Further Topics in Social Network Analysis
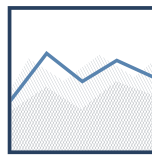# Centrality

Dominik Batorski   Michał Bojanowski   Bartosz Chroł   Kamil Filipek

Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw

WARSAW
SCHOOL of
DATA
ANALYSIS

Contents

The goal in many studies is to identify the most important actor(s) in the network. The most important actors exercise control over others or influence their behaviors to achieve private goals. However, the notion of "importance" may be defined in very different ways. Consequently, it can also be measured in many different ways in social network analysis. In general, sociological theory posits that important actors are those, who face minimal number of *constraints* and have many *opportunities* to act. Important actors are called "central" in the terminology of SNA.

Historically, being "central" in a network usually referred to being involved in many ties, because it made an actor prominent in a network, more visible to others, and so on (Wasserman and Faust 1994: 173). Over time, many other definitions of being central (or centrality) have been developed. In this chapter we will focus on a couple of different definitions and measures of centrality in social networks. Four concepts of centrality will be discussed further:

a) Degree centrality
b) Closeness centrality
c) Betweenness centrality
d) Eigenvector centrality

To get more information on centrality measures see Wasserman and Faust (1994 Chapter 5).

```
# Load necessary R packages
library(igraph)
library(isnar)
library(scales)
```

1   Degree centrality

The simplest measure of centrality is degree centrality. In undirected graphs, actors having more ties have better opportunities to act as they have more choices. In a directed graph, in-degree and out-degree can be considered separately to differentiate between having many incoming relations, i.e. "popularity", and having many outgoing relations, i.e. "sociality".
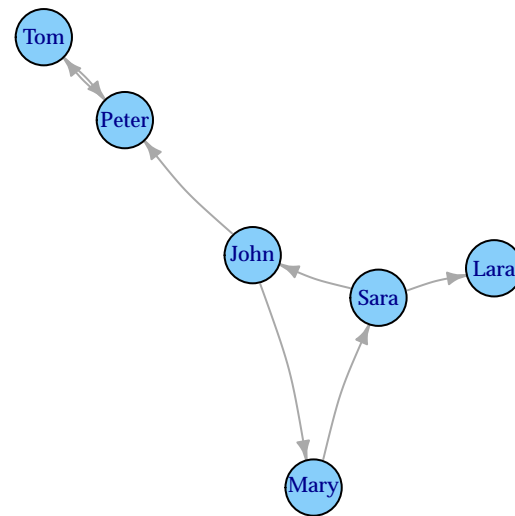
Consider the two simple networks presented below.

```
g1 <- graph.formula(
  Mary ---+ Sara,
  Sara ---+ Lara,
  Sara ---+ John,
  John ---+ Mary,
  John ---+ Peter,
  Peter ---+ Tom,
  Tom ---+ Peter
)
# undirected version
g1u <- as.undirected(g1)
```

The network g1u is an undirected version of g1, where a line is present in g1u whenever there is an arc in g1.
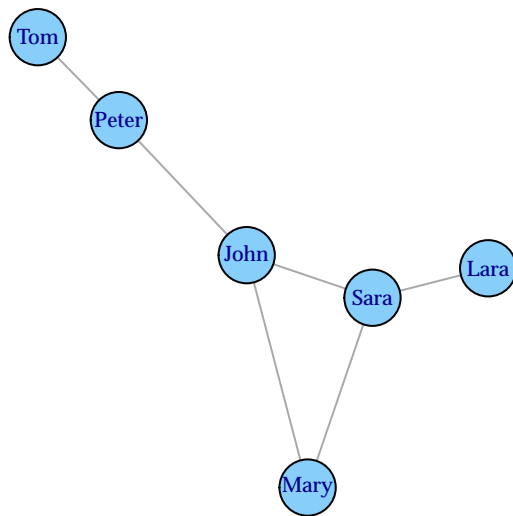
In an undirected graph the most central actors have the highest degree value. In our example these will be John and Sara, both of them have degree 3.

```
degree(g1u)
```

```
## Mary  Sara  Lara  John Peter   Tom
##    2     3     1     3     2     1
```

Now, let us consider the directed graph above.

In this graph relations between actors have directions, so we can analyze in-degree and out-degree separately specifying the `mode` argument.

```
degree(g1)    # total
```

```
## Mary  Sara  Lara  John Peter   Tom
##    2     3     1     3     3     2
```

```
degree(g1, mode = "out")
```

```
## Mary  Sara  Lara  John Peter   Tom
##    1     2     0     2     1     1
```

```
degree(g1, mode = "in")
```

```
## Mary  Sara  Lara  John Peter   Tom
##    1     1     1     1     2     1
```

First measure gives us the total number relations (sum of in- and out-degree) for each actor. Again, Sara and John are actors with the highest total degree of 3. They also score the highest when considering out-degrees, both have 2 outgoing relations. However, it is Peter who has the largest number of incoming relations (in-degree): 2.

Degree centrality has also a relative, or normalized, variant. The degree of each actor is divided by the number of all possible relations he may have in this particular network, so $N - 1$ where N is the number of actors. To calculate normalized version set the argument `normalized` to `TRUE`. In our example networks the relative degree centrality scores are:

```
# undirected network
degree(g1u, normalized=TRUE)
```

```
##  Mary  Sara  Lara  John Peter   Tom
##   0.4   0.6   0.2   0.6   0.4   0.2
```

```
# directed network, in-degree centrality
degree(g1, mode="in", normalized=TRUE)
```

```
##  Mary  Sara  Lara  John Peter   Tom
##   0.2   0.2   0.2   0.2   0.4   0.2
```

So, for example, normalized in-degree of Peter is 0.4 meaning that he was nominated by 40% of others in this network.

In social science literature value of in-degree is often treated as indicator of popularity or prestige. For example, in IT Department actors having high in-degree value might be treated as mentors or worthy employees. They share knowledge or skills with other employees asking them for help. On the other side, out-degree can be treated as indicator of power or influence. Actors with high value of out-degree might influence behaviors of other actors in a network. For example, managing staff give its subordinates instructions and commands on how to fill up various tasks. It is worth to mention that under optimal conditions the most worthy managers have both high value of in- and out-degree.

2   Closeness centrality

Basic rationale behind closeness centrality is that all pairs of actors in a network are separated by measurable distances. Actor with the shortest paths to all other nodes in a graph occupy central position measured by closeness centrality. The most evident example of actor being closer to other actors is a node located in the center of a star network.

Let us illustrate this concept using the example network from above, and think of the edges as steps of a walks around the network that start from Sara.

```
# list of edge sequences for shortest paths originating from Sara
sp <- shortest_paths(g1, from=V(g1)["Sara"], output="epath")$epath
lens <- sapply(sp, length)
ecol <- rep("grey", ecount(g1))
ecol[ unique(sapply(sp[lens > 0], "[", 1)) ] <- "brown"
ecol[ unique(sapply(sp[lens > 1], "[", 2)) ] <- "red"
ecol[ unique(sapply(sp[lens > 2], "[", 3)) ] <- "orange"
plot(g1, main="Walks from Sara to others", layout=lay, edge.curved=0.2,
     edge.color=ecol)
legend("topright", lty=1, lwd=2, col=c("brown", "red", "orange"),
       bty="n", legend=1:3, title="Steps from Sara")
```

Walks from Sara to others
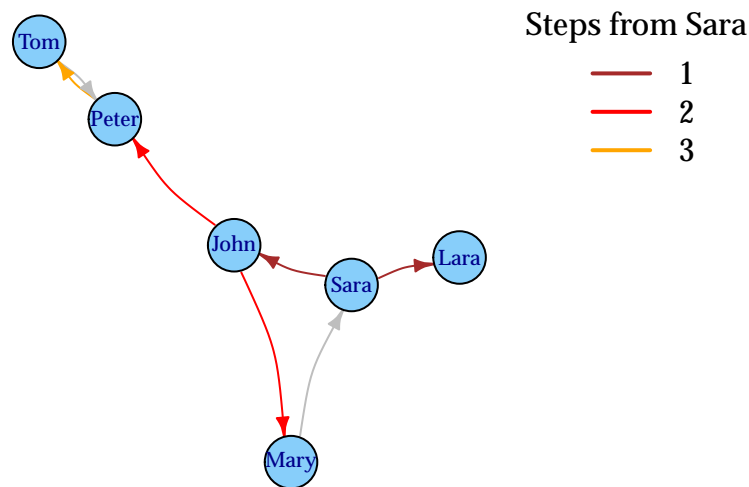
For each actor, apart from Sara, we can calculate the length of the walk to that actor starting from Sara:

- Lara and John are 1 step away from Sara
- Mary and Peter are 2 steps away from Sara
- Tom is 3 steps away from Sara

```
# vector of lengths of shortest walks originating from Sara to all others
d <- shortest.paths(g1, v=V(g1)["Sara"], mode="out")
d
```

```
##      Mary Sara Lara John Peter Tom
## Sara    2    0    1    1     2   3
```

We can sum-up these distances to calculate how far away, in total, Sara is from others in this particular network. It is:

```
# sum of the shortest walk lengths
sum(d)
```

```
## [1] 9
```

The sum itself can be interpreted as a measure of "decentrality". Closeness Centrality is defined as an inverse of the sum of the distances. For Sara it will be equal to:

```
1/sum(d)
```

```
## [1] 0.1111111
```

We can calculate closeness centralities for all actors in the network using `closeness` function:

```
closeness(g1, mode = "out")
```

```
##       Mary       Sara       Lara       John      Peter        Tom
## 0.08333333 0.11111111 0.03333333 0.11111111 0.04000000 0.04000000
```

The `mode` argument determines how the distances between actors are calculated. If it is `"out"` or `"in"` the centrality is based on walks, respectively, originating or terminating on a focal actor. If `mode` is `"all"`, than shortest paths are considered (directionality of the ties is ignored). For comparison:

```
closeness(g1, mode = "in")
```

```
##       Mary       Sara       Lara       John      Peter        Tom
## 0.04761905 0.04761905 0.05555556 0.04761905 0.07692308 0.06250000
```

```
closeness(g1, mode = "all")
```

```
##       Mary       Sara       Lara       John      Peter        Tom
## 0.11111111 0.12500000 0.08333333 0.14285714 0.11111111 0.07692308
```

To consider possible interpretations, let us assume that the network ties represent knowledge flows, i.e., an arrow from Sara to John represents the fact that John goes often to Sara for advice. As a consequence, advice (knowledge) can be thought to "flow" from Sara to John.

Closeness centrality of Sara when considering *incoming ties* (`mode="in"`) could be interpreted as the extent, to which Sara is a "sink" in the overall process of advice flow in the network. In other words, that she tends to receive advice from others, who themselves seek advice from others, who seek advice from others, and so on. Her position allows her to receive more directed and indirect advice than others in the network.

Closeness centrality of Sara when considering *outgoing ties* (`mode="out"`) can be then interpreted as an extent to which she is a "source" of advice for others (directly or indirectly) in the network.

3   Betweenness centrality

Structural advantage in a network is often based on opportunity to mediate between others. Some actors depend on others as they are connected through them with distant nodes. Betweenness centrality returns the number of times an actor acts as a bridge along the shortest path between pairs of nodes. Thus, sometimes it is not important how many ties an actor has or how close he is to other nodes in a network. Rather, of interest is the extent to which others are indirectly connected to one another through him. This measure has been explained in e.g. Freeman (1979).

Let us take a look at the created above:
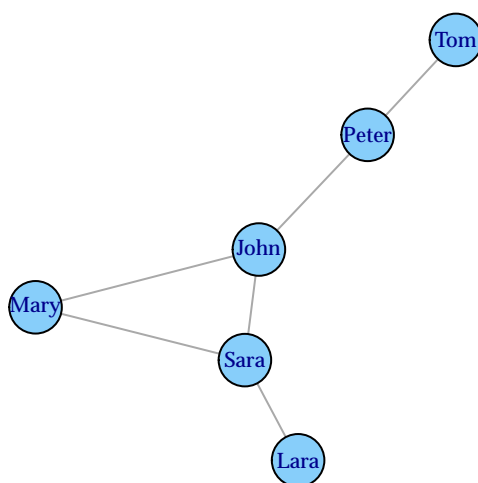
The most obvious cases are probably Lara and Tom. Both of them have only one connection and reside on a "boundary" of this network. Consequently, they do not mediate any indirect connections between others. Their

betweenness scores will be 0. A less obvious case is Mary. She has two connections to Sara and John. However, she is a rather redundant mediator: Sara and John have a direct relationship between one another, so do not need Mary as a mediator. In fact, any shortest path connecting a pair of actors does not go through Mary. For example, if we would like to trace the shortest path between, say, Lara and Tom, we would traverse Sara, John, and Peter. Consequently, Mary's betweenness will be also 0.

All the remaining actors in this network will play a role of mediators, and to a different extent. We may calculate how many pairs of actors and shortest paths between them have to traverse a given actor. Let us take a closer look at Sara. Sara is a true "gate keeper" for Lara: all shortest paths that go to Lara need to pass through Sara. We can see that:

1. There are four other actors in the network apart from Lara and Sara.
2. These four actors (Mary, John, Peter, and Tom) can reach each other directly or indirectly without Sara being involved.

Therefore, Sara lies on four shortest paths: Lara-Mary, Lara-John, Lara-Peter, and Lara-Tom.

Betweenness Centrality of an actor is a number of shortest paths in the network involving that actor.

Betweenness Centrality can be calculated using function `betweenness()`.

```
betweenness(g1u)
```

```
##  Mary  Sara  Lara  John Peter   Tom
##     0     4     0     6     4     0
```

From these results we can see that John has the highest betweenness centrality. He mediates all contacts between two groups of actors:

1. Peter and Tom.
2. Lara, Sara, and Mary.

All shortest paths from group (1) to group (2) go through John. As there are two actors in group (1) and three actors in group (2), John's betweenness centrality is $2 \times 3 = 6$.

We can show betweenness scores on the picture by making more central nodes bigger. As raw betweenness scores are from 0 to 6, we need to rescale them somewhat so that they work well as node sizes. Default node size (`vertex.size`) is 15, so let's rescale them to the interval [15; 30]:

```
# Rescale betweenness  using rescale() from package "scales"
b <- rescale(betweenness(g1u), c(25, 45))
range(b)
```

```
## [1] 25 45
```

```
plot( g1u, vertex.size=b )
```
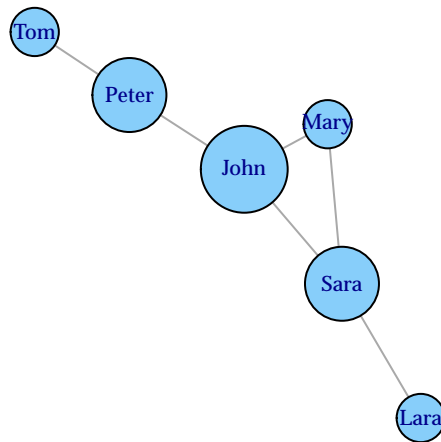
In directed networks we have an option whether or not to take tie directionality into account. In other words, when should we calculate the number of shortest *paths* (undirected) or shortest *walks* (directed). Using the directed network created earlier:
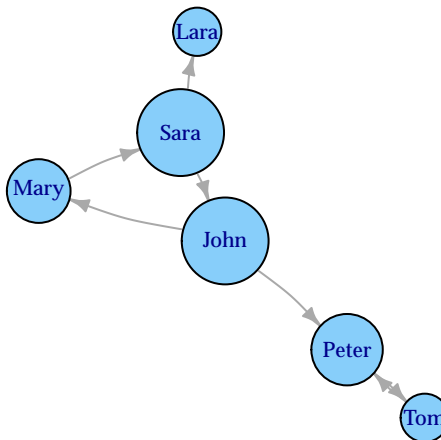
```
betweenness(g1, directed=FALSE) # identical to undirected above
```

```
##  Mary  Sara  Lara  John Peter   Tom
##     0     4     0     6     4     0
```

```
betweenness(g1, directed=TRUE)
```

```
##  Mary  Sara  Lara  John Peter   Tom
##     2     5     0     5     3     0
```

```
plot(g1, vertex.size=rescale(betweenness(g1), c(25, 45)),
     edge.curved=0.1 )
```


We can see that, for example, Mary now has a betweenness of 2 because:

1. She lies on a shortest walk from John to Sara (John -> Mary -> Sara).
2. She lies on a shortest walk from John to Lara (John -> Mary -> Sara -> Lara).

It may make sense to analyze shortest walks if the directionality of the ties reflects some sort of flow (e.g. knowledge as described in the section on closeness centrality).

For more details see e.g. Borgatti (2005) for a comparison of different network centrality measures vis a vis some paradigmatic dynamic processes of network flow.

```
# TODO delete me
graph <- g1u
```

4 Eigenvector centrality

Eigenvector centrality was developed by Bonacich (2007). The basic rationale behind this measure is an actor is more central the more central are his network neighbors. More precisely, actor's centrality is proportional to centrality scores of his/her network peers. This seemingly circular definition has an exact solution in that such defined actors' centralities are equal to the values of the first eigenvector of the graph's adjacency matrix. Actors with high eigenvector centralities are those which are connected to many other actors which are, in turn, connected to many others, and so on.

Eigenvector centralities can be calculated with the function `evcent()`. The value returned is a list containing more details regarding the eigenvalue decomposition, but the centrality scores themselves are stored in a component `vector`:

```
ecent <- evcent(g1u)
ecent$vector
```

```
##      Mary      Sara      Lara      John      Peter       Tom
## 0.8426258 0.9668575 0.4142136 1.0000000 0.5247172 0.2247953
```

```
plot(g1u, vertex.size=rescale(ecent$vector, c(25, 45)))
```
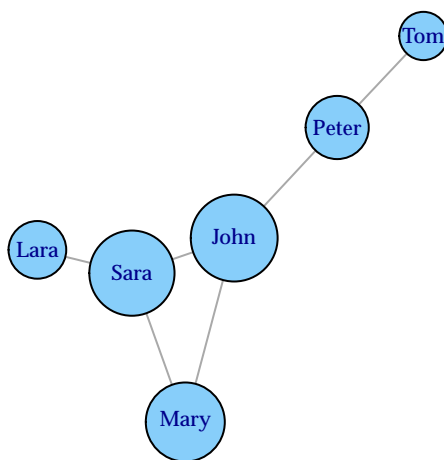

As in betweenness centrality, John is the most central person in the graph, followed by Sara. However, now Mary is third most central person in a graph, while previously she was one of the least central. The reason for this difference is that Mary is directly connected to most central actors (according to this measure) in the network.

5 Extended example on a real network

Let us now calculate all the measures defined above on a real network. Consider the network of judges from a Polish regional court. Two judges are connected, if they have ruled on at least one case together. We want to find

the most important, most central judges in the network according to previously shown measures. Our analysis will be limited to the largest component of a bigger network because it is unclear how to compare, for example, closeness scores for unconnected components.

First, we extract the largest connected component:

```
data(judge_net)
cl <- clusters(judge_net)
gj <- induced.subgraph(judge_net, cl$membership == which.max(cl$csize))
gj$layout <- layout.fruchterman.reingold(gj)
```

Second, we calculate values of different centrality measures:

```
deg <- degree(gj)
cl <- closeness(gj)
between <- betweenness(gj)
ecent <- evcent(gj)$vector
```

Below you could see our network with nodes colored according to their (normalized) centrality scores. The most central nodes are red and the least central are blue. Next to that, we show distributions of (unnormalized) centralities.
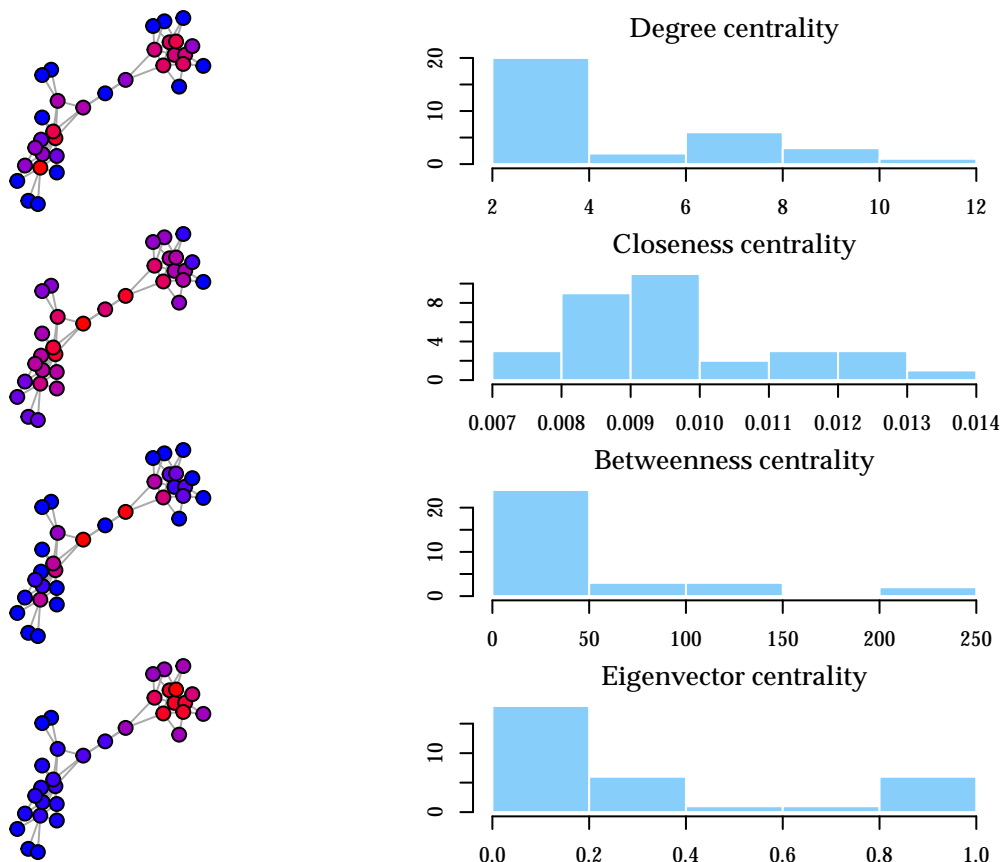
You could see that each measure marks different nodes as the most central. Nodes in the centers of two clusters have clearly the highest degrees. In closeness centrality the most central nodes lie in the "visual" center of the network, between two bigger clusters. They are reasonably close to all other nodes, while nodes from each cluster always have longer path to the other cluster. Also betweenness centrality marks those nodes central, but unlike closeness it is more rigid – only nodes forming the bridge between two clusters are central. The one that is connected to both of them, but none of the others, has betweenness score 0. Looking on the histograms you could say that closeness

centrality is more uniformly distributed, while betweenness centrality resembles more power–law distribution. Eventually eigenvector centrality completely ignores one cluster and treats nodes is the second cluster as central, even though it is smaller.

To sump up, you must remember that various centrality measures take into account different properties. Therefore in the first place you should decide what properties you are interested in and then find appropriate centrality measure, and not the other way round.

References

Bonacich, Phillip. 2007. "Some Unique Properties of Eigenvector Centrality." *Social Networks* 29 (4). Elsevier: 555–64.

Borgatti, Stephen P. 2005. "Centrality and Network Flow." *Social Networks* 27 (1). Elsevier: 55–71.

Freeman, Linton C. 1979. "Centrality in Social Networks Conceptual Clarification." *Social Networks* 1 (3). Elsevier: 215–39.

Wasserman, Stanley, and Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. New York: Cambridge University Press.