

University of Tirana

Faculty of Economy

Departament of statistics and Applied Infomatics

Linear Programming

Course Work

Theme : Infection Network

Prepared by: Rigers Zaimi Group: IE\_English

* Introduction

In this project I am going to use Graph theory to analyze the risk of infection and its mitigation at a school, based on the data on how they interact with each other, using R language and basic graph theory concepts such as density, transitivity, and different types of centralities.

The network is undirected with weighted edges and includes 2 types of Nodes:  
**Student and Teachers**.

The dataset is available at the project folder edited from the original version from [vCIES2020](https://github.com/Arizonagong/vCIES2020_Network-Analysis). It includes a node list made of 243 with 2 attributes (Class, Gender) for each, and the edge list with 125774 connections with repetition.

**Software Used:**

* R for Windows 4.1.1
* Rtools 4.0
* RStduio – 2021.09.1- 372
* Packages Used:
* iGraph
* Cairo
* ggscater
* magrittr

Note: Using Notepad++, the R code is saved as an .HTML file and then pasted to the Word document so that it supports syntax highlighting. **=> All the images are hyperlinks to hi-res versions <=**

* **Importing of the Dataset:**

Both the nodes and edges data are read from .csv files using the default R libraries.

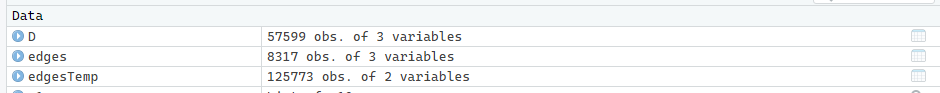
nodes **<-** read.csv**(**"nodeData.csv"**) #nodeData stored in nodes**

edgesTemp **<-** read.csv**(**"edgeData.csv"**) #Temporary holder for the edge data**

D **<-** as.data.frame**(**table**(**edgesTemp**)) #Edge data is passed as an array**

edges **<-** subset**(**D, Freq **>** 0**) #A new subset is formed where the frequency tells the weight of the edge**

The reduction in dataset size can be seen here. From 125773 objects to 8317. The extra variable dictates the frequency.



Text

Description automatically generated with medium confidenceCalling the head function (returns the first 6 elements for a vector, table, list, data frame etc.) for edgesTemp and edges shows us how the structure has changed from the above operation. For each repetitive edge between 2 elements its frequency is incremented by 1. Source dictated the starting node and target the ending one but since the graph is undirected the order won’t matter.

**The Freq variable will act as the weight of the edges.**

* **Creating the iGraph object**

g1 – This is the main igraph object that will be used during this project.

g1 **<-** graph.data.frame**(**edges, directed **=** **FALSE**,nodes**) #Important to specify #that this is an undirected network**

g1 #displays the first 21 edges

E**(**g1**)$**weight **<-** E**(**g1**)$**Freq

# The $Freq data of the edges is saved as $weight

# because many of the default centrality functions

# automatically scan for it as an argument

class**(**g1**)** # Checks the class of the object



**The igraph object is created successfully.**

To check is the weights are converted successfully we can call this function:

g1**[**c**(**1**:**15**)**,c**(**1**:**15**)]** # Returns a 15x15 sparse matrix with the corresponding weights for each edge

Text

Description automatically generated with low confidence

**Plotting the basic graph:**

By default, the normal graph generated by RStudio has a low basic resolution. Using the Cairo library, the IDE can generate a much higher quality render of the graph by using anti-aliasing which proves to be useful with many edges. The color of the edges is assigned through a for loop [function](file:///C:\Users\Rigers\Desktop\RProject\images\colorFunc.PNG)(1).

set.seed**(**222**)** #Using a constant seed makes it easier to analyze the graph

plot**(**g1,

edge.color **=** "black", #color of the edges

vertex.label **=** **NA**, #remove label names/id

edge.width **=** E**(**g1**)$**weight**/**300, #The width of the edge is based on the weight

layout **=** layout.fruchterman.reingold**(**g1**)**

#Force-directed graph drawing - As few crossing edges as possible, and all the edges are roughly the same size

**)**

classHolder **<-** c**(**"1A", "1B","2A","2B","3A", "3B", "4A", "4B","5A", "5B","Teacher"**)**

colorHolder **<-**c**(**"pink","yellow","red","green","brown","orange","purple","grey","yellowgreen","navajowhite1","blue"**)**

legend**(**x **=** "bottomleft", legend **=** classHolder, fill **=** colorHolder, title **=** "Classes:"**)** #legend for the graph

[Chart, scatter chart

Description automatically generated](file:///C:\Users\Rigers\Desktop\RProject\images\graph0.PNG)

There are 242 nodes in the dataset, starting from node with index 1:242 we assign a color**(1)** based on the Class attribute of each student. So that Students in the same ***$Class*** will have the same color.

The last else dictates blue for teachers since their $Class attribute is “Teachers”.

* **Density -** Density can be a good measure to predict hotspots where the infection can spread easily. The density of a graph is a measure of how many ties between nodes exist compared to how many ties between nodes are possible. The denser a network the easier the infection can spread.

edge\_density**(**g1**)** # Edge density of the entire dataset

# create a subgraph with only students from class 1A

A1 **<-** induced\_subgraph**(**g1, V**(**g1**)[**Class **==**"1A"**]**, impl **=** c**(**"auto"**))**

edge\_density**(**A1**)**

# create a subgraph with only students from class 2B

B2 **<-** induced\_subgraph**(**g1, V**(**g1**)[**Class **==**"2B"**]**, impl **=** c**(**"auto"**))**

edge\_density**(**B2**)**

Text

Description automatically generated

**Conclusion:** Edge density is much higher within members of the same class, so would the risk of infection. That would explain the groups that are visible in the graph. A clustering function can be implemented for better visualization of these groups with higher density.

#Clustering algo based on density

louvainCluster **<-** cluster\_louvain**(**g1, weights **=** E**(**g1**)$**weight**)**

communities**(**louvainCluster**)** # Gives a list of all communities

plot**(**louvainCluster,g1,…

[Diagram

Description automatically generated with medium confidence](file:///C:\Users\Rigers\Desktop\RProject\images\graph1.PNG)

Here are shown the communities formed by the clustering algorithm. There is an overlap of communities between parallel classes. In this case the **light blue community is a merging of Classes 2A and 2B.**

This happens because the Louvain algorithm evaluates how much more densely connected the nodes within a community are, compared to how connected they would be in a random network. Parallel classes are naturally more connected with each other than their seniors or juniors and will have a higher density.

A2andB2 **<-** induced\_subgraph**(**g1, V**(**g1**)[**Class **==**"2A" **|** Class **==** "2B"**]**, impl **=** c**(**"auto"**))**

edge\_density**(**A2andB2**)**

A2andB4 **<-** induced\_subgraph**(**g1, V**(**g1**)[**Class **==**"2A" **|** Class **==** "4B"**]**, impl **=** c**(**"auto"**))**

edge\_density**(**A1andB2**)**

A picture containing graphical user interface

Description automatically generated

**Here is shown that the density between parallel classes 2A and 2B (0.618) is higher than that of 2A and 4B (0.5149). Hence the result that they are grouped together.**

* **Transitivity**

The transitivity of a graph is based on the relative number of triangles in the graph, compared to total number of connected triples of nodes. Practically this will find the probability that a group of 3 nodes are connected each with the other.

# transitivity of the entire graph

trans **<-** transitivity**(**g1, type **=** "global", weights **=** E**(**g1**)$**weight**)**

trans

# transitivity of class 1A

trans1A **<-** transitivity**(**A1, type **=** "global", weights **=** E**(**g1**)$**weight**)**

trans1A

Graphical user interface, text

Description automatically generated with medium confidence

Shown by transitivity too the risk of infection inside one class is higher than that of the whole dataset.

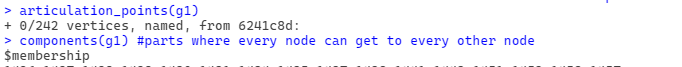
* **Articulation Points**

Articulation points are specific nodes that if removed would cause the number of connected components to increase by 1 thus breaking a connected graph into multiple parts. If such existed, its removal would have a positive effect in the lowering of the infection rate.

#These are nodes that if removed would increase the number of connected components in the system

articulation\_points**(**g1**)**

components**(**g1**)** #parts where every node can get to every other node



As seen here no such nodes exists. (0/242 vertices). There is only one component, and every node is part of it. The output of the ***components(g1)*** is given below. There is no node that if removed would add another component so the infection rate can’t be lowered this way.

**Conclusion**: The results from transitivity and density show that some restrictions should be made inside classed to lower the infection rate since their contact is much more frequent. Some attention should be also paid on the activity between parallel classes.

* **Degree Centrality – immediate risk of a student/teachers catching or spreading the infection**

The degree centrality is a good measure for individuals who are more likely to spread or perhaps be infected since it shows the number of edges that stem from each node (since the graph is undirected). The higher the degree the more in contact with other members of the school this individual is and that would contribute to a higher infectivity rate.

g1degree **<-** degree**(**g1**) # Find the degree for our graph**

V**(**g1**)$**degree **<-** g1degree **# Assigns the degree to vertexes**

sortg1degree **<-** sort**(**g1degree**) # Sorts the degrees in increasing order**

tail**(**sortg1degree**) # Prints the 6 biggest variables**

V**(**g1**)$**degree

[Scatter chart

Description automatically generated](images/degree.PNG)On the plotting function add: vertex.size **=** V**(**g1**)$**degree**/**15, **# Vertex size scales with degree score**

The nodes with the bigger degree score can be assumed as the most popular individuals and have contact with the most people. They can serve as better infection spreaders thus contributing more on the rate of the spread of the infection.

[The black](images/degree%20set%20color.PNG) nodes denote the 6 nodes with the highest degree. 2 are in class 1B, 1 in 1A and 3 on class 3B where the biggest degree centrality score is also located.

Table

Description automatically generated

* **Betweenness Centrality**

Betweenness centrality tells us the number of times a node acts as a bridge along the shortest path between other nodes. These nodes would make for the highest infections spreaders since they act as a middling so much.

#acts as a middling node in the shortest path

g1between **<-** betweenness**(**g1, directed **=** F**)**

V**(**g1**)$**betweenness **<-** g1between

sortg1between **<-** sort**(**g1between**)**

head**(**sortg1between)

tail**(**sortg1between**)**

V**(**g1**)$**betweenness

vertex.size **=** V**(**g1**)$**betweenness**/**40

Some nodes have a betweenness of 0 meaning that they don’t act as bridge in the shortest path between other students, they wouldn’t pose much risk of spreading the infection. The nodes with the highest betweenness value [(black)](images/Betweenness%20set%20color.PNG) show the students that act more often as a bridge to other students.

The teachers(blue) have a low betweenness level given by their small size on the graph.

**[A picture containing text, map, indoor

Description automatically generated](images/Betweenness.PNG)**

**A picture containing letter

Description automatically generated**

**Eigen Centrality – Not a good predictor**

Eigen Centrality measures the **transitive** influence of nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores. The centrality of each actor is proportional to the sum of the centralities of those actors to whom he or she is connected. It measures the influence of a node in the network (Prestige score).

g1eigen **<-** evcent**(**g1**)$**vector

V**(**g1**)$**Eigen **<-** g1eigen

sortEigen **<-** sort**(**g1eigen**)**

tail**(**sortEigen**)**

V**(**g1**)$**Eigen On the plotting function add: vertex.size **=** sqrt**(**V**(**g1**)$**Eigen**)\***10

[A close-up of a plant

Description automatically generated with low confidence](file:///C:\Users\Rigers\Desktop\RProject\images\Eigen.PNG)

Class 1B has the bigger eigen centrality. These nodes are themselves connected to other important nodes in the network. But why is this?

**Eigenvector Centrality isn’t a good predictor.** Its recursive definition of eigenvector centrality sometimes results in highlighting densely connected subsets of a network. **This would explain the high Eigen Centrality of class 1B that got further empathized with every iteration of the algorithm.**

Text

Description automatically generated with low confidence

meanEigen **<-** mean**(**V**(**g1**)$**Eigen**)** # find the mean of V(g1)$Eigen

meanEigen # show the mean

meanEigen1B **<-** mean**(**V**(**g1**)[**Class **==** "1B"**]$**Eigen**)** # find the mean of -;- for Class 1B

meanEigen1B # show the mean of 1B

meanEigennot1B **<-** mean**(**V**(**g1**)[**Class **!=** "1B"**]$**Eigen**)** # find the mean of -;- for not 1B

meanEigennot1B # show the mean

Below can be shown that the mean of $Eigen for class B1 is much higher than the $Eigen mean of notB1. This would explain the bigger clustering.

Graphical user interface, text, application

Description automatically generated with medium confidence

* **PageRank Centrality**

PageRank Centrality – Instead of Eigenvector centrality, Google’s pageRank variation can be used to find the most influential node in the network.

g1Pagerank **<-** page\_rank**(**

g1, vids **=** V**(**g1**)**,directed **=** F, damping **=** 0.85,

weights **=** E**(**g1**)$**weight**)** %>% unlist**()** #Pipe from magrittr,

V**(**g1**)$**pagerank **<-** g1Pagerank # Store the score as an attribute

sortPage **<-** sort**(**g1Pagerank**)** # Stores and sorts the results

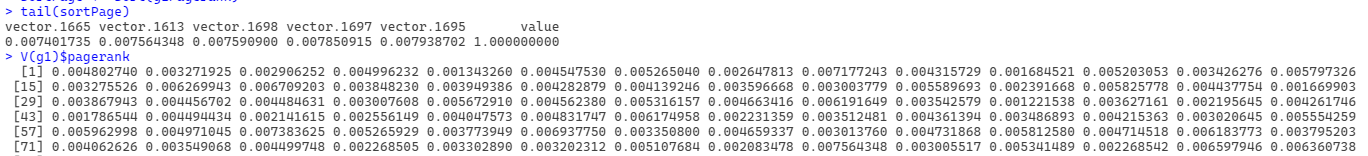
tail**(**sortPage**)** # Returns the 6 largest centrality values

V**(**g1**)$**pagerank # Returns all centrality values

On the plotting function add: vertex.size **=** V**(**g1**)$**pagerank %>% as.numeric**()** **\***1000,

[Map, scatter chart

Description automatically generated](file:///C:\Users\Rigers\Desktop\RProject\images\Pagerank.PNG)



The nodes returned by the tail function are the most influential in the network.

**The results from this graph are quite different from that of the normal eigenvector centrality where all the highest centrality nodes where members of the same class.**

**# Get the values of pageRank from index 1 to 242**

g1Pagerank **<-** g1Pagerank**[**1**:**242**]**

**# Place the centralities scores in a dataFrame**

corFrame **<-** data.frame**(**g1close,g1degree, g1between, g1eigen, g1Pagerank**)**

**# Find the correlation**

cor**(**corFrame**)# Returns the correlation matrix**



This matrix shows the results form analyzing the correlation between the different types of centralities. The eigenvector centrality can be shown to be the one with the lowest correlation with every other type of centrality.

References:

[R: The R Project for Statistical Computing](https://www.r-project.org/)

[RStudio | Open source & professional software for data science teams - RStudio](https://www.rstudio.com/)

[Using Rtools4 on Windows](https://cran.r-project.org/bin/windows/Rtools/rtools40.html)

[igraph R manual pages](https://igraph.org/r/doc/graph_from_data_frame.html)

[Infection in Social Networks: Using Network Analysis to Identify High-Risk Individuals | American Journal of Epidemiology | Oxford Academic (oup.com)](https://academic.oup.com/aje/article/162/10/1024/65086)

[(PDF) Network Centrality and Super-Spreaders in Infectious Disease Epidemiology (researchgate.net)](https://www.researchgate.net/publication/259077966_Network_Centrality_and_Super-Spreaders_in_Infectious_Disease_Epidemiology)

[PageRank centrality & EigenCentrality - Cambridge Intelligence (cambridge-intelligence.com)](https://cambridge-intelligence.com/eigencentrality-pagerank/)

[Eigenvector Centrality - Neo4j Graph Data Science](https://neo4j.com/docs/graph-data-science/current/algorithms/eigenvector-centrality/#:~:text=Eigenvector%20Centrality%20is%20an%20algorithm,who%20themselves%20have%20high%20scores.)

[tsiR: An R package for time-series Susceptible-Infected-Recovered models of epidemics (plos.org)](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0185528)

[Infection in Social Networks: Using Network Analysis to Identify High-Risk Individuals | American Journal of Epidemiology | Oxford Academic (oup.com)](https://academic.oup.com/aje/article/162/10/1024/65086)

[Add titles to a plot in R software - Easy Guides - Wiki - STHDA](http://www.sthda.com/english/wiki/add-titles-to-a-plot-in-r-software)

[Pagerank Algorithm To Analyze Packages in R Using Puzzle (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2015/04/application-pagerank-algorithm/)

[Definition of the transitivity of a graph - Math Insight](https://mathinsight.org/definition/transitivity_graph)

[Transitivity in a Graph | The Geography of Transport Systems (transportgeography.org)](https://transportgeography.org/contents/methods/graph-theory-measures-indices/transitivity-graph/)

[2.9 Density | Social Networks: An Introduction (bookdown.org)](https://bookdown.org/omarlizardo/_main/2-9-density.html?fbclid=IwAR2kHox8-cW8kMONVlEyIs3oxAkdWiONwr-DYZWdTmHcLGTrSTCz3djuc9A)

[Networks -> Centrality -> Degree (analytictech.com)](http://www.analytictech.com/ucinet/help/3ava_zr.htm)

[subset function - RDocumentation](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/subset)

[Force-directed graph drawing - Wikipedia](https://en.wikipedia.org/wiki/Force-directed_graph_drawing#Advantages)

[r - Which layout should I use to get non-overlapping edges in igraph? - Stack Overflow](https://stackoverflow.com/questions/13961913/which-layout-should-i-use-to-get-non-overlapping-edges-in-igraph)

[Betweenness Centrality - Neo4j Graph Data Science](https://neo4j.com/docs/graph-data-science/current/algorithms/betweenness-centrality/)

[Centrality - Wikipedia](https://en.wikipedia.org/wiki/Centrality#PageRank_centrality)

[Closeness Centrality - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/computer-science/closeness-centrality)