Web Science: Assignment #5

Alexander Nwala

Mohd. Nauman Siddique

Saturday, March 24, 2019

Contents

Problem 1 **3**

Problem 2 **7**

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. 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?

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/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-dete
9478989#9478989
http://stackoverflow.com/questions/5822265/are-there-implementations-of-algorithms-for-com
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_betweennes

SOLUTION

The solution to the problem is as below:

1. **Drawing Connected Karate Club**: I used the networkX library to load the karate club dataset and plot all the 34 nodes in a network graph where every edge shows the connection between the nodes as per the karate dataset. The node 0 and node 33 are labelled as Mr. Hi and John A. In, total the dataset has 78 edges and 34 nodes. Figure 1 shows the network graph for the Karate Club. The function *connected_graph()* was used to draw this graph.

2. **Split in Karate Club**: Figure 2 shows Karate Club network graph after the conflict between Mr. Hi and John A. The supporters of Mr. Hi are colored with red while the supporters of John A. are colored by blue. The affiliations of nodes has been calculated using Girvan-Newman algorithm using the maximum betweeness algorithm which requires deleting of nodes which have have togetherness until the graph splits into two partitions. The function *disconnected_graph()* has been used to plot this network graph.

3. **Two connected components of Karate Club**: Figure 3 shows the two connected components of the Karate Club after a split. The function *multiple_iterations()* has been used to draw this plot. The function can also be used to find further splits till each node becomes separate from each other. This has has also been used to generate graphs shows in Problem 2 for splits 3, 4, 5, and 6. This shows the total split of the group into 2 splits which has been done using Girvan-Newman algorithm which used maximum betweeness to find partitions in a graph.

```
def connected_graph():
    G = nx.karate_club_graph()
    pos = nx.spring_layout(G)
    print("Node Degree")
    for v in G: print('%s %s' % (v, G.degree(v)))
    labels = {}

    for i in range(0, G.number_of_nodes()):
        labels[i] = i
        if i == 0:
            labels[i] = "Mr. Hi"
        elif i == 33:
            labels[i] = "John A."
    nx.draw_networkx(G, pos, with_labels=False)
    nx.draw_networkx_labels(G, pos, labels, font_size=16)
    plt.axis('off')
    plt.show()
```

```
def most_central_edge(G):
    centrality = nx.edge_betweenness_centrality(G, weight='weight')
    return max(centrality, key=centrality.get)


def disconnected_graph():
    G = nx.karate_club_graph()
    print("Node Degree")
    for v in G:
        print('%s %s' % (v, G.degree(v)))
    comp = girvan_newman(G,most_valuable_edge=most_central_edge)
    partition_dict = {}
    list_items = []
    for c in next(comp):
        print(c)
        list_items.append(list(c))
    for items in list_items:
        for i in items:
            partition_dict[i] = list_items.index(items)

    print(partition_dict)
```

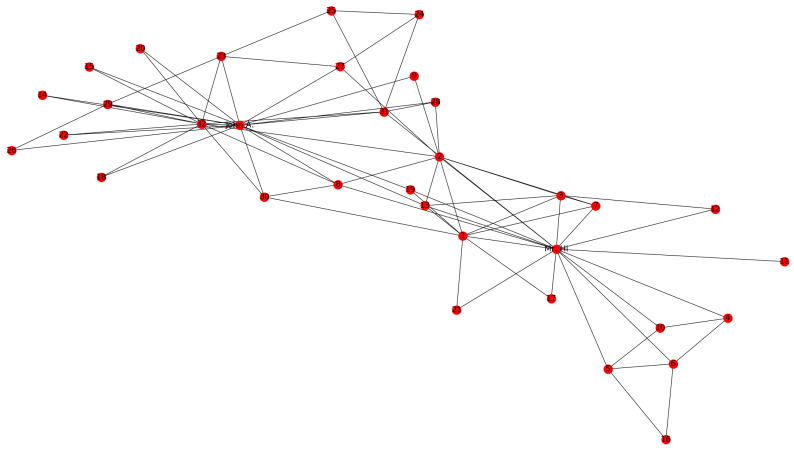

Figure 1: Connected Karate Club

```
    labels = {}

    for i in range(0, G.number_of_nodes()):
        labels[i] = i
        if i == 0:
            labels[i] = "Mr. Hi"
        elif i == 33:
            labels[i] = "John A."

    pos = nx.spring_layout(G)
    nx.draw_networkx_nodes(G, pos, node_size=600 ,cmap=plt.cm.RdYlBu, node_color=
        list(partition_dict.values()))
    nx.draw_networkx_edges(G, pos, alpha=0.3)
    nx.draw_networkx_labels(G, pos, labels,font_size=16)

    plt.axis('off')
    plt.show()
```

```
def multiple_iterations():
    G = nx.karate_club_graph()
    print("Node Degree")
    for v in G:
        print('%s %s' % (v, G.degree(v)))
    # comp = girvan_newman(G,most_valuable_edge=most_central_edge)
    comp = girvan_newman(G)
    comp1 = girvan_newman(G)
    comp2 = girvan_newman(G)
    list_items = []
```

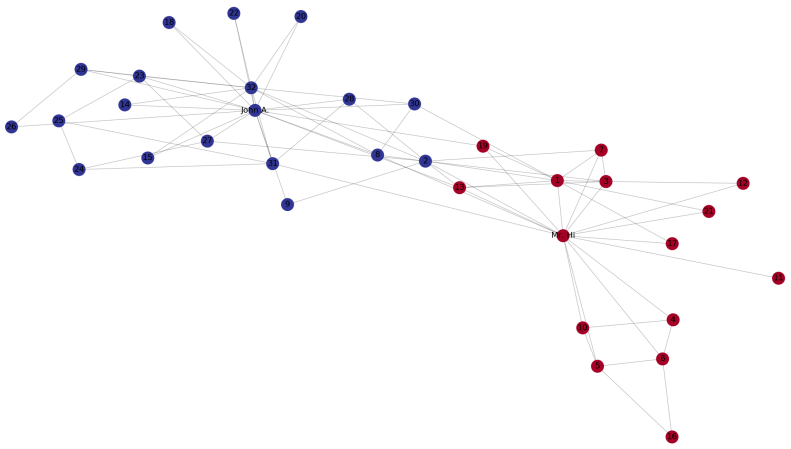

Figure 2: Karate Club graph showing affinity of nodes to each source, where blue nodes support John A. and red nodes support Mr. Hi

```
    for c in comp:
        list_temp = []
        for itr in c:
            list_temp.append(list(itr))
        list_items.append(list(list_temp))
    counter = 0
    for items in list_items:
        print("Iteration")
        partition_dict = {}
        for i in items:
            for j in i:
                partition_dict[j] = items.index(i)
        print(partition_dict)
        temp_G = copy.deepcopy(G)
        print(temp_G.edges())
        for key_i in partition_dict:
            for key_j in partition_dict:
                if temp_G.has_edge(key_i, key_j) and partition_dict[key_i] !=
                    partition_dict[key_j]:
                    temp_G.remove_edge(key_i, key_j)
        labels = {}
        print(temp_G.edges())
        for i in range(0, temp_G.number_of_nodes()):
            labels[i] = i
            if i == 0:
                labels[i] = "Mr. Hi"
            elif i == 33:
                labels[i] = "John A."
        if counter > 1:
```

```
            figure(figsize=(20, 20))
        else:
            figure(figsize=(15, 15))
        pos = nx.spring_layout(temp_G)
        nx.draw_networkx_nodes(temp_G, pos, node_size=600, cmap=plt.cm.RdYlBu,
            node_color=list(partition_dict.values()))
        nx.draw_networkx_edges(temp_G, pos, alpha=0.3)
        nx.draw_networkx_labels(temp_G, pos, labels, font_size=16)

        plt.axis('off')
        file_name = "Girvan_Newman" + str(counter) + ".png"
        plt.savefig(file_name)
        if counter < 4:
            counter = counter + 1
        else:
            exit()
```

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

If the group splits become more nuanced, the group split for 3 would look like the network graph in figure 4. Similarly for split into 4 incan be viewed in figure 5, split for 5 in figure 6 ,and split for 6 in figure 7.

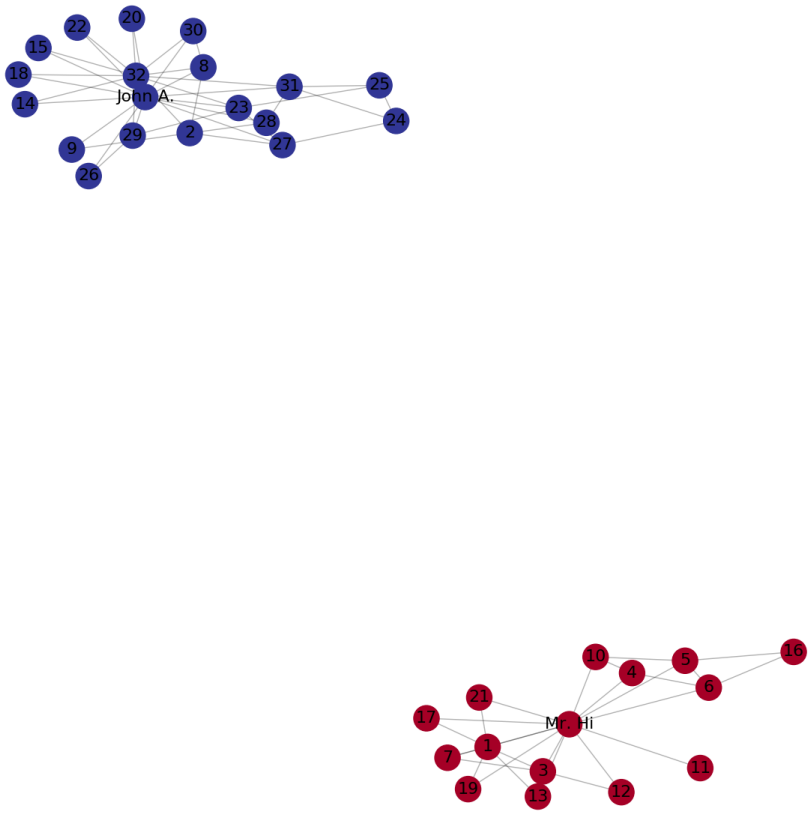

Figure 3: Two connected components from the Karate Club graph

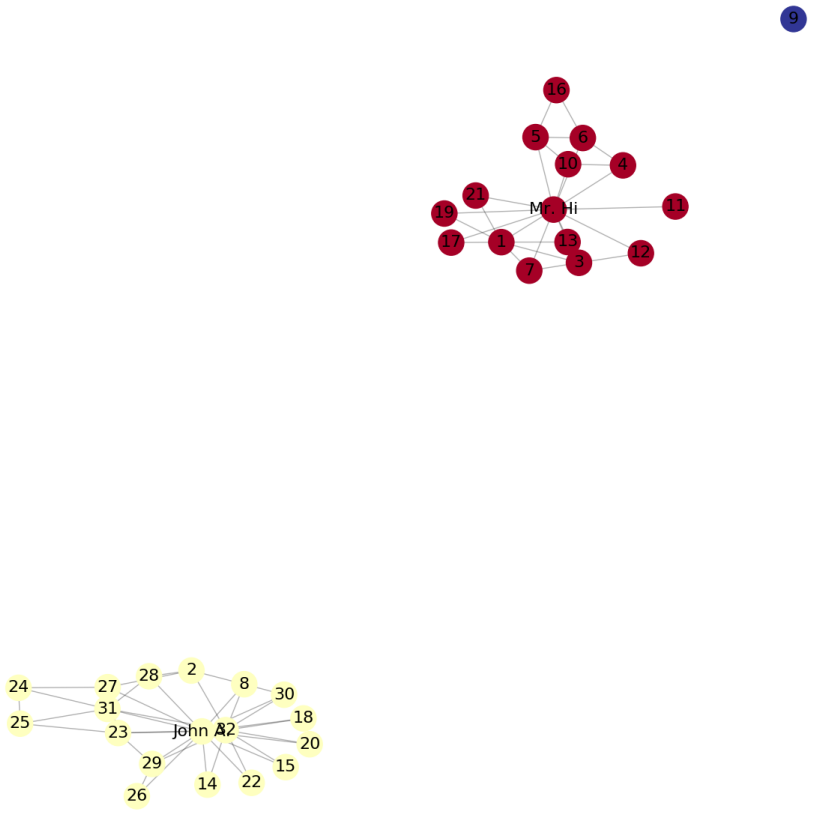

Figure 4: Karate Club split in 3 groups

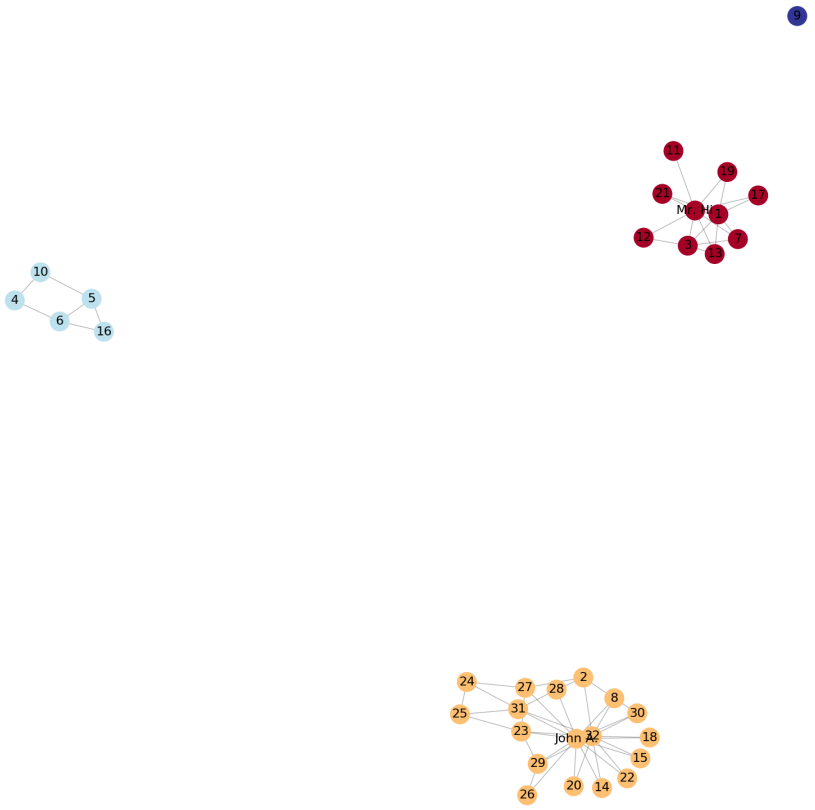

Figure 5: Karate Club split in 4 groups

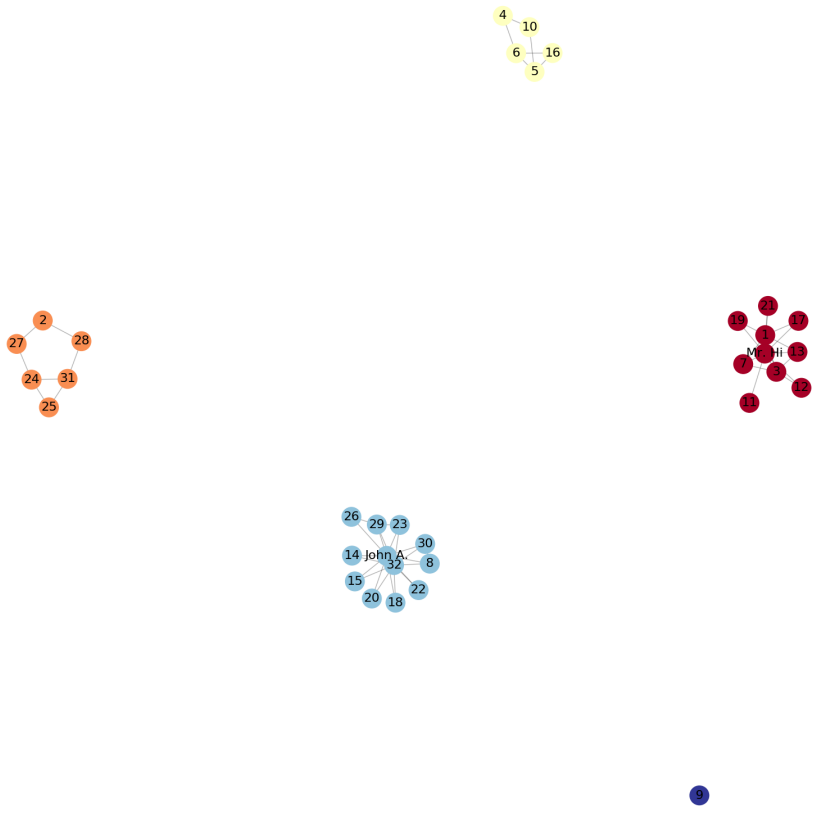

Figure 6: Karate Club split in 5 groups

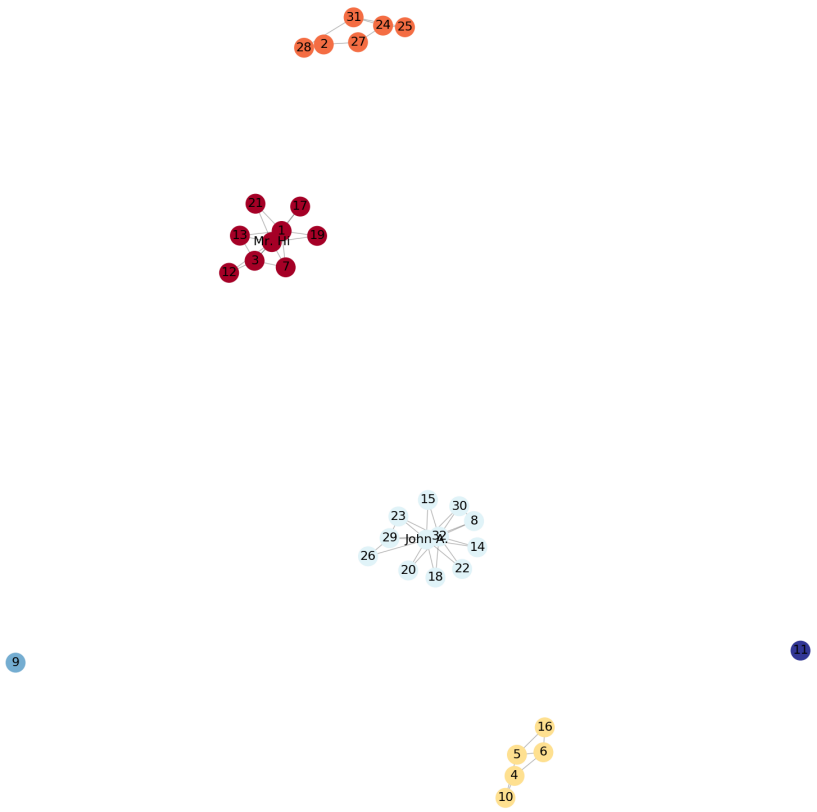

Figure 7: Karate Club split in 6 groups