



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Detección de comunidades

Paola Bermolen

paola@fing.edu.uy

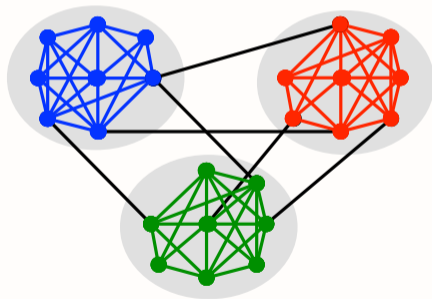
29 de agosto de 2022



- 1 **Estructura en comunidades**
- 2 **Ejemplos de comunidades en redes**
- 3 **Algoritmos de detección de comunidades**
 - Método de Girvan-Newman
 - Máxima Modularidad
 - Partición espectral de grafos
- 4 **Medidas de desempeño**

Comunidades en redes

- ▶ Rol importante de las redes: relacionar lo **local** con lo **global**
 - ⇒ explicar cómo procesos a nivel de nodo se extienden a la población
- ▶ Usualmente se piensan las redes (sociales) con la siguiente estructura

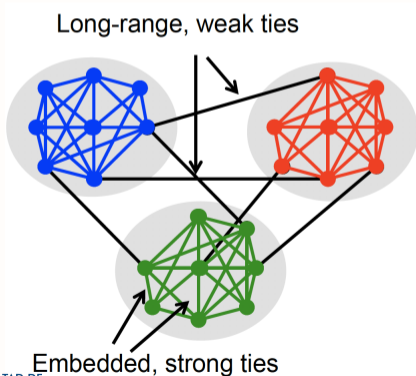


Contexto y motivación

- ▶ En los 60s., M. Granovetter entrevistó a gente que había cambiado de trabajo
- ▶ Consultó sobre cómo habían encontrado sus nuevos trabajos
 - ⇒ muchos se habían enterado a través de contactos personales
- ▶ Sorprendente: los contactos eran a menudo **conocidos** más que **amigos**
- ▶ Si los amigos cercanos tienen la mayor motivación para ayudar
 - ⇒ ¿porqué los conocidos más distantes aportan información crucial?
- ▶ M. Granovetter, *Getting a job: A study of contacts and careers*. University of Chicago Press, 1974

Respuesta de Granovetter

- Relaciona dos perspectivas diferentes acerca de los conocidos lejanos
 - **Estructural:** énfasis en cómo las amistades se extienden a través de la red
 - **Interpersonal:** consecuencias locales de que la amistad sea fuerte o débil
- Entrecruce entre rol estructural e informacional de una arista



- Respuesta trascendió la pregunta concreta sobre la búsqueda de trabajo
- **Manera general de pensar la arquitectura de redes sociales**

Respuesta de Granovetter

- ▶ Relaciona dos perspectivas diferentes acerca de los conocidos lejanos
 - **Estructural**: énfasis en cómo las amistades se extienden a través de la red
 - **Interpersonal**: consecuencias locales de que la amistad sea fuerte o débil

▶ Entrecruce entre rol estructural e informacional de una arista

1) **Aristas estructuralmente incluidas** en una comunidad:

⇒ tienden a ser socialmente fuertes; y

⇒ son altamente redundantes en términos de acceso a la información

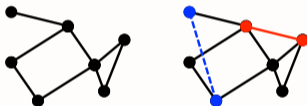
2) **Aristas largo alcance** que abarcan distintas partes de la red

⇒ tienden a ser socialmente débiles; y

⇒ permiten acceso a información útil/novedosa (e.g., trabajo nuevo)

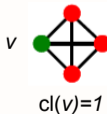
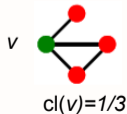
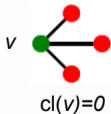
Cierre de triángulos

- ▶ Un principio básico de la formación de redes es el **cierre de triángulos**
“Si dos personas tienen un amigo en común, hay mayores chances de que se hagan amigos en el futuro”
- ▶ Aristas que aparecen (dinámica) en redes sociales tienden a cerrar triángulos



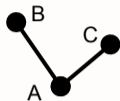
⇒ Más probable que aparezca la arista **roja** que la **azul**

- ▶ Prevalencia de cierre de triángulos se mide con el **clustering coefficient**



Cierre de triángulos

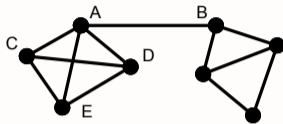
- ¿Porqué es natural que aparezcan cierres de triángulos?



- 1) Aumento de **oportunidades** para que B y C se conozcan
⇒ ambos pasan tiempo con A
 - 2) Hay una base preexistente de **confianza mutua** entre B y C
⇒ ambos tienen a A como amigo común
 - 3) A tiene **incentivos** para juntar a B y C
⇒ falta de amistad entre B y C puede ser fuente de estrés para A
- esta premisa se basa en teorías anteriores de psicología social (Heider 1958)

Puentes

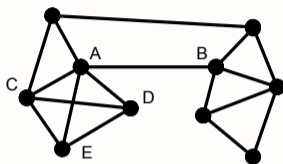
- ▶ Ejemplo de red social simple:



- ▶ A tiene aristas con C,D, y E formando un grupo **bien tejido**
 - ⇒ A,C,D, y E probablemente expuestos a **opiniones similares**
- ▶ Arista entre A y B permite **alcanzar a una parte diferente de la red**
 - ⇒ B ofrece a A **opciones que en otro caso no hubiera conocido**
- ▶ Arista A-B se llama **punto**, si se saca la red se desconecta
 - ⇒ Componentes gigantes sugieren que los **puentes son bastante raros**

Puentes locales

- ▶ En realidad, la red social es mucho más grande y podría ser así

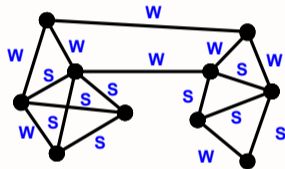


⇒ sin saberlo A y B, pueden existir caminos más largos entre ellos

- ▶ **Def. Alcance** de la arista (u, v) : distancia entre u y v si la arista se elimina
- ▶ **Def.** Un **puente local** es una arista con alcance > 2
 - ⇒ extremos de la arista sin amigos en común
 - ⇒ **Ejemplo:** la arista A-B es un puente local con alcance 3
- ▶ Los puentes locales no son parte de triángulos

Propiedad fuerte de cierre de triángulos

- ▶ Categorizar todas las aristas de la red de acuerdo a su fuerza
 - ⇒ Aristas **fuertes** corresponden a **amistades**
 - ⇒ Aristas **débiles** corresponden a **conocidos**



- ▶ Oportunidad, confianza e incentivos promueven aristas fuertes
 - ⇒ sugiere hipótesis cualitativa llamada **strong triadic closure**
- “Dos aristas fuertes implica una tercer arista que cierra el triángulo”*
- ▶ **Abstracción para analizar las consecuencias de lazos débiles y fuertes**

Puentes locales y lazos débiles

- a) Distinción local interpersonal entre aristas \Rightarrow lazos fuertes/débiles (estática)
- b) Noción global estructural \Rightarrow existencia/ausencia de puentes locales

Teorema

Si un nodo verifica la propiedad fuerte de cierre de triángulos y forma parte de al menos dos aristas fuertes, entonces todo puente local incidente a él es débil

- Relaciona las perspectivas **estructurales** e **interpersonales** en amistades
- Vuelta a la búsqueda de trabajo: los puentes locales conectan con nueva información
 - \Rightarrow Conceptualmente el *alcance* refiere a su debilidad como lazo social
 - \Rightarrow Un poco paradójico, el doble rol habla de “**fortaleza de los lazos débiles**”

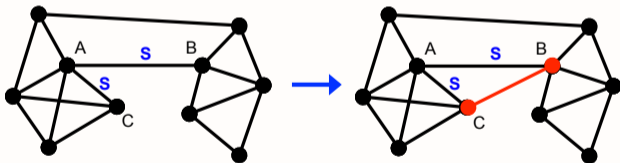
Easley & Kleinberg'10

Prueba por absurdo

- Supongo que el nodo **A** tiene dos aristas fuertes
- Supongo además que **A** cumple la propiedad fuerte de cierre de triángulos



- Sea **A-B** un puente local y un lazo fuerte



⇒ Arista B-C tiene que existir

- Contradice que A-B es un puente local (C es un amigo en común)

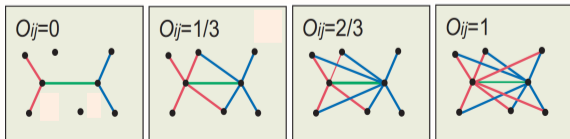
Estructura y fuerza de los lazos en redes grandes

- ▶ ¿Se puede testear la teoría de Granovetter's con **datos en redes reales**?
 - ⇒ Imposible por décadas: falta de datos en redes sociales de gran escala
- ▶ Ejemplo más actual: redes **“who-calls-whom”** con ambos ingredientes clave
 - ⇒ estructura de la red de comunicación entre pares de personas
 - ⇒ tiempo total de habla como medida de fortaleza del lazo
- ▶ J. P. Onella et al., “Structure and tie strengths in mobile communication networks,” *PNAS*, vol. 104, pp. 7332-7336, 2007
 - red celular que incluye $\approx 20\%$ de la población del país

Generalizando lazos débiles y puentes locales

- Hasta ahora hablamos de nociones dicotómicas
 - ⇒ las aristas son fuertes o débiles, son puentes locales o no
- Es conveniente tener medidas que tengan una **graduación más regular**
- Valores **númericos de fortaleza** para los lazos ⇒ duración de las llamadas
 - ⇒ Ordenar las aristas por su fortaleza y registrar su percentil
- **Generalizar puentes locales** ⇒ **superposición de vecindario** de la arista (i, j)

$$O_{ij} = \frac{|n(i) \cap n(j)|}{|n(i) \cup n(j)|}; \quad n(i) := \{j \in V : (i, j) \in E\}$$

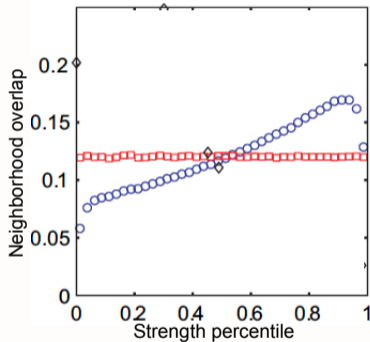


Propiedad:

$O_{ij} = 0$ si (i, j)
es un puente
local

Resultados empíricos

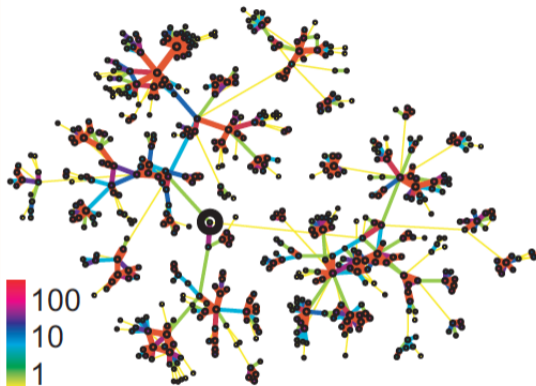
- **Comportamiento esperado:** O_{ij} creciente con la fuerza de los lazos
⇒ Confirmado muy claramente por los datos (puntos \circ)[final?]



- Permutación aleatoria de fortaleza de los datos (puntos \square)
⇒ rompe el acoplamiento de O_{ij} y la fortaleza de los lazos

Red celular y fortaleza de los lazos

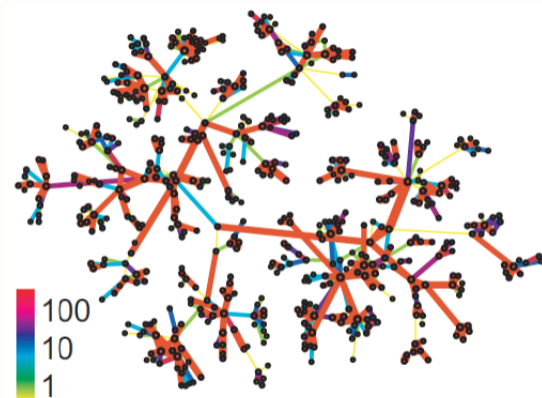
- Red celular con rango de colores para la fortaleza de los lazos



- 1) Lazos más fuertes más estructuralmente arraigados (intra comunidades)
- 2) Lazos más débiles acordes con aristas de largo alcance (inter comunidades)

Permutación aleatoria de la fortaleza de los lazos

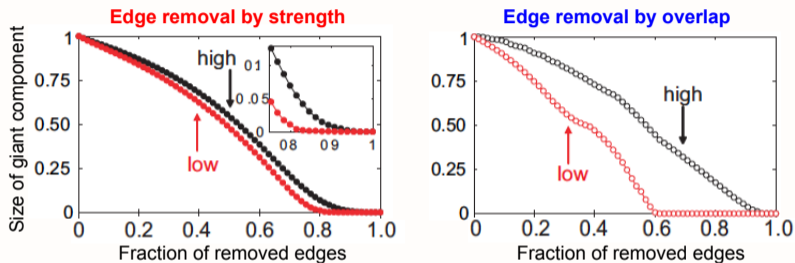
- Misma red celular con **permutación aleatoria de los valores de fortaleza**



- No hay relación aparente entre las funciones estructurales e interpersonales de las aristas

Lazos débiles uniendo comunidades

- **Hipótesis:** lazos débiles -> largo alcance unen comunidades de lazos fuertes



- 1) Eliminar aristas una a la vez empezando con la más fuerte

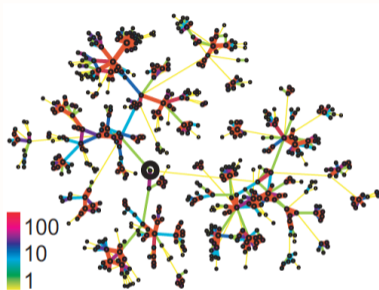
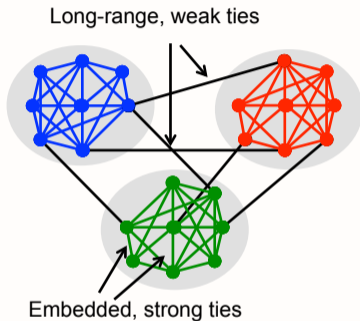
⇒ la componente gigante se achica sostenidamente

- 2) Idem empezando con la más débil

⇒ la componente gigante se achica más rápidamente y punto de quiebre

¿Y la vaca?

- Inicio: usualmente se piensan las redes (sociales) con la siguiente estructura



- Vimos que la imagen conceptual tiene soporte en la teoría de Granovetter's sobre la **fortaleza de los lazos débiles**

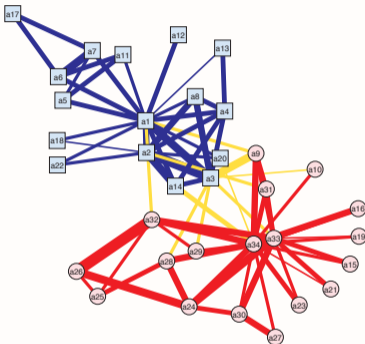
⇒ aunque no es el único intento de formalización...

Ejemplos de comunidades en redes

- 1 Estructura en comunidades
- 2 Ejemplos de comunidades en redes
- 3 Algoritmos de detección de comunidades
 - Método de Girvan-Newman
 - Máxima Modularidad
 - Partición espectral de grafos
- 4 Medidas de desempeño

Zachary's karate club

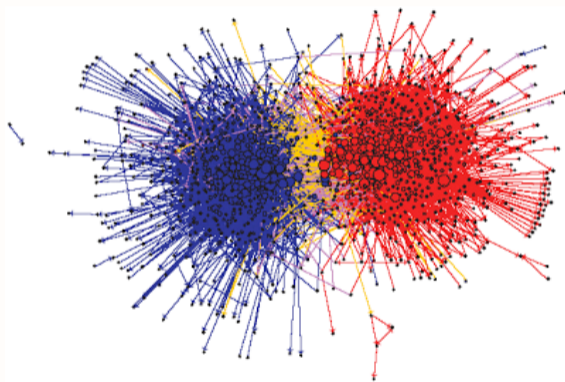
- Interacciones sociales en un club de karate en los 70s



- Zachary fue testigo de la partición en dos grupos durante su estudio
 - ⇒ ejemplo de juguete, clásico algoritmo de detección de comunidades
 - ⇒ “ground truth” de pertenencia a las comunidades (usualmente inexistente)

Blogs políticos

- ▶ Blogs políticos para las elecciones presidenciales de 2004 en USA



- ▶ Estructura de blogs **liberales** y **conservadores** es aparente

Red eléctrica

- ▶ Dividir la red eléctrica en áreas con interacción inter-áreas mínimas

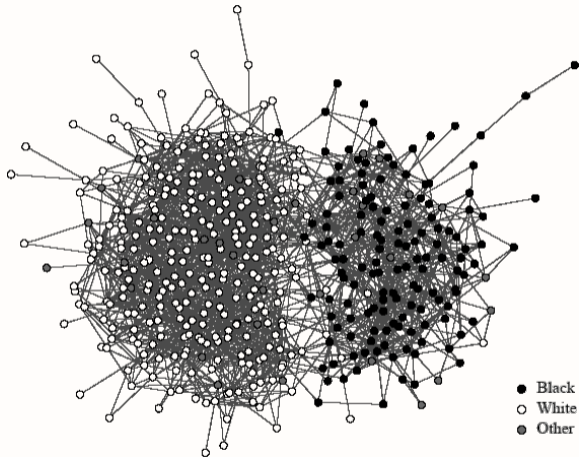


- ▶ **Aplicaciones:**

- Definir áreas de control para la estimación del estado del sistema eléctrico
- Cálculo paralelo del flujo de potencia
- Control de las islas para evitar la propagación de los apagones

Estudiantes de liceo

- ▶ Red de interacción social entre liceales



- ▶ **Assortative mixing** fuerte con raza como característica

Físicos trabajando en Network Science

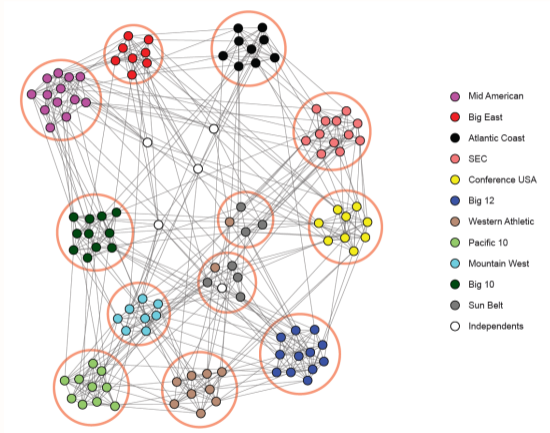
- Red de coautoría de físicos con publicaciones en análisis de redes



- Comunidades *bien tejidas* (temas) son evidentes a partir de la estructura de

Fútbol universitario

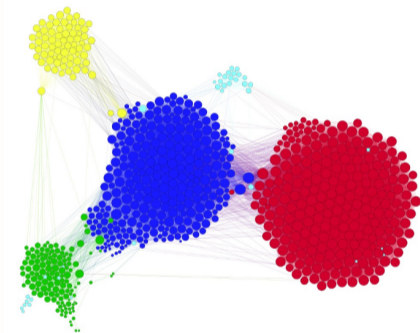
- ▶ Vértices son equipos de fútbol y aristas son partidos en Otoño '00



- ▶ Comunidades son las confederaciones y los equipos

Amistades en Facebook

- ▶ Facebook (subgrado inducido) con 744 vértices y 30K aristas



- ▶ Identificar círculos sociales a los cuáles pertenece los amigos
⇒ empresa, liceo, basketball club, squash club (?), familia

Algoritmos de detección de comunidades

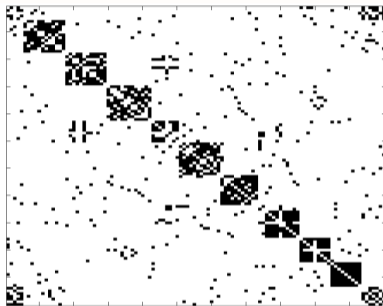
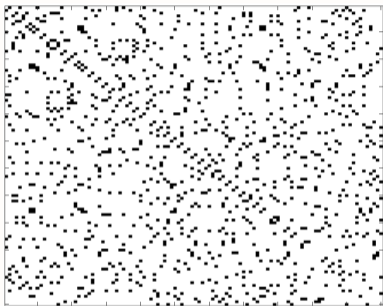
- 1 Estructura en comunidades
- 2 Ejemplos de comunidades en redes
- 3 Algoritmos de detección de comunidades
 - Método de Girvan-Newman
 - Máxima Modularidad
 - Partición espectral de grafos
- 4 Medidas de desempeño

Detección de comunidades y partición de grafos

- ¿Cómo podemos **identificar automáticamente** comunidades?
- **Detección de comunidades** es un problema difícil de *clustering*:
 - C1)** No hay consenso en la definición de comunidad
 - C2)** Selección de subconjuntos de nodos generalmente intratable
 - C3)** Falta de *ground-truth* para validación
- **Partición de grafos**: Separar V en un **número dado** de grupos disjuntos de **tamaño dado**
- **Detección de comunidades**: Número y tamaño de los grupos es **desconocido** en detección de comunidades
 - ⇒ Identificar las *fallas naturales* por las que se separa una red

Detección de comunidades

- ▶ Dado un grafo $G(V, E)$ con matriz de adyacencia \mathbf{A} (izq.)



- ▶ Encontrar permutación de filas/columnas que revelen una **estructura diagonal de bloques** (der.)

Ejemplo: ejemplo de fútbol universitario que vimos antes [Mateos-Giannakis'12]

Particiones de grafos

- ▶ **Bisección de grafos:** partir V en dos grupos tal que,
 - grupos V_1 and V_2 son disjuntos
 - grupos tienen igual tamaño , i.e., $|V_1| = |V_2| = N_v/2$
 - la cantidad de aristas entre vértices en grupos diferentes es mínima
- ▶ Fácil de describir pero difícil de resolver

$$\text{Número de particiones posibles } V : \binom{N_v}{N_v/2} \approx \frac{2^{N_v}}{\sqrt{N_v}}$$

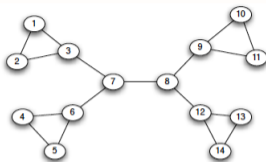
⇒ Usando la fórmula de Stirling's $N_v! \approx \sqrt{2\pi N_v} (N_v/e)^{N_v}$

⇒ **búsqueda exhaustiva solo posible para redes pequeñas**

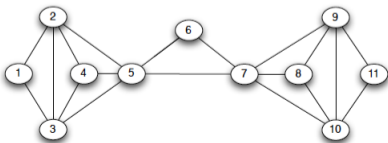
- ▶ No se conoce un algoritmo eficiente (i.e., tiempo polinomial de resolución)
 - ⇒ Excelentes heurísticas, relajando criterios de partición

Fuerza de los lazos débiles

- Puentes locales conectan partes con poca interacción entre sí



- Buena idea: eliminar los puentes locales y que aparezcan las comunidades
- Mala idea
 - ¿qué hacer si hay varios puentes locales? cuál elegir...
 - ¿y si no hay ningún puente local? igual puede haber divisiones naturales

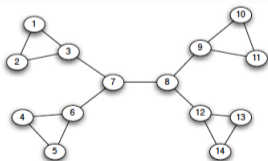


Centralidad betweenness de aristas

- Buena idea: identificar lazos débiles/puentes locales a través de aristas con centralidad betweenness alta

Definición (Centralidad betweenness de aristas)

Se define de manera análoga contando la fracción de caminos más cortos entre pares de vértices a los cuales pertenece la arista



$$C_{Be}(7 - 8) = 7 \times 7 = 49$$

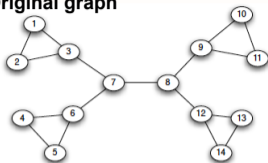
$$C_{Be}(3 - 7) = 3 \times 11 = 33 \text{ idem } 6 - 7, 8 - 9, 8 - 12$$

$$C_{Be}(1 - 3) = 12 \text{ idem } 3, 6, 9, 12 \text{ en sus triángulos}$$

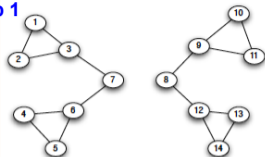
el resto tiene centralidad 1

Algoritmo en acción para el ejemplo

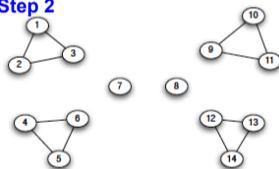
Original graph



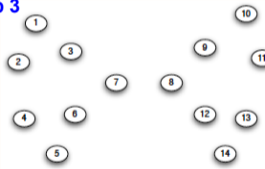
Step 1



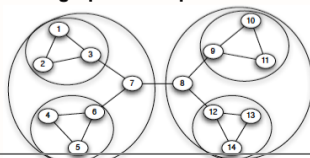
Step 2



Step 3

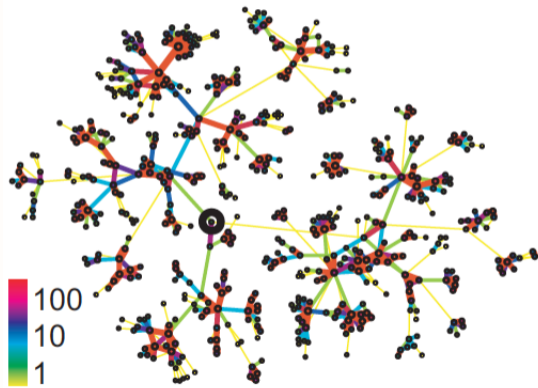


Nested graph decomposition

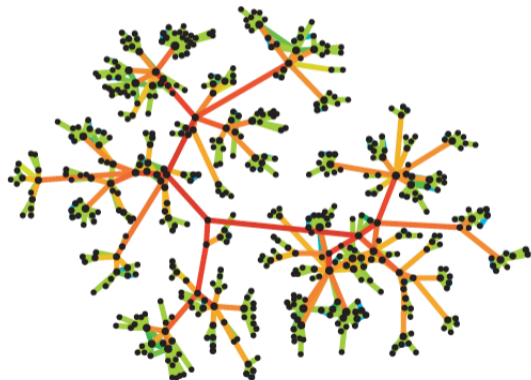


Centralidad betweenness de aristas

- Red celular de antes coloreada con fortaleza de los lazos y con centralidad betweenness



Edge strength



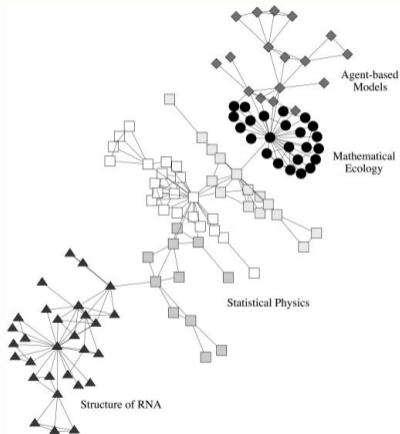
Edge betweenness

Método de Girvan-Newman

- ▶ **Método de Girvan-Newman** es conceptualmente muy simple
 - ⇒ encontrar y eliminar aristas entre subgrupos cohesivos
- ▶ **Algoritmo:** Repetir hasta que no queden aristas
 - ⇒ Calcular la centralidad betweenness $c_{Be}(e)$ para todas las aristas
 - ⇒ Eliminar arista(s) con $c_{Be}(e)$ más alta
 - ⇒ **Identificar como comunidades a las componentes conectadas**
- ▶ **Método divisivo:** comienza con la red completa y se va partiendo a medida que el algoritmo avanza
- ▶ **Partición anidada:** comunidades más grandes potencialmente incluyen grupos más densos
- ▶ M. Girvan and M. Newman, “Community structure in social and biological networks,” *PNAS*, vol. 99, pp. 7821-7826, 2002

Red de colaboración científica

- Red de coautorías de los científicos del Santa Fe Institute



- Las comunidades detectadas se identifican con diferentes disciplinas

Clustering Jerárquico

- Modificación iterativa de sucesivas particiones candidatas
 - **Divisivas**: refinamiento sucesivo de las particiones mediante la separación
 - **Aglomerativas**: aumento sucesivo de las particiones mediante la unión
- En cada paso, se modifican las particiones de forma de **minimizar un costo**
- Medidas de (dis)similaridad x_{ij} entre pares de vértices v_i y v_j :
 - distancia Euclidea (dis-similaridad) $x_{ij} = \sqrt{\sum_{k \neq i, j} (A_{ik} - A_{jk})^2}$
 - equivalencia regular o estructural
 - muchas más...
- Se obtiene una jerarquía completa de particiones anidadas del grafo
⇒ desde $\{\{v_1\}, \dots, \{v_{N_v}\}\}$ hasta V

Clustering aglomerativo

- ▶ Los **algoritmos de clustering jerárquico agglomerativo** funcionan así
 1. Elegir una métrica de (dis)similaridad y calcularla para todos los pares de vértices
 2. Asignar cada vértice a un grupo propio
 3. Unir el par de grupos con la menor (dis)similaridad
 4. Calcular la (dis)similaridad entre el nuevo grupo y todos los otros
 5. Repetir desde el paso 3 hasta que todos los vértices pertenezcan al mismo grupo
- ▶ Se necesita:
 - medida de (dis)similaridad de vértices (costo)
 - ordenar para elegir la menor (dis)similaridad
 - medida de (dis)similaridad entre conjuntos de vértices

Clustering aglomerativo

► Medidas de (dis)similaridad de grupos a partir de las medidas entre vértices

- **Single linkage:** (dis)similaridad de grupo x_{G_i, G_j}^{SL} alta si un par de vértices tiene (dis)similaridad alta

$$x_{G_i, G_j}^{SL} = \max_{u \in G_i, v \in G_j} x_{uv}$$

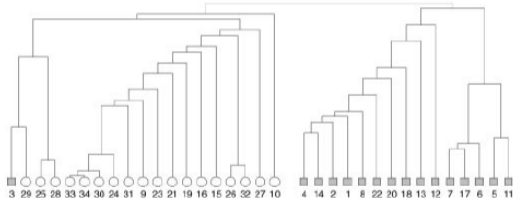
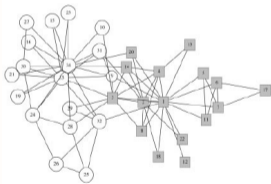
- **Complete linkage:** para tener (dis)similaridad x_{G_i, G_j}^{CL} alta, todos los pares de vértices tienen que tener (dis)similaridad alta

$$x_{G_i, G_j}^{CL} = \min_{u \in G_i, v \in G_j} x_{uv}$$

- **Otras medidas:** ponderar por tamaño de los grupos, (dis)similaridad entre centroides, etc.

Dendrograma

- ▶ Particiones jerárquicas usualmente se representan con **dendrogramas**
- ▶ Muestra grupos encontrados en la red a cada paso del algoritmo
 - ⇒ **División de la red con diferentes resoluciones**
- ▶ **Ejemplo:** Algoritmo de Girvan-Newman para el Zachary's karate club



- ▶ Otro problema: ¿Cuál es la división óptima (más útil) en algún sentido?
 - ⇒ es necesario definir métricas de la calidad del clustering

Modularidad

- Modularidad medida de *assortative mixing* - cohesividad de grupos
 - ⇒ **Intuición:** densidad de aristas en las comunidades mayor que la esperada
- Sea G un grafo y una partición $s \in S$. **Modularidad:**

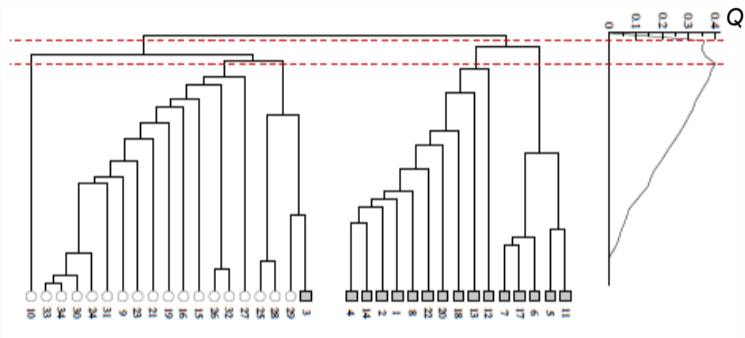
$$Q(G, S) = \frac{1}{2N_e} \sum_{s \in S} \sum_{i, j \in s} \left[A_{ij} - \frac{d_i d_j}{2N_e} \right] \in [-1, 1]$$

⇒ $\delta_{g_i g_j} = 1$ si i, j están en la misma partición ($i, j \in S$)

- Podemos evaluar la modularidad de cada partición en un dendrograma
- **Máximo valor** indica la “mejor” estructura de comunidades

Calidad del clustering

- ▶ **Ejemplo:** algoritmo de Girvan-Newman's para el Zachary's karate club



- ▶ **Buena idea** ¿Porqué no directamente optimizar $Q(G, S)$ sobre todas las posibles particiones S ?

Máxima Modularidad

- ▶ Sea g_i el grupo al que pertenece el vértice i :

$$Q(G, S) = \frac{1}{2N_e} \sum_{i,j \in V} \left[A_{ij} - \frac{d_i d_j}{2N_e} \right] \mathbf{1}_{\{g_i = g_j\}}$$

- ▶ Definir por comodidad los sumandos $B_{ij} := A_{ij} - \frac{d_i d_j}{2N_e}$
⇒ observar que las sumas marginales de B_{ij} se anulan,

$$\sum_j B_{ij} = \sum_j A_{ij} - \frac{d_i}{2N_e} \sum_j d_j = d_i - \frac{d_i}{2N_e} 2N_e = 0$$

Partición del grafo

- ▶ Consideremos (por simplicidad) dividir la red en dos grupos
- ▶ Se definen **variables binarias de pertenencia a las comunidades** por vértice

$$s_i = \begin{cases} +1, & \text{vértice } i \text{ pertenece al grupo 1} \\ -1, & \text{vértice } i \text{ pertenece al grupo 2} \end{cases}$$

- ▶ Usando la igualdad $\frac{1}{2}(s_i s_j + 1) = \mathbf{1}_{\{g_i=g_j\}}$, la modularidad es

$$\begin{aligned} Q(G, S) &= \frac{1}{2N_e} \sum_{i,j \in V} \left[A_{ij} - \frac{d_i d_j}{2N_e} \right] \mathbf{1}_{\{g_i=g_j\}} \\ &= \frac{1}{4N_e} \sum_{i,j \in V} B_{ij} (s_i s_j + 1) \end{aligned}$$

- ▶ Usando que $\sum_j B_{ij} = 0$ se obtiene una expresión más sencilla

$$Q(G, S) = \frac{1}{4N_e} \sum_{i,j \in V} B_{ij} s_i s_j$$

Optimización de la modularidad

- ▶ Sea $\mathbf{B} \in \mathbb{R}^{N_v \times N_v}$ la **matriz de modularidad** con entradas $B_{ij} := A_{ij} - \frac{d_i d_j}{2N_e}$
⇒ Las particiones S quedan definidas por el vector $\mathbf{s} = [s_1, \dots, s_{N_v}]^T$

- ▶ Modularidad es una forma cuadrática

$$Q(G, S) = \frac{1}{4N_e} \sum_{i,j \in V} B_{ij} s_i s_j = \frac{1}{4N_e} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

- ▶ Modularidad como criterio para la bisección de un grafo se traduce en:

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s} \in \{\pm 1\}^{N_v}} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

⇒ restricciones binarias! $\mathbf{s} \in \{\pm 1\}^{N_v}$ (vértices de un hypercubo)

- ▶ **Mala idea:** optimización de la modularidad es NP-hard [Brandes et al '06]

Just relax!

- ▶ Relajar las restricciones $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg \max_{\mathbf{s}} \mathbf{s}^T \mathbf{B} \mathbf{s}, \quad \text{restringido a } \mathbf{s}^T \mathbf{s} = N_v$$

- ▶ Sea λ el **multiplicador de Lagrange** asociado a la restricción $\mathbf{s}^T \mathbf{s} = N_v$
⇒ Condiciones de optimalidad resultan en:

$$\nabla_{\mathbf{s}} \left[\mathbf{s}^T \mathbf{B} \mathbf{s} + \lambda(N_v - \mathbf{s}^T \mathbf{s}) \right] = \mathbf{0} \Rightarrow \mathbf{B} \mathbf{s} = \lambda \mathbf{s}$$

- ▶ Conclusión: \mathbf{s} es un **vector propio de \mathbf{B} con valor propio λ**
- ▶ Nuevamente ¿qué vector propio elegir?
⇒ en el óptimo la función objetivo vale $\mathbf{B} \mathbf{s} = \lambda \mathbf{s}$ entonces

$$\mathbf{s}^T \mathbf{B} \mathbf{s} = \lambda \mathbf{s}^T \mathbf{s} = \lambda$$

- ▶ Para maximizar la modularidad, elegir el **vector propio dominante de \mathbf{B}**

Maximización espectral de la modularidad

- ▶ Sea \mathbf{u}_1 el vector propio dominante de \mathbf{B} , con entrada i -ésima $[\mathbf{u}_1]_i$
 - ⇒ No se puede elegir $\mathbf{s} = \sqrt{N_v} \mathbf{u}_1$ porque $\mathbf{u}_1 \neq \{\pm 1\}^{N_v}$
 - ⇒ **Best effort**: maximizar $\mathbf{s}^\top \mathbf{u}_1$ (minimizar ángulo) resulta en

$$s_i = \text{sign}([\mathbf{u}_1]_i) := \begin{cases} +1, & [\mathbf{u}_1]_i > 0 \\ -1, & [\mathbf{u}_1]_i \leq 0 \end{cases}$$

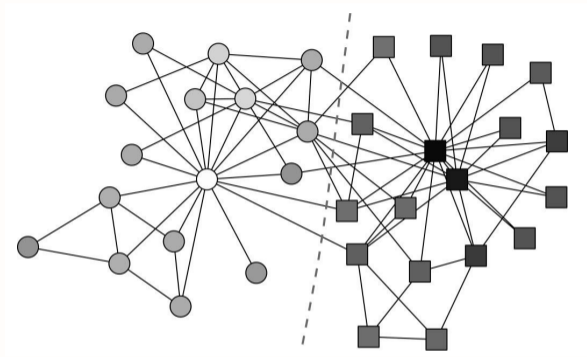
- ▶ Maximización espectral de la modularidad

s1: Calcular la matriz de modularidad \mathbf{B} con entradas $B_{ij} = A_{ij} - \frac{d_i d_j}{2N_e}$

s2: Hallar el vector propio dominante \mathbf{u}_1 de \mathbf{B} (e.g., método de las potencias)

s3: Decidir la pertenencia al cluster del vértice i como $s_i = \text{sign}([\mathbf{u}_1]_i)$

Ejemplo: Zachary's karate club

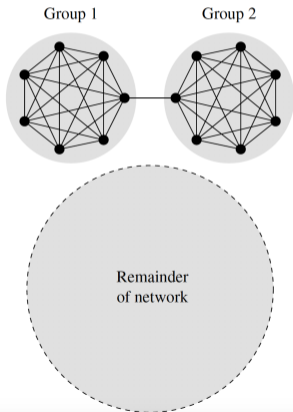


► Maximización espectral de la modularidad

- forma de los vértices indica pertenencia a la comunidad
- línea punteada indica la partición encontrada por el algoritmo
- colores de los vértices indican la fuerza de la pertenencia

Resolución límite de la modularidad máxima

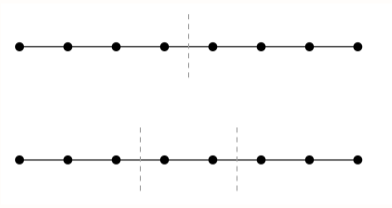
- El método de modularidad máxima tiene problema de **resolución límite**
- Incapaz de detectar comunidades muy chicas respecto al tamaño de la red



- $\Delta Q = \frac{1}{2m} \left(1 - \frac{k_1 k_2}{2m} \right)$ cambio al juntar grupos
- Si $\Delta Q > 0$ conviene juntarlos
- Comunidades con $k_1 k_2 < 2m$ no se detectan
- Ejemplo: si $m = 5000$ y $k_1, k_2 < 100$

Múltiples comunidades

- ▶ Primer idea: bisecciones sucesivas (cuidado! consider cambios en la modularidad al dividir un grupo)



⇒ arriba la mejor bisección

⇒ abajo la partición óptima sin imponer cantidad de comunidades

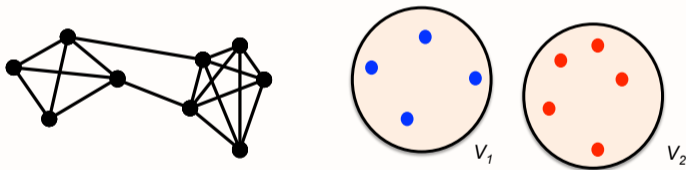
- ▶ Maximizar directamente sobre un número desconocido de grupos: problema de optimización
- ▶ Adaptar la maximización espectral, Zhang & Newman '15

Múltiples comunidades

- ▶ Más utilizado **Algoritmo de Louvain**. Blondel et. al.'08
- ▶ Método aglomerativo:
 1. Definir N_v grupos individuales.
 2. Calcular ΔQ al mover i al grupo de cada uno de sus vecinos
 3. Mover i al grupo que tenga el mayor incremento la modularidad.
 4. Los grupos resultantes se toman como nuevas unidades (self-loops y aristas con pesos) y se repite el paso 2.
 6. Terminar si ningún movimiento aumenta la modularidad
- ▶ Ventaja de ser muy rápido e implementado en muchos paquetes (networkx)

Bisección de grafos

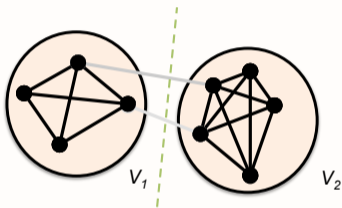
- ▶ Sea $G(V, E)$ un grafo no dirigido
- ▶ **Bisección del grafo** dividir V en dos grupos tales que:
 - Grupos V_1 y $V_2 = V_1^C$ son disjuntos
 - Grupos tienen tamaño dado, i.e., $|V_1| = N_1$ y $|V_2| = N_2$



- ▶ ¿Cuál es un buen criterio para partir el grafo?
⇒ Vimos la modularidad, veamos otro(s)

Corte de grafos

- **Desiderata:** elementos de cada grupo cumplan con
 - ⇒ estar bien conectados entre ellos; y
 - ⇒ estar relativamente bien separados del otro grupo



Definición

Un **corte** C del grafo en grupos V_1 y V_2 es el número de aristas entre los grupos

$$C := \text{cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij}$$

De corte de grafos...

- ▶ **Criterio natural de bisección:** minimizar corte entre V_1 y $V \setminus V_1$
- ▶ Se definen variables binarias de pertenencia por vértice a los grupos

$$s_i = \begin{cases} +1, & \text{vértice } i \text{ pertenece al grupo 1} \\ -1, & \text{vértice } i \text{ pertenece al grupo 2} \end{cases}$$

- ▶ Sea g_i el grupo al que pertenece el vértice i , se cumple que

$$\mathbb{I}\{g_i \neq g_j\} = \frac{1}{2}(1 - s_i s_j) = \begin{cases} 1, & i \text{ y } j \text{ en grupos diferentes} \\ 0, & i \text{ y } j \text{ en el mismo grupo} \end{cases}$$

- ▶ Se puede expresar los *cortes* en términos de las variables s_i :

$$C = \sum_{i \in V_1, j \in V_2} A_{ij} = \frac{1}{2} \sum_{i, j \in V} A_{ij} (1 - s_i s_j)$$

... al Laplaciano del grafo

- ▶ Primer sumando en $C = \frac{1}{2} \sum_{i,j} A_{ij}(1 - s_i s_j)$ es

$$\sum_{i,j \in V} A_{ij} = \sum_{i \in V} d_i = \sum_{i \in V} d_i s_i^2 = \sum_{i,j \in V} d_i s_i s_j \mathbb{I}\{i = j\} \quad \text{usamos } s_i^2 = 1$$

- ▶ El corte resulta

$$C = \frac{1}{2} \sum_{i,j \in V} (d_i \mathbb{I}\{i = j\} - A_{ij}) s_i s_j = \frac{1}{2} \sum_{i,j \in V} L_{ij} s_i s_j$$

- ▶ Corte en términos de las entradas L_{ij} del **Laplaciano del grafo** $\mathbf{L} = \mathbf{D} - \mathbf{A}$, i.e.,

$$C(\mathbf{s}) = \frac{1}{2} \mathbf{s}^\top \mathbf{L} \mathbf{s}, \quad \mathbf{s} := [s_1, \dots, s_{N_V}]^\top$$

- ▶ Maximizar modularidad $Q(\mathbf{s}) \propto \mathbf{s}^\top \mathbf{B} \mathbf{s}$ vs. Minimizar cortes $C(\mathbf{s}) \propto \mathbf{s}^\top \mathbf{L} \mathbf{s}$

Minimizar cortes del grafo

- ▶ Dado que si $|V_1| = N_1$ entonces $|V_2| = N_2 = N - N_1$, tenemos la restricción

$$\sum_{i \in V} s_i = \sum_{i \in V_1} (+1) + \sum_{i \in V_2} (-1) = N_1 - N_2 \Rightarrow \mathbf{1}^T \mathbf{s} = N_1 - N_2$$

- ▶ **Criterio de minimizar cortes** para la bisección de grafos se traduce en

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \{\pm 1\}^{N_v}} \mathbf{s}^T \mathbf{L} \mathbf{s}, \quad \text{sujeto a } \mathbf{1}^T \mathbf{s} = N_1 - N_2$$

- ▶ Nuevamente, restricciones binarias $\mathbf{s} \in \{\pm 1\}^{N_v}$ hacen que la minimización de cortes sea compleja

⇒ **Relajar restricciones binarias** como en maximización de la modularidad

Función de costo equivalente

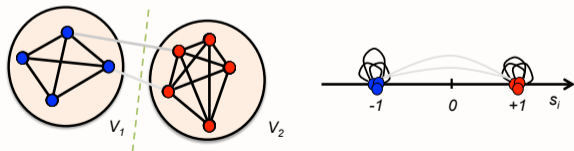
- Vimos que $\mathbf{s}^\top \mathbf{L} \mathbf{s} = \sum_{(i,j) \in E} (s_i - s_j)^2$, de donde la formulación resulta:

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \{\pm 1\}^{N_v}} \sum_{(i,j) \in E} (s_i - s_j)^2, \quad \text{sujeto a } \mathbf{1}^\top \mathbf{s} = N_1 - N_2$$

- ¿Cómo se interpreta esta función de costo equivalente?

⇒ Aristas que unen vértices en el mismo grupo no aportan a la suma

⇒ Aristas que unen vértices en grupos diferentes aportan 4 a la suma



► **Minimizar cortes:** asignar valores s_i a vértices i tal que pocas aristas crucen el 0

Relajar restricciones binarias

- ▶ Relajar las restricciones $\mathbf{s} \in \{\pm 1\}^{N_v}$ a $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = 1$

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \mathbf{s}^T \mathbf{L} \mathbf{s}, \quad \text{sujeto a } \mathbf{1}^T \mathbf{s} = N_1 - N_2 \text{ y } \mathbf{s}^T \mathbf{s} = 1$$

⇒ se resuelve directo usando multiplicadores de Lagrange

- ▶ Caracterización de la **solución** $\hat{\mathbf{s}}$ [Fiedler '73]:

$$\hat{\mathbf{s}} = \mathbf{v}_2 + \frac{N_1 - N_2}{N_v} \mathbf{1}$$

⇒ El **segundo vector propio más chico** \mathbf{v}_2 de \mathbf{L} verifica que $\mathbf{1}^T \mathbf{v}_2 = 0$

⇒ Mínimo corte resulta $C(\hat{\mathbf{s}}) = \hat{\mathbf{s}}^T \mathbf{L} \hat{\mathbf{s}} = \mathbf{v}_2^T \mathbf{L} \mathbf{v}_2 \propto \lambda_2$

- ▶ Si G está desconectado sabemos que $\lambda_2 = 0 = C(\hat{\mathbf{s}})$

⇒ Si G es susceptible de bisección, el corte es pequeño y también lo es λ_2

Bisección espectral de grafos

- ▶ ¿Cómo obtener las etiquetas binarias de la partición $\mathbf{s} \in \{\pm 1\}^{N_v}$ a partir de $\hat{\mathbf{s}} \in \mathbb{R}^{N_v}$?

⇒ nuevamente, maximizar al medida de similitud $\mathbf{s}^\top \hat{\mathbf{s}}$

$$s_i = f(\mathbf{v}_2) := \begin{cases} +1, & [\mathbf{v}_2]_i \text{ entre las } N_1 \text{ coordenadas más grandes de } \mathbf{v}_2 \\ -1, & \text{otro caso} \end{cases}$$

- ▶ Algoritmo espectral de bisección de grafos

s1: Calcular el laplaciano \mathbf{L} con entradas $L_{ij} = D_{ij} - A_{ij}$

s2: Hallar el vector propio asociado al segundo valor propio más chico \mathbf{v}_2 de \mathbf{L}

s3: Proponer pertenencia del vértice i como $\bar{s}_i = f(\mathbf{v}_2)$ (o $\underline{s}_i = f(-\mathbf{v}_2)$)

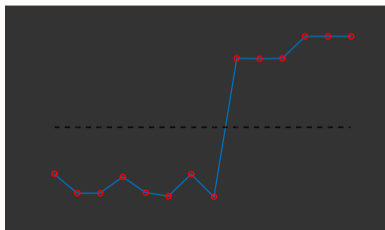
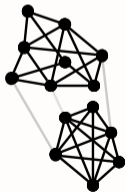
s4: Elegir entre $\bar{\mathbf{s}}$ y $\underline{\mathbf{s}}$ el que minimiza $C(\mathbf{s})$

Spectral gap en las entradas del vector de Fiedler

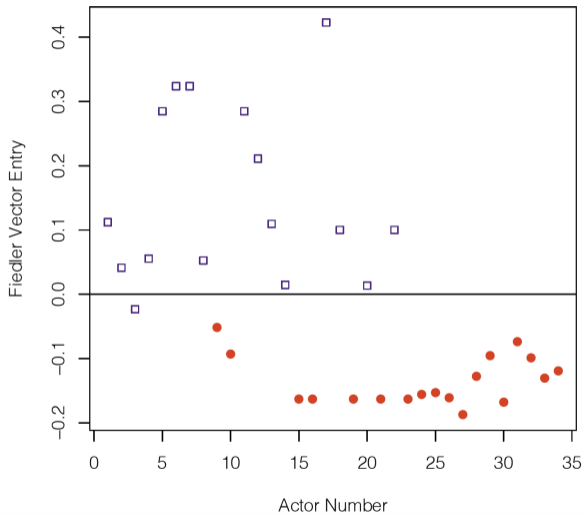
- ▶ Sea G no conectado con dos componentes conexas
 - L es diagonal con bloques, dos vectores propios más chicos indican grupos, i.e.,

$$\mathbf{v}_1 = [1, 1, \dots, 1, 0, \dots, 0]^\top \text{ and } \mathbf{v}_2 = [0, 0, \dots, 0, 1, \dots, 1]^\top$$

- ▶ Si G is conectado pero pasible de bisección $\mathbf{v}_1 = \mathbf{1}$ y $\lambda_2 \approx 0$
 - Además, $\mathbf{1}^\top \mathbf{v}_2 = \sum_i [\mathbf{v}_2]_i = 0 \Rightarrow$ entradas positivas y negativas en \mathbf{v}_2

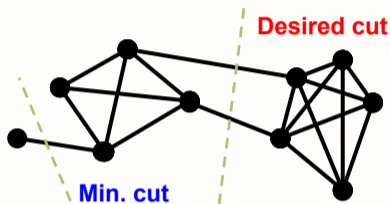


Vector de Fiedler para el Zachary Karate Club



Tamaño de comunidades desconocido

- Problema: bisección del grafo con **tamaño de los grupos desconocidos**
- Minimizar el corte del grafo puede no tener más sentido!



⇒ Costo $C := \sum_{i \in V_1, j \in V_2} A_{ij}$ indiferente a la estructura interna de los grupos

- Un mejor criterio es el **corte ponderado** R definido como

$$R := \frac{C}{|V_1|} + \frac{C}{|V_2|}$$

- **Particiones balanceadas:** comunidades pequeñas son penalizadas en costo

Minimización de corte ponderado

- ▶ Fijar una partición S de G en los grupos V_1 y V_2
- ▶ Definir $\mathbf{f} : \mathbf{f}(S) = [f_1, \dots, f_{N_v}]^T \in \mathbb{R}^{N_v}$ con entradas

$$f_i = \begin{cases} \sqrt{\frac{|V_2|}{|V_1|}}, & \text{si vértice } i \text{ pertenece a } V_1 \\ -\sqrt{\frac{|V_1|}{|V_2|}}, & \text{si vértice } i \text{ pertenece a } V_2 \end{cases}$$

- ▶ Se pueden probar las siguientes propiedades:
 1. $\sum_i f_i = 0$, i.e., $\mathbf{1}^T \mathbf{f} = 0$
 2. $\|\mathbf{f}\|^2 = N_v$
 3. $\mathbf{f}^T \mathbf{L} \mathbf{f} = C \left(\frac{N_2}{N_1} + \frac{N_1}{N_2} + 2 \right) = N_v R(S)$
- ▶ De 1-3 se deduce que la **minimización de corte ponderado** es equivalente a

$$\min_{\mathbf{f}} \mathbf{f}^T \mathbf{L} \mathbf{f}, \quad \text{sujeto a } \mathbf{1}^T \mathbf{f} = 0 \text{ y } \mathbf{f}^T \mathbf{f} = N_v$$

Bisección espectral y por corte ponderado

- ▶ Minimización por corte ponderado es también NP. Relajación:

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \mathbb{R}^{N_v}} \mathbf{s}^\top \mathbf{L} \mathbf{s}, \quad \text{sujeto a } \mathbf{1}^\top \mathbf{s} = 0 \text{ y } \mathbf{s}^\top \mathbf{s} = N_v$$

- ▶ Se obtiene la misma partición \hat{S} con el algoritmo de **bisección espectral**

s1: Calcular el Laplaciano \mathbf{L} con entradas $L_{ij} = D_{ij} - A_{ij}$

s2: Hallar el segundo vector propio más chico \mathbf{v}_2 de \mathbf{L}

s3: Asignar pertenencia al grupo del vértice i como $s_i = \text{sign}([\mathbf{v}_2]_i)$

- ▶ Criterio alternativo es el **corte normalizado** NC definido como

$$NC = \frac{C}{\text{vol}(V_1)} + \frac{C}{\text{vol}(V_2)}, \quad \text{vol}(V_i) := \sum_{v \in V_i} d_v, \quad i = 1, 2$$

⇒ corresponde a usar el **Laplaciano normalizado** $\mathbf{D}^{-1} \mathbf{L}$

Múltiples comunidades

- ▶ Se define el corte ponderado de una partición U en K grupos $\{\mathcal{V}_i\}_{i=1}^K$ como

$$R(U) := \sum_{i=1}^K \frac{C(\mathcal{V}_i, \mathcal{V}_i^c)}{|\mathcal{V}_i|}$$

- ▶ Problema de la minimización de la relajación del corte ponderado resulta:

$$\hat{\mathbf{U}} = \arg \min_{\mathbf{U} \in \mathbb{R}^{N_v \times K}} \text{trace}(\mathbf{U}^T \mathbf{L} \mathbf{U}), \quad \text{s. to } \mathbf{U}^T \mathbf{U} = \mathbf{I}$$

- ▶ Partición \hat{U} dada por el algoritmo de clustering espectral

s1: Calcular el Laplaciano \mathbf{L} con entradas $L_{ij} = D_{ij} - A_{ij}$

s2: Hallar los K vectores propios más pequeños $\mathbf{v}_1, \dots, \mathbf{v}_K$ of \mathbf{L}

s3: Definir $\hat{\mathbf{U}} = [\mathbf{v}_1, \dots, \mathbf{v}_K]$, *embedding* del vértice i como la fila $\hat{\mathbf{u}}_i^T \in \mathbb{R}^{1 \times K}$

S4: Asignar vértices a comunidades vía K -means en los embeddings

Clustering espectral

- ▶ ¿Porqué funciona la partición espectral de grafos?
- ▶ Observar que:

$$\text{traza}(\hat{\mathbf{U}}^T \mathbf{L} \hat{\mathbf{U}}) = \sum_{(i,j) \in \mathcal{E}} A_{ij} \|\hat{\mathbf{u}}_i^T - \hat{\mathbf{u}}_j^T\|^2$$

- ⇒ Embeddings están cerca en \mathbb{R}^K si i, j están bien conectados en G
- ⇒ Se conocen como Laplacian eigenmaps [Belkin-Niyogi'01]

- ▶ **Para recordar:** codificar al estructura del grado en embeddings de baja dimensión



Otros algoritmos...

- ▶ Hay muchos otros algoritmos, con enfoques similares o muy distintos
 - Metodo de Radicchi propone otra definición de betweenness
 - Métodos basados en teoría de la información: InfoMap
 - Métodos basados en inferencia estadística (necesita modelo de grafo)
 - Métodos clásicos aplicados a vectores de características (u otros vectores asociados)
 - Clustering jerárquico: basta variar medida de (dis)similaridad entre vértices y entre grupo de vértices para obtener un nuevo método
- ▶ No hablamos ni de complejidad ni de garantías teóricas...
- ▶ Detección de otro tipo de estructuras: comunidades con solapamiento, estructuras *core-periphery*
- ▶ Mark Newman: Networks (2018)

Medidas de desempeño

- ▶ ¿Cuál es el **mejor algoritmo de detección de comunidades**?
 - ⇒ depende: buen desempeño, rigor matemático, velocidad de procesamiento, escalabilidad...
- ▶ ¿Cómo evaluar el **buen desempeño**?
 1. testear en **redes reales** donde se conoce la estructura de comunidades
 2. testear en redes sintéticas
- ▶ Testeo en **redes reales**: probar muchos métodos y si la mayoría coincide tomarlo como *ground-truth*
- ▶ Ejemplos que vimos antes:
 - ⇒ karate club! (sociedad “Karate Club Club”)
 - ⇒ Blogs políticos
 - ⇒ Equipos deportivos
 - ⇒ Red social de delfines (Lusseau’03): similar a ZKC

Medidas de desempeño

- ▶ Testeo en **redes sintéticas**: modelos de grafos aleatorios con estructura de comunidades predeterminada (*coming soon!*)
- ▶ ¿Cómo medir desempeño cuando la detección no es perfecta?
- ▶ **Fracción de vértices bien clasificados**:
 - ⇒ máximo valor sobre las permutaciones de etiquetas (mínimo por azar)
 - ⇒ pueden detectarse diferente cantidad de comunidades
- ▶ **Rand Index**: *fracción de vértices en la misma (o distinta) comunidad en la red test que son asignados a la misma (o distinta) comunidad por el algoritmo*

$$R = 2 \frac{(s + d)}{N_v(N_v - 1)} \quad s = \sum_{i < j} \delta_{g_i g_j} \delta_{t_i t_j} \quad y \quad d = \sum_{i < j} (1 - \delta_{g_i g_j})(1 - \delta_{t_i t_j})$$

⇒ $0 \leq R \leq 1$ e independencia de las etiquetas

▶ Información mutua normalizada