Old Dominion Univeristy

CS 495: Introduction to Web Science
Instructor: Michael L. Nelson, Ph.D
Fall 2014 4:20pm - 7:10pm R, ECSB 2120

Assignment # 6

GeorgeC. Micros UIN: 00757376

Honor Pledge

I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagIiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violations of the Honor Code. I will report to a hearing if summoned.

Signed _____

October 30, 2014

George C. Micros

Written Assignment 6

Fall 2014
CS 495: Introduction to Web Science
Dr. Michael Nelson

October 30, 2014

Contents

Chapter 1
Written Assignment 6

1.1 Question 1

1.1.1 The Question

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.
Useful sources include:
* Original paper
http://aris.ss.uci.edu/~lin/76.pdf
* 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
http://networkx.github.io/documentation/latest/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

1.1.2 The Answer

The intuitive approach to determining communities that arise from the original graphs is to successively remove edges of high importance until two disconnected clusters are produced. This idea is very broad are requires the definition of a few concepts. Edges is high importance have large flow and connect large areas of the graph. By removing these edges parts of the graph that are not clustered become disconnected as they were using these main edges to remain connected. The idealized scenario is that of two clusters connected by the edge of two bridge nodes. In this situation the edge is service the path spanning across the two clusters and will have large importance in the structure of the group.

The idea of edge importance can be quantified with the definition of edge betweenness. Edge betweenness is the amount of “flow” and edge carries between all pairs of nodes where a signle unit of flow between two nodes divides itself evenly among all shortest paths between the nodes. The central idea of this definition is that flow can be represented by the number of shortest paths that span an edge. If an edge is used frequently in the shortest path between two nodes then it is important or central to the graph. Betweenness centrality is one measure of the centrality of a graph.

The divisive approach to community detection is centered on this notion of removing nodes or edges that are central to the graph structure and observing the resulting graph for patterns. This is confirmed if we consider what the characteristics we expect in the factions contained in a graphs. Factions can be thought is as cliques were nodes are highly connected to each other and therefore have a high clustering coefficient. By removing the more prominent edges that are used by highly used to connect sections of the graph we remove the connections with low clusting and the result should make the groups of high connectivity more obvious.

The Girvan-Newman algorithm is the quantification of these idea and can be summarized concisely in a few steps.

- Compute the betweenenss of all existing edges in the network
- Remove the edge with the highest betweenness

- Repeat this process until the desired number of distinct clusters has formed.

Following this algorithm leads to a result that is a fair apporximation of the actual division of the karate club. However, the result does contain some misclassifications.

More accurate techiques exist that attemp to optimize a given quanity. One such method is the leading eigenvector method. This method optimizes the modality of the graph, which is a measure of the clustering or density of modules of the graph. The modularity matrix is defined:

$$B_{vw} = A_{vw} - \frac{k_v k_w}{2m}$$

By decomposing this matrix into eigenvectors the main components of modularity can be determined. The largest eigenvector is used to split the graph repeatedly in the direction of highest modality. Using this method resulted in an accurate prediction of the actual data of the karate club.

```
#! /usr/bin/Rscript

library("igraph")
library("igraphdata")

graphics.off()

data(karate)
g <- karate
h <- karate
f <- karate
g<-set.vertex.attribute(g,"name",index=(2:33), (2:33))
colrs <- c("red", "blue")[get.vertex.attribute(g,"Faction")]
plot(g,vertex.color=colrs, vertex.size=25,layout=layout.fruchterman.reingold(g,niter=500,area=
    vcount(g)^2.3,repulserad=vcount(g)^2), main="Original")
repeat
{
  a <- edge.betweenness(g)
    g <- delete.edges(g,which.max(a))

    if(clusters(g)$no == 2)
    {
        #colrs <- c("red","blue")[get.vertex.attribute(g,"Faction")]
        colrs <- c("red", "blue")[clusters(g)$membership]
        h<-set.vertex.attribute(h,"name",index=(2:33), (2:33))
        dev.new()
        plot(h,vertex.color=colrs, vertex.size=25,layout=layout.fruchterman.reingold(g,niter=500,
            area=vcount(g)^2.3,repulserad=vcount(g)^2), main="Girvan-Newman")
        break;
    }
}

j <- leading.eigenvector.community(h, steps=1, weights=get.edge.attribute(h,"weight"))
f <- set.vertex.attribute(f,"Faction", index=(1:34), j$membership);
colrs <- c("red", "blue")[get.vertex.attribute(f, "Faction")]
dev.new()
plot(h,vertex.color=colrs, vertex.size=25,layout=layout.fruchterman.reingold(g,niter=500,area=
    vcount(g)^2.3,repulserad=vcount(g)^2), main="Leading Eigenvector - 2 class")

# 3 class

j <- leading.eigenvector.community(h, steps=3, weights=get.edge.attribute(h,"weight"))
f <- set.vertex.attribute(f,"Faction", index=(1:34), j$membership);
colrs <- c("red", "blue", "green")[get.vertex.attribute(f, "Faction")]
dev.new()
plot(h,vertex.color=colrs, vertex.size=25,layout=layout.fruchterman.reingold(g,niter=500,area=
    vcount(g)^2.3,repulserad=vcount(g)^2), main="Leading Eigenvector - 3 Class")
# 4 class

j <- leading.eigenvector.community(h, steps=4, weights=get.edge.attribute(h,"weight"))
f <- set.vertex.attribute(f,"Faction", index=(1:34), j$membership);
colrs <- c("red", "blue", "green", "yellow")[get.vertex.attribute(f, "Faction")]
dev.new()
plot(h,vertex.color=colrs, vertex.size=25,layout=layout.fruchterman.reingold(g,niter=500,area=
    vcount(g)^2.3,repulserad=vcount(g)^2), main="Leading Eigenvector - 4 Class")

# 5 class

j <- leading.eigenvector.community(h,  weights=get.edge.attribute(h,"weight"))
f <- set.vertex.attribute(f,"Faction", index=(1:34), j$membership);
colrs <- c("red", "blue", "green", "yellow", "pink")[get.vertex.attribute(f, "Faction")]
dev.new()
```

```
58 plot(h,vertex.color=colrs, vertex.size=25,layout=layout.fruchterman.reingold(g,niter=500,area=
       vcount(g)^2.3,repulserad=vcount(g)^2), main="Leading Eigenvector - 5 Class")
```

Listing 1: R code that performs that community detection

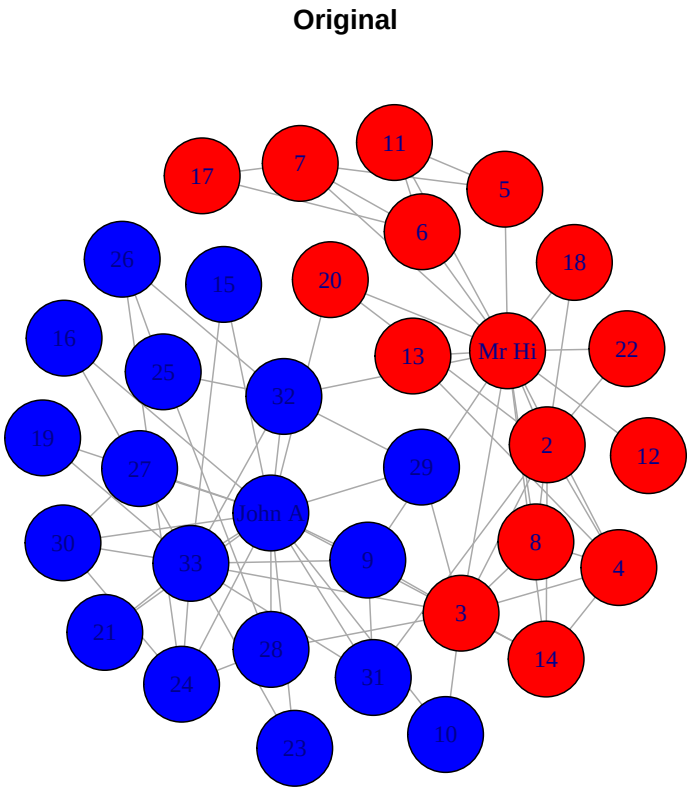

Fig. 1.1: Original partinioning based on data

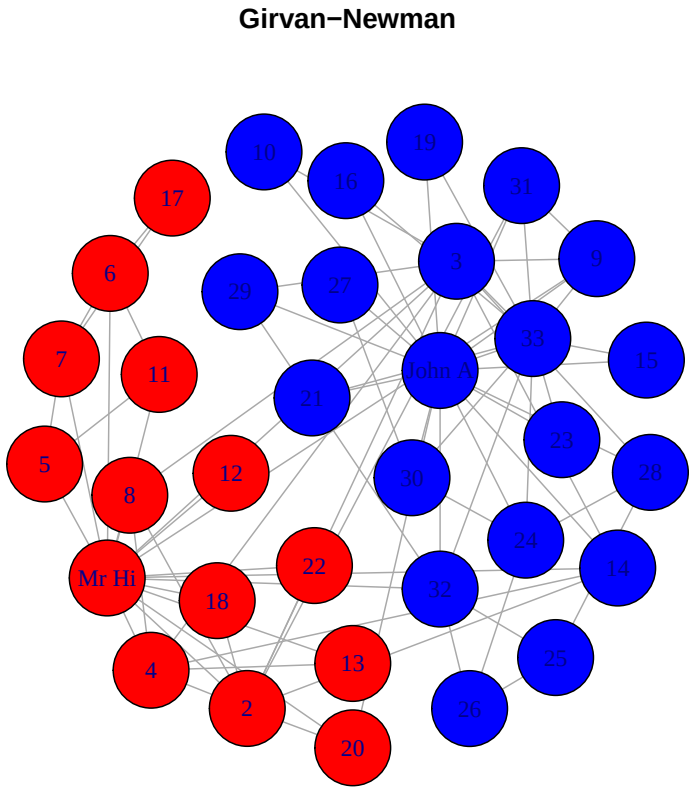

Fig. 1.2: Predicted partitioning based on Girvan-Newman

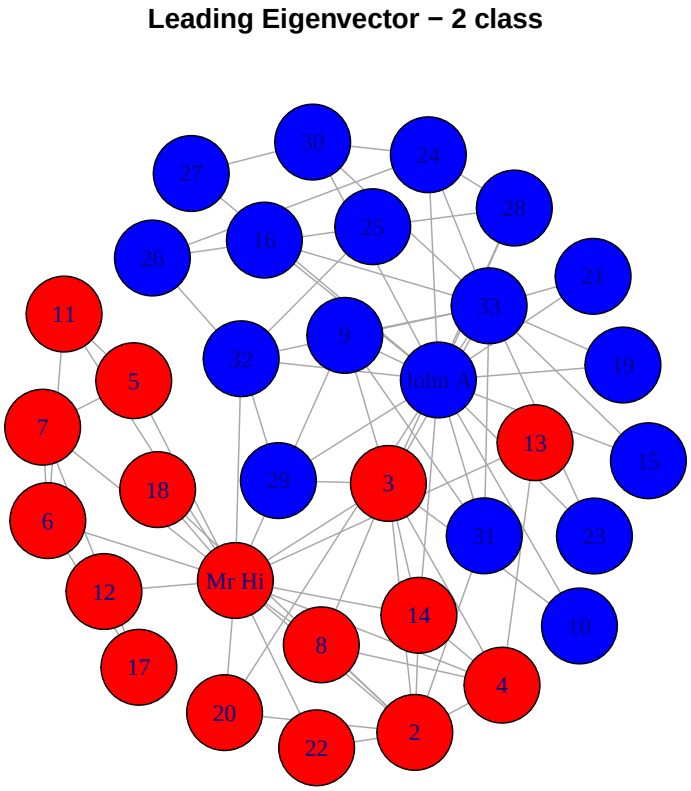

Fig. 1.3: Predicted partitioning based on Leading Eigenvector Method

1.2 Question 2

1.2.1 The Question

Using your facebook account, repeat question #1 (if you have > 50 friends).
Start at: `https://developers.facebook.com/docs/graph-api/reference/v2.1/user/friends`
or perhaps:
`http://socialnetimporter.codeplex.com/`

1.2.2 The Answer

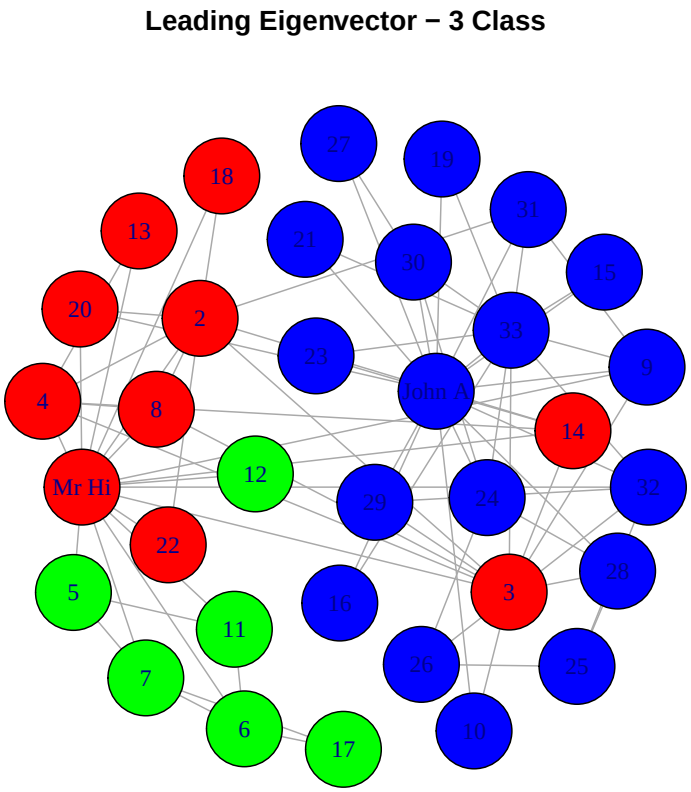

Fig. 1.4: Original partinioning based on data

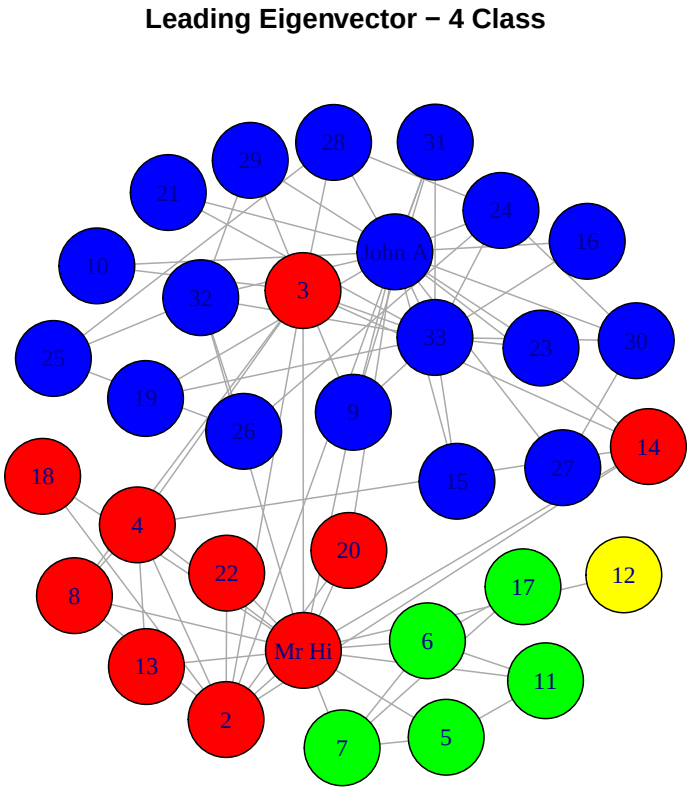

Fig. 1.5: Predicted partitioning based on Girvan-Newman

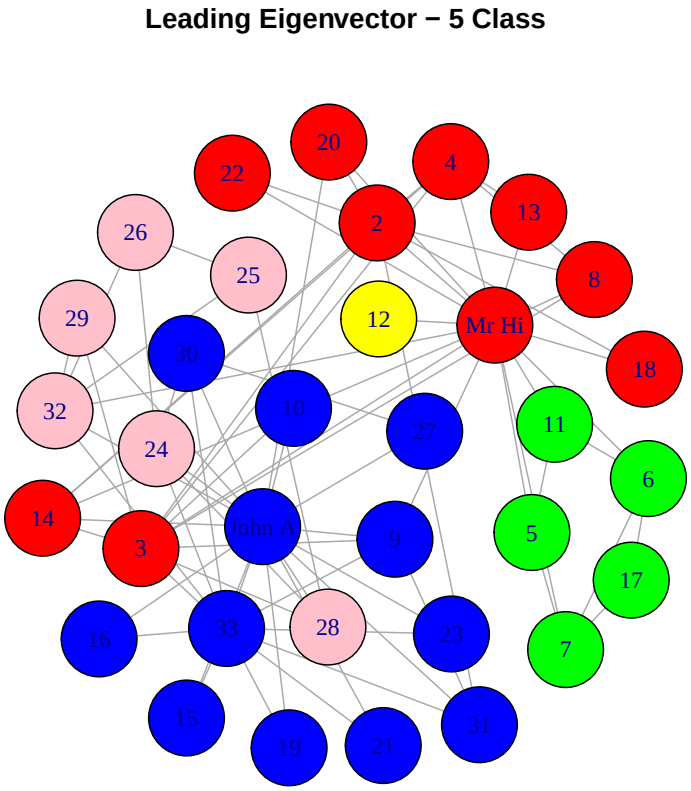

Fig. 1.6: Predicted partitioning based on Leading Eigenvector Method

References

1. Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." Proceedings of the National Academy of Sciences 99.12 (2002): 7821-7826.
2. https://stat.ethz.ch/pipermail/r-help//2012-August/333656.html
3. http://digitalpbk.com/perl/perl-script-check-google-pagerank
4. http://www.google.com
5. http://jakevdp.github.io/blog/2012/10/14/scipy-sparse-graph-module-word-ladders/
6. http://curl.haxx.se/docs/httpscripting.html
7. http://www.crummy.com/software/BeautifulSoup/bs4/doc/
8. http://www.rmi.net/~lutz/
9. http://www.cs.cornell.edu/home/kleinber/networks-book/
10. http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/
11. http://www.cs.odu.edu/~mklein/cs796/lecture/