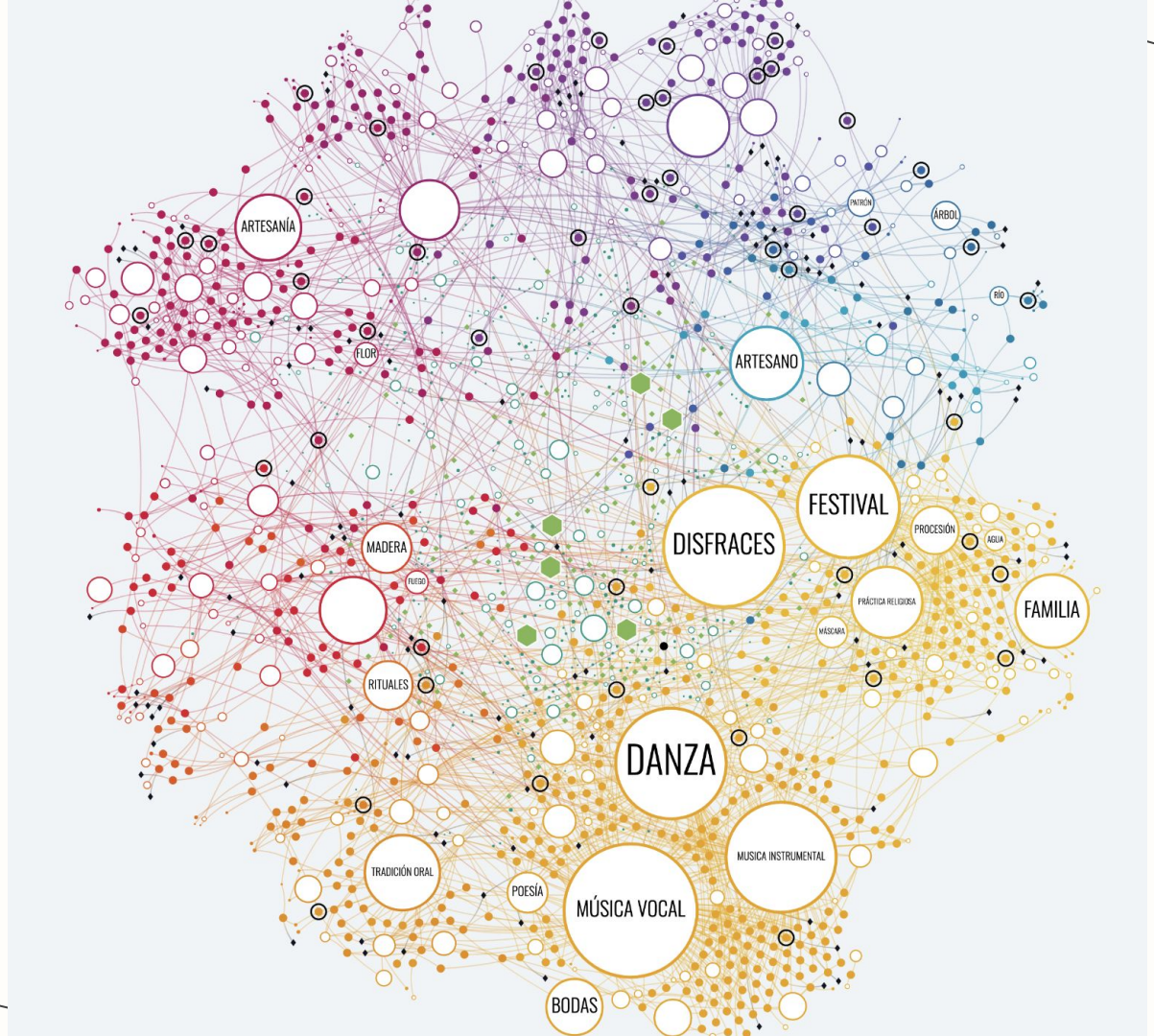
Búsqueda de Comunidades

Alondra Berzunza

Detección de Comunidades

La detección de comunidades, también conocida como partición de grafos, hace referencia al conjunto de técnicas no supervisadas utilizadas para encontrar grupos de nodos altamente interconectados. Más concretamente, la detección de comunidades trata de identificar subconjuntos de nodos que están muy conectados entre ellos y, al mismo tiempo, poco conectados con el resto de nodos de la red.

La detección de comunidades tiene multitud de aplicaciones, tales como la identificación de grupos de personas con intereses similares, la clasificación de proteínas con funciones relacionadas y la agrupación de sitios web con temas comunes.


ARTESANÍA
FLOR
MADERA
FUEGO
RITUALES
TRADICIÓN ORAL
POESÍA
BODAS
MÚSICA VOCAL
DANZA
MUSICA INSTRUMENTAL
DISFRACES
ARTESANO
PATRÓN
ÁRBOL
RÍO
FESTIVAL
PROCESIÓN
AGUA
PRÁCTICA RELIGIOSA
MÁSCARA
FAMILIA

Detección de Comunidades

La Detección de Comunidades es como encontrar amigos en una gran multitud.

Imagínate que entras a una habitación llena de gente y quieres saber quién conoce a quién. La detección de comunidades nos ayuda a ver qué personas se juntan más entre sí que con otras. En una red, algunos nodos (personas) tienen conexiones más fuertes con ciertos nodos que con otros.

Eso es lo que trata de averiguar la detección de comunidades.

Importancia

Entender cómo se forman y funcionan las comunidades puede ayudar en muchas áreas:

- **Salud**: Encontrar grupos de pacientes con enfermedades similares.
- **Ecología**: Rastrear poblaciones de animales en diferentes hábitats.
- **Aprendizaje Automático**: Mejorar los algoritmos al saber cómo se agrupan los datos.
- **Descubrimiento de Medicamentos**: Identificar objetivos para nuevos medicamentos.
- **Investigación del Cerebro:** Estudiar cómo interactúan diferentes partes del cerebro.

Al identificar comunidades, podemos obtener ideas sobre cómo se comportan los sistemas, lo que puede llevar a mejores decisiones, estrategias e incluso datos curiosos sobre dinámicas sociales.

Desafío de la Detección de Comunidades

No siempre sabemos cuántas comunidades hay o cuán grandes son. Es como intentar adivinar cuántas gomitas hay en un frasco sin mirar. Debido a esta incertidumbre, los investigadores suelen usar métodos basados en suposiciones (heurísticas) para abordar el problema.

Una forma popular de medir qué tan buena es una comunidad implica algo llamado "Modularidad." Una modularidad más alta es mejor, ya que muestra conexiones fuertes dentro de la comunidad pero conexiones más débiles hacia el exterior.

Detección de Comunidades

En la actualidad, la detección de comunidades es uno de los tópicos con mayor importancia por las aplicaciones que puede tener en redes sociales, economía, marketing, política, búsqueda de información, minería de datos, adquisición de conocimiento, análisis de datos, informática, física, sociología, etc.

Algunas de las aplicaciones se han combinado para desarrollar un papel principal en los últimos años, como el caso de las redes sociales y la política. Por estos motivos, la detección de comunidades es un problema ampliamente investigado, abierto y de relevancia.

Este problema puede ser modelado como uno de optimización multiobjetivo, ya que comúnmente se plantean varios objetivos que están en conflicto para lograr un buen conjunto de soluciones, en particular, se modela como un problema de partición de un grafo.

Detección de Comunidades

El problema de agrupamiento trata de encontrar la mejor partición C de un grafo G(V,A) con nodos V y aristas A. Algunos algoritmos hacen suposiciones a priori de la estructura del grafo y esto puede afectar el rendimiento y la calidad de las soluciones.

El problema de detección de comunidades en redes sociales lo abordamos desde una estrategia multiobjetivo por la naturaleza de las redes sociales, ya que es difícil encontrar una red social representada por grafos disjuntos, lo común es tener una red social representada por un grafo conexo que contenga la mayoría de las relaciones entre todos los nodos.

En el caso de que pudiéramos garantizar que el grafo es disjunto, podría aplicarse un método de agrupamiento de un solo objetivo.

Pero en el caso de un grafo conexo, es deseable tener más de una métrica para medir la calidad del agrupamiento. A veces es necesario agregar nuevos objetivos para elevar la calidad de los resultados.

Detección de Comunidades

Usualmente, el problema de detección de comunidades trata de encontrar, en un grafo $G(A,V)$, una partición C compuesta por k conjuntos disjuntos de nodos v tal que cada $v \in V$, que minimice el número de aristas a entre grupos y maximice el número de aristas a dentro de cada grupo para cada $a \in A$. La ecuación (1) muestra el primer objetivo a maximizar, que es una modificación a la ecuación (2) $Mcut$, ya que Mcut por sí sola debe ser minimizada:

$$f_1 = \frac{1}{1+Mcut}, \tag{1}$$

$$Mcut_k = \frac{cut(C_1, \bar{C}_1)}{W(C_1)} + \frac{cut(C_2, \bar{C}_2)}{W(C_2)} + \cdots + \frac{cut(C_k, \bar{C}_k)}{W(C_k)}, \tag{2}$$

$$W(C_i) = cut(C_i, C_i), \tag{3}$$

$$cut(C_i, \bar{C}_i) = \sum_{u \in C_1, v \in \bar{C}_1} W_{uv}, \tag{4}$$

para toda $i \in \{1,...,k\}$ y $k > 1$. $f1$ toma valores en el intervalo (0,1], ya que $Mcutk$ toma valores en el intervalo $[0, \infty)$

Detección de Comunidades

El mejor caso de $Mcutk$ ocurre cuando la ecuación (4) de cada termino es igual a cero, es decir, $cut(Ci,\bar{C}\ i) = 0$, para toda $i \in \{1,...,k\}$.
Esto quiere decir que no existe ninguna arista a que vaya desde algún nodo a otro nodo de otro grupo, lo cual puede verse como que ninguna arista es cortada por la agrupación.
El caso neutral está dado cuando el número de aristas entre grupos es igual al número de aristas internas de cada grupo, con lo cual cada término $cut(Ci,\bar{C}\ i)/W(Ci) = 1$, teniendo como resultado $Mcutk = m$, en donde m es el número de nodos de G. Finalmente, se tendría que $f1 = 1/1+m$.

Esta observación es importante ya que $f1$ toma valores en el intervalo (0,1] y sería fácil pensar que el caso neutral estaría dado por $f1 = 0.5$.
La última consideración con respecto a esta función se muestra en la Fig. 1 y hace evidente un caso especial en la ecuación (3) y por el cual se considera $Wuu = 1$, para toda $u \in \{1,...,m\}$.

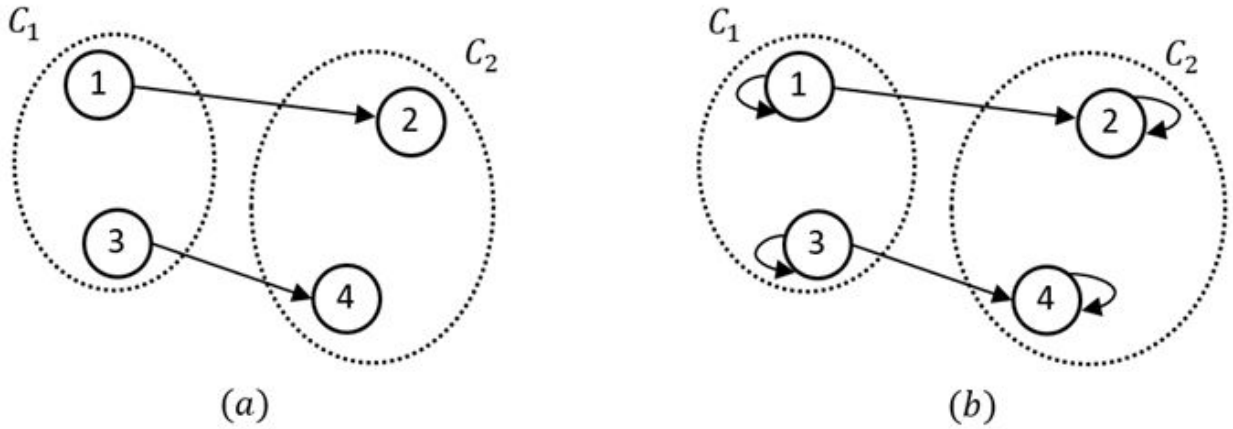


Fig. 1. En (a) se muestra el caso especial de un agrupamiento donde $W(C_1) = 0$ y $W(C_2) = 0$, ya que para C_1 y C_2 no hay relaciones internas. Para evitar la división entre cero en los términos de $Mcut_k$ se considera implícitamente el bucle $W_{uu} = 1$, para todos los nodos de G como se muestra en (b), de esta forma se tiene $W(C_1) = 2$ y $W(C_2) = 2$.

Detección de Comunidades

El segundo objetivo por maximizar es Global Silhouette, está definido en la ecuación (5):

$$f_2 = Global\ Silhouette\ = \frac{\sum_{i \in V} S(i)}{|V|}, \tag{5}$$

$$S(i) = \begin{cases} \frac{a(i)-b(i)}{\max\{a(i),b(i)\}}, para\ |C_l| > 1 \\ 0, para\ |C_l| = 1 \end{cases}, \forall\ i\ \in \{1, \dots, |V|\}, \tag{6}$$

$$a(i) = \frac{\sum_{j \in C_l} W_{ij}}{|C_l|}, i\ \in C_l, \tag{7}$$

$$b(i) = \max\{d(i, C_m)\}, m\ \in \{1, \dots, |C|\}, m \neq l, \tag{8}$$

$$d(i, C_m) = \frac{\sum_{j \in C_m} W_{ij}}{|C_m|}, \tag{9}$$

Detección de Comunidades

f2 toma valores en el intervalo (−1,1], ya que cada término de la ecuación (5) también toma valores en el intervalo (−1,1] y $f2$ es el promedio los mismos. En la ecuación (7) se calcula el promedio de aristas dentro del grupo al que pertenece el nodo i. La ecuación (8) selecciona el promedio máximo de aristas que existen entre el nodo i y el resto de los grupos, cada promedio es calculado con la ecuación (9). El mejor caso de $Global\ Silhouette$ ocurre cuando la ecuación (6) toma el valor de uno y la ecuación (8) toma el valor de cero, es decir, $S(i) = 1$, dado $b(i) = 0$, para toda $i \in \{1,...,|V|\}$.

Esto quiere decir que no existe ninguna arista a que vaya desde algún nodo a otro nodo de otro grupo. El caso neutral está dado cuando el número de aristas entre grupos es igual al número de aristas internas de cada grupo, con lo cual para cada término $S(i)$ de la ecuación (5), $a(i) = b(i)$, teniendo como resultado $Global\ Silhouette = 0$. Para el caso que el grupo Cl solo contará con un nodo se le asigna a $S(i)$ el valor cero.

La última consideración con respecto a esta función se muestra en la Fig. 2 y hace evidente un caso especial por el cual, al igual que el primer objetivo, se considera

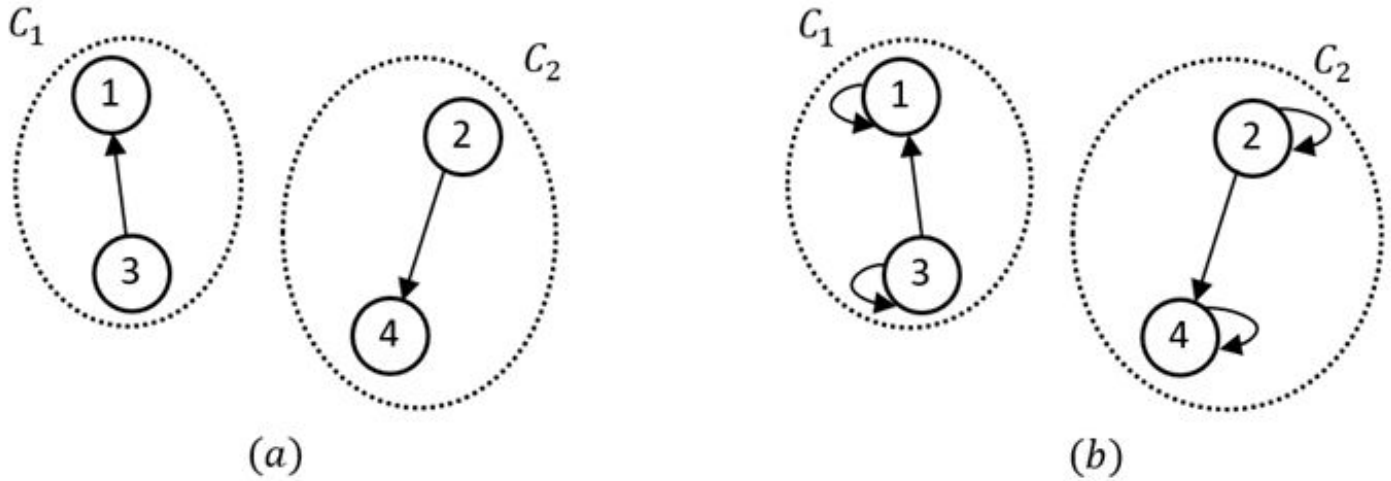


Fig. 2. En (a) se muestra el caso especial de un agrupamiento donde tenemos $a(1) = 0$ y $b(1) = 0$, es decir, que el nodo 1 no tiene relación con ningún otro, ya sea de su mismo grupo o de otro grupo. Para evitar la división entre cero en $S(i)$, se considera implícitamente el bucle $W_{uu} = 1$, para todos los nodos de G como se muestra en (b), en donde tenemos $a(1) = 0.5$ y $b(1) = 0$.

Algoritmos de Detección de Comunidades

Existen diferentes métodos de detección de comunidades dependiendo del tipo de definición con la cual se defina la comunidad (local, global y basada en similitud). Los métodos también se pueden diferenciar por el método de solución del problema, los cuales pueden ser enfoques jerárquicos, modulares o espectrales. Además, se puede considerar otra clasificación en función de si permiten detectar comunidades sin solapamiento o con solapamiento.

Algoritmos de Detección de Comunidades

- **Comunidades sin solapamiento:** Este tipo comunidades se caracteriza por la generación de una cantidad determinada de comunidades, en las cuales cada individuo sólo pueden formar parte de una comunidad. Este tipo de comunidades se caracterizan por la presencia explícita de una jerarquía.

- **Comunidades con solapamiento:** En este tipo de comunidades los individuos pueden formar parte de más de una comunidad. Por ejemplo, las redes sociales un individuo puede formar parte de la comunidad “familia” y a su vez hacer parte de la comunidad “amigos”.

Está clasificación no es totalmente excluyente; es decir, se pueden presentar métodos bajo un enfoque modular que considere solapamientos o un método jerárquico sin solapamientos.

Algoritmo de Girvan–Newman.

Este método detecta comunidades eliminando iterativamente las aristas con mayor centralidad de intermediación. Si bien es eficaz para identificar estructuras comunitarias, es computacionalmente costoso y no escala bien a redes grandes.

Algoritmo rápido de Newman (Newman-Girvan): Este método es una extensión del algoritmo de Girvan-Newman y optimiza la modularidad de forma más eficiente. Utiliza un enfoque voraz para dividir el grafo en comunidades mediante el análisis de la puntuación de modularidad de las diferentes particiones.

Método de Newman y Girvan

Uno de los algoritmos más populares es el introducido en el trabajo de Newman y Girvan que utilizan la medida de intermediación o betweenness.

De manera general, esta medida consiste en identificar las aristas que cumplen la condición de puente entre cualquier par de nodos logrando la menor cantidad de saltos entre ellos. Es de destacar que es muy importante este método porque marcó el inicio del estudio de detección de comunidades en el área de la Física.

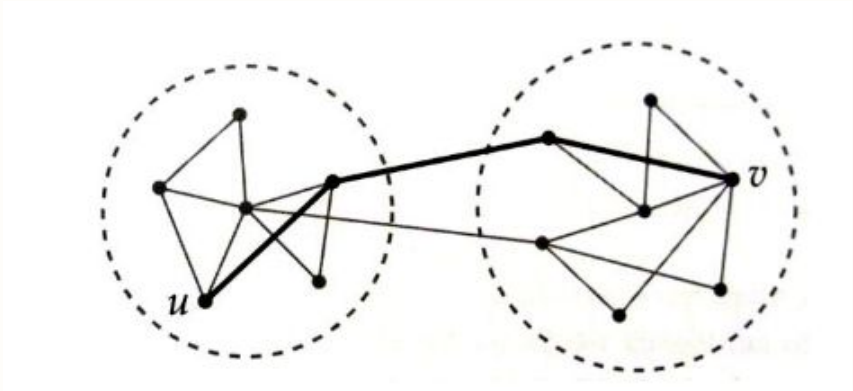
Método de Newman y Girvan

De manera general, los pasos del algoritmo son:

1. Calcular el índice de betweenness de todas las aristas de la red.
2. Eliminar las aristas con mayor índice de betweenness (en caso de empate, se escoge una de manera aleatoria).
3. Recalcular el índice de betweenness en toda la red.
4. Retornar al paso 2.

Método de Newman y Girvan

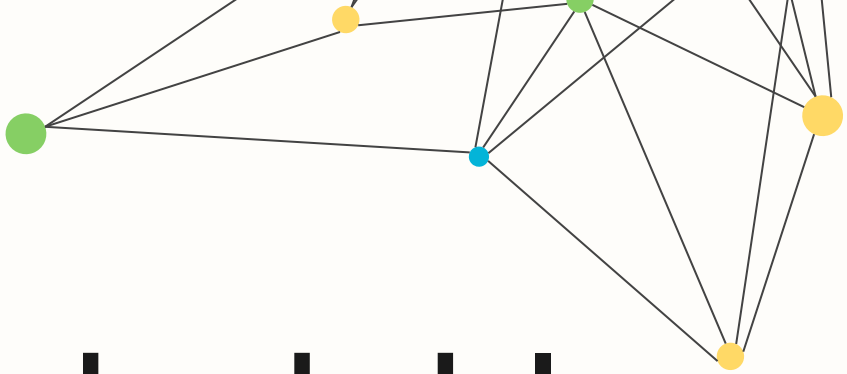
De una manera simplificada, su objetivo es encontrar aquellas aristas por las cuales existe un mayor “tráfico” o nivel de conexión entre nodos de la red que forman parte de diferentes comunidades. Cada vez que se encuentra una arista con altos valores de índice de betweenness esta se elimina y luego se recalcula el índice para todas las aristas de la red. Este proceso continuará de manera iterativa hasta que sobre la red se desarrollen subgrafos que se encuentren aislados del resto de la red. Este proceso se repite recursivamente sobre cada subgrafo hasta la construcción del dendrograma completo.



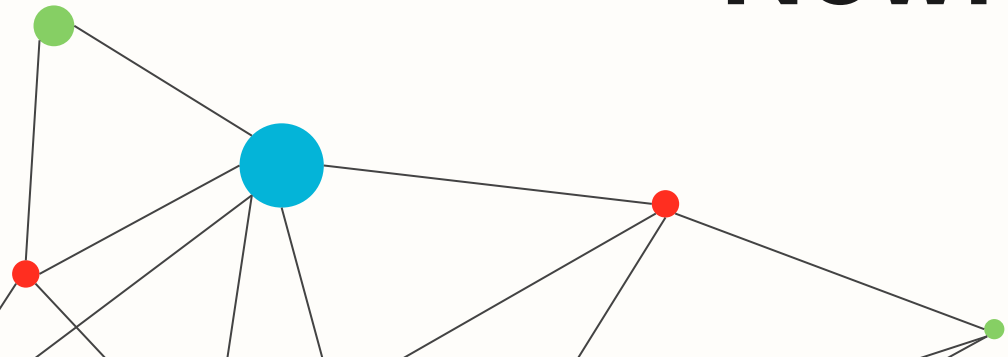
Método de Newman y Girvan

Uno de los grandes inconvenientes de este tipo de algoritmos es su complejidad computacional, la cual puede variar dependiendo del método empleado para evaluar la medida de betweenness.

Por lo general, esta complejidad está en el orden de $O(mn(m+n))$ dado un grafo G con m aristas y n nodos, por lo que este método no se aconseja en uso de redes con una alta cantidad de nodos y aristas. A pesar de estos inconvenientes, cabe resaltar los buenos resultados que proporciona el algoritmo.



Revisar el Notebook del Algoritmo de Girvan Newman



Algoritmo de Louvain

El algoritmo de Louvain tiene como objetivo optimizar un concepto matemático denominado modularidad, una medida que se utiliza en la teoría de grafos para evaluar la calidad de una partición.

La modularidad se define como la diferencia entre el número de enlaces observado dentro de las comunidades y el número de enlaces esperados por azar. Una alta modularidad indica que las comunidades encontradas tienen más conexiones que las esperadas por azar y, por tanto, son compactas y bien definidas.

La modularidad tiene un rango de valores entre -0.5 y 1 y se define mediante la siguiente fórmula:

$$M = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

Algoritmo de Louvain

dónde

- A_{ij} matriz de adyacencia (nodos i y j). Puede ser de un grafo ponderado.
- k_i y k_j son la suma de los pesos de la matriz de adyacencia de los ejes que conectan a los nodos i y j. Si el grafo es no ponderado, equivale al grado del nodo, es decir, el número de conexiones.
- m es la suma de todos los pesos de la matriz de adyacencia. En un grafo no ponderado, es igual al número de ejes (L).
- c_i y c_j son las comunidades a las que pertenecen los nodos i y j.
- δ es la función delta de Kronecker. Tiene valor 1 si los nodos i y j pertenecen a la misma comunidad y 0 en caso contrario. Por lo tanto, la fórmula sólo aplica si los nodos pertenecen a la misma comunidad.

Para un grafo no ponderado, la modularidad se puede simplificar a la siguiente fórmula:

$$M = \frac{1}{2L} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2L} \right] \delta(c_i, c_j)$$

dónde L es el número total de enlaces del grafo.

Algoritmo de Louvain

Una forma sencilla de entender la ecuación es la siguiente:

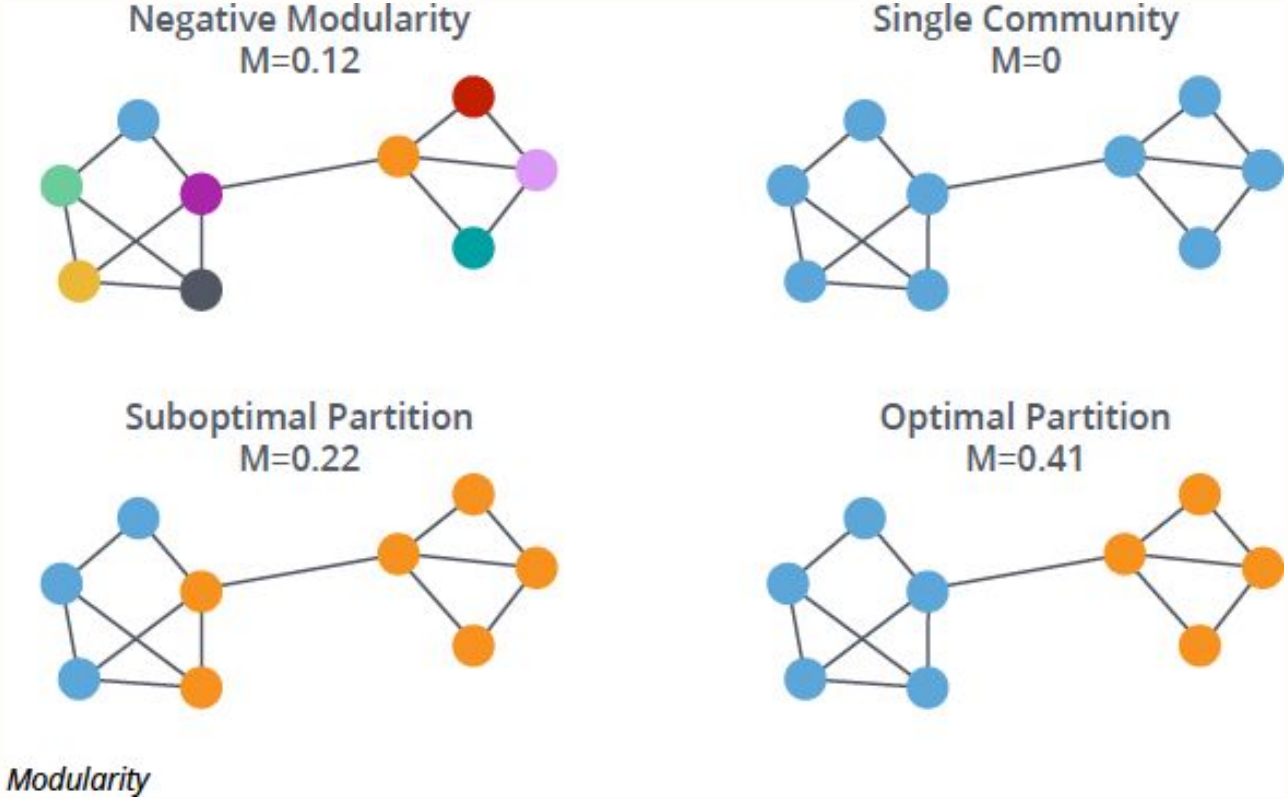
1. Para encontrar las comunidades de mayor calidad, se quiere maximizar la modularidad.
2. La fórmula únicamente toma valores distintos de cero cuando los nodos pertenecen a la misma comunidad, de lo contrario el delta de Kroneker vale 0.
3. Maximizar la ecuación implica que el primer término debe ser lo más alto posible, mientras que el segundo término debe ser lo más bajo posible.
4. ∑ijAij es el número de enlaces existentes dentro de una comunidad.

Algoritmo de Louvain

5. El segundo término de la ecuación representa el número de enlaces esperados por azar entre dos nodos. El número de enlaces posibles entre dos nodos es proporcional al producto de sus grados. Por ejemplo, si el grado de alguno de los nodos es 0, el número de enlaces esperados por azar es 0 (ya que uno de los nodos no tiene enlaces). El producto de los grados se divide por el número total de enlaces (2L) para convertirlo en una proporción.

6. Si el número de enlaces de la comunidad es mayor que el esperado por azar, el valor de la modularidad es positivo. Cuanto mayor sea la diferencia entre el número de enlaces dentro de la comunidad y el número de enlaces esperados por azar, mayor la modularidad.

Algoritmo de Louvain



Algoritmo de Louvain

Su funcionamiento se divide en dos fases: una fase de agrupamiento y una fase de refinamiento.

Fase de agrupamiento

En la fase de agrupamiento, cada nodo se asigna a su propia comunidad. Luego, se itera sobre cada nodo del grafo y se calcula la ganancia en modularidad que se obtendría al mover el nodo a una comunidad diferente. Si se obtiene una ganancia positiva, se mueve el nodo a la comunidad correspondiente. Esto se repite hasta que ya no se pueden obtener ganancias adicionales.

Algoritmo de Louvain

Fase de refinamiento

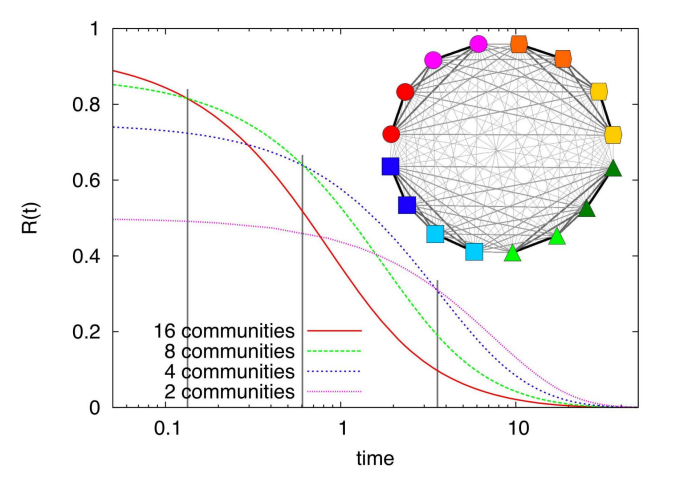
La segunda fase consiste en construir una nueva red cuyos nodos son las comunidades encontradas en la primera fase. Los pesos de los enlaces entre los nuevos nodos se calculan sumando del peso de los enlaces entre los nodos de las dos comunidades correspondientes. Al final de esta fase, se obtiene una partición del grafo con comunidades más compactas y mejor definidas que las obtenidas en la fase de agrupamiento.

Las dos fases anteriores se ejecutan hasta que no se obtiene ninguna ganancia de modularidad (o es inferior a un umbral). El resultado final es una partición del grafo en comunidades.

Algoritmo de Louvain

El algoritmo de Louvain tiene un parámetro llamado resolution (resolución) que controla el grado de resolución de las comunidades detectadas. Un valor de resolución alto produce comunidades más pequeñas y específicas, mientras que un valor bajo produce comunidades más grandes y generales. Este es un parámetro importante que debe ser ajustado para cada caso.

Es importante tener en cuenta que el valor de resolución óptimo depende de cada caso y del tamaño de las comunidades que se desean detectar. Por lo tanto, se recomienda experimentar con diferentes valores de resolución para encontrar la configuración que mejor se adapte a los objetivos de la detección de comunidades en un grafo determinado.



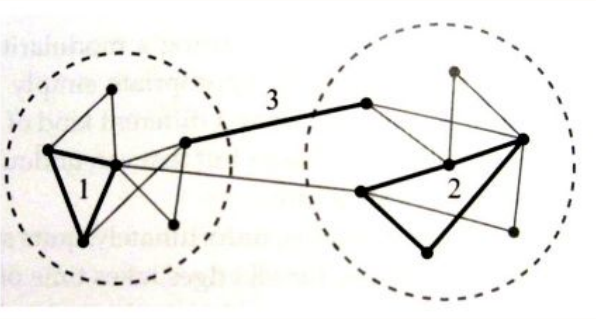
Método de Radicchi

El algoritmo de Radicchi consiste en detectar aristas dentro de los agrupamientos donde se forman bucles. Como las comunidades son conjuntos de nodos altamente interconectados, es de esperar que aparezcan en ellas bucles de aristas y nodos. Por el contrario, las aristas que conectan nodos entre agrupaciones es poco probable que estén involucradas en bucles.

Basándose en estas idea, el algoritmo de Radicchi propone un coeficiente de agrupamiento de aristas en el que los valores pequeños permiten identificar qué aristas forman parte de un mismo agrupamiento.

Este coeficiente de agrupamiento de aristas se basa en el principio del coeficiente de agrupamiento de nodos propuesto por Watts y Strogatz en 1998, el cual consiste en la relación entre el número de triángulos que incluyen un nodo y el número de posibles triángulos que se pueden formar.

Método de Radicchi



Se puede observar el principio del coeficiente de agrupación de manera gráfica. El grado de atenuación permite identificar el valor del coeficiente de agrupamiento, cuanto más oscuro su coeficiente es mayor.

Método de Radicchi

De manera formal, el coeficiente de agrupamiento se define como:

$$C_{ij} = \frac{z_{ij} + 1}{\min(k_i - 1, k_j - 1)}$$

Donde z_{ij} es el número de triángulos a los que los nodos i y j pertenecen y k_i corresponde al grado del nodo i. Esta medida se basa en el hecho de que los bordes que conectan las comunidades tienden a exhibir un pequeño valor de éste coeficiente. El valor $+1$ presente en $z_{ij} + 1$ se utiliza para penalizar el caso en el que se presenten varias aristas con cero triángulos

Método de Radicchi

El coeficiente de agrupamiento tiene valores prácticamente inversos a los obtenidos por el algoritmo de Girvan y Newman: cuando se obtienen valores de betweenness altos, lo más probable es que se obtengan bajos valores de agrupamiento y viceversa.

Una ventaja de este método es su velocidad, su complejidad algorítmica es de orden *O(n^2)*. pero el algoritmo tiende a fallar si el coeficiente promedio de agrupamiento de la red es pequeño, entonces el coeficiente será pequeño para todas las aristas.

Este método tiene la ventaja de que funciona en redes con gran número de triángulos k-cliques en una primera instancia, lo que lo hace idóneo para implementarlo sobre redes sociales.

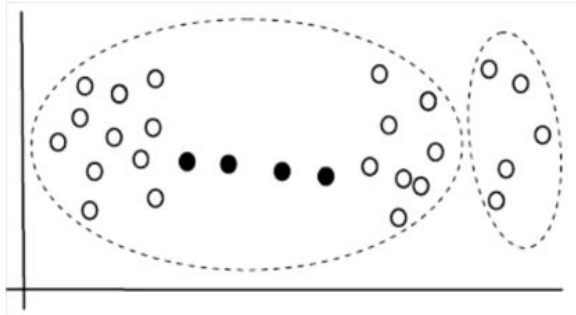
Método de conexión simple (simple-link)

El método de conexión simple (simple-link) para el clustering jerárquico consiste en encontrar la pareja de nodos más cercana (los dos nodos más similares entre sí que pertenezcan a diferentes clusters y fusionar dichos clusters).

De manera formal, la distancia mínima entre dos clusters es:

$$d(C_i, C_j) = \min d(x, x'), x \in C_i, x' \in C_j$$

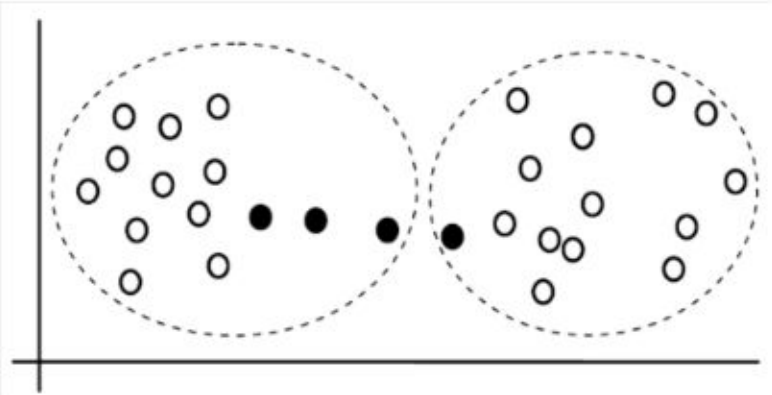
La complejidad computacional de este método es de *O(n^2)*, donde es el número de nodos.
Uno de los problemas de este método se conoce como el problema del encadenamiento.
Este problema consiste en que se conforman grupos de forma elíptica.



Método de conexión completa (complete-link)

El método de conexión completa (complete-link) para el clustering jerárquico es muy similar a método simple link, con la diferencia que su objetivo no es encontrar al nodo más cercano, sino encontrar conjuntos tipo clique. El método fusiona los dos cluster cuyos nodos más alejados tienen la distancia más pequeña.
De manera formal, la distancia máxima entre dos clusters es: $d(C_i, C_j) = \max d(x, x'), x \in C_i, x' \in C_j$

La complejidad computacional de este método es del orden de *O(n^2 log n)* donde n es el número de nodos de la red.
Este método tiende a formar grupos más compactos y con diámetros similares. También es más robusto frente el problema del encadenamiento, pero puede tener problemas con valores atípicos.



Método de conexión media (average-link)

El método de conexión media (average-link), también conocido como UMPGA, para el clustering jerárquico tiene en consideración el caso de los valores atípicos del método de conexión completa y la propiedad de encadenamiento de método simple. La distancia entre dos grupos se calcula como la distancia media de todas las distancias por pares entre los nodos en dos grupos.
De manera formal, la distancia media entre dos clusters es:

$$d(C_i, C_j) = \frac{\sum x \in C_i, x' \in C_j d(x, x')}{|C_i| \cdot |C_j|}$$

La complejidad computacional de este método es del orden de *O(n^2 log n)* donde n es el número de nodos de la red.

Aparte de las medidas de conexión analizadas anteriormente como los métodos de conexión simple, máxima y media; se emplean otros métodos basados en medidas geométricas o el cálculo de error cuadrático medio como los métodos de centroide, mediada y Ward.

Método SimRank

Su idea principal es que dos objetos (nodos) son similares si son referenciados por objetos similares.

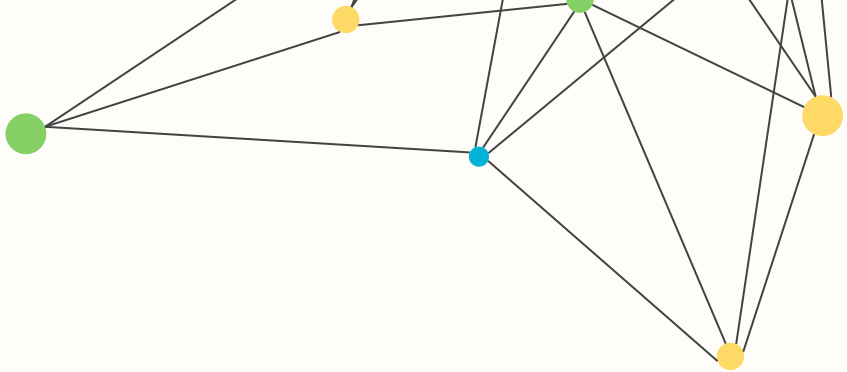
Asigna una puntuación numérica a la similitud entre cada par de nodos en un grafo. Se basa en el contexto estructural de los nodos y opera bajo el principio de recursión:

- **Base de la similitud:** Dos nodos son idénticos si son el mismo nodo, en cuyo caso su similitud es máxima (generalmente 1).
- **Similitud recursiva:** Para dos nodos diferentes, su similitud se define como la similitud promedio de los nodos que apuntan a ellos (sus nodos de entrada o "referenciadores"), atenuada por un factor de decaimiento o constante de amortiguación (generalmente denotado como C, entre 0 y 1).

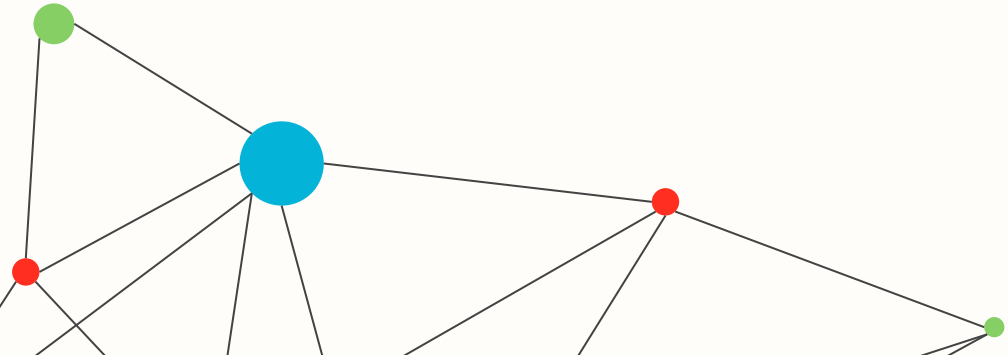
Método SimRank

El cálculo de SimRank se realiza típicamente mediante un proceso iterativo:

1. Se inicializa la similitud de todos los pares de nodos a un valor base (por ejemplo, 0 para pares distintos y 1 para pares idénticos).
2. En cada iteración, se actualizan las puntuaciones de similitud basándose en los valores de la iteración anterior y la estructura del grafo.
3. El proceso converge a una puntuación de similitud final para cada par de nodos.



Revisar el Notebook del Algoritmo de SimRank



Método Fast Greedy

El algoritmo se basa en el algoritmo de Girvan y Newman. En su ejecución, se produce la división de los nodos en comunidades, independientemente si estas divisiones son naturales o no. Para determinar la calidad de las divisiones, se utiliza el concepto de Q modularidad , el cual se define de la siguiente manera:

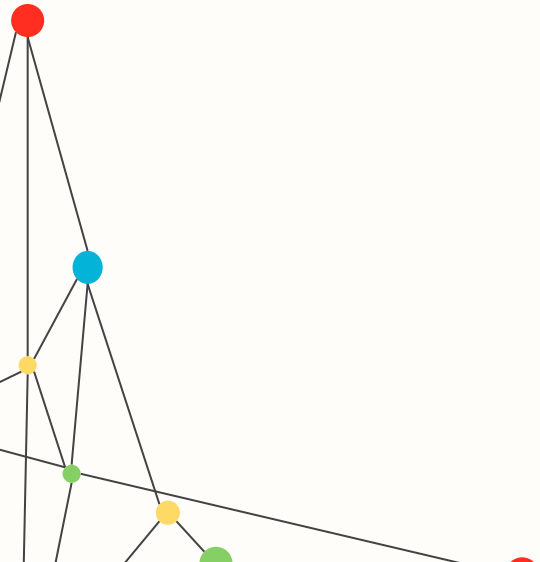
$$Q = \sum_i (e_{ii} - a_i^2)$$

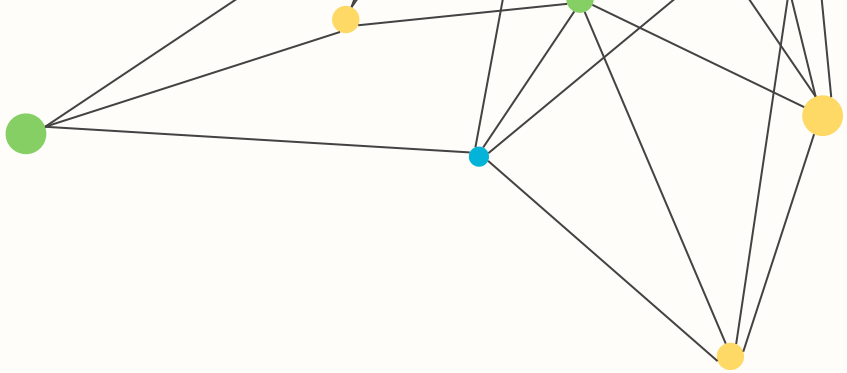
donde i, j son comunidades de la red, e_{ii} son las aristas que hay en la comunidad i, y $a_i = \sum_j e_{ij}$.

Si al momento de realizar las divisiones en la red, el valor resultante es menor que la cantidad esperada al azar, esta modularidad es Q=0. Valores distintos de 0 indican desviaciones de la aleatoriedad y con valores superiores a 0.3 se aprecia una estructura de comunidad significativa.

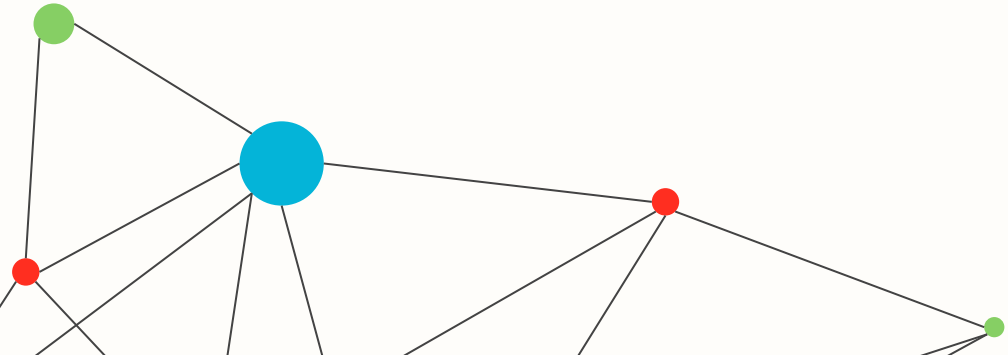
Algoritmo de Kernighan – Lin

Tarea: Investigar en qué consiste este algoritmo.



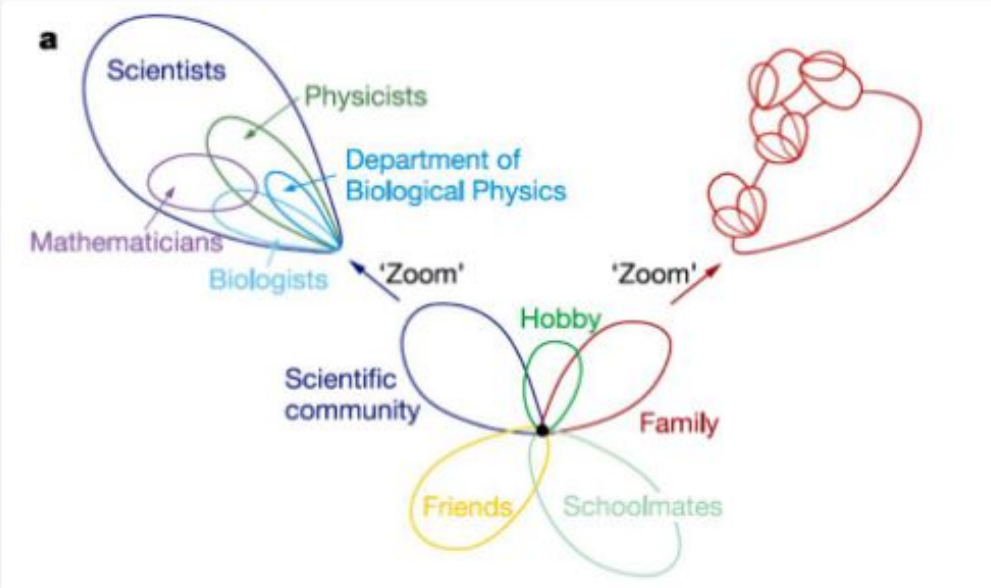


Detección de comunidades con solapamiento

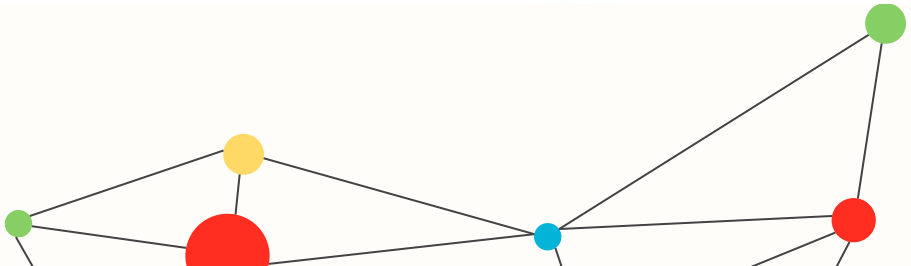


Detección de comunidades con solapamiento

La presencia de comunidades solapadas (overlapping communities) es un fenómeno común cuando se analizan redes sociales. Por ejemplo, en una red social, una persona puede pertenecer a varias comunidades como su familia, amigos, hobbies, trabajo; encontrar este tipo de comunidades con las técnicas tradicionales es algo para lo que no están preparadas.



Se puede ver como un nodo pertenece a varias comunidades. Estas comunidades no son únicas, también forman parte de comunidades más grandes, lo que significa que éstas se solapan entre sí.



Detección de comunidades con solapamiento

Las técnicas descritas anteriormente están enfocadas a la detección clásica de comunidades, en las que un nodo puede pertenecer exclusivamente a una comunidad.

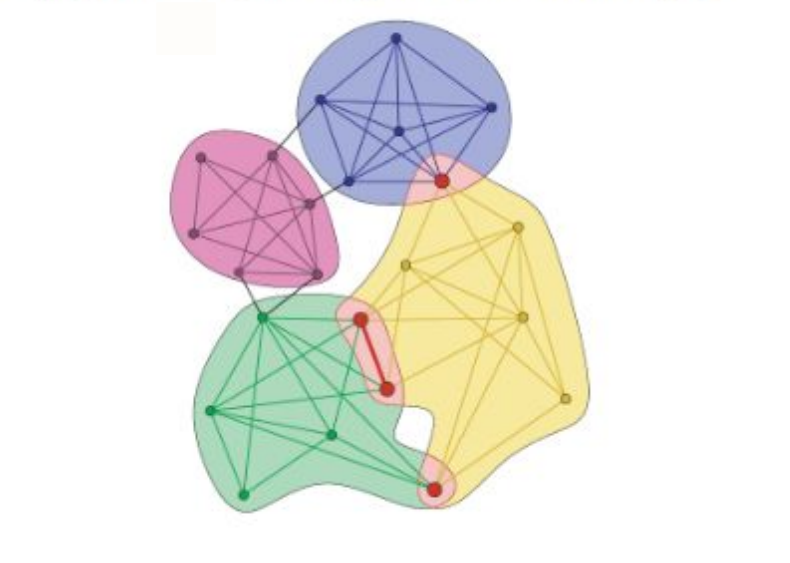
A continuación se expondrá algunas de las técnicas más representativas para la detección de comunidades solapadas.

Adyacencia de subconjuntos

Dos conjuntos se consideran adyacentes si se superponen entre sí con tanta fuerza como sea posible; es decir, si comparten k - 1 nodos. Si se elimina un enlace de un k- clique, esto conduce a la formación de dos (k - 1) -cliques adyacentes que comparten (k - 2) nodos.

La unión de estos conjuntos adyacentes conforma una cadena de k-cliques. Dos cliques se encuentran conectados si forman parte de la misma cadena. Finalmente, una comunidad la conforman el conjunto de k-cliques interconectados sobre una misma cadena.

Adyacencia de subconjuntos



Ejemplo de una comunidad solapada con k=4. La comunidad inferior derecha (amarilla) se encuentra solapada con la comunidad superior derecha (azul) a través de un nodo. En cambio, se encuentra solapada con la comunidad inferior izquierda (verde) a través de tres nodos y un enlace entre dos de los nodos compartidos

Clique Percolation Method

Una de las técnicas más populares es la propuesta por Palla et al. en 2005 llamada método de percolación de cliques o, en inglés, clique percolation method (CPM), el método se basa en el fenómeno de que las aristas en un grafo altamente conectado tiende a formar cliques.

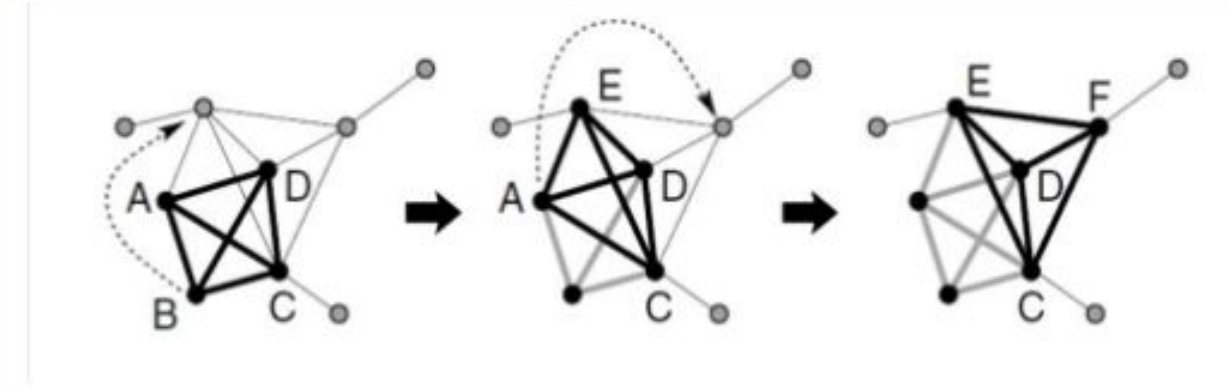
De manera opuesta, las aristas que conectan nodos de diferentes comunidades no tienden a formar este tipo de estructuras. Este concepto tiene una base similar a la propuesta para el método de agrupamiento de Radicchi.

Las comunidades detectadas usando este método basadas en la idea clave de formación de cliques, se construyen a partir de bloques adyacentes de un mismo tamaño , que corresponden a k-cliques.

Clique Percolation Method

El algoritmo CPM consiste en encontrar los k-cliques adyacentes que permitan conformar una cadena. Cuando no sea posible extender la cadena de adyacencias, se da por formada la comunidad.

Dentro de esta comunidad, es posible “rotar” o “pivotar” los k-cliques al largo de la cadena tan solo reemplazando un nodo del clique.



Clique Percolation Method

En la anterior se puede apreciar el proceso de formación de manera más didáctica.

En el paso 1 se tiene un clique con un k=4 formado por los nodos A-B-C-D.

En el paso 2 se “rota” a un nodo adyacente cumpliendo con la definición de adyacencia, esta rotación da lugar a un clique formado por los nodos A-E-D-C; este proceso se repite hasta que no sea posible continuar con las “rotaciones”.

En el paso 3 se puede ver la última formación de nodos E-D-C-F.

Después de terminar el proceso, se obtiene una cadena de cliques que da lugar a una comunidad formada por los nodos A-B-C-D-E-F.

Clique Percolation Method

De manera general, el algoritmo CPM consiste en tres pasos:

- Buscar todos los cliques de tamaño en todo el grafo.
- Construir cadenas de cliques.
- Todos los nodos que intervienen en los cliques de la cadena forman parte de la comunidad.

Para encontrar los cliques adyacentes, se emplea una matriz de adyacencia de cliques, cuyo tamaño corresponde al número de cliques presente en el grafo. A continuación, se representa el número de nodos compartidos por cada par de cliques. Por último, se eliminan de la matriz de adyacencia aquellas celdas cuyo valor sea menor o igual a (k - 1). Con esta matriz, las comunidades solapadas se pueden determinar fácilmente.

Clique Percolation Method

Este algoritmo tiene una complejidad exponencial, determinada por el número de cliques presentes en la red y el tamaño de la red (nodos, aristas). El número de cliques es muy difícil de determinar con antelación, pero en varios casos donde las redes contaban con 10^5 nodos se puede resolver el problema en tiempos razonables.

En la actualidad, hay una implementación práctica del método CPM llamada CFinder.

CPM Secuencial

Este método funciona de manera opuesta al CPM.

Se basa en la idea de buscar comunidades de k-cliques por medio de la inserción de aristas en un grafo vacío. Cada vez que se añade una nueva arista al grafo, se comprueba la formación de k-cliques. A la par de la detección de los k-cliques, se construye el dendrograma correspondiente. El algoritmo ha sido especialmente diseñado para el análisis de redes ponderadas. También es posible su uso en redes no ponderadas, donde la arista por analizar es seleccionada de manera aleatoria.

CPM Secuencial

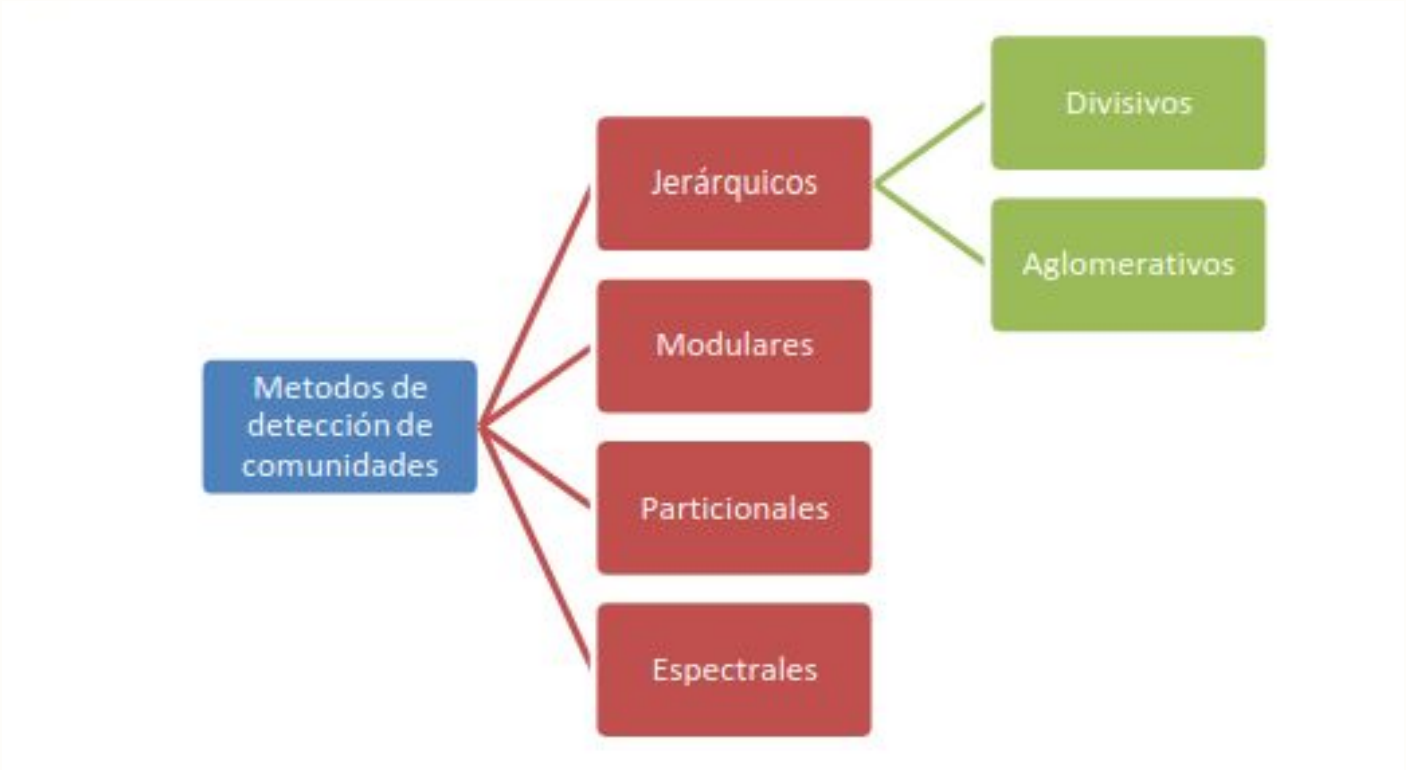
El método se compone principalmente de dos pasos:

- Detección de k-cliques por la inserción de un enlace.
- Seguimiento y fusión de k-comunidades mediante el procesamiento de los k- cliques.
-

Todo el proceso se repite por cada k-clique introducido. Una vez se han formado los k-cliques, las k-comunidades pueden ser identificadas en el grafo elaborado. Además, se puede hacer el seguimiento al proceso de formación de cliques y detenerlo en cualquier momento, lo que puede ser útil para redes muy densas.

La complejidad computacional de este método es prácticamente lineal, con respecto al número de k-cliques.

Resumen


Metodos de detección de comunidades
Jerárquicos
Divisivos
Aglomerativos
Modulares
Particionales
Espectrales

Resumen

Método	Algoritmo	Año	Ref.	S	D	P	Complejidad
Jerárquico	Newman y Girvan	2002	[25]				$O(n^3)$
	Radicchi	2004	[61]		X	X	$O(n^2)$
	WERN-Kpath	2012	[12]		X	X	$O(km)$
	PAM	2013	[90]	X	X	X	$O(k(n-k)^2)$
	SLINK	1971	[73]			X	$O(n^2)$
	CLINK	1976	[13]			X	$O(n^2 \log n)$
	UPGMA	1958	[39]			X	$O(n^2 \log n)$
	FAC-EG	2008	[44]			X	$O(m)$
	HC-PIN	2011	[31]			X	$O(\bar{k}^2 m)$
	Zhao - Zhang	2011	[91]	X		X	$O(n^2 \log n)$
	EAGLE	2009	[71]	X	X	X	$O(n^2 + (h+n)s) + O(n^2 s)$
Modular	Fast Greedy	2004	[50]			X	$O(n^2)$
	Clauset et al.	2004	[7]			X	$O(m \log^2 n)$
	RG	2010	[56]		X	X	$O(m+n)$
	FPMQA	2013	[6]			X	$O((k^{max})^2)$
Particional	k-medias	1967	[40]			X	$O(tkn)$
	Kernighan – Lin	1970	[33]				$O(n^2 \log n)$
	LMAP	2010	[69]		X	X	$O(n^3 \log n)$
Espectral	Espectral Estándar	1973	[15]			X	$O(n^3)$
	GANC	2011	[57]			X	$O(n \log^2 n)$
Solapamiento	CPM	2005	[57]	X			$O({n_c}^2)$
	SCPM	2008	[35]	X		X	$O(n_c)$
	MOSES	2010	[41]	X			$O(n)$
	Shang et al.	2010	[70]	X		X	$O(n^2)$
	CONA	2011	[87]	X			$O(tn_c n^2)$

Fecha de entrega:
12 de Noviembre 2025

Práctica 10:

De los todos los Notebooks de la unidad 5,
entregar la interpretación de los resultados.

Fecha de entrega:
17 de Noviembre 2025

Práctica 11:

Elige un algoritmo distinto a los que están en los Notebooks, con los datos de Facebook realiza la detección de comunidades y realiza la comparación con el algoritmo expuesto.