Statistical Analysis of Network Data with R

Glossary of terms

Hwang Charm Lee

April 14, 2017

The Elements of Network Terminology

Creating Network Graphs

Undirected and Directed Graph

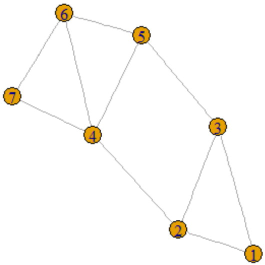
- $G = (V, E)$
  - Structure consisting of a set V of vertices and a set E of edges, where elements of E are unordered pairs $\{\mu, \nu\}$ of distinct vertices $\mu, \nu \in V$.

1. Order : The number of vertices $N_\nu = |V|$
2. Size : The number of edges $N_e = |E|$

- Undirected graph
  - There is no ordering in the vertices defining an edge.

Creating Network Graphs

```
g <- graph.formula(1-2, 1-3, 2-3, 2-4, 3-5,
                   4-5, 4-6, 4-7, 5-6, 6-7)
```



Creating Network Graphs

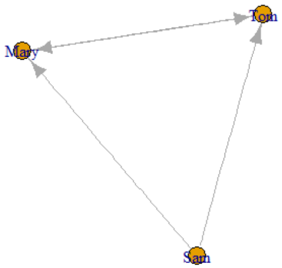
Undirected and Directed Graph

- Directed graph(Digraph)
  - A graph G for which each edge in E has an ordering to its vertices.

1. Arc
   - Directed edge with the direction of an arc (μ, ν) read from left(tail) to right(head).
2. Mutual
   - Digraphs may have two arcs between a pair of vertices, with the vertices playing opposite roles of head and tail for the respective arcs.

Creating Network Graphs

```
dg <- graph.formula(Sam-+Mary, Sam-+Tom, Mary++Tom)
```



Representations for Graphs

Basic formats

1. Edge list
   - An edge list is a simple two-column list of all vertex pairs that are joined by an edge.

```
E(dg)
```

```
## + 4/4 edges (vertex names):
## [1] Sam ->Mary Sam ->Tom  Mary->Tom  Tom ->Mary
```

Representations for Graphs

Basic formats

2. Adjacency matrix

```
get.adjacency(g)
```

```
## 7 x 7 sparse Matrix of class "dgCMatrix"
##   1 2 3 4 5 6 7
## 1 . 1 1 . . . .
## 2 1 . 1 1 . . .
## 3 1 1 . . 1 . .
## 4 . 1 . . 1 1 1
## 5 . . 1 1 . 1 .
## 6 . . . 1 1 . 1
## 7 . . . 1 . 1 .
```

Operations on Graphs

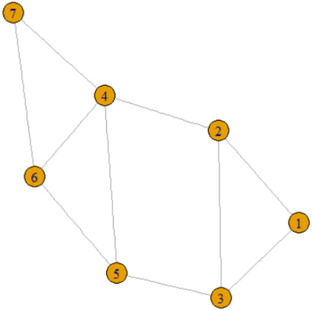
Subgraph

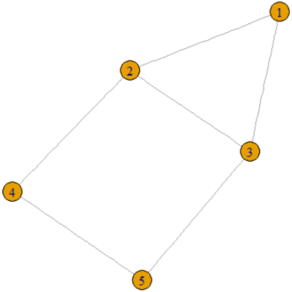
1. Subgraph
   - A graph $H = (V_H, E_H)$ is a subgraph of another graph $G = (V_G, E_G)$ if $V_H \subseteq V_G$ and $E_H \subseteq E_G$.

2. Induced subgraph
   - A subgraph $G' = (V', E')$, where $V' \subseteq V$ is a prespecified subset of vertices and $E' \subseteq E$ is the collection of edges to be found in G among that subset of vertices.

Operations on Graphs

```r
h <- g - vertices(c(6,7))
```





Talking about Graphs

Basic Graph Concepts

- Loop
  - Edge for which both ends connect to a single vertex.
- Multi-edge graph
  - A graph has pairs of vertices with more than one edge between them.
- Multi-graph
  - An object with either of loop or multi-edge property is called a multi-graph.
- Simple-graph, Proper edge
  - A graph that is not a multi-graph is called a simple graph, and its edges are referred to as proper edges.

Talking about Graphs

Connectivity

- Adjacency(vertices)
  - Two vertices $\mu, \nu \in V$ are said to be adjacent if joined by an edge in E.
- Adjacency(edges)
  - Two edges $e_1, e_2 \in E$ are adjacent if joined by a common endpoint in V.

- Neighbor
  - Adjacent vertices are also referred to as neighbors.
  - e.g. the three neighbors of vertex 5 in our toy graph g are 3, 4 and 5.
- Incident
  - A vertex $\nu \in V$ is incident on an edge $e \in E$ if ν is an endpoint of e.

Talking about Graphs

Connectivity of a graph

- Degree of a vertex(d_ν)
  - The number of edges incident on ν.

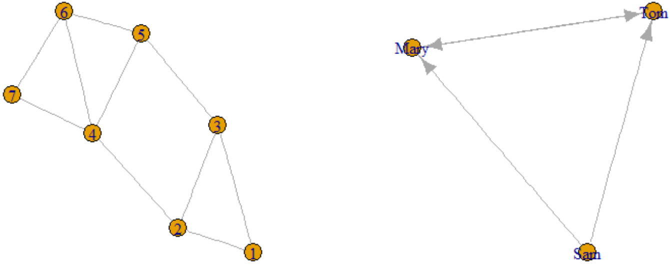
```
degree(g)
```

```
## 1 2 3 4 5 6 7
## 2 3 3 4 3 3 2
```

```
degree(dg)
```

```
##   Sam Mary  Tom
##     2    3    3
```

Talking about Graphs



Talking about Graphs

Connectivity of a graph

- In-degree(d_ν^{in}) and Out-degree(d_ν^{out})

```
degree(dg, mode='in')
```

```
##   Sam Mary  Tom
##     0    2    2
```

```
degree(dg, mode='out')
```

```
##   Sam Mary  Tom
##     2    1    1
```

Talking about Graphs

Movement of a graph

- Walk
  - A walk on a graph G, from ν_0 to ν_1, is an alternating sequence $\{\nu_0, e_1, \nu_1, e_2, \nu_2, \cdots, \nu_{l-1}, e_l, \nu_l\}$, where the endpoints of e_i are $\{\nu_{i-1}, \nu_i\}$. The length of this walk is said to be l.

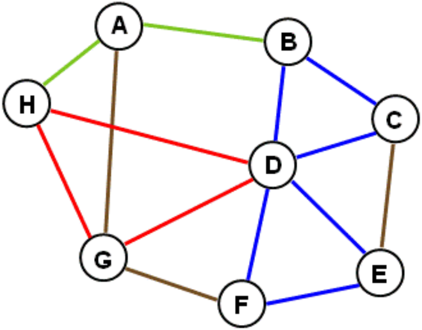
1. Trail : A trail is a walk in which all edges are distinct.
2. Path : A path is a trail in which all vertices are distinct.
3. Closed walk : A sequence of vertices starting and ending at the same vertex.

Talking about Graphs

Movement of a graph

- Cycle
  - A walk of length at least three, for which the beginning and ending vertices are the same, but for which all other vertices are distinct from each other.
- Circuit
  - A trail for which the beginning and ending vertices are the same.
  - A circuit can be a closed walk allowing repetitions of vertices but not edges.
  - However, it can also be a simple cycle, so explicit definition is recommended when it is used.

Talking about Graphs



Talking about Graphs

Movement of a graph

- Reachable
  - If there exists a walk from μ to ν, a vertex ν in a graph G is said to be reachable from another vertex μ.
- Connected
  - If every vertex is reachable from every other.
- Component
  - A component of a graph is a maximally connected subgraph.
  - It is a connected subgraph of G for which the addition of any other remaining vertex in V would ruin the property of connectivity.

Talking about Graphs

Movement of a digraph

- There are two variations of the concept of connectedness.
  1. Weakly connected
     - ★ If its underlying graph, the result of stripping away the labels 'tail' and 'head' from G, is connected.
  2. Strongly connected
     - ★ If every vertex v is reachable from every μ by a directed walk.

Talking about Graphs

Distance

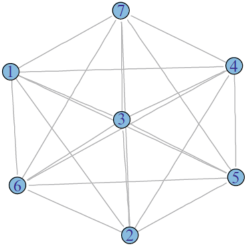
- Geodesic distance
  - The length of the shortest path(s) between the vertices
    - (which we set equal to ∞ if no such path exists).
- Diameter
  - The value of the longest distance in a graph.

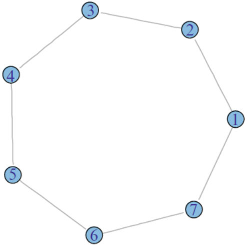
Special type of Graphs

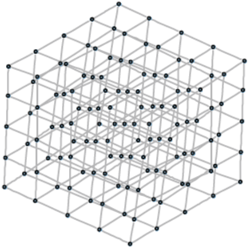
Special type

- Complete graph
  - A complete graph is a graph where every vertex is joined to every other vertex by an edge.
- Clique : A complete subgraph.
- Regular graph
  - A graph in which every vertex has the same degree.
  - e.g.
    - Ring : 2-regular
    - Lattice : 4-regular(the standard infinite lattice)

Special type of Graphs







Special type of Graphs

Special type

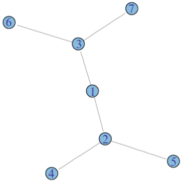
- Tree
  - A connected graph with no cycles.
  - In other words, any acyclic connected graph is a tree.
- Forest
  - The disjoint union of such tree graphs.
- Directed tree
  - A digraph whose underlying graph is a tree.

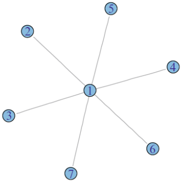
Special type of Graphs

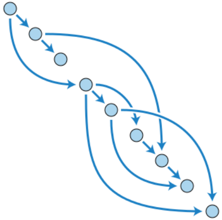
Special type

- K-star
  - A case of a tree, consisting only of one root and k leaves.
- DAG(directed acyclic graph)
  - A finite directed graph with no directed cycles
  - Underlying graph is not necessarily a tree, in that replacing the arcs with undirected edges may leave a graph that contains cycles.

Special type of Graphs







Special type of Graphs

Special type

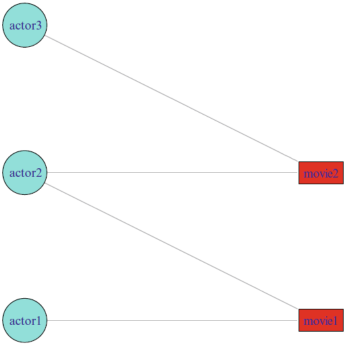
- Bipartite
  - The vertex set V may be partitioned into two disjoint sets, say V_1 and V_2, and each edge in E has one endpoint in V_1 and the other in V_2.
  - Typically are used to represent 'membership' networks.

- Projection
  - A graph $G_1 = (V_1, E_1)$ may be defined on the vertex set V_1 by assigning an edge to any pair of vertices that both have edges in E to at least one common vertex in V_2.

Special type of Graphs

- Movie-Actor example

```
g.bip <- graph.formula(actor1:actor2:actor3, movie1:movie2,
                        actor1:actor2 - movie1, actor2:actor3 -
```

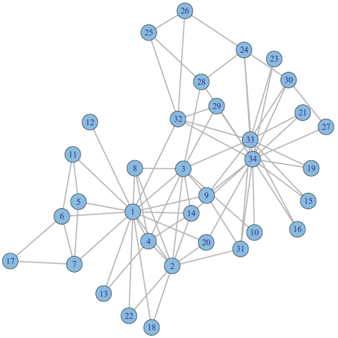


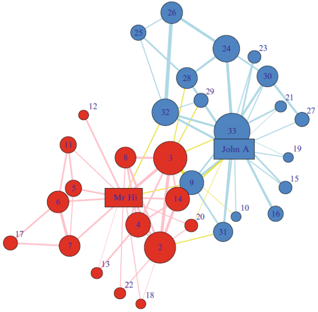
Descriptive Analysis of Network Graph Characteristics

Karate network example

1. Network description
   - Nodes represent members of a karate club observed by Zachary for 2 years during the 1970s, and links connecting two nodes indicate social interactions between the two corresponding members.
   - This dataset is composed of being split into two different clubs, due to a dispute between the administrator "John A" and instructor "Mr. Hi".
2. Data set
   - The data can be summarized as list of integer pairs. Each integer represents one karate club member and a pair indicates the two members interacted.
   - Node 1 stands for the instructor, node 34 for the president.

Karate network example



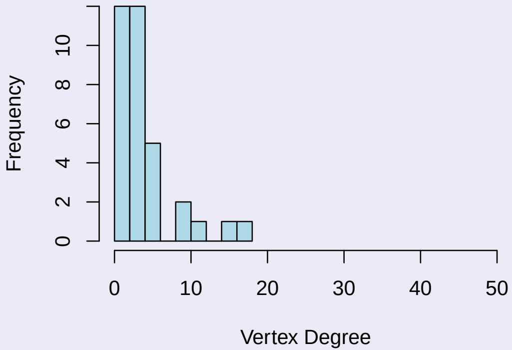


1. Vertices are sized in proportion to their degree.
2. The relative frequency of interactions is shown using edge thickness.
3. A change in vertex shape indicates the faction leaders.
4. Colors are used to distinguish membership in the factions as well as edges joining within versus between the same factions.

Vertex and Edge Characteristics

Vertex Degree

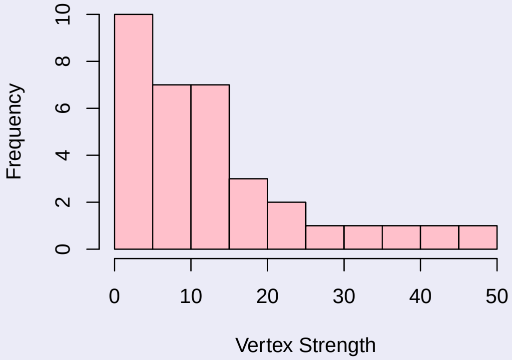
- Degree distribution of G
  - f_d : The fraction of vertices $\nu \in V$ with degree $d_v = d$.
  - The collection $\{f_d\}$, $d \geq 0$ is called the degree distribution of G, and is simply a rescaling of the set of degree frequencies, formed from the original degree sequence.



Vertex and Edge Characteristics

Vertex Degree

- Vertex strength
  - For weighted networks, vertex strength is obtained by summing up the weights of edges incident to a given vertex.
  - Sometimes it's called the weighted degree distribution.



Vertex and Edge Characteristics

Vertex Centrality

1. Motivation
   - Someone may have question Which actors in a social network seem to hold the 'reins of power'?
   - Measures of centrality are designed to quantify such notions of 'importance' and thereby facilitate the answering of such questions.

2. Vertex centrality
   - The most popular centrality measures are Closeness, Betweenness, and Eigenvector centrality.

Vertex and Edge Characteristics

Vertex Centrality

1. Closeness centrality[1]

$$C_{Cl}(\nu) := \frac{1}{\sum_{\mu \in V} dist(\nu, \mu)}$$

- Capture the notion that a vertex is 'central' if it is 'close' to many other vertices.
- The standard approach is to let the centrality vary inversely with a measure of the total distance of a vertex from all others.

Vertex and Edge Characteristics

Vertex Centrality

2. Betweenness centrality[2]

$$c_{CB}(\nu) := \sum_{s \neq t \neq \nu \in V} \frac{\delta(s, t|\nu)}{\delta(s, t)}$$

   - $\delta(s, t)$:= the # of shortest paths between s and t.
   - $\delta(s, t|\nu)$:= the # of shortest paths between s and t that pass through ν

- Aimed at summarizing the extent to which a vertex is located 'between' other pairs of vertices.
- 'Importance' relates to where a vertex is located with respect to the paths in the network graph.

Vertex and Edge Characteristics

Vertex Centrality(Continued)

3. Eigenvector centrality[3]

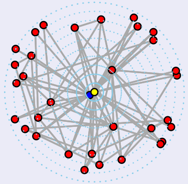
$$c_{E_i}(\nu) := \alpha \sum_{\{\mu,\nu\} \in E} c_{E_i}(\mu)$$

- The vector $c_{E_i} = (c_{E_i}(1), \cdots, c_{E_i}(N_\nu))^T$ is the solution to the eigenvalue problem $Ac_{E_i} = \alpha^{-1} c_{E_i}$
- Optimal choice of α^{-1} is the largest eigenvalue of A.

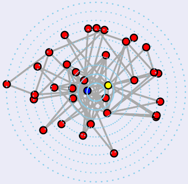
Vertex and Edge Characteristics

Vertex Centrality(Continued)

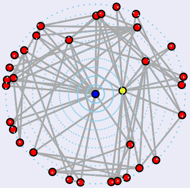
Degree



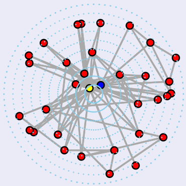
Closeness



Betweenness



Eigenvalue



Vertex and Edge Characteristics

Characterizing Edges

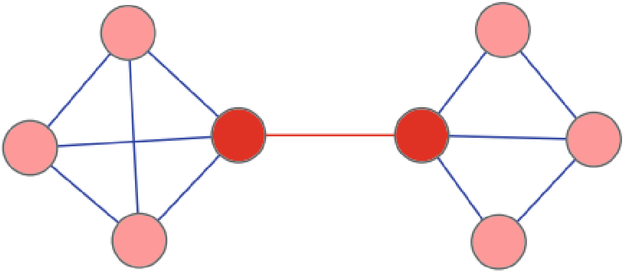
- Edge betweenness centrality[4]

$$E_{CB}(e) := \sum_{s,t \in V} \delta(s, t|e)$$

  - $\delta(s, t)$:= the # of shortest paths between s and t.
  - $\delta(s, t|e)$:= The # of shortest paths between s and t that pass through e
    - where e $= \{\mu, \nu\}$, for any $\mu, \nu \in V$

- It extends vertex betweenness centrality in a straightforward manner.
- The edge betweenness centrality is defined as the # of the shortest paths that go through an edge in a graph or network.
- An edge with a high edge betweenness centrality score represents a bridge-like connector between two parts of a network.

Vertex and Edge Characteristics



- This figure illustrates an example of eight nodes in a network, and the red edge between two red nodes has a high edge betweenness centrality value of "16".

Vertex and Edge Characteristics

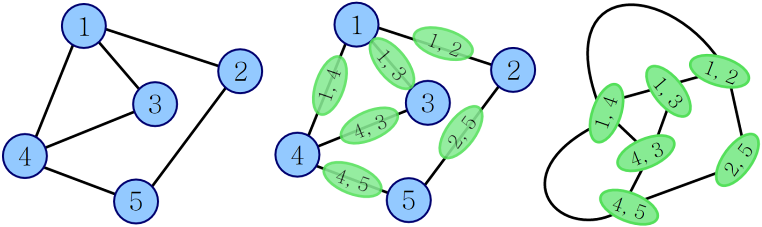
- e.g. for Karate network
  - eb <- edge.betweenness(karate)
  - E(karate)[order(eb, decreasing=T)[1:3]]
  - Edge sequence:
  - [53] John A – Actor 20
  - [14] Actor 20 – Mr Hi
  - [16] Actor 32 – Mr Hi
- Note that actor 20 plays a key role from this perspective in facilitating the direct flow of information between the head instructor (Mr Hi, vertex 1) and the administrator (John A, vertex 34).

Vertex and Edge Characteristics

Characterizing Edges

- Line graph
  - $G' = (V', E')$, the line graph of G, is obtained essentially by changing vertices of G to edges, and edges, to vertices.
  - More formally, the vertices $\nu' \in V'$ represent the original edges $e \in E$, and the edges $e' \in E'$ indicate that the two corresponding original edges in G were incident to a common vertex in G.

Vertex and Edge Characteristics



Reference

1. G. Sabidussi, The centrality index of a graph. Psychometrika 31, 581–683 (1966).
2. L. Freeman, A set of measures of centrality based on betweenness. Sociometry 40(1), 35–41 (1977).
3. P. Bonacich, Factoring and weighting approaches to status scores and clique identification. J. Math. Socio. 2(1), 113–120 (1972)
4. Werner Dubitzky, Encyclopedia of Systems Biology. Springer New York. 647-648 (2013).