge Betweenness cluster=2
karate_ceb = karate.community_edge_betweenness(clusters=2, directed=False, weights=
    karate.es["weight"]).as_clustering()
```

```
#Community Edge Betweenness cluster=3
karate_ceb3 = karate.community_edge_betweenness(clusters=3, directed=False, weights=
    karate.es["weight"]).as_clustering()

#Community Edge Betweenness cluster=4
karate_ceb4 = karate.community_edge_betweenness(clusters=4, directed=False, weights=
    karate.es["weight"]).as_clustering()

#Community Edge Betweenness cluster=5
karate_ceb5 = karate.community_edge_betweenness(clusters=5, directed=False, weights=
    karate.es["weight"]).as_clustering()

for x in range(0,34):
  if (karate.vs[x]["Faction"] == 1.0):
     karate_actual_MrHi.append(x)
  if (karate.vs[x]["Faction"] == 2.0):
     karate_actual_John.append(x)


#Initial
plot(karate, "graphs/beforesplit.pdf", layout = layout, margin = 20)

#Community Edge Betweenness
plot(karate_ceb, "graphs/predictedsplit.pdf", layout = layout, margin = 20)

#Community Edge Betweenness, cluster = 3
plot(karate_ceb3, "graphs/predictedsplit3.pdf", layout = layout, margin = 20)

#Community Edge Betweenness, cluster = 4
plot(karate_ceb4, "graphs/predictedsplit4.pdf", layout = layout, margin = 20)

#Community Edge Betweenness, cluster = 5
plot(karate_ceb5, "graphs/predictedsplit5.pdf", layout = layout, margin = 20)


#accuracy
diff = list(set(karate_actual_MrHi) - set(karate_ceb[0])) + list(set(karate_ceb[0]) -
    set(karate_actual_MrHi))

diff.sort()

accuracy = round(((34-len(diff))/34.0)*100, 1)

print "Actual Mr. Hi:\t", karate_actual_MrHi
print "Predic Mr. Hi:\t", karate_ceb[0]


print "Actual John:\t", karate_actual_John
print "Predic John:\t", karate_ceb[1]
print "\nDifference:\t", diff
print "Accuracy: \t",  accuracy
```

Problem 1 continued on next page...

Note the difference between the actual and predicted split were nodes 2 and 13 resulting in 94.1 percent accuracy

Below is the karate club below the split and the predicted graph of the split:

```
francis@soda-pop: ~/ODU/CS595/Assign6
francis@soda-pop:~/ODU/CS595/Assign6$ ./karate_club_igraph.py
Actual Mr. Hi:  [0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 16, 17, 19, 21]
Predic Mr. Hi:  [0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 16, 17, 19, 21]
Actual John:    [8, 9, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 3
2, 33]
Predic John:    [2, 8, 9, 13, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30
, 31, 32, 33]

Difference:     [2, 13]
Accuracy:       94.1
francis@soda-pop:~/ODU/CS595/Assign6$
```

Figure 1: Code Running

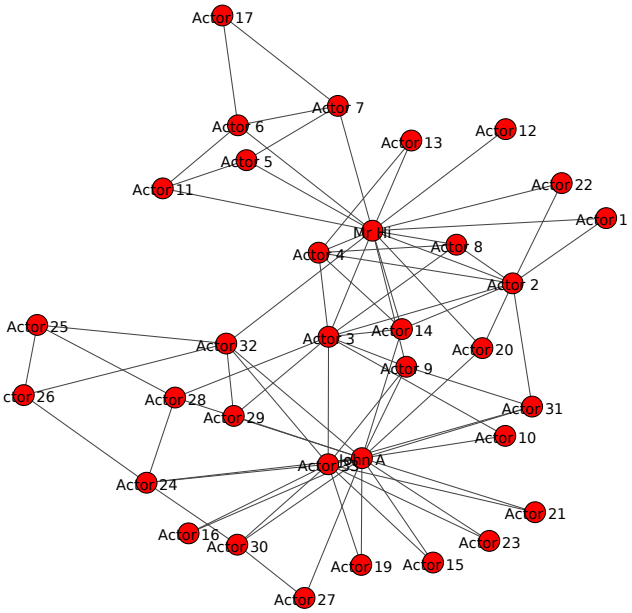

Figure 2: Before Split

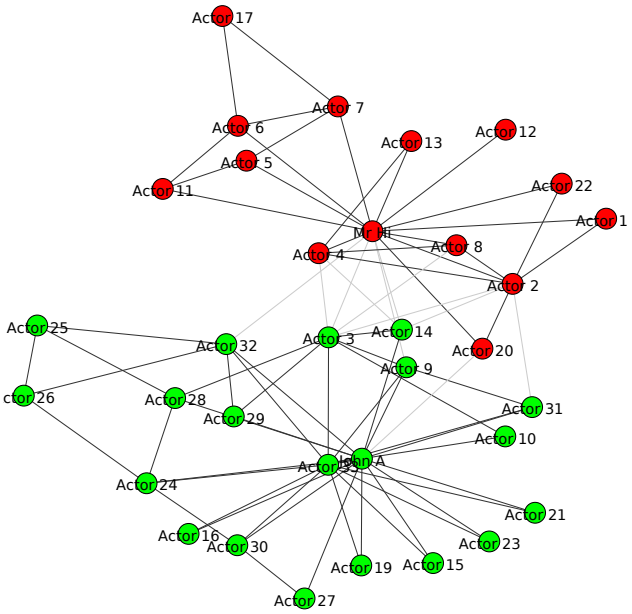

Figure 3: Predicted Split

Problem 2

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 code in problem 1 also executes for clusters of size 3, 4, and 5.

Below is the the projected way it would look

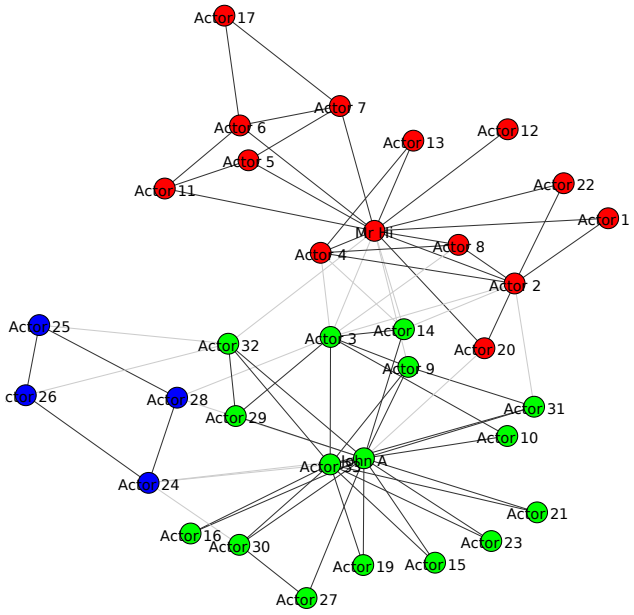

Figure 4: Projected 3 groups

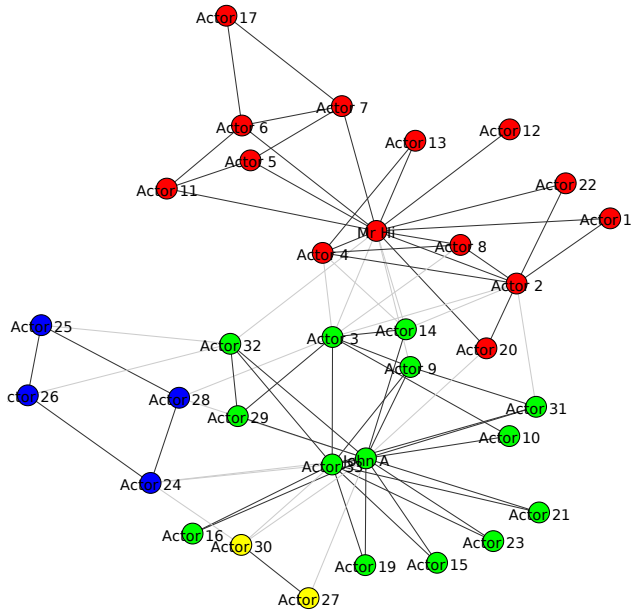

Figure 5: Projected 4 groups

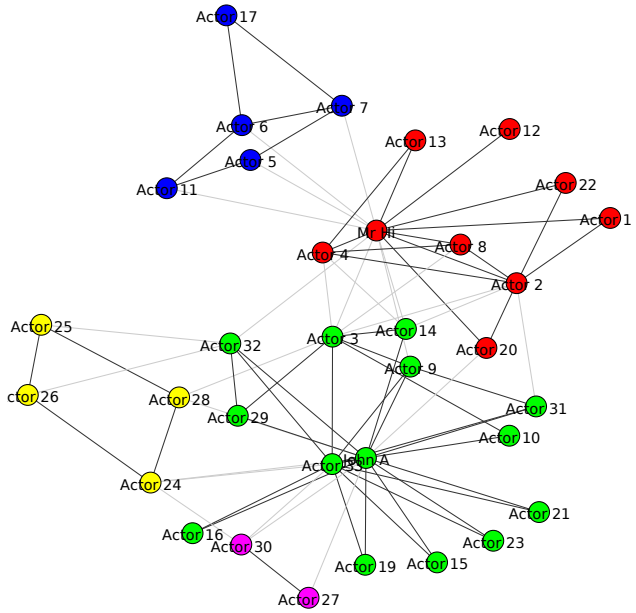

Figure 6: Projected 5 groups

References

[1] python-igraph Manual. http://igraph.org/python/doc/tutorial/tutorial.html.

[2] python-igraph Manual. http://igraph.org/python/doc/igraph.GraphBase-class.htmlcommunity$_e dge_b etweenness$.

[3] Wikipedia - GirvanNewman algorithm. http://en.wikipedia.org/wiki/Girvan%E2%80%93Newman_algorithm, 2004.

[4] Tamas Nepusz, Gabor Csardi. python-igraph v0.6 documentation. http://www.cs.rhul.ac.uk/home/tamas/development/igraph/tutorial/tutorial.html, 2010.