Social Network Analysis: Centrality Measures

Donglei Du
(ddu@unb.ca)

Faculty of Business Administration, University of New Brunswick, NB Canada Fredericton
E3B 9Y2

Table of contents

What is centrality? I

- Centrality measures address the question:
  "Who is the most important or central person in this network?"
- There are many answers to this question, depending on what we mean by importance.
- According to Scott Adams, the power a person holds in the organization is inversely proportional to the number of keys on his keyring.
  - A janitor has keys to every office, and no power.
  - The CEO does not need a key: people always open the door for him.
- There are a vast number of different centrality measures that have been proposed over the years.

What is centrality? II

- According to Freeman in 1979, and evidently still true today: "There is certainly no unanimity on exactly what centrality is or on its conceptual foundations, and there is little agreement on the proper procedure for its measurement."
- We will look at some popular ones...

Centrality measures

- Degree centrality
- Closeness centrality
- Betweeneness centrality
- Eigenvector centrality
- PageRank centrality
- ...

Degree centrality for undirected graph I

- The nodes with higher degree is more central.
- Let $A \in \mathbb{R}^{n\times n}$ be the adjacency matrix of a undirected graph. Let $k \in \mathbb{R}^n$ be the degree vector. Let $e \in \mathbb{R}^n$ be the all-one vector. Then

$$k = Ae$$

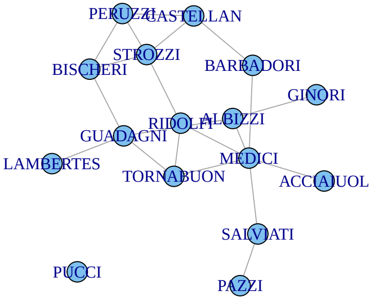
  - For comparison purpose, we can standardize the degree by dividing by the maximum possible value $n-1$.
  - Degree is simply the number of nodes at distance one.
- Though simple, degree is often a highly effective measure of the influence or importance of a node:
  - In many social settings people with more connections tend to have more power and more visible.

Degree centrality for undirected graph II

- Group-level centralization: degree, as an individual-level centrality measure, has a distribution which can be summarized by its mean and variance as is commonly practiced in data analysis.

An example: The Padgett Florentine families: Business network

```
rm(list = ls()) # clear memory
library(igraph) # load packages
load("./R code/padgett.RData") # load data
plot(padgett$PADGB) # plot the business graph
```



An example: the Padgett Florentine families:Marriage network

```
plot(padgett$PADGM) # plot the marriage graph
```

An example: Degree centrality for the Padgett Florentine families: business netowrk

```
# calculate the degree centrality for business network
deg_B <- degree(padgett$PADGB, loops = FALSE)
sort(deg_B, decreasing = TRUE)

##    MEDICI  GUADAGNI   STROZZI   ALBIZZI  BISCHERI CASTELLAN   PERUZZI
##         6         4         4         3         3         3         3
##   RIDOLFI TORNABUON BARBADORI  SALVIATI ACCIAIUOL    GINORI LAMBERTES
##         3         3         2         2         1         1         1
##     PAZZI     PUCCI
##         1         0

# calculate the standardized degree centrality
deg_B_S <- degree(padgett$PADGB, loops = FALSE)/(vcount(padgett$PADGM) - 1)
sort(deg_B_S, decreasing = TRUE)

##    MEDICI  GUADAGNI   STROZZI   ALBIZZI  BISCHERI CASTELLAN   PERUZZI
##   0.40000   0.26667   0.26667   0.20000   0.20000   0.20000   0.20000
##   RIDOLFI TORNABUON BARBADORI  SALVIATI ACCIAIUOL    GINORI LAMBERTES
##   0.20000   0.20000   0.13333   0.13333   0.06667   0.06667   0.06667
##     PAZZI     PUCCI
##   0.06667   0.00000
```

An example: Degree centrality for the Padgett Florentine families: marriage network

```
# calculate the degree centrality for business network
deg_M <- degree(padgett$PADGM, loops = FALSE)
sort(deg_M, decreasing = TRUE)

##    MEDICI BARBADORI LAMBERTES   PERUZZI  BISCHERI CASTELLAN    GINORI
##         5         4         4         4         3         3         2
##  GUADAGNI     PAZZI  SALVIATI TORNABUON ACCIAIUOL   ALBIZZI     PUCCI
##         2         1         1         1         0         0         0
##   RIDOLFI   STROZZI
##         0         0

# calculate the standardized degree centrality
deg_M_S <- degree(padgett$PADGM, loops = FALSE)/(vcount(padgett$PADGB) - 1)
sort(deg_M_S, decreasing = TRUE)

##    MEDICI BARBADORI LAMBERTES   PERUZZI  BISCHERI CASTELLAN    GINORI
##   0.33333   0.26667   0.26667   0.26667   0.20000   0.20000   0.13333
##  GUADAGNI     PAZZI  SALVIATI TORNABUON ACCIAIUOL   ALBIZZI     PUCCI
##   0.13333   0.06667   0.06667   0.06667   0.00000   0.00000   0.00000
##   RIDOLFI   STROZZI
##   0.00000   0.00000
```

Outdegree centrality and indegree prestige for digraph I

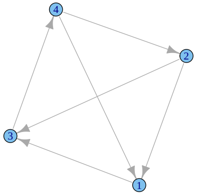
- The nodes with higher outdegree is more central (choices made).
- The nodes with higher indegree is more prestigious (choices received).
- Let $A \in \mathbb{R}^{n \times n}$ be the adjacency matrix of a directed graph. Let $k^{in}, k^{out} \in \mathbb{R}^n$ be the indegree and outdegree vectors respectively. Let $e \in \mathbb{R}^n$ be the all-one vector. Then

$$\begin{aligned} k^{out} &= A^T e \quad \text{(column sum of } A\text{);} \\ k^{in} &= Ae \quad \text{(row sum of } A\text{).} \end{aligned}$$

- **Note**: The adjacency matrix in directed graph has the counter-intuitive convention where $A_{ij} = 1$ iff there is a link from j to i.

An example

```r
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(0, 1, 0, 1,
    0, 0, 0, 1,
    1, 1, 0, 0,
    0, 0, 1, 0), # the data elements
  nrow=4, # number of rows
  ncol=4, # number of columns
  byrow = TRUE)# fill matrix by rows
g <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
degree(g, mode='in')

## [1] 2 1 2 1

degree(g, mode='out')

## [1] 1 2 1 2

plot(g) # plot the graph
```



Closeness centrality for undirected graph

- The farness/peripherality of a node v is defined as the sum of its distances to all other nodes
- The closeness is defined as the inverse of the farness.

$$closeness(v) = \frac{1}{\sum_{i \neq v} d_{vi}}$$

  - For comparison purpose, we can standardize the closeness by dividing by the maximum possible value $1/(n-1)$
  - If there is no (directed) path between vertex v and i then the total number of vertices is used in the formula instead of the path length.
- The more central a node is, the lower its total distance to all other nodes.
- Closeness can be regarded as a measure of how long it will take to spread information from v to all other nodes sequentially.

Example: Closeness centrality for the Padgett Florentine families

```r
rm(list = ls()) # clear memory
library(igraph) # load packages
load("./R code/padgett.RData") # load data
# calculate the closeness centrality
sort(closeness(padgett$PADGB), decreasing = TRUE)

##    MEDICI   RIDOLFI   ALBIZZI TORNABUON  GUADAGNI BARBADORI   STROZZI
##  0.024390  0.022727  0.022222  0.022222  0.021739  0.020833  0.020833
##  BISCHERI CASTELLAN  SALVIATI ACCIAIUOL   PERUZZI    GINORI LAMBERTES
##  0.019608  0.019231  0.019231  0.018519  0.018519  0.017241  0.016949
##     PAZZI     PUCCI
##  0.015385  0.004167

# calculate the standardized closeness centrality
close_B_S <- closeness(padgett$PADGB) * (vcount(padgett$PADGB) - 1)
sort(close_B_S, decreasing = TRUE)

##    MEDICI   RIDOLFI   ALBIZZI TORNABUON  GUADAGNI BARBADORI   STROZZI
##    0.3659    0.3409    0.3333    0.3333    0.3261    0.3125    0.3125
##  BISCHERI CASTELLAN  SALVIATI ACCIAIUOL   PERUZZI    GINORI LAMBERTES
##    0.2941    0.2885    0.2885    0.2778    0.2778    0.2586    0.2542
##     PAZZI     PUCCI
##    0.2308    0.0625
```

Betweenness centrality

- Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.
- It was introduced as a measure for quantifying the control of a human on the communication between other humans in a social network by Linton Freeman.
- In this conception, vertices that have a high probability to occur on a randomly chosen shortest path between two randomly chosen vertices have a high betweenness.

Betweenness centrality I

- The betweenness of a vertex v in a graph $G := (V, E)$ with V vertices is computed as follows:
  - For each pair of vertices (s, t), compute the shortest paths between them.
  - For each pair of vertices (s, t), determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).
  - Sum this fraction over all pairs of vertices (s, t).
- More compactly the betweenness can be represented as:

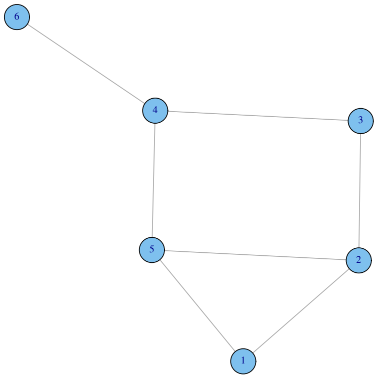
$$Betwenness(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

  - where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

Betweenness centrality II

- The betweenness may be normalized by dividing through the number of pairs of vertices not including v, which for directed graphs is $(n-1)(n-2)$ and for undirected graphs is $(n-1)(n-2)/2$.

An example I



- The node betweenness for the graph on the left:

Node	Betwenness
1	0
2	1.5
3	1
4	4
5	3
6	0

How to find the betweeness in the example?

- For example: for node 2, the $(n-1)(n-2)/2 = 5(5-1)/2 = 10$ terms in the summation in the order of 13, 14, 15, 16, 34, 35, 36, 45, 46, 56 are

$$\frac{1}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{1}{2} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} = 1.5.$$

- Here the denominators are the number of shortest paths between pair of edges in the above order and the numerators are the number of shortest paths passing through edge 2 between pair of edges in the above order.

Betweenness centrality for the Padgett Florentine families

```
rm(list = ls()) # clear memory
library(igraph) # load packages
load("./R code/padgett.RData") # load data
# calculate the betweenness centrality
sort(betweenness(padgett$PADGB), decreasing = TRUE)

##    MEDICI  GUADAGNI   ALBIZZI  SALVIATI   RIDOLFI  BISCHERI   STROZZI
##    47.500    23.167    19.333    13.000    10.333     9.500     9.333
## BARBADORI TORNABUON CASTELLAN   PERUZZI ACCIAIUOL    GINORI LAMBERTES
##     8.500     8.333     5.000     2.000     0.000     0.000     0.000
##     PAZZI     PUCCI
##     0.000     0.000

# calculate the standardized Betweenness centrality
betw_B_S <- 2*betweenness(padgett$PADGB)/((vcount(padgett$PADGB) - 1)*(vcount(padgett$PADGB)
sort(betw_B_S, decreasing = TRUE)

##    MEDICI  GUADAGNI   ALBIZZI  SALVIATI   RIDOLFI  BISCHERI   STROZZI
##   0.45238   0.22063   0.18413   0.12381   0.09841   0.09048   0.08889
## BARBADORI TORNABUON CASTELLAN   PERUZZI ACCIAIUOL    GINORI LAMBERTES
##   0.08095   0.07937   0.04762   0.01905   0.00000   0.00000   0.00000
##     PAZZI     PUCCI
##   0.00000   0.00000
```

Eigenvector centrality for undirected graph I

- Let $\mathbf{x}$ be eigenvector of the largest eigenvalue λ of the non-negative adjacency matrix A of the undirected graph $G = (V, E)$.
- The eigenvector centrality of node i is equal to the leading eigenvector x_i of (column) stochastic matrix $N := AD^{-1}$ (whose leading eigenvalue is 1):

$$N\mathbf{x} = \mathbf{x}$$

- Consider a particular node i with its neighboring nodes $N(i)$:

$$x_i = \sum_{j \in N(i)} x_j = \sum_j A_{ij} x_j$$

Eigenvector centrality for undirected graph II

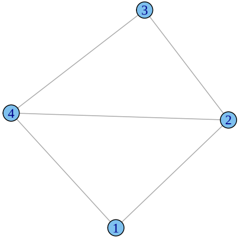
- The eigenvector centrality defined in this way depends both on the number of neighbors $|N(i)|$ and the quality of its connections $x_j,\ j \in N(i)$.

Why the leading eigenvector?

- Suppose we want to choose an eigenvector $\mathbf{x}$ to define a centrality measure, then a necessary condition is $x \in \mathbb{R}_n^+$.
- For non-negative matrix, the leading eigenvector is non-negative (see ▸ Appendix A (Slide 68) for background information on non-negative, irreducible and primitive matrices).

A toy example

```r
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(0, 1, 0, 1,
    1, 0, 1, 1,
    0, 1, 0, 1,
    1, 1, 1, 0), # the data elements
  nrow=4, # number of rows
  ncol=4, # number of columns
  byrow = TRUE)# fill matrix by rows
g <- graph.adjacency(adj, mode="undirected") # create igrpah object from adjacency matrix
plot(g) # plot the graph
```



A toy example

```
D <- diag(1/degree(g), 4) #degree diagonal matrix
D

##      [,1]   [,2] [,3]   [,4]
## [1,]  0.5 0.0000  0.0 0.0000
## [2,]  0.0 0.3333  0.0 0.0000
## [3,]  0.0 0.0000  0.5 0.0000
## [4,]  0.0 0.0000  0.0 0.3333

N <- adj %*% D # PageRank matrix
N

##      [,1]   [,2] [,3]   [,4]
## [1,]  0.0 0.3333  0.0 0.3333
## [2,]  0.5 0.0000  0.5 0.3333
## [3,]  0.0 0.3333  0.0 0.3333
## [4,]  0.5 0.3333  0.5 0.0000

y <- eigen(N) # find the eigenvalues and eigenvectors
y$val # the eigenvalues

## [1]  1.000e+00 -6.667e-01 -3.333e-01  3.088e-17

y$vec # the eigenvectors

##         [,1] [,2]       [,3]       [,4]
## [1,] -0.3922 -0.5 -1.233e-32 -7.071e-01
## [2,] -0.5883  0.5 -7.071e-01  1.091e-16
## [3,] -0.3922 -0.5  0.000e+00  7.071e-01
## [4,] -0.5883  0.5  7.071e-01  7.544e-17
```

Eigenvector centrality for the Padgett Florentine families

```
rm(list = ls()) # clear memory
library(igraph) # load packages
load("./R code/padgett.RData") # load data
# calculate the degree centrality
sort(evcent(padgett$PADGB)[[1]], decreasing = TRUE)

##    MEDICI   STROZZI   RIDOLFI TORNABUON  GUADAGNI  BISCHERI   PERUZZI
## 1.000e+00 8.273e-01 7.937e-01 7.572e-01 6.719e-01 6.572e-01 6.408e-01
## CASTELLAN   ALBIZZI BARBADORI  SALVIATI ACCIAIUOL LAMBERTES    GINORI
## 6.020e-01 5.669e-01 4.920e-01 3.391e-01 3.071e-01 2.063e-01 1.741e-01
##     PAZZI     PUCCI
## 1.041e-01 6.191e-17

sort(evcent(padgett$PADGM)[[1]], decreasing = TRUE)

##   PERUZZI LAMBERTES CASTELLAN BARBADORI  BISCHERI    MEDICI  GUADAGNI
## 1.000e+00 9.236e-01 8.305e-01 8.290e-01 7.311e-01 5.121e-01 4.993e-01
##    GINORI TORNABUON     PAZZI  SALVIATI ACCIAIUOL   ALBIZZI     PUCCI
## 4.046e-01 1.545e-01 1.545e-01 1.545e-01 2.354e-17 2.354e-17 2.354e-17
##   RIDOLFI   STROZZI
## 2.354e-17 2.354e-17
```

PageRank centrality I

- Google's PageRank is a variant of the Eigenvector centrality measure for directed network.
- Basic PageRank.
  - Whenever a node i has no outgoing link, we addd a self loop to i such that $k_i^{in} = k_i^{out} = 1$. Therefore $A_{ii} = 1$ for such nodes in the adjacency matrix.
  - Let D be the diagonal matrix of outdegrees where each element $D_{ii} = k_i$
  - Define a column stochastic matrix

$$\boxed{N = AD^{-1}}$$

  - The PageRank centrality of node i is equal to the leading eigenvector x_i of matrix N (The leading eigenvalue is 1):

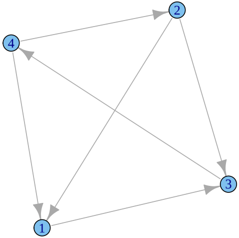
$$x = Nx$$

PageRank centrality II

- **Note**: The adjacency matrix in directed graph has the counter-intuitive convention where $A_{ij} = 1$ iff there is a link from j to i.

A toy example for the basic PageRank

```
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(0, 1, 0, 1,
    0, 0, 0, 1,
    1, 1, 0, 0,
    0, 0, 1, 0), # the data elements
  nrow=4, # number of rows
  ncol=4, # number of columns
  byrow = TRUE)# fill matrix by rows
g <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g) # plot the graph
```



A toy example for the basic PageRank

```
D <- diag(1/pmax(degree(g, mode = "out"), 1), 4) #degree diagonal matrix
D

##      [,1] [,2] [,3] [,4]
## [1,]    1  0.0    0  0.0
## [2,]    0  0.5    0  0.0
## [3,]    0  0.0    1  0.0
## [4,]    0  0.0    0  0.5

N <- adj %*% D # PageRank matrix
N

##      [,1] [,2] [,3] [,4]
## [1,]    0  0.5    0  0.5
## [2,]    0  0.0    0  0.5
## [3,]    1  0.5    0  0.0
## [4,]    0  0.0    1  0.0

y <- eigen(N) # find the eigenvalues and eigenvectors
y$val # the eigenvalues

## [1]  1.0000+0.0000i -0.3403+0.8166i -0.3403-0.8166i -0.3194+0.0000i

y$vec # the eigenvectors

##           [,1]            [,2]            [,3]       [,4]
## [1,] 0.4472+0i -0.2864-0.1910i -0.2864+0.1910i  0.4249+0i
## [2,] 0.2981+0i -0.1408-0.3378i -0.1408+0.3378i -0.7518+0i
## [3,] 0.5963+0i -0.2204+0.5288i -0.2204-0.5288i -0.1534+0i
## [4,] 0.5963+0i  0.6476+0.0000i  0.6476+0.0000i  0.4803+0i
```

Scaling PageRank centrality

- The scaling PageRank
  - Construct the positive linear combination

$$\boxed{M = \alpha N + \frac{1-\alpha}{n} ee^T}$$

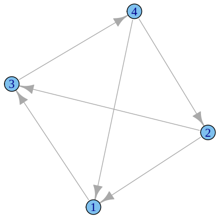
  - The Scaling PageRank centrality of node i is equal to the leading eigenvector x_i of matrix M:

$$x = Mx$$

- **Note**: The adjacency matrix in directed graph has the counter-intuitive convention where $A_{ij} = 1$ iff there is a link from j to i.

A toy example for the scaling PageRank with damping factor $\alpha = 0.85$

```r
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(0, 1, 0, 1,
    0, 0, 0, 1,
    1, 1, 0, 0,
    0, 0, 1, 0), # the data elements
  nrow=4, # number of rows
  ncol=4, # number of columns
  byrow = TRUE)# fill matrix by rows
g <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g) # plot the graph
```

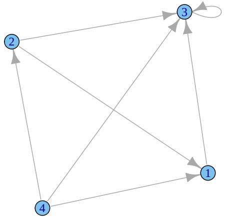


A toy example for the scaling PageRank with damping factor $\alpha = 0.85$

```
D <- diag(1/pmax(degree(g, mode = "out"), 1), 4) #degree diagonal matrix
D

##      [,1] [,2] [,3] [,4]
## [1,]    1  0.0    0  0.0
## [2,]    0  0.5    0  0.0
## [3,]    0  0.0    1  0.0
## [4,]    0  0.0    0  0.5

N <- adj %*% D # PageRank matrix
N

##      [,1] [,2] [,3] [,4]
## [1,]    0  0.5    0  0.5
## [2,]    0  0.0    0  0.5
## [3,]    1  0.5    0  0.0
## [4,]    0  0.0    1  0.0

Eye <- matrix(rep(1, 16), nrow = 4, ncol = 4, byrow = TRUE) # create a 4x4 all-one matrix
alpha <- 0.85 # damping factor
M <- alpha * N + (1 - alpha) * Eye/4
y <- eigen(M) # find the eigenvalues and eigenvectors
y$val # the eigenvalues

## [1]  1.0000+0.0000i -0.2892+0.6941i -0.2892-0.6941i -0.2715+0.0000i

y$vec # the eigenvectors

##           [,1]            [,2]            [,3]       [,4]
## [1,] 0.4552+0i -0.2864-0.1910i -0.2864+0.1910i  0.4249+0i
## [2,] 0.3194+0i -0.1408-0.3378i -0.1408+0.3378i -0.7518+0i
## [3,] 0.5958+0i -0.2204+0.5288i -0.2204-0.5288i -0.1534+0i
## [4,] 0.5795+0i  0.6476+0.0000i  0.6476+0.0000i  0.4803+0i
```

Why scaling? if you run the basic PageRank for this modified example...

```
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(0, 1, 0, 1,
    0, 0, 0, 1,
    1, 1, 1, 1,
    0, 0, 0, 0), # the data elements
  nrow=4, # number of rows
  ncol=4, # number of columns
  byrow = TRUE)# fill matrix by rows
g <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g) # plot the graph
```



Why scaling? if you run the basic PageRank for this modified example...

```
D <- diag(1/pmax(degree(g, mode = "out"), 1), 4) #degree diagonal matrix
D

##      [,1] [,2] [,3]   [,4]
## [1,]    1  0.0    0 0.0000
## [2,]    0  0.5    0 0.0000
## [3,]    0  0.0    1 0.0000
## [4,]    0  0.0    0 0.3333

N <- adj %*% D # PageRank matrix
N

##      [,1] [,2] [,3]   [,4]
## [1,]    0  0.5    0 0.3333
## [2,]    0  0.0    0 0.3333
## [3,]    1  0.5    1 0.3333
## [4,]    0  0.0    0 0.0000

y <- eigen(N) # find the eigenvalues and eigenvectors
y$val # the eigenvalues

## [1] 1 0 0 0

y$vec # the eigenvectors

##      [,1]    [,2]       [,3]    [,4]
## [1,]    0  0.7071 -7.071e-01  0.7071
## [2,]    0  0.0000  5.669e-292  0.0000
## [3,]    1 -0.7071  7.071e-01 -0.7071
## [4,]    0  0.0000  0.000e+00  0.0000
```

Leaking problem due to reducibility I

- Note that the previosu example shows that Node 3 gets all weights!
- The problem comes from the structure of the graph: it is not strongly connected, implying that N is reducible.
- The Perron-Frobenius theorem offers a way to gurantee a positive leading eignevector (see Appendix A (Slide 68)).
- Therefore we should try to revise N to generate a new matrix which is regular (or more stongly positive).
  - The scaling PageRank matirx $M > 0$.

Now, run the scaling PageRank for this modified example...

```
Eye <- matrix(rep(1, 16), nrow = 4, ncol = 4, byrow = TRUE) # create a 4x4 all-one matrix
alpha <- 0.85 # damping factor
M <- alpha * N + (1 - alpha) * Eye/4
M

##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0375 0.4625 0.0375 0.3208
## [2,] 0.0375 0.0375 0.0375 0.3208
## [3,] 0.8875 0.4625 0.8875 0.3208
## [4,] 0.0375 0.0375 0.0375 0.0375

y <- eigen(M) # find the eigenvalues and eigenvectors
y$val # the eigenvalues

## [1]  1.000e+00+0.000e+00i  5.039e-07+8.728e-07i  5.039e-07-8.728e-07i
## [4] -1.008e-06+0.000e+00i

y$vec # the eigenvectors

##            [,1]                  [,2]                  [,3]           [,4]
## [1,] 0.08061+0i -7.071e-01+0.000e+00i -7.071e-01-0.000e+00i  7.071e-01+0i
## [2,] 0.05657+0i -8.384e-07-1.452e-06i -8.384e-07+1.452e-06i -1.677e-06+0i
## [3,] 0.99416+0i  7.071e-01+0.000e+00i  7.071e-01+0.000e+00i -7.071e-01+0i
## [4,] 0.04408+0i  2.982e-12-5.165e-12i  2.982e-12+5.165e-12i  5.965e-12+0i
```

Comparison among centrality measures for the Padgett Florentine families

- Let us look at the business ties network of the Padgett Florentine families
- The top three ranks by different methods are summarized as follows:

Rank	Degree	Closeness	Betweenness	Eigenvector	PageRank
1	MEDICI	MEDICI	MEDICI	MEDICI	MEDICI
2	GUADAGNI	RIDOLFI	GUADAGNI	STROZZI	GUADAGNI
3	STROZZI	ALBIZZI	ALBIZZI	RIDOLFI	STROZZI

- Deciding which are most appropriate for a given application clearly requires consideration of the context.

Correlation analysis among centrality measures for the Padgett Florentine families

```
rm(list = ls()) # clear memory
library(igraph) # load packages
load("./R code/padgett.RData") # read in the Padgett Florentine families network
# calculate the degree centrality
deg_B <- degree(padgett$PADGB, loops = FALSE)
sort(deg_B, decreasing = TRUE) # sort the nodes in decreasing order

##    MEDICI  GUADAGNI   STROZZI   ALBIZZI  BISCHERI CASTELLAN   PERUZZI
##         6         4         4         3         3         3         3
##   RIDOLFI TORNABUON BARBADORI  SALVIATI ACCIAIUOL    GINORI LAMBERTES
##         3         3         2         2         1         1         1
##     PAZZI     PUCCI
##         1         0

# calculate the standardized degree centrality
deg_B_S <- degree(padgett$PADGB, loops = FALSE)/(vcount(padgett$PADGM) - 1)
sort(deg_B_S, decreasing = TRUE) # sort the nodes in decreasing order

##    MEDICI  GUADAGNI   STROZZI   ALBIZZI  BISCHERI CASTELLAN   PERUZZI
##   0.40000   0.26667   0.26667   0.20000   0.20000   0.20000   0.20000
##   RIDOLFI TORNABUON BARBADORI  SALVIATI ACCIAIUOL    GINORI LAMBERTES
##   0.20000   0.20000   0.13333   0.13333   0.06667   0.06667   0.06667
##     PAZZI     PUCCI
##   0.06667   0.00000
```

Correlation analysis among centrality measures for the Padgett Florentine families

```
# calculate the closeness centrality
close_B <- closeness(padgett$PADGB)
sort(close_B, decreasing = TRUE)

##    MEDICI   RIDOLFI   ALBIZZI TORNABUON  GUADAGNI BARBADORI   STROZZI
##  0.024390  0.022727  0.022222  0.022222  0.021739  0.020833  0.020833
##  BISCHERI CASTELLAN  SALVIATI ACCIAIUOL   PERUZZI    GINORI LAMBERTES
##  0.019608  0.019231  0.019231  0.018519  0.018519  0.017241  0.016949
##     PAZZI     PUCCI
##  0.015385  0.004167

# calculate the standardized closeness centrality
close_B_S <- closeness(padgett$PADGB) * (vcount(padgett$PADGB) - 1)
sort(close_B_S, decreasing = TRUE)

##    MEDICI   RIDOLFI   ALBIZZI TORNABUON  GUADAGNI BARBADORI   STROZZI
##    0.3659    0.3409    0.3333    0.3333    0.3261    0.3125    0.3125
##  BISCHERI CASTELLAN  SALVIATI ACCIAIUOL   PERUZZI    GINORI LAMBERTES
##    0.2941    0.2885    0.2885    0.2778    0.2778    0.2586    0.2542
##     PAZZI     PUCCI
##    0.2308    0.0625
```

Correlation analysis among centrality measures for the Padgett Florentine families

```
# calculate the Betweenness centrality
betw_B <- betweenness(padgett$PADGB)
sort(betw_B, decreasing = TRUE)

##    MEDICI  GUADAGNI   ALBIZZI  SALVIATI   RIDOLFI  BISCHERI   STROZZI
##    47.500    23.167    19.333    13.000    10.333     9.500     9.333
## BARBADORI TORNABUON CASTELLAN   PERUZZI ACCIAIUOL    GINORI LAMBERTES
##     8.500     8.333     5.000     2.000     0.000     0.000     0.000
##     PAZZI     PUCCI
##     0.000     0.000

# calculate the standardized Betweenness centrality
betw_B_S <- 2 * betweenness(padgett$PADGB)/((vcount(padgett$PADGB) - 1) * (vcou
sort(betw_B_S, decreasing = TRUE)

##    MEDICI  GUADAGNI   ALBIZZI  SALVIATI   RIDOLFI  BISCHERI   STROZZI
##   0.45238   0.22063   0.18413   0.12381   0.09841   0.09048   0.08889
## BARBADORI TORNABUON CASTELLAN   PERUZZI ACCIAIUOL    GINORI LAMBERTES
##   0.08095   0.07937   0.04762   0.01905   0.00000   0.00000   0.00000
##     PAZZI     PUCCI
##   0.00000   0.00000
```

Correlation analysis among centrality measures for the Padgett Florentine families

```
# calculate the Eigenvector centrality
eigen_B <- evcent(padgett$PADGB)
sort(eigen_B[[1]], decreasing = TRUE)
```

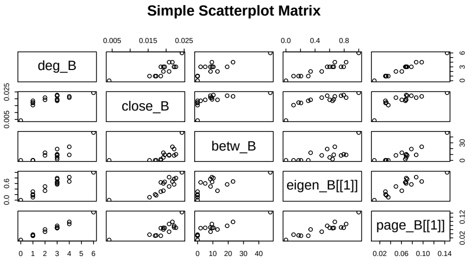
```
##    MEDICI   STROZZI   RIDOLFI TORNABUON  GUADAGNI  BISCHERI   PERUZZI
##    1.0000    0.8273    0.7937    0.7572    0.6719    0.6572    0.6408
## CASTELLAN   ALBIZZI BARBADORI  SALVIATI ACCIAIUOL LAMBERTES    GINORI
##    0.6020    0.5669    0.4920    0.3391    0.3071    0.2063    0.1741
##     PAZZI     PUCCI
##    0.1041    0.0000
```

Correlation analysis among centrality measures for the Padgett Florentine families

```
# calculate the PageRank centrality
page_B <- page.rank(padgett$PADGB)
sort(page_B[[1]], decreasing = TRUE)

##    MEDICI  GUADAGNI   STROZZI   ALBIZZI TORNABUON   RIDOLFI CASTELLAN
##  0.144373  0.097424  0.087226  0.078339  0.070574  0.068885  0.068644
##  BISCHERI   PERUZZI  SALVIATI BARBADORI     PAZZI    GINORI LAMBERTES
##  0.068180  0.067203  0.060696  0.049803  0.035697  0.032097  0.030604
## ACCIAIUOL     PUCCI
##  0.030354  0.009901
```

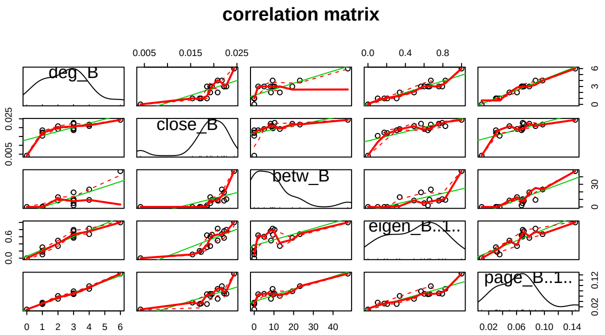
Correlation analysis among centrality measures for the Padgett Florentine families

```
df <- data.frame(deg_B_S, close_B_S, betw_B_S, eigen_B[[1]], page_B[[1]])
Pearson_correlation_matrix <- cor(df) # Pearson correlation matrix
Spearman_correlation_matrix <- cor(df, method = "spearman") # Spearman correlation matrix
cor(df, method = "kendall") # Kendall correlation matrix

##              deg_B_S close_B_S betw_B_S eigen_B..1.. page_B..1..
## deg_B_S       1.0000    0.6976   0.6680       0.8620      0.8991
## close_B_S     0.6976    1.0000   0.6905       0.7459      0.6611
## betw_B_S      0.6680    0.6905   1.0000       0.5570      0.6963
## eigen_B..1..  0.8620    0.7459   0.5570       1.0000      0.7000
## page_B..1..   0.8991    0.6611   0.6963       0.7000      1.0000

# Basic Scatterplot Matrix
pairs(~deg_B+close_B+betw_B+eigen_B[[1]]+page_B[[1]],data=df, main="Simple Scatterplot Matrix")
```



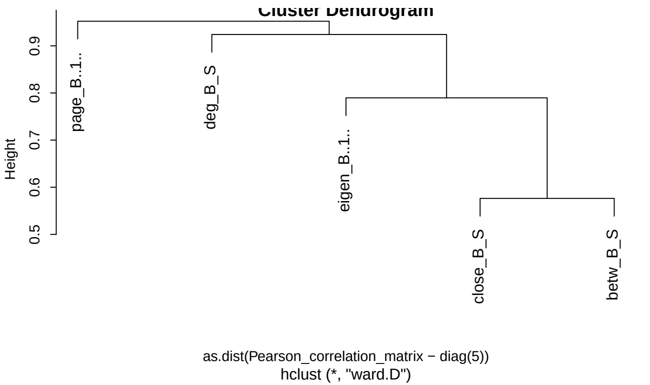
Correlation analysis among centrality measures for the Padgett Florentine families

```
# Scatterplot Matrices from the car Package, include lowess and linear best
# fit #lines, and boxplot, densities, or histograms in the principal
# diagonal, as well as #rug plots in the margins of the cells.
library(car)
## Warning: package 'car' was built under R version 3.0.2
scatterplotMatrix(~deg_B + close_B + betw_B + eigen_B[[1]] + page_B[[1]], data = df, main = "correl
```



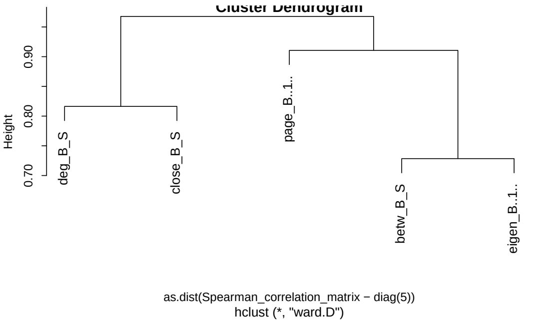
Correlation analysis among centrality measures for the Padgett Florentine families

```
# Classifcication based on correaltion coefficient Ward Hierarchical
# Clustering
fit_pearson <- hclust(as.dist(Pearson_correlation_matrix - diag(5)), method = "ward")

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

plot(fit_pearson) # display dendogram
```



Classification of centrality measures base don the correlation analysis of the Padgett Florentine families

```
groups <- cutree(fit_pearson, k = 3) # cut tree into 5 clusters
fit_spearman <- hclust(as.dist(Spearman_correlation_matrix - diag(5)), method = "ward")

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

plot(fit_spearman) # display dendogram
y<-eigrn(adj)

## Error:  could not find function "eigrn"
```



Comparing the three most popular centrality measures

- Generally, the 3 centrality types will be positively correlated
- When they are not (low) correlated, it probably tells you something interesting about the network

	Low degree	Low closeness	Low betweenness
High degree		Embedded in cluster that is far from the rest of the network	Ego's connections are redundant - communication bypasses him/her
High closeness	Key player tied to important/active alters		Probably multiple paths in the network, ego is near many people, but so are many others
High betweenness	Ego's few ties are crucial for network flow	Ego monopolizes the ties from a small number of people to many others	

A word for future by Wasserman and Faust (Social Network Analysis, Cambridge University Press, 1994: pp730) I

- "..., we do not expect that the most fruitful development in descriptive techniques will be the continued addition of yet another definition of centrality measure or yet another subgroup definition or yet another definition of equivalence. Rather, we expect that careful assessment of the usefulness of current methods in substantive and theoretical applications will be helpful in determining when, and under what conditions, each method is useful (perhaps in conjunction with statistical assumptions). Considerable work also needs to be done on measurement properties (such as sampling variability) of the current measures."

Extensions

- Weighted network
- Bipartitite and hypergraph
- Dynamic network

Extensions to weighted network

- Reduce to unweighted network so the standard techniques for unweighted graphs can be applied (Newman, 2004)
    - Assume positive weights, we can map from a weighted network to an unweighted multigraph
    - Formally, every edge of positive integer weight $w \in \mathbb{N}^+$ is replaced with w parallel edges of weight 1 each, connecting the same vertices.

Extensions to bipartitie network: affiliation network

- Reduce to unweighted network so the standard techniques for unweighted graphs can be applied

Extensions to dynamic

- Some work but largely open

Hypergraph

- An (undirected) hypergraph $(V;E)$ is a set system with ground set V as hypervertrices and $E = \{E_1, \ldots, E_m\}$ $(E_j \subseteq 2^V)$ as hyperedges.
- Equivalently, hypergraph can be represented by the incidence matrix $H_{n\times m}$ such that

$$H_{ij} = \begin{cases} 1, & \text{if } v_i \in E_j; \\ 0, & \text{otherwise,} \end{cases}$$

- Equivalently, hypergraph can be understood as a bipartitie graph (V, E) as the partition of nodes.

Hypergraph degree

- Let 1_m and 1_n be the all one vectors.
- Node degree:

$$D_v = H1_m$$

- Edge degree:

$$D_e = H^t 1^n$$

- If edge degree are all equal to 2, then we obtain the normal graph.

Eigenvector centrality for hypergraph

- There are many possibile definitions, the simplest one is to project the hypergraph to two normal graphs:
- For the incidence matrix $H_{n\times m}$ of hypergraph (V, E), then

$$\begin{aligned} A_v &:= HH^t \\ A_e &:= H^t H \end{aligned}$$

  are the adjacency matrices of two normal graphs on node sets V and E respectively.
- Define two (column) stochastic matrices:

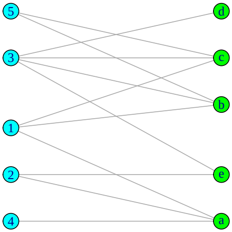
$$\begin{aligned} N_v &:= A_v D_v^{-1} \\ N_e &:= H^t H D_e^{-1} \end{aligned}$$

- Define the node and edge centrality measures respectively.

$$\begin{aligned} N_v x &= x \\ N_e y &= y \end{aligned}$$

An example

```
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from edge list
from<- c(1,1,1,2,2, 3,3,3,3,4,5,5)
to<- c("a","b","c", "a", "e", "b","c","d","e","a","b","c")
edgelist_df <-data.frame(from, to)
g<- graph.data.frame(edgelist_df,directed=FALSE)
V(g)$type <- V(g)$name %in% edgelist_df[,1]#add the type vertex attribute to create a biaprtite graph
lay <- layout.bipartite(g)
plot(g, layout=lay[,2:1],vertex.color=c("green","cyan")[V(g)$type+1])# plot the graph

proj<-bipartite.projection(g) # find the two projected normal graphs
g1<-proj$proj1
g2<-proj$proj2
```



continue

```
Nv <- t(get.stochastic(g1,sparse=FALSE)) #column stochastic matrix
Nv

##        a    b    c    e      d
## a 0.0000 0.25 0.25 0.25 0.0000
## b 0.3333 0.00 0.25 0.25 0.3333
## c 0.3333 0.25 0.00 0.25 0.3333
## e 0.3333 0.25 0.25 0.00 0.3333
## d 0.0000 0.25 0.25 0.25 0.0000

yv <- eigen(Nv) # find the eigenvalues and eigenvectors
yv$val # the eigenvalues

## [1]  1.000e+00 -5.000e-01 -2.500e-01 -2.500e-01  4.411e-17

yv$vec # the eigenvectors

##         [,1]    [,2]       [,3]       [,4]       [,5]
## [1,] -0.3693  0.5477  2.719e-17  6.701e-17 -7.071e-01
## [2,] -0.4924 -0.3651 -9.065e-18 -8.165e-01  3.107e-17
## [3,] -0.4924 -0.3651  7.071e-01  4.082e-01  3.107e-17
## [4,] -0.4924 -0.3651 -7.071e-01  4.082e-01  3.107e-17
## [5,] -0.3693  0.5477  2.719e-17  6.701e-17  7.071e-01
```

continue

```
Ne <- t(get.stochastic(g2,sparse=FALSE)) #column stochastic matrix
Ne

##      1      2      3   4   5
## 1 0.00 0.3333 0.3333 0.5 0.5
## 2 0.25 0.0000 0.3333 0.5 0.0
## 3 0.25 0.3333 0.0000 0.0 0.5
## 4 0.25 0.3333 0.0000 0.0 0.0
## 5 0.25 0.0000 0.3333 0.0 0.0

ye <- eigen(Ne) # find the eigenvalues and eigenvectors
ye$val # the eigenvalues

## [1]  1.0000 -0.6076 -0.5000  0.2743 -0.1667

ye$vec # the eigenvectors

##         [,1]       [,2]       [,3]       [,4]    [,5]
## [1,] -0.6172  2.941e-16 -8.165e-01 -1.027e-16  0.5345
## [2,] -0.4629  6.199e-01  2.283e-16  4.493e-01 -0.5345
## [3,] -0.4629 -6.199e-01  1.746e-16 -4.493e-01 -0.5345
## [4,] -0.3086 -3.401e-01  4.082e-01  5.460e-01  0.2673
## [5,] -0.3086  3.401e-01  4.082e-01 -5.460e-01  0.2673
```

Eigenvector centrality for hypergraph

- Here is another way to project by taking into consideration of the edge degree.

$$P = HD_e^{-1}H^tD_v^{-1}$$

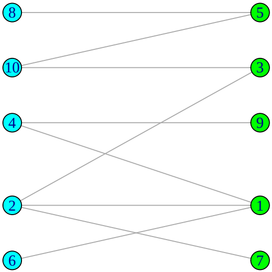
- Then P is a (column) stochastic matrix, and we define the node eigenvector measure as the leading eigenvector (which is 1):

$$Px = x$$

- Assume regularity and aperiodicity (Perron-Frobenius theorem), there is a unique x.
- Otherwise, we can add a damping factor, liek the PageRank, to gurantee uniqueness.

An example

```
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from edge list
typ<- rep(0:1,length=10)
edg<- c(1,2,1,4,1,6,3,2,3,10,5,8,5,10,7,2,9,4)
#edg<- c(1,6,1,7,1,8,2,6,2,10,3,7,3,8,3,9,3,10,4,6,5,7,5,8)
g<- graph.bipartite(typ,edg)
lay <- layout.bipartite(g)
plot(g, layout=lay[,2:1],vertex.color=c("green","cyan")[V(g)$type+1])# plot the graph
```



continue

```
H<-get.incidence(g,sparse=FALSE) #incidence matrix of a bipartite network
e<-rep(1,5)
degv<-1./(H %*% e)
Dv <- diag(degv[,1]) #node degree diagonal matrix
dege<-1./(t(H) %*% e)
De <- diag(dege[,1]) #edge degree diagonal matrix
P <- H %*% De %*% t(H) %*% Dv # PageRank matrix
P

##     [,1]   [,2] [,3]   [,4] [,5]
## 1 0.6111 0.1667 0.00 0.3333  0.5
## 3 0.1111 0.4167 0.25 0.3333  0.0
## 5 0.0000 0.2500 0.75 0.0000  0.0
## 7 0.1111 0.1667 0.00 0.3333  0.0
## 9 0.1667 0.0000 0.00 0.0000  0.5

y <- eigen(P) # find the eigenvalues and eigenvectors
y$val # the eigenvalues

## [1] 1.00000 0.83010 0.50000 0.19537 0.08564

y$vec # the eigenvectors

##        [,1]     [,2]       [,3]    [,4]     [,5]
## [1,] 0.6882  0.58930 -1.415e-16 -0.7740  0.23655
## [2,] 0.4588 -0.22856 -5.000e-01  0.4141  0.71770
## [3,] 0.4588 -0.71339  5.000e-01 -0.1867 -0.27007
## [4,] 0.2294  0.05512 -5.000e-01  0.1231 -0.58903
## [5,] 0.2294  0.29754  5.000e-01  0.4235 -0.09515
```

Non-negative, irreducible and regular matrices

- **Non-negative** matrices:

$$\boxed{A \geq 0, \text{ (element-wise)}}$$

- **Irreducible** matrices: for any pair of i, j:

$$\boxed{\begin{aligned} A &\geq 0 \\ (A^{k_{ij}})_{ij} &> 0, \text{ for some } k_{ij} \geq 1 \\ &\Updownarrow \\ \forall \text{ permutation matrix } P : P^T A P &\neq \begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix} \end{aligned}}$$

- **Regular** matrices (a.k.a. primitive matrices):

$$\boxed{\begin{aligned} A &\geq 0 \\ A^k &> 0, \text{ for some } k \geq 1 \end{aligned}}$$

- Obviously

Regular $\Longrightarrow$ Irreducible $\Longrightarrow$ Non-ngeative

Go Back Go Back

Graph interpretation I

- Let $G = (V, E)$ be the induced directed graph from matrix A such that $V = \{1, \ldots, n\}$ and an arc $(i, j) \in E$ iff $A^T_{ij} > 0$.
- A is irreducible iff G is strongly connected.
- A is regular iff G is strongly connected and the greatest common divisor (gcd) of all cycle lengths in G is one (a.k.a. aperiodic). Go Back

Period I

- Given a non-negative matrix A, for any $i \in \{1, \ldots, n\}$, define the period of index i to be the greatest common divisor of all natural numbers k such that $(A^k)_{ii} > 0$
- When A is irreducible, the period of every index is the same and is called the period of A.
  - Or equivalently, the period can be defined as the greatest common divisor of the lengths of the closed directed paths in G.
  - If the period is 1, A is aperiodic $\Longrightarrow$ A is regular (or primitive).

Go Back

Spectral radius for matrix $A \in \mathbb{C}^{n\times n}$ with spectrum $\lambda_1, \ldots, \lambda_n$ I

- The spectral radius $\rho(A)$ of A is defined as:

$$\rho(A) \stackrel{\text{def}}{=} \max_i(|\lambda_i|) \underbrace{=}_{\text{Gelfand's formula}} \lim_{k\to\infty} \|A^k\|^{1/k}.$$

  for any matrix norm $||\cdot||$

- The power of A satisfies that

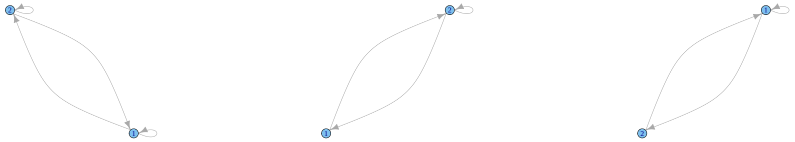
$$\lim_{k\to\infty} A^k = 0 \text{ if and only if } \rho(A) < 1.$$

  Moreover, if $\rho(A) > 1$, $\|A^k\|$ is not bounded for increasing k values. Go Back

Examples: regular

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ (positive) $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

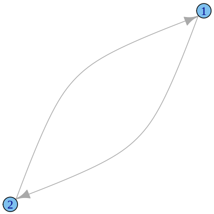
```r
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(1, 1,
    1, 1), # the data elements
  nrow=2, # number of rows
  ncol=2, # number of columns
  byrow = TRUE)# fill matrix by rows
g1 <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g1,edge.curved=TRUE) # plot the graph

adj<-matrix(
  c(0, 1,
    1, 1), # the data elements
  nrow=2, # number of rows
  ncol=2, # number of columns
  byrow = TRUE)# fill matrix by rows
g2 <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g2,edge.curved=TRUE) # plot the graph

adj<-matrix(
  c(1, 1,
    1, 0), # the data elements
  nrow=2, # number of rows
  ncol=2, # number of columns
  byrow = TRUE)# fill matrix by rows
g3 <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g3,edge.curved=TRUE) # plot the graph
```



Examples: Irreducible, but not regular

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

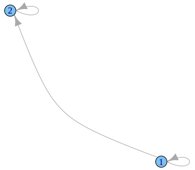
```
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(0, 1,
    1, 0), # the data elements
  nrow=2, # number of rows
  ncol=2, # number of columns
  byrow = TRUE)# fill matrix by rows
g <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g,edge.curved=TRUE) # plot the graph
```

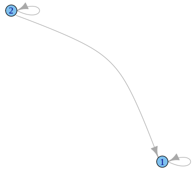


Examples: reducible

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

```
rm(list=ls()) #remove ALL objects
library(igraph)
#Generate graph object from adjacency matrix: igraph has the regular meaning
adj<-matrix(
  c(1, 0,
    1, 1), # the data elements
  nrow=2, # number of rows
  ncol=2, # number of columns
  byrow = TRUE)# fill matrix by rows
g1 <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g1,edge.curved=TRUE) # plot the graph

adj<-matrix(
  c(1, 1,
    0, 1), # the data elements
  nrow=2, # number of rows
  ncol=2, # number of columns
  byrow = TRUE)# fill matrix by rows
g2 <- graph.adjacency(t(adj), mode="directed") # create igrpah object from adjacency matrix
plot(g2,edge.curved=TRUE) # plot the graph
```





Observation

- These example show that both the existence and position of zeros matter!
- ◂ Go Back

Perron-Frobenius theorem I

- A testament that beautiful maths tends to be useful and useful maths tends to be beautiful eventually.
- Both German Mathematicians:
  - Oskar Perron (1880-1975): published 18 of his 218 papers after 84 years old
  - Ferdinand Georg Frobenius (1849-1917):
- Regular matrices share the same properties as positive matrices.
- Irreducible matrices sahre most of the properties of positive matrices
- Non-negative matrices has the weakest results.
- Refs: for more details, refer to Carl D. Meyer (`http://www.matrixanalysis.com/DownloadChapters.html`: Chapter 8) Go Back

Perron-Frobenius theorem: **Positive and Regular** matrix A with spectral radius $\rho(A) = r$ I

1. The number r is a positive real number such that any other eigenvalue λ (possibly, complex) is strictly smaller than r in absolute value, $|\lambda| < r$.
2. The eigenvalue r is simple. Both right and left eigenspaces associated with r are one-dimensional.
3. A has a left eigenvector v with eigenvalue r whose components are all positive.
4. A has a right eigenvector w with eigenvalue r whose components are all positive.
5. The only eigenvectors whose components are all positive are those associated with the eigenvalue r.

Go Back

Perron-Frobenius theorem: **irreducible** matrix A with period h and spectral radius $\rho(A) = r$ I

6. Suppose the left and right eigenvectors for A are normalized so that $w^T v = 1$. Then

$$\lim_{k\to\infty} A^k / r^k = v w^T,$$

7. Collatz-Wielandt formula:

$$r = \max_{x \gneq 0} \min_{i: x_i \neq 0} \frac{[Ax]_i}{x_i} = \min_{x \gneq 0} \max_{i: x_i \neq 0} \frac{[Ax]_i}{x_i}$$

8. The Perron-Frobenius eigenvalue satisfies the inequalities

$$\min_i \sum_j a_{ij} \leq r \leq \max_i \sum_j a_{ij}.$$

Go Back

Perron-Frobenius theorem: **irreducible** matrix A with period h and spectral radius $\rho(A) = r$ I

1. The number r is a positive real number and it is an eigenvalue of the matrix A.
2. The eigenvalue r is simple. Both right and left eigenspaces associated with r are one-dimensional.
3. A has a left eigenvector v with eigenvalue r whose components are all positive.
4. A has a right eigenvector w with eigenvalue r whose components are all positive.
5. The only eigenvectors whose components are all positive are those associated with the eigenvalue r.

Go Back

Perron-Frobenius theorem: **irreducible** matrix A with period h and spectral radius $\rho(A) = r$ I

6. Matrix A has exactly h eigenvalues with absolute value r:

$$\{re^{i\frac{2\pi k}{h}}\}_{0 \le k \le h-1} = \{r, re^{i\frac{2\pi}{h}}, \ldots, re^{\frac{2\pi(h-1)}{h}}\}$$

7. Let $\omega = 2\pi/h$. Then the matrix A is similar to $e^{i\omega}A$, consequently the spectrum of A is invariant under multiplication by $e^{i\omega}$ (corresponding to the rotation of the complex plane by the angle ω). Go Back

Perron-Frobenius theorem: **irreducible** matrix A with period h and spectral radius $\rho(A) = r$

8. If $h > 1$ then there exists a permutation matrix P such that

$$PAP^{-1} = \begin{pmatrix} 0 & A_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & A_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & A_{h-1} \\ A_h & 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

where the blocks along the main diagonal are zero square matrices.

Further properties: **irreducible** matrix A with period h and spectral radius $\rho(A) = r$

1. $(I + A)^{n-1} > 0$.
2. Wielandt's theorem. If $|B| < A$, then $\rho(B) \leq \rho(A)$.
3. If some power A^k is reducible, then it is completely reducible, i.e. for some permutation matrix P, it is true that:

$$PAP^{-1} = \begin{pmatrix} A_1 & 0 & 0 & \dots & 0 \\ 0 & A_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & A_d \end{pmatrix}$$

   where A_i are irreducible matrices having the same maximal eigenvalue. The number of these matrices d is the greatest common divisor of k and h.
4. If $c(x) = x^n + c_{k_1} x^{n-k_1} + c_{k_2} x^{n-k_2} + \ldots + c_{k_s} x^{n-k_s}$ is the characteristic polynomial of A in which the only nonzero coefficients are listed, then $h = \gcd(k_1, \ldots, k_s)$

Go Back

Further properties: **irreducible** matrix A with period h and spectral radius $\rho(A) = r$

5. Cesáro averages:

$$\lim_{k\to\infty} \frac{\sum_{i=0}^{k} \left(\frac{A}{r}\right)^k}{k} = \frac{vw^T}{w^T v} > 0.$$

6. For $h = 1$:

$$\lim_{k\to\infty} \left(\frac{A}{r}\right)^k = \frac{vw^T}{w^T v} > 0.$$

7. The adjoint matrix for $(r - A)$ is positive.
8. If A has at least one non-zero diagonal element, then A is regular.
9. If $0 \leq A < B$, then $r_A \leq r_B$. Moreover, if A is irreducible, then the inequality is strict: $r_A < r_B$.

Go Back

Perron-Frobenius theorem: **non-negative** matrix A with spectrum $|\lambda_1| \leq \ldots \leq |\lambda_n|$

1. $\lambda_n \geq \max\{|\lambda_1|, \ldots, |\lambda_{n-1}|\}$
2. There exists left and right eigenvectors $u, w^T \in \mathbb{R}$ of λ_n that are nonnegative (not necessarily unique, or strictly positive):

$$\begin{aligned} Au &= \lambda_n u, \\ w^T A &= \lambda_n w^T \end{aligned}$$

3. Collatz-Wielandt min-max formula

$$\lambda_n = \max_{x \gneq 0} \min_{i: x_i \neq 0} \frac{[Ax]_i}{x_i}$$

Go Back

References I

Luenberger, D. (1979). Introduction to dynamic systems: theory, models, and applications.

Newman, M. E. (2004). Analysis of weighted networks. *Physical Review E*, 70(5):056131.