

Centralidade_Questionario_BrunoFBessa_5881890

May 19, 2021

0.1 SME0130 - Redes Complexas

Structure of networks: Network Centrality

Professor: Francisco Aparecido Rodrigues, francisco@icmc.usp.br. Estudante: Bruno F. Bessa (num. 5881890), bruno.fernandes.oliveira@usp.br Universidade de São Paulo, São Carlos, Brasil.

```
[1]: import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import math
from scipy import stats
```

```
[2]: def get_graph_from_data_file(file_name='lesmis.txt', ncols=3):

    '''
    Defines a NetworkX graph based on data from file.
    Plots a visual representation of the graph
    '''

    file_path = 'data/' + file_name

    if ncols == 2:
        G = nx.read_edgelist("data/powergrid.txt", nodetype=int)
    else:
        G = nx.read_edgelist(file_path, nodetype=int, data= (('weight', float),))

    pos = nx.spring_layout(G)
    nx.draw(G, pos, node_color='b', node_size=50, with_labels=False)

    G = G.to_undirected()
    G.remove_edges_from(nx.selfloop_edges(G))
    Gcc = sorted(nx.connected_components(G), key=len, reverse=True)
    G = G.subgraph(Gcc[0])
    G = nx.convert_node_labels_to_integers(G, first_label=0)

    return G
```

```
[3]: def get_correlation_plot(x, y, x_label, y_label):

    fig= plt.figure(figsize=(10,6))
    plt.plot(x, y, '-o', color='gray', markersize=10, linewidth=0,
             markerfacecolor='lightgray',
             markeredgecolor='black',
             markeredgewidth=2)

    pearson_corr = np.corrcoef(x, y)[0,1]
    spearman_corr, spearman_pval = stats.spearmanr(x, y)
    plt.title("Corr. plot for {} and {}".format(x_label, y_label), fontsize=20)
    plt.ylabel(x_label, fontsize=15)
    plt.xlabel(y_label, fontsize=15)

    plt.show()
    print("Pearson corr.: {:.2f}, Spearman corr.: {:.2f} (p-val: {:.3f})".
    ↪format(pearson_corr,

    ↪        spearman_corr,

    ↪        spearman_pval))
```

0.2 Questions

1 - A média da medida eigenvector centrality da rede USairport500 é igual a quanto? Considere apenas o maior componente.

A medida “eigenvector centrality” é proporcional ao número de visitas em cada nó em um caminho aleatório de tamanho infinito. Um valor alto para a medida indica que o referido nó é conectado a muitos nós que, por sua vez, possuem alto valor de eigenvector centrality.

Sendo A a matriz de adjacência, para um dado vértice i (com vizinhos k), o valor de eigenvector centrality para o este nó é dado por:

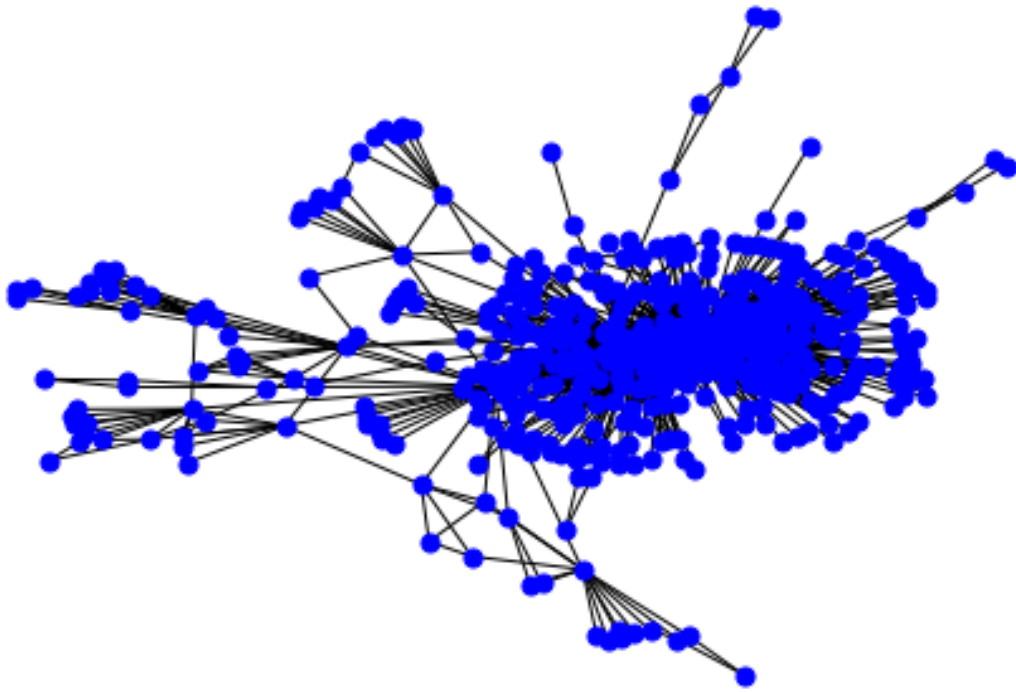
$$x_i = \frac{1}{\lambda} \sum_{k=1}^N A_{k,i} x_k$$

Esta medida está implementada na função “eigenvector_centrality” da biblioteca NetworkX.

```
[4]: G = get_graph_from_data_file('USairport500.txt')

x = list(dict(nx.nx.eigenvector_centrality(G, max_iter=1000)).values())
print('Média de eigenvector centrality para a rede USairport500 é {:.3f}'.
    ↪format(np.mean(x)))
```

Média de eigenvector centrality para a rede USairport500 é 0.023

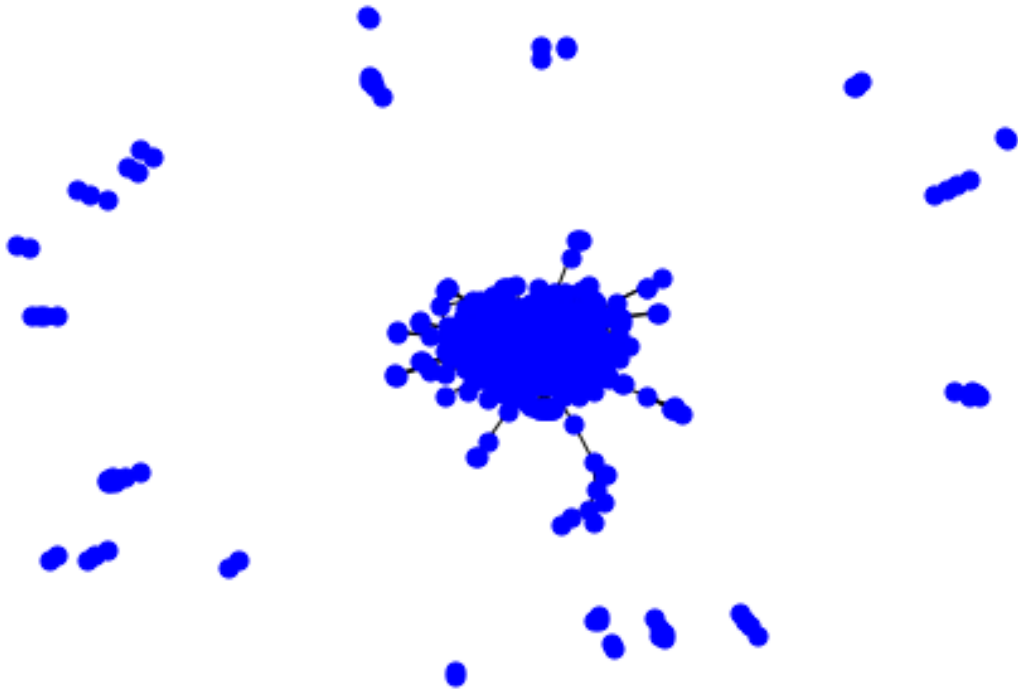


2 - Calcule a correlação de Pearson entre a medida betweenness centrality e grau para a rede hamsterster.

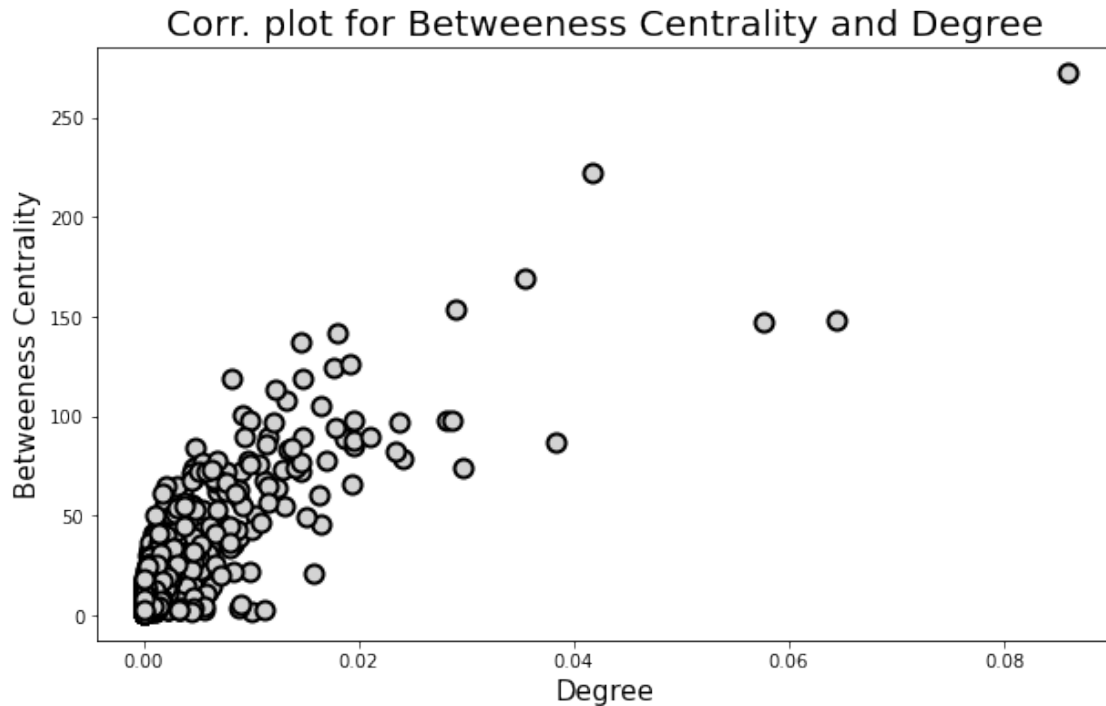
Betweenness centrality a importância de um vértice para comunicação (tráfego) entre diferentes hubs. Considera a quantidade de menores caminhos (η) que passam pelo vértice i conectando dois hubs a e b . Está implementada na função “betweenness centrality” da biblioteca NetwrkX e é dado por:

$$B_i = \sum_{(a,b)} \frac{\eta_{a,i,b}}{\eta_{a,b}}$$

```
[5]: G = get_graph_from_data_file('hamsterster.txt')
```



```
[6]: b = list(nx.betweenness centrality(G).values())  
k = list(dict(G.degree(G)).values())  
  
get_correlation_plot(b, k, "Betweenness Centrality", "Degree")
```



Pearson corr.: 0.82, Spearman corr.: 0.86 (p-val: 0.000)

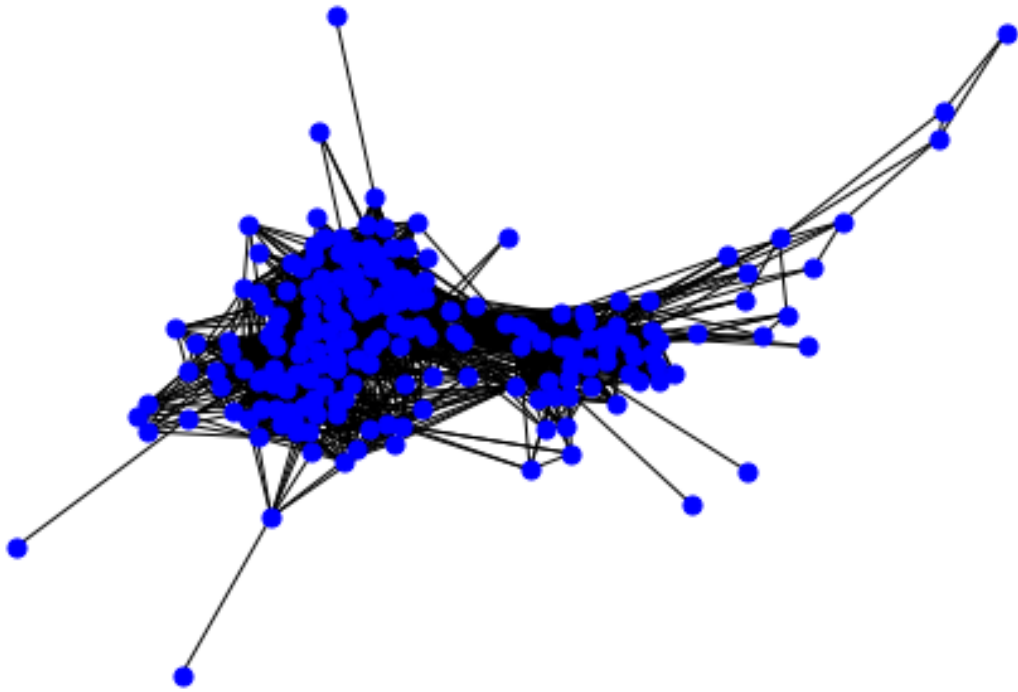
3 - Calcule o coeficiente de correlação de Spearman entre as medidas closenes centrality e k-core para a rede “Jazz”.

A medida k-core de um grafo G é o maior subgrafo conectado (componente) de G em que todos os vértices tenham ao menos grau k . Permite que sejam isolados e identificados os hubs periféricos ao serem eliminadas conexões de menor grau. Com a função “core number” da biblioteca NetworkX recebemos um dicionário com os subgrafos de cada grau k possível.

Closeness centrality é uma medida que prioriza a proximidade (distâncias curtas) de um nó com suas conexões. Está implementada função “closeness centrality” da biblioteca NetworkX e é dado por:

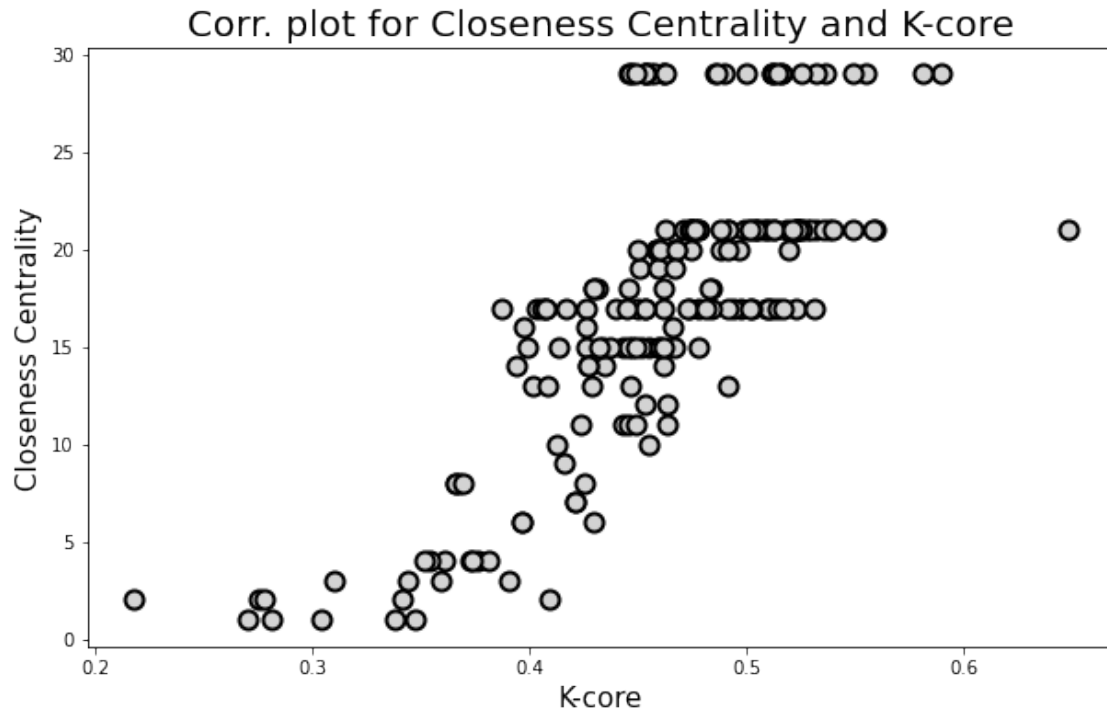
$$C_i = \frac{N}{\sum_{j=1, j \neq i}^N d_{i,j}}$$

```
[7]: G = get_graph_from_data_file('jazz.txt')
```



```
[8]: cc = list(nx.closeness centrality(G).values())
      kc = list(dict(nx.core_number(G)).values())

      get_correlation_plot(cc, kc, "Closeness Centrality", "K-core")
```



Pearson corr.: 0.75, Spearman corr.: 0.74 (p-val: 0.000)

4 - Escolha as medidas de centralidade.

- Usa os menores caminhos de modo a determinar a carga (em termos da quantidade de tráfego) de cada vértice. → Betweenness Centrality
- É calculada a partir do espectro da matriz de adjacência. → Eigenvector Centrality
- É uma medida de centralidade local, pois alguns vértices apresentando altos valores dessa medida podem estar na borda da rede. → Degree
- Um vértice é considerado central se estiver conectado a outros vértices centrais. → PageRank
- É calculada a partir da exponencial da matriz de probabilidade de transição. → Accessibility