

Proyecto de Análisis de Algoritmos

Reporte

Francisco Valente Castro
Vanessa Avalos Piñón

Noviembre 2014

1 Algoritmos de Centralidad

Dada una red social, ¿cuál de sus nodos es el más central? Esta pregunta se ha hecho muchas veces en la sociología, la psicología y ciencias de la computación, y varias medidas de centralidad (también conocidos como los índices de centralidad, o ranking) proponen dar cuentas de la importancia de los nodos de una red.

1.1 Degree

La centralidad de grado de un nodo v en un gráfica se define como

$$C_D = \deg(v),$$

el cual es el número de aristas incidentes a dicho vértice.

1.2 Closeness

La centralidad de cercanía de un nodo v es el recíproco de la suma de las distancias más cortas de v a todos los demás $n - 1$ nodos. Dado que la suma de las distancias depende del número de nodos en el grafo, la cercanía se normaliza por la suma de las distancias mínimas posibles $n - 1$. Se tiene que,

$$C_{Cl}(v) = \frac{n - 1}{\sum_{u=1}^{n-1} d(v, u)}$$

Donde :

- $d(v, u)$ es la distancia del camino más corto entre v y u .
- n es el número de nodos.

1.3 Betweenness

La intermediación del nodo v cuantifica el número de veces que el nodo v actúa como un puente a lo largo del camino más corto entre dos nodos.

La centralidad de intermediación de un nodo v es la suma de de todos los pares de los caminos más cortos que pasan a traves de v .

El algoritmo **Betweenness** calcula el camino más corto para los nodos.

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

Donde :

- V es el conjunto de nodos
- $\sigma(s,t)$ es el número de caminos más cortos entre un nodo s y t
- $\sigma(s,t|v)$ es el número de esos caminos que pasan a través de algún nodo v distinto de s, t

Si $s = t$, $\sigma(s,t) = 1$, y si $v \in s, t$, $\sigma(s,t|v) = 0$.

2 Metodología

2.1 Generar el grafo

Se generan grafos con ayuda de la librería NETWORKX y dos bases de datos libres llamadas MusicBrainz y Echonest. Cada una de estas bases de datos son manejadas desde Python con sus respectivas librerías "pyechonest" , "musicbrainzngs" que se pueden obtener facilmente con el programa de instalación "pip".

Para generar el grafo respecto a un artista se agrega un nodo que representa al artista. Después se utilizan las relaciones entre artistas de la base de datos MusicBrainz para definir a sus nodos adyacentes y agregarlos al grafo. Estas relaciones pueden ser "member of band" , "tribute" , "collaboration" , "sibling" , etc , las que se tomaron como aristas, les predefinimos un peso dependiendo de nuestro criterio (Un peso menor nos dice que los artistas se han relacionado más). Después recursivamente generamos un grafo a partir de los nuevos nodos agregados. Limitamos la profundidad de la búsqueda.

Para generar el grafo respecto a un género se obtienen ciertos números de artistas representantes de un género de la base de datos EchoNest (se limita el número de artistas con un parámetro). Y cada uno de ellos se expande una vez obtenido datos de artistas similares de la misma base de datos (La medida de "similaridad" depende completamente de la base EchoNest).

2.2 Algoritmos de Centralidad

En el algoritmo **Degree** se utiliza una función de NETWORKX,

```
DiGraph.degree_iter(nbunch=None, weight=None)
```

La cual nos regresa una iterador para (node,degree).

Para el algoritmo de **Closeness** se calculan las distancias de los caminos más cortos con una función de NETWORKX

```
nx.single_source_dijkstra_path_length(G, source, cutoff=None, weight='weight')
```

Donde

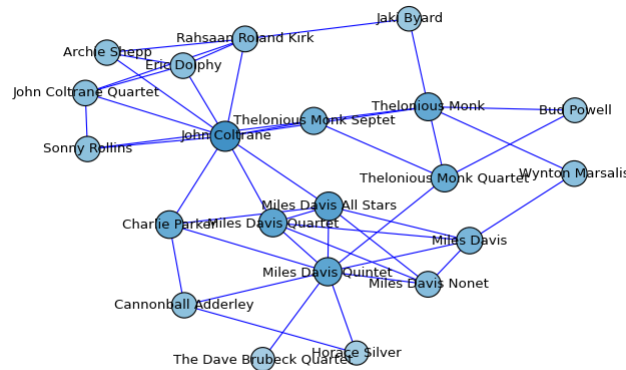
- G : Una gráfica de NetworkX
- $source$: Etiqueta de nodo
- $weight$: Un string opcional (default='weight')
Peso de la arista
- $cutoff$: integer or float, optional
Profundidad para detener el algoritmo.

Finalmente para el algoritmo **Betweenness** se desarrollan dos funciones para encontrar los caminos más cortos, un *BFS* y el algoritmo de *Dijkstra* y se realizan funciones para sumar dichas distancias y normalizarlas.

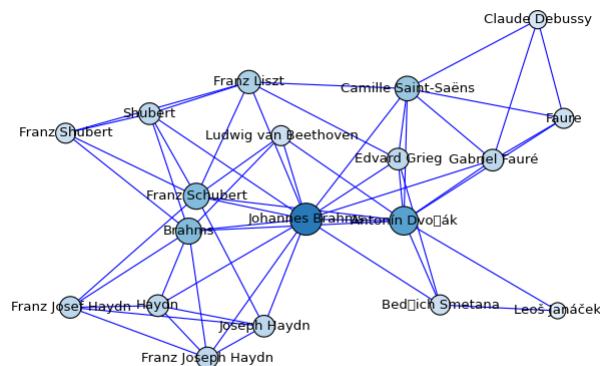
Los algoritmos regresan un diccionario de nodos con medidas centralidad como valores.

3 Resultados

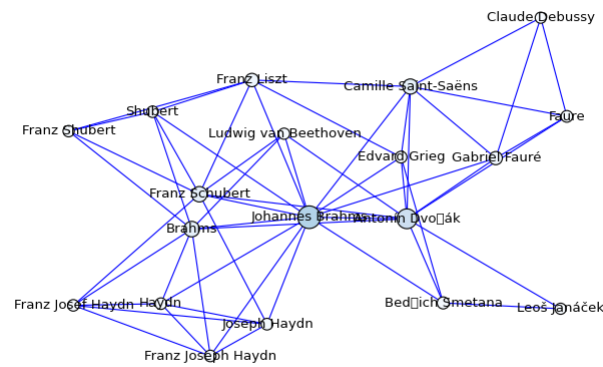
Closeness



Degree



Betweenness



4 Conclusiones

Según la gráfica que se tenga, se determinara qué algoritmo de centralidad es más conveniente para encontrar los nodos más relevantes.

5 Referencias

- <http://static.echonest.com/SixDegreesOfBlackSabbath/>
- <http://musicmachinery.com/2010/05/20/six-degrees-of-black-sabbath/>
- <http://www.inf.uni-konstanz.de/algo/publications/b-vspbc-08.pdf>
- <http://networkx.lanl.gov/reference/generated/>
- <https://networkx.github.io>