Alexander J Freund, Timothy A. Romer, Dat Luong, Jaxson Wirth

Miami University

Department of Computer Science and Software Engineering

Oxford, OH, USA

freundaj@miamioh.edu

1. **Introduction:**

What should CIA do to identify the leader of a terrorist group if the resources spent on investigating a person is costly? Where should a series of hospitals be constructed to maximize the number of people that are located near one? By measuring the centralities of a network, we can tell how important/ central a node is with respect to the rest of the network. Centrality measurement methods have been used for biological networks and applied to gene regulatory networks (Koshützki and Schreiber) and also used to assess the prospects for criminal intelligence (Sparrow) or even solving a financial crisis (Kuzubas et al.).

Calculating these centrality measures can be quite expensive in a computational sense which insinuates that it takes a lot of time to calculate. To extend this even further, ranks are taken based on the calculation so nodes in the network each have a rank to compare rather than comparing the centrality values. The goal of this research is to figure out if certain centrality ranks can be accurately predicted using ranks of other centrality variants so researchers can spend less time generating repetitive information. To do this, we will train and test various machine learning models.

In this paper, we will be identifying various centrality measures and identifying whether a calculated metric can be predicted by other centrality metrics, with regards many types of network. Specifically, the research is as follows:

* Examines 7 different metrics for measuring centrality
* Assesses the correlation between metrics
* Computes the prediction of a metric based on other metrics
* Computes the accuracy of such prediction to identify whether that metrics is truly worth doing and interpreting

The organization of this paper is as follows. In the next section, we will provide a background on the metrics we will be using and the mathematical definitions behind them. Section two will also contain examples of a randomly generated graph and the centrality calculations/ranks will be discussed. Following this, the last section will provide detailed methods on how we generate different types of networks, compute centralities, compute the correlations, and prediction of the metrics.

1. **Background:**

Centrality is the measure of how central a particular node in a graph or network is and the general assumption is that it is typically located in the center of a network or graph (Freeman). There are many forms of centrality measure including simplistic measure such as degree, closeness, betweenness, and load. There are also more complicated measures that have been developed throughout history for other applications but can be related to network centrality such Page Rank.

In order to know if certain centrality measures are able to predict other centrality measures using different machine learning models, the centrality measures will have to be calculated on random graphs that are generated a specified amount of times. These random graphs feature scale-free, small-world, random, and scale-free small-world networks. Below will be a discussion on the different centrality measures that are used in this research.

**IIA. Degree Centrality:**

In network science, degree centrality has traditionally been considered to be the simplest measure of centrality and first item to look at when examining centrality (Opsahl et al.). Degree centrality can be defined in English to be the ability for a node to receive information that is flowing through a network. This is measured by the number of links that node has to other nodes (Opsahl et al.).

The degree centrality for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where N is the is the total number of vertices (Opsahl et al.).

Below, an example of a randomly generated network and the degree centrality associated with some of the nodes will be demonstrated.

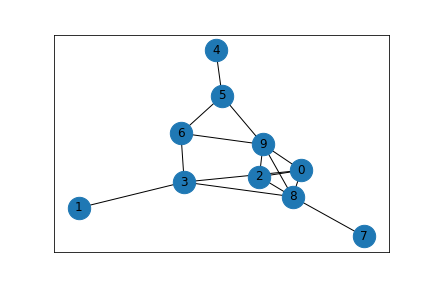


Fig. 1. A randomly generated graph with 10 vertices and 15 edges to demonstrate degree centrality.

In Figure 1, the vertices with the highest degree centrality would be vertices nine and eight with a degree centrality value of 0.5556. This value was calculated examining how many edges are connected to the vertex. By summing up those edges, then dividing by the one less than the total amount of nodes, you get the degree centrality for that vertex. The vertices with the lowest degree centrality are vertices one, seven, and four with a degree centrality of 0.1111.

**IIB. Closeness Centrality:**

Within graphs, a node is considered to have a high value of ‘closeness’ if the node has a relatively low average of shortest path distance to all other nodes (Rochat). This is insinuating that a node with a high value is generally closer to all other nodes in the graph. To calculate this value, take one less than the number of nodes in a graph and divide it by the sum of the shortest path between a node and all other nodes in a graph.

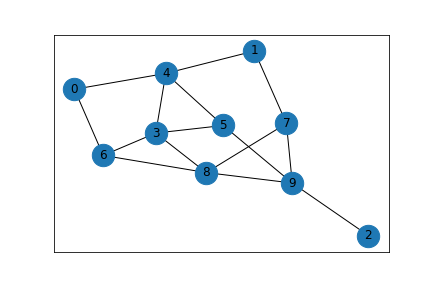


Fig. 2. A randomly generated graph with 10 vertices and 15 edges to demonstrate closeness centrality.

The closeness centrality can be defined mathematically as follows. The closeness centrality for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where is the number of vertices in a graph and is the shortest path function between vertex y and x (Rochat).

In Figure 2, the vertex with the highest closeness centrality is vertex eight with a value of 0.6429. This value was calculated using the formula above by dividing one less than the number of nodes by the sum of the shortest distance from that node and all other nodes in the graph. The vertex with the lowest closeness centrality is vertex two with a value of 0.3913.

**IIC. Betweenness Centrality:**

The betweenness centrality measure is the measure of how many times that a particular node occurs on the shortest path between two other nodes. This centrality measure is similar to the closeness centrality because both of them involve the calculation of the shortest path between nodes. To calculate this value, take the sum of the shortest paths between nodes s and t that pass-through node v then divide the number of shortest paths between nodes s and t. (Brandes, "Maintaining the duality of closeness and betweenness centrality.").

The betweenness centrality can be defined mathematically as follows. The closeness centrality for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where is the number of shortest paths between s and t given they contain vertex v and is the number of shortest paths between s and t (Brandes, "Maintaining the duality of closeness and betweenness centrality.").

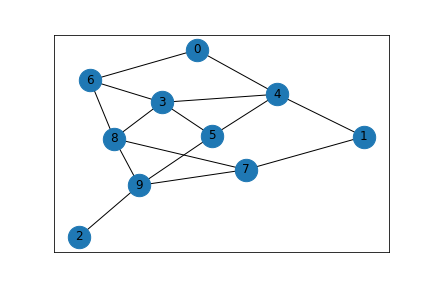


Fig. 3. A randomly generated graph with 10 vertices and 15 edges to demonstrate betweenness centrality.

Based on the graph depicted in Figure 3, the vertex with the highest betweenness centrality value is vertex nine with a value of 0.2639. This value was found by summing the shortest path between two nodes that involve a node then divide it by all shortest paths between two nodes. The vertex with the lowest centrality rating is vertex two with a value of 0.0000.

**IID. Load Centrality:**

The load centrality measure is similar to the betweenness centrality in that it measures the amount of flow that goes through a particular node; however, the load centrality measures the unit amount of information that get split between other nodes. Information is continually split between adjacent nodes until the target is reached. The total amount of information that passes through the node is defined as its load (Brandes, "On variants of shortest-path betweenness centrality and their generic computation.").

The betweenness centrality can be defined mathematically as follows. The closeness centrality for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where N is the number of vertices, is the vertex, and is defined to be 0.80. (Goh).

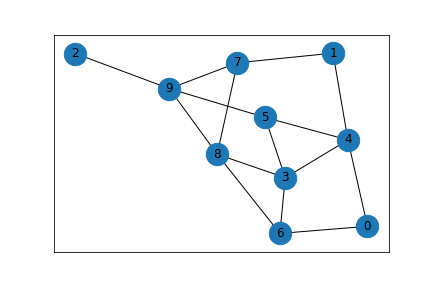


Fig. 4. A randomly generated graph with 10 vertices and 15 edges to demonstrate load centrality.

Based on the mathematical formula defined above, the graph depicted in Figure 4, and the implementation of this centrality calculation in the network python library, the node with the highest level of load is node nine with a value of 0.2639 and the node with the lowest level of load is node two with a load centrality value of 0.

**IIE. Local Reaching Centrality:**

The local reaching centrality is the measure for a node and its proportion of all other nodes that are reachable in a graph of that particular node. This gives a fundamental assumption that all nodes that are reachable for a node are located in some finite distance away (Mones et al.).

The betweenness centrality can be defined mathematically as follows. The closeness centrality for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where is the distance formula and is the number of vertices in a graph (Mones et al.).

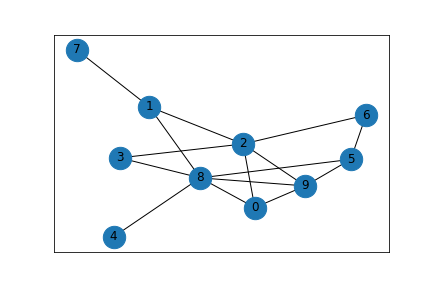


Fig. 5. A randomly generated graph with 10 vertices and 15 edges to demonstrate load centrality.

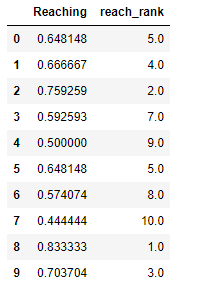


Table 1. A displaying showing the local reaching calculation and rank for all nodes depicted in figure 5.

As you can see in the results of the calculation of the graph in Figures 5 and Table 1, the node with the highest reaching is node 8 and the node with the lowest reaching is node 7. It is interesting to note that node 7 and node 4 are very removed from the graph as they have a degree of one and they both have the lowest reach rank.

**IIF. Harmonic Centrality:**

The harmonic centrality measure is similar to the closeness centrality however it addresses the issues of unreachable nodes. The harmonic difference will correct the issues with the average shortest path measure because disconnected nodes can have a potentially misleading value because the average distance could be low if the graph is almost entirely disconnected (Boldi and Vigna).

The harmonic centrality can be defined mathematically as follows. The closeness centrality for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where is the shortest average distance function (Boldi and Vigna).

Below is an example of a randomly generated network and the betweenness centrality associated with some of the nodes will be demonstrated.

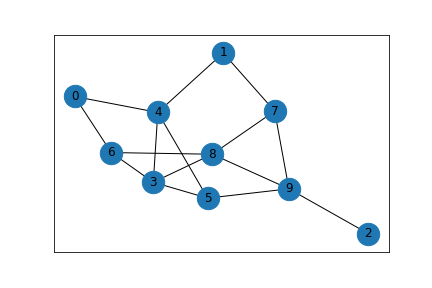


Fig. 6. A randomly generated graph with 10 vertices and 15 edges to demonstrate harmonic centrality.



Table 3. Correlation table for closeness rank and harmonic rank for the graph depicted in figure 6.

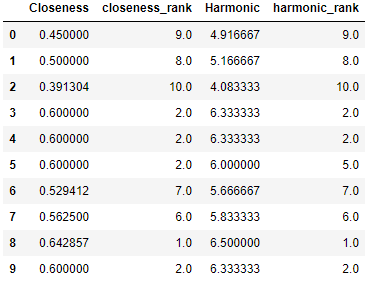


Table 2. Pandas DataFrame with closeness centrality data and harmonic centrality data for the graph depicted in figure 6.

The graph depicted in Figure 6 is a simple graph with all of the nodes connected to one another. So, it should be expected that the results of calculating the harmonic centrality to be similarly ranked as the closeness centrality ranking. In fact, there is an expectation for harmonic centrality to be closely correlated to closeness centrality (Boldi and Vigna). As you can see in Table 3, the closeness ranks, and the harmonic ranks are very close together and the nodes typically have the same rank and show a strong correlation as shown in Table 3.

**IIG. Page Rank:**

The page rank centrality measure has many applications including being the basis for how Google designed its search function (Page et al.). Page rank will rank websites based on the quality of websites that reference that particular website. In terms of networks, page rank will work in a very similar sense as it works in populating a search result. Page rank can be calculated iteratively and will return a probability that a node is accessed via another link. The page rank value at any particular time can be shown as where is the node that is accessed. The page rank for a vertex , for a given graph , where the graph can be defined as for vertices and edges is defined as where is the number of links from a node and c is a factor for normalization. (Page). Although this is a simplified definition of the page rank calculation, it should suffice for our simplified network application rather than ranking web pages for a search engine (Page et al.).

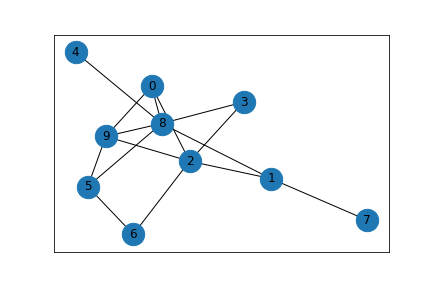


Fig. 7. A randomly generated graph with 10 vertices and 15 edges to demonstrate betweenness centrality.

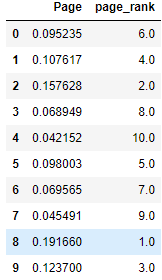


Table 4. A randomly generated graph with 10 vertices and 15 edges to demonstrate betweenness centrality.

The networkx python library provides a page rank function that will be calculated on a specific graph. For the graph in Figure 7, the table in Table 4 shows the rank and page rank value for all of the nodes. The node with the highest page rank value is node 8 and the node with the lowest page rank value is node 4.

1. **Methods:**

**IIIA. Resources Used:**  
   
 In order to work with network based systems and analyzing them computationally, we worked with four important Python libraries; Pandas, Numpy, and NetworkX, and SKLearn Pandas and Numpy primarily are our methods to work with data in general, NetworkX is used to generate network data with associated methods, and SKLearn serves as our method of machine learning.

**IIIB. Data Generation**  
   
 Our data comes from generating data for each of the combinations of scale free and small world; being random, only scale free, only small world, and both scale free and small world. We generate 100 graphs for each type, further subdivided in that by taking 25 of each specified size within the array [100, 200, 400, 800]; thus creating a total of 400 different graph samples.  
 The specific data we are working with is found from a variety of NetworkX libraries that allow us to take the data given a graph. The specifics we are working with are as follows; both degree and closeness rank are taken from NetworkX’s degree centrality method, betweenness ranking is taken from the betweenness centrality method, load rank is taken from the load centrality method, and reach rank is taken from the global reaching centrality.  
 Using this data we then calculated correlations, between each element. To calculate this we simply used Pandas’ correlation method. From there we specified to calculate for these three different correlations: pearson, spearman, and kendall.

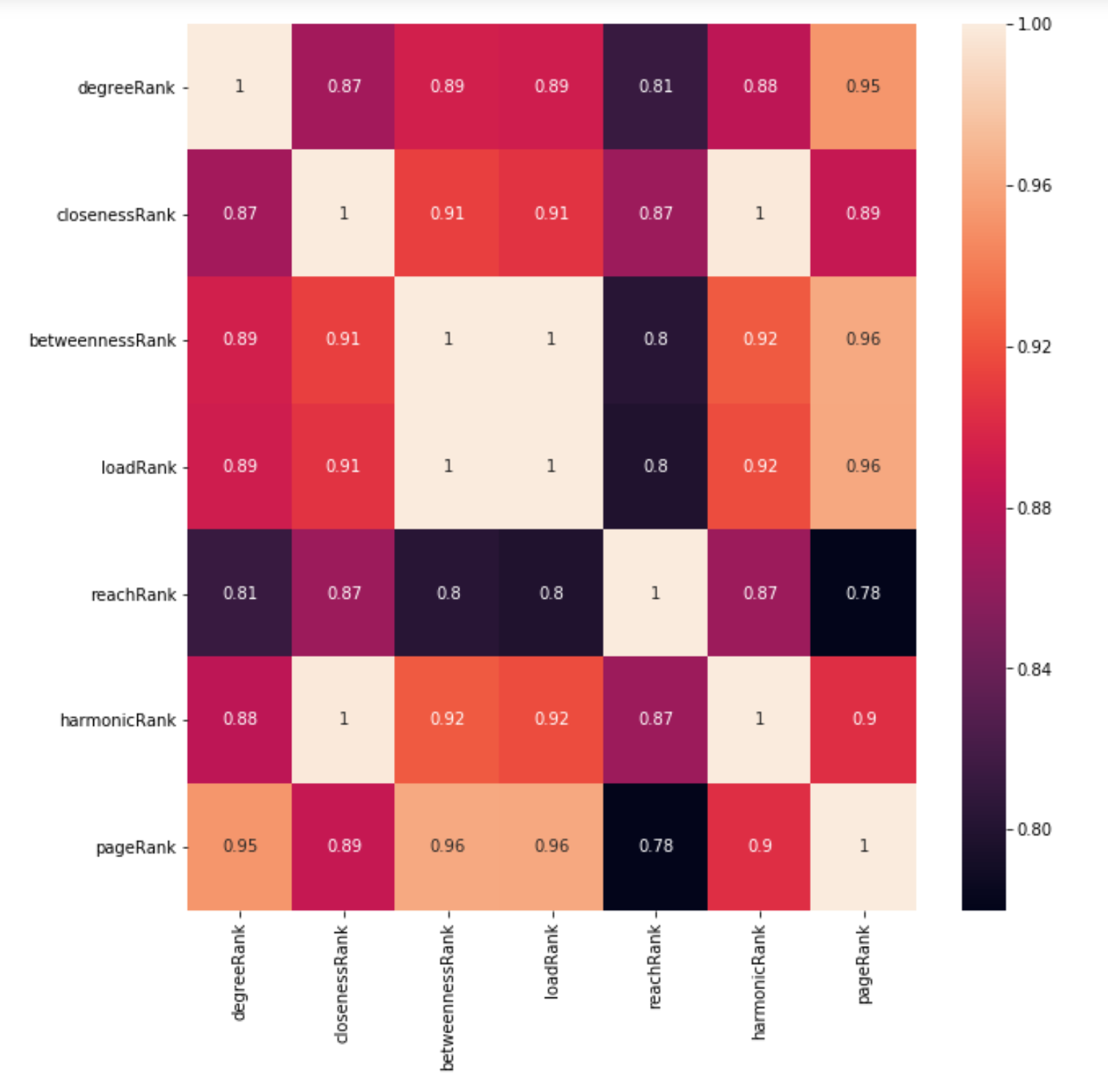


Fig. 8. Correlation Heatmap

1. Data Preparation  
     
    As with all forms of machine learning, in order to produce accurate results it is important that we thoroughly clean the data. The process is rather consistent across each separate machine learning process we use, and will be detailed in this section.  
    The first part we must go about doing is creating the class outcome columns. The first step of this is to create a dictionary with the max ranking node for each network and centrality type. From there we simply create a binary list for each network type, where the list is comprised of 1 if a network is within the top 25%, and a 0 otherwise.  
      
    Following from that, we must simply split the data and balance it. We split the data with an 80:20 training:testing ratio. The balancing is a simple process of removing the majority until it has the same population as the minority.  
    Finally, we must split the training data into 10 training and validation folds.
2. Machine Learning  
     
    Our machine learning process is completed via SKLearn, where we use three different classifiers. The classifiers used are the random forest classifier, decision tree classifier, and the support vector classifier. These classifiers are all supplied via SKLearn.
3. **References:**

[1] Koschützki, Dirk, and Falk Schreiber. “Centrality Analysis Methods for Biological Networks and Their Application to Gene Regulatory Networks.” Gene Regulation and Systems Biology, vol. 2, 2008, doi:10.4137/grsb.s702.

[2] Sparrow, Malcolm K. “The Application of Network Analysis to Criminal Intelligence: An Assessment of the Prospects.” Social Networks, North-Holland, 3 July 2002, https://www.sciencedirect.com/science/article/pii/037887339190008H.

[3] Kuzubaş, Tolga Umut, et al. “Network Centrality Measures and Systemic Risk: An Application to the Turkish Financial Crisis.” Physica A: Statistical Mechanics and Its Applications, North-Holland, 12 Mar. 2014, https://www.sciencedirect.com/science/article/pii/S0378437114002003.

[4] Boldi, Paolo, and Sebastiano Vigna. "Axioms for centrality." *Internet Mathematics* 10.3-4 (2014): 222-262.

[5] Brandes, Ulrik. "On variants of shortest-path betweenness centrality and their generic computation." *Social Networks* 30.2 (2008): 136-145.

[6] Brandes, Ulrik, Stephen P. Borgatti, and Linton C. Freeman. "Maintaining the duality of closeness and betweenness centrality." *Social Networks* 44 (2016): 153-159.

[7] Freeman, Linton C. "Centrality in social networks conceptual clarification." *Social networks* 1.3 (1978): 215-239.

[8] Goh, K-I., Byungnam Kahng, and Doochul Kim. "Universal behavior of load distribution in scale-free networks." *Physical Review Letters* 87.27 (2001): 278701.

[9] Langville, Amy N., and Carl D. Meyer. "A survey of eigenvector methods for web information retrieval." *SIAM review* 47.1 (2005): 135-161.

[10] Mones, Enys, Lilla Vicsek, and Tamás Vicsek. "Hierarchy measure for complex networks." *PloS one* 7.3 (2012): e33799.

[11] Opsahl, Tore, Filip Agneessens, and John Skvoretz. "Node centrality in weighted networks: Generalizing degree and shortest paths." *Social networks* 32.3 (2010): 245-251.

[12] Page, Lawrence, et al. *The PageRank citation ranking: Bringing order to the web*. Stanford InfoLab, 1999.

[13] Rochat, Yannick. *Closeness centrality extended to unconnected graphs: The harmonic centrality index*. No. CONF. 2009.