Redes bi-partidas

Nuestro objeto de estudio de partida son redes bi-partidas. Las redes bi-partidas son una clase de relojes complejas en las que el conjunto de nodos de la red está dividido en dos clases (o modos), A y B, y solo hay enlaces entre nodos de clases distintas. En nuestro caso las dos clases serian

A: amplificadores de información o audiencia.
B: fuente de información relevante.

Un caso clásico viene de analisis de audiencias de medios de información. Por ejemplo, los nodos de clase B serían paginas web y los nodos de clase A serían los usuarios que han visitado las páginas. Se pone un enlace entre el nodo que representa un usuario e un nodo que representa una pagina web, si ese usuario ha visitado esa pagina.

En el caso concreto de conversaciones en Twitter, se suele hablar de crowdsourced elites, es decir, actores que en la conversación han recibido un número de retuits relevante y por lo tanto se pueden considerar como los agentes más influyentes en la conversación. Estos serían nuestros nodos de clase B. Los actores que retuitean a estos son los amplificadores, nuestros nodos de clase A.

Una red bi-partida puede ser transformada en una red de una sola clase de nodos. Acción que se llama “proyección”. Es decir, podemos proyectar una red-bipartida sobre los nodos de la clase A o de la clase B, eligiendo la regla con la que ponemos enlaces entre nodos.

En el caso del analisis de audiencia, se proyecta sobre los nodos de clase B y el enlace entre dos nodos de clase B tiene un peso que representa el tamaño de la audiencia compartida entre los dos medios. En el caso de conversaciones en Twitter queremos hacer lo mismo. Hablamos en este caso de audience overlap networks. El paper de referencia es este[1 https://academic.oup.com/joc/article-abstract/68/1/26/4858530]:

Audience overlap network

SIguiendo a [1] nos calculamos la matrix de correlación phi en la que cada elemento phi_ij es el coeficiente phi calculado para los nodos i y j, que representa la correlación estadistica entre sus audiencias. Llegado aquí, a diferencia de [1], no queremos hacer un tresholding de la matriz phi paraa construir una red, sino que queremos manipular

de alguna manera esta matriz para utilizarla como matriz de distancias en un algoritmo de clustering jerárquico.

Clustering jerárquico

El clearing jerárquico es un metodo estadistico de analisis de cluster cuyo objetivo es construir una jerarquia de clusters. Hay dos estrategias posibles: aglomeravias y divisivas.

Las estrategias aglomerantes son estrategias bottom up que empiezan con un conjuntos de datos que van aglomerando a cada paso en clusters anidados.

Las divisivas son top down que empieza considerando todo el conjunto de datos como un cluster que van partiendo.

Nosotros nos centraremos en las estrategias aglomerantes.

Para decidir que clusters tienen que ser juntados en un único cluster en cada paso hay que definir una medida de disimilaridad o de distancia entre conjuntos de datos. Hay varias métricas propuestas en literatura (centroides, ward, etc.)

Clustering jerarquico en audience overlap networks.

Nuestra audience overlap network está codificada por la matriz de correlación phi. Esta matriz no puede ser usada directamente como medidas de disimilaridad, entre otras razones porque no está definida positiva.

En realidad, por razones que ya discutiremos en detalle, también nos interesa que nuestra medida de disimilaridad sea efectivamente una distancia.

Se puede demostrar que a partir del coeficiente phi_ij podemos obtener la distancia euclídea entre i y j via la formula

d_ij=\sqrt{2(1-phi_ij)}.

Esta es la distancia que queremos utilizar.

Una vez que tengo la matriz de distancia puedo aplicar el algoritmo de clustering jerárquico que quiera.

Polarización

Si tengo un conjunto de actores divididos en bloques, el grado de polarizacion del sistema se puede medir como [2ref. https://www.jstor.org/stable/2951734?casa_token=0P_uOoyPO_gAAAAA%3AtG6gcx-Ykt0P4ndySiKcEY2OtKi5kMiCt5m-pZjpC2bzcbJN8f1S2s8ybpaBo244ld3IDSln6aGsFGnHnISM9ATPjmTfxqda9OW5wNsBRIPYyLuGdfw&seq=13#metadata_info_tab_contents].

P=\sum_{uv}\pi_u^{1+\alpha}\pi_v d(u,v)

Donde la suma es sobre todo los bloques, \pi_u es el tamaño del bloque u, d(u,v) es la distancia entre los bloques u y v, y alpha \in (0,1.6].

Lo que hacemos....

Hasta ahora, lo que hemos hecho ha sido

Construir la jerarquia de cluster de la audience overlap network tomando la distancia definida arriba y un metodo de clustering dado (en concreto ward). Luego calculamos la polarización tomando como distancia entre los clusters la distancia final en el dendograma producido por el algoritmo.

Lo que queremos explorar en tu tfm...

Un algoritmo de cluster a cada paso tiene que tener una regla para calcularse las distancia entre el nuevo cluster que ha creado y los demás. La idea es utilizar directamente los elementos de la medida de polarización.

Es decir, si he formado en nuevo cluster u, la distancia entre u y v será

\pi_u^{1+\alpha}\pi_vd(c_u,c_v) + \pi_v^{1+\alpha}\pi_ud(c_u,c_v)

Donde c_u y c_v son los centroides del cluster u y del cluster v respectivamente.

A cada paso, el algoritmo, para decidir que cluster formar tomará el que tenga menor distancia con los que ya existen.