Block 4: Community detection

ELEC 573: Network Science and Analytics

Santiago Segarra

Electrical and Computer Engineering
Rice University
segarra@rice.edu

Fall 2021

Local density

Local density, clustering coefficient and group centrality

Case study: Diagnosing Alzheimer's disease

Network connectivity and assortativity mixing

Case study: Analysis of an epileptic seizure

Network community detection

Modularity maximization

Spectral graph partitioning

- Many network analytic questions pertain to network cohesion

Example

- Q1: Do common friends of an actor end up being friends?
- Q2: What collections of proteins in a cell work closely together?
- Q3: Does Web page structure separate relative to content?
- Q4: What portion of the Internet topology constitutes a 'backbone'?

Network cohesion

- Many network analytic questions pertain to network cohesion

Example

- Q1: Do common friends of an actor end up being friends?
- Q2: What collections of proteins in a cell work closely together?
- Q3: Does Web page structure separate relative to content?
- Q4: What portion of the Internet topology constitutes a 'backbone'?

- Definitions of network cohesion depend on the context

  $\Rightarrow$ Scale from local (e.g., triads) to global (e.g., giant components)

  $\Rightarrow$ Specified explicitly (e.g., cliques) or implicitly (e.g., clusters)

Cohesive subgroups

- Cohesive subgroups defined by social network analysts as:
  'Actors connected via dense, directed, reciprocated relations'
- Allow sharing information, creating solidarity, collective actions
  Ex: religious cults, terrorist cells, sport clubs, military platoons, ...

Cohesive subgroups

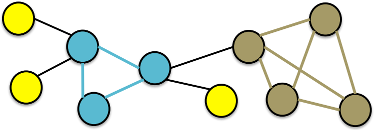
- Cohesive subgroups defined by social network analysts as:
  'Actors connected via dense, directed, reciprocated relations'
- Allow sharing information, creating solidarity, collective actions
  Ex: religious cults, terrorist cells, sport clubs, military platoons, ...
- Desirable properties of a cohesive subgroup
  - $\Rightarrow$ Familiarity (degree);
  - $\Rightarrow$ Reachability (distance);
  - $\Rightarrow$ Robustness (connectivity); and
  - $\Rightarrow$ Density (edge density)

Cohesive subgroups

- Cohesive subgroups defined by social network analysts as:
  'Actors connected via dense, directed, reciprocated relations'
- Allow sharing information, creating solidarity, collective actions
  Ex: religious cults, terrorist cells, sport clubs, military platoons, ...
- Desirable properties of a cohesive subgroup
  - $\Rightarrow$ Familiarity (degree);
  - $\Rightarrow$ Reachability (distance);
  - $\Rightarrow$ Robustness (connectivity); and
  - $\Rightarrow$ Density (edge density)
- Natural to think of cliques, i.e., complete subgraphs of G

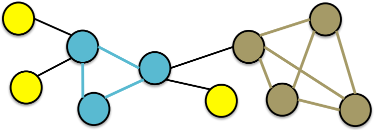
Local density and cliques

- Large cliques are rare; single missing edge destroys property



Local density and cliques

- Large cliques are rare; single missing edge destroys property

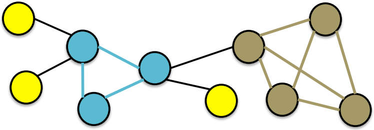


- Sufficient condition for the existence of a size-n clique

$$N_e > \frac{N_v^2}{2}\frac{(n-2)}{(n-1)}, \text{ while sparse graphs have } N_e = O(N_v)$$

Local density and cliques

- Large cliques are rare; single missing edge destroys property



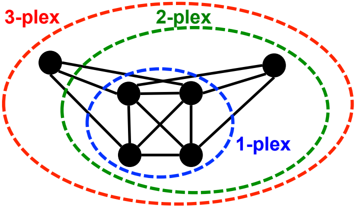
- Sufficient condition for the existence of a size-n clique

$$N_e > \frac{N_v^2}{2}\frac{(n-2)}{(n-1)}, \text{ while sparse graphs have } N_e = O(N_v)$$

- Complexity of clique-related algorithms varies widely
  - Is $U \subseteq V$ a clique? Is it maximal? $O(N_v + N_e)$ complexity
  - Identifying all triangles in G? $O(N_v^3)$ ($O(N_v^{\sqrt{2}})$ for sparse graphs)
  - Does G have a maximal clique of size $\geq n$? NP-complete

Relaxing cliques by familiarity

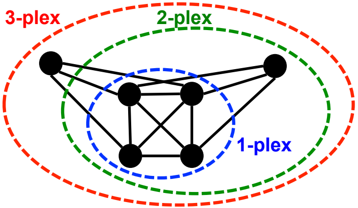
- Cliques tend to be an overly restrictive notion of cohesiveness. Relax!
- **Def:** An induced subgraph $G'(V', E')$ is a k-plex if $d_v(G') \geq |V'| - k$ for all $v \in V'$, and G' is maximal



$\Rightarrow$ Degrees are in the induced subgraph G', not in G

Relaxing cliques by familiarity

- Cliques tend to be an overly restrictive notion of cohesiveness. Relax!
- **Def:** An induced subgraph $G'(V', E')$ is a k-plex if $d_v(G') \geq |V'| - k$ for all $v \in V'$, and G' is maximal



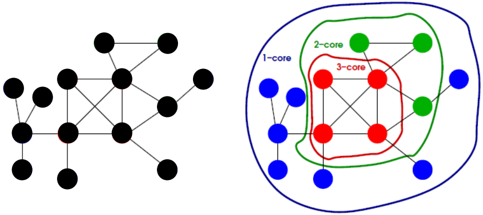
$\Rightarrow$ Degrees are in the induced subgraph G', not in G

- No vertex is missing more than $k - 1$ of its possible $|V'| - 1$ edges

  $\Rightarrow$ A clique is a 1-plex

- Complex: problems involving k-plexes scale like clique counterparts

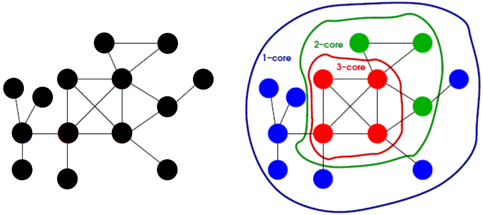
- The *k*-core decomposition. A dual notion of cohesiveness



- **Def:** An induced subgraph $G'(V', E')$ is a k-core if $d_v(G') \geq k$ for all $v \in V'$, and G' is maximal

The k-core decomposition

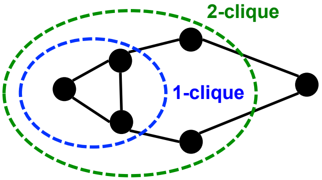
- The *k*-core decomposition. A dual notion of cohesiveness



- **Def:** An induced subgraph $G'(V', E')$ is a k-core if $d_v(G') \geq k$ for all $v \in V'$, and G' is maximal
- Hierarchy: larger "coreness" $\Rightarrow$ larger degrees and centrality
- Algorithm: recursively prune all vertices of degree less than k

  $\Rightarrow$ Complexity $O(N_v + N_e)$, very efficient for sparse graphs

- Idea: specify that any two actors are no more than k hops away
- **Def:** An induced subgraph $G'(V', E')$ is a k-clique if $d(u, v) \leq k$ for all $u, v \in V'$

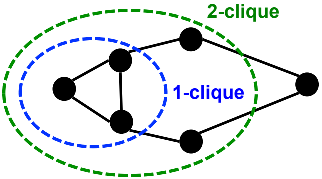


$\Rightarrow$ Useful if important social processes occur via intermediaries

$\Rightarrow$ $\text{diam}(G')$ may exceed k, if distances used are in G

Relaxing cliques by reachability

- Idea: specify that any two actors are no more than k hops away
- **Def:** An induced subgraph $G'(V', E')$ is a k-clique if $d(u, v) \leq k$ for all $u, v \in V'$



$\Rightarrow$ Useful if important social processes occur via intermediaries

$\Rightarrow$ $\text{diam}(G')$ may exceed k, if distances used are in G

- Likewise, a k-club is a subgraph G' with $\text{diam}(G') \leq k$

$\Rightarrow$ k-clubs are k-cliques but the converse is not true, in general

- A natural measure of density of a subgraph $G'(V', E')$ is

$$\text{den}(G') = \frac{|E'|}{|V'|(|V'|-1)/2} \in [0, 1]$$

$\Rightarrow$ Quantifies how close is G' to being a clique

Quantifying local density

- A natural measure of density of a subgraph $G'(V', E')$ is

$$\mathrm{den}(G') = \frac{|E'|}{|V'|(|V'|-1)/2} \in [0, 1]$$

$\Rightarrow$ Quantifies how close is G' to being a clique

- $\mathrm{den}(G')$ is just a rescaling of the average degree $\bar{d}(G')$

$$\bar{d}(G') = \frac{1}{|V'|}\sum_{v \in V'} d_v = \frac{2|E'|}{|V'|} \Rightarrow \mathrm{den}(G') = \frac{\bar{d}(G')}{|V'|-1}$$

Quantifying local density

- A natural measure of density of a subgraph $G'(V', E')$ is

$$\mathrm{den}(G') = \frac{|E'|}{|V'|(|V'|-1)/2} \in [0, 1]$$

$\Rightarrow$ Quantifies how close is G' to being a clique

- $\mathrm{den}(G')$ is just a rescaling of the average degree $\bar{d}(G')$

$$\bar{d}(G') = \frac{1}{|V'|}\sum_{v \in V'} d_v = \frac{2|E'|}{|V'|} \Rightarrow \mathrm{den}(G') = \frac{\bar{d}(G')}{|V'|-1}$$

- Flexibility in choosing G' to measure local density via $\mathrm{den}(G')$

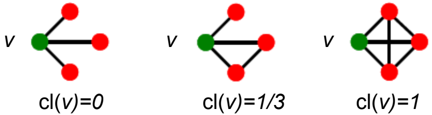
$\Rightarrow$ Use v's egonet G'_v, subgraph induced by v and its neighbors

$\Rightarrow$ Density of the overall graph G is $\mathrm{den}(G) = \frac{2N_e}{N_v(N_v-1)}$

- Q: What fraction of v's neighbors are themselves connected?
- **Def:** The clustering coefficient $\text{cl}(v)$ of $v \in V$ is

$$\text{cl}(v) = \frac{2|E_v|}{d_v(d_v - 1)} \in [0, 1]$$

$\Rightarrow$ $|E_v|$ is the number of edges among v's neighbors

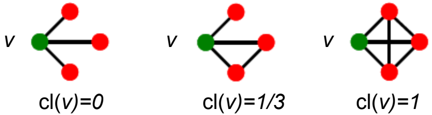


Clustering coefficient

- Q: What fraction of v's neighbors are themselves connected?
- **Def:** The clustering coefficient $\text{cl}(v)$ of $v \in V$ is

$$\text{cl}(v) = \frac{2|E_v|}{d_v(d_v - 1)} \in [0, 1]$$

$\Rightarrow$ $|E_v|$ is the number of edges among v's neighbors



- The global clustering coefficient is usually defined as

$$\text{cl}(G) = \frac{3 \times \ \# \text{ triangles}}{\# \text{ triplets of connected nodes}}$$

- Which is in general different from the (average) local coefficient

$$\bar{\text{cl}} = \frac{1}{N_v} \sum_{v \in V} \text{cl}(v)$$

Local efficiency

- An alternative to the local clustering coefficient
  - $\Rightarrow$ Measure of how efficiently information is exchanged
- Define the efficiency of a graph G as

$$E(G) = \frac{1}{N_v(N_v - 1)} \sum_{u \neq v \in G} \frac{1}{d(u, v)}$$

- Based on this, we define the local efficiency at node v as

$$E(v) = E(G_v), \qquad G_v = (\mathcal{N}(v) \backslash \{v\}, E_{\mathcal{N}(v) \backslash \{v\} \times \mathcal{N}(v) \backslash \{v\}})$$

- G_v is the subgraph induced by the neighborhood of v
  - $\Rightarrow$ When v is deleted

Extending centrality to vertex groups

- Capture the importance of node subgroups [Everett et al'99]
- Q1: Are engineers more popular than accountants in an organization?
- Q2: How do we select board members with most business influence?
- Group centrality measures to generalize vertex centrality

Extending centrality to vertex groups

- Capture the importance of node subgroups [Everett et al'99]
- Q1: Are engineers more popular than accountants in an organization?
- Q2: How do we select board members with most business influence?
- Group centrality measures to generalize vertex centrality
- Ex: Consider subgraph $G'(V', E')$ induced by node subset V'
  - Let $U_{V'} \subset V \setminus V'$ with edges to members of V'
- Group degree centrality of node subset V'

$$d_{V'} = |U_{V'}|$$

$\Rightarrow$ Number of non-group nodes connected to G'

Group centrality measures

- **Def:** Distance from $v \in V$ to a group of nodes $V' \subset V$ is

$$d_*(v, V') = \min_{u \in V'} d(u, v)$$

- Group closeness centrality of node subset V'

$$c_{Cl}(V') = \frac{1}{\sum_{u \in V \setminus V'} d_*(u, V')}$$

Group centrality measures

- **Def:** Distance from $v \in V$ to a group of nodes $V' \subset V$ is

$$d_*(v, V') = \min_{u \in V'} d(u, v)$$

- Group closeness centrality of node subset V'

$$c_{Cl}(V') = \frac{1}{\sum_{u \in V \setminus V'} d_*(u, V')}$$

- Group betweenness centrality of node subset V'

$$c_{Be}(V') = \sum_{s \neq t \in V \setminus V'} \frac{\sigma(s, t|V')}{\sigma(s, t)}$$

  - $\sigma(s, t)$ is the total number of $s - t$ shortest paths $(s, t \in V \setminus V')$
  - $\sigma(s, t|V')$ is the number of $s - t$ shortest paths through $v \in V'$

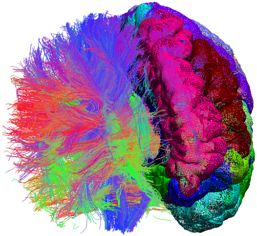
You are here

Wk.	Date	Topic	HW	Project
1	23-Aug	Introduction to course	HW0 out	
2	30-Aug	Graph theory	HW0 solutions posted	
3	6-Sep	LABOR DAY (no class)	HW1 out	
4	13-Sep	Centrality measures / Community detection		
5	20-Sep	Community detection		
6	27-Sep	Signal Processing and Deep learning for graphs	HW1 due	
7	4-Oct	Signal Processing and Deep learning for graphs	HW2 out	
8	11-Oct	FALL BREAK (no class)		
9	18-Oct	Network models	HW2 due	
10	25-Oct	Network models	HW3 out	Project proposal due
11	1-Nov	Epidemics		
12	8-Nov	Inference of network topologies, features, and processes	HW3 due	
13	15-Nov	Inference of network topologies, features, and processes		
14	22-Nov	Inference of network topologies, features, and processes		Project progress report
15	29-Nov	Inference of network topologies, features, and processes		
	13-Dec	Project presentation (video recording) and final report due		

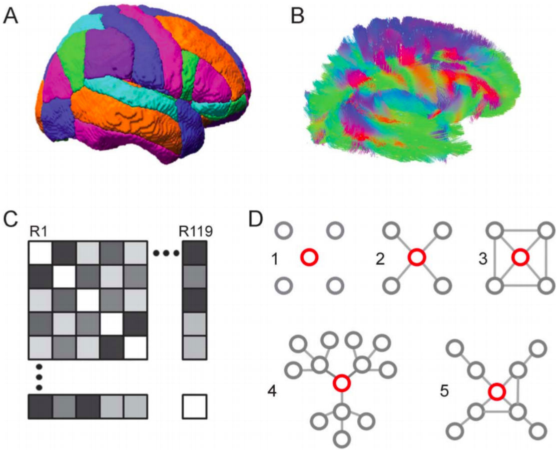
Networks for diagnosis

- Detect Alzheimer disease vs non-Alzheimer degeneration

  $\Rightarrow$ Binary classification problem
- Data: Diffusion-weighted imaging tractography

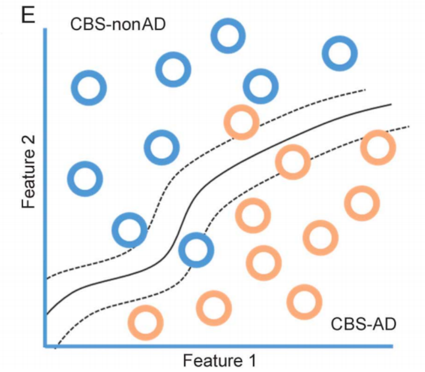


- Use network-based statistics as a feature generation process
- Medaglia, Huang, Segarra, et al., "Brain Network Efficiency is Influenced by Pathological Source of Corticobasal Syndrome", *Neurology*, 2017.



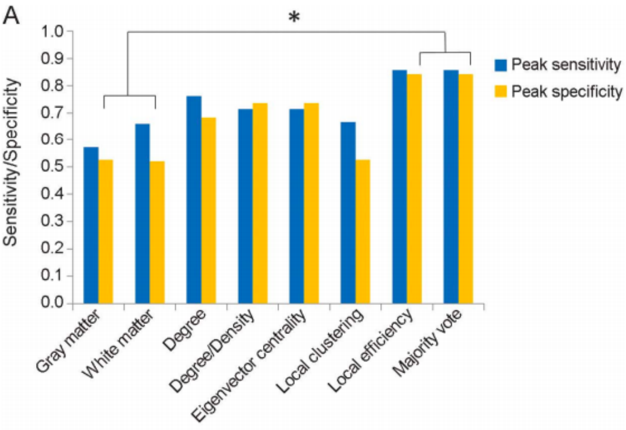
- 1) white matter, 2) degree, 3) clustering, 4) eigenvector, 5) efficiency
- Every feature choice leads to a multidimensional representation

Kernelized SVM



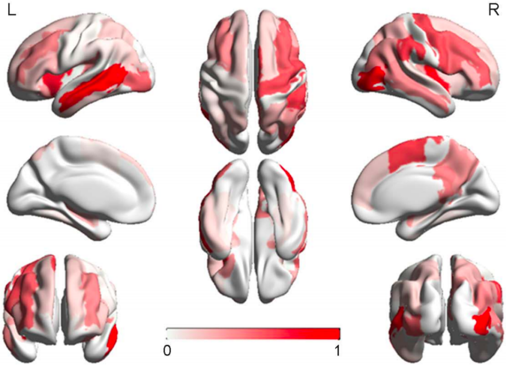
- Great for binary classification of high-dimensional vectors
- Kernel allows for non-linear decision boundaries

Results



- Local efficiency minimizes the classification error
- Network-based features outperform node-centric features

SVM and interpretability



- Spatial distribution of weights for the local efficiency classifier
  - $\Rightarrow$ Left middle temporal gyrus
  - $\Rightarrow$ Bilateral insula
  - $\Rightarrow$ Right lateral temporal-parietal-occipital

Local density, clustering coefficient and group centrality

Case study: Diagnosing Alzheimer's disease

Network connectivity and assortativity mixing

Case study: Analysis of an epileptic seizure

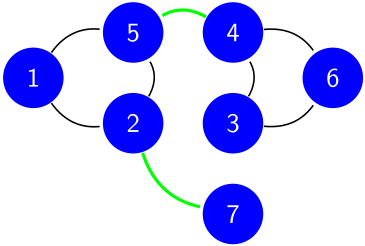
Network community detection

Modularity maximization

Spectral graph partitioning

Network connectivity and robustness

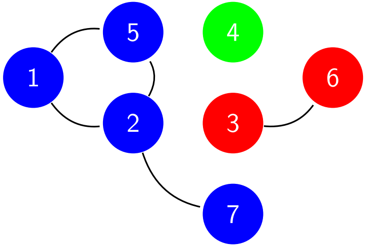
- Connectivity relevant when taking a larger, global perspective
  - Q: Does a given graph G separate into different subgraphs?
  - If it does not, a 'less robust' network is closer to splitting
- **Def:** Graph is connected if $\exists$ walks joining each vertex pair



$\Rightarrow$ If bridge edges are removed, the graph becomes disconnected

Connected components

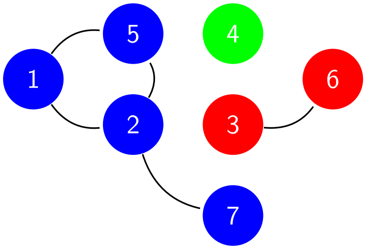
- A component is a maximally-connected subgraph



- In figure $\Rightarrow$ Components are $\{1, 2, 5, 7\}$, $\{3, 6\}$ and $\{4\}$
  $\Rightarrow$ Subgraph $\{3, 4, 6\}$ not connected, $\{1, 2, 5\}$ not maximal

Connected components

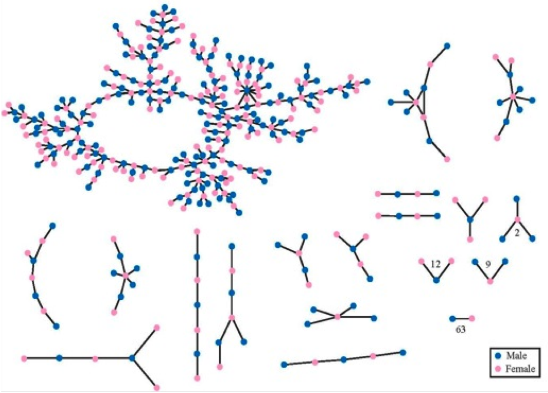
- A component is a maximally-connected subgraph



- In figure $\Rightarrow$ Components are $\{1, 2, 5, 7\}$, $\{3, 6\}$ and $\{4\}$
  $\Rightarrow$ Subgraph $\{3, 4, 6\}$ not connected, $\{1, 2, 5\}$ not maximal
- Disconnected graphs have 2 or more components
  - $\Rightarrow$ Number of components $=$ Multiplicity of eigenvalue 0 for $\mathbf{L}$
  - $\Rightarrow$ Largest component often called giant component
- Check for connectivity, identify components with DFS, BFS: $O(N_v)$

Giant connected components

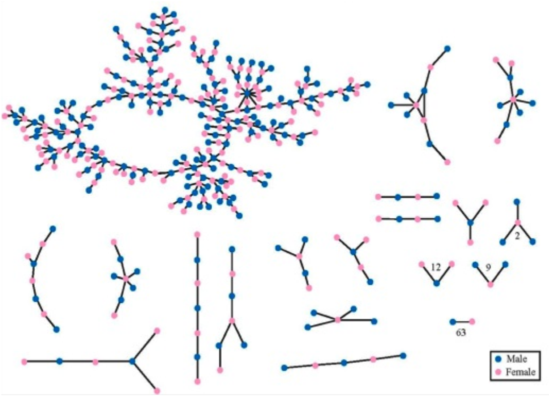
- Large real-world networks typically exhibit one giant component
- Ex: romantic relationships in a US high school [Bearman et al'04]



- Q: Why do we expect to find a single giant component?

Giant connected components

- Large real-world networks typically exhibit one giant component
- Ex: romantic relationships in a US high school [Bearman et al'04]



- Q: Why do we expect to find a single giant component?
- A: Well, it only takes one edge to merge two giant components

Average path length and small world

- Giant components tend to exhibit the small world property
- Small refers to the average path length

$$\bar{\ell} = \binom{N_v}{2}^{-1} \sum_{u \neq v \in V} d(u, v) = O(\log N_v)$$

Ex: facilitates spread of gossip, diseases, search for WWW content

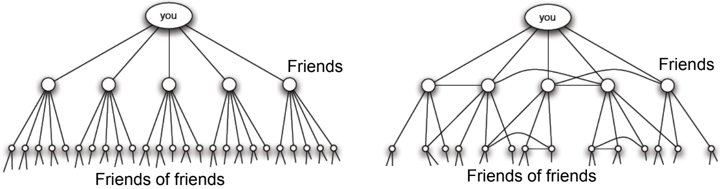
Average path length and small world

- Giant components tend to exhibit the small world property
- Small refers to the average path length

$$\bar{\ell} = \binom{N_v}{2}^{-1} \sum_{u \neq v \in V} d(u,v) = O(\log N_v)$$

Ex: facilitates spread of gossip, diseases, search for WWW content

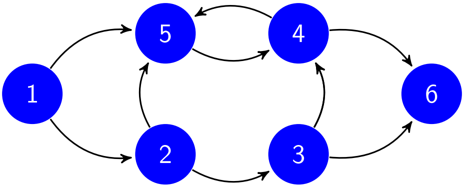
- Not too surprising that the property holds. Informal argument:



- If $d_v = d$, after h_* hops have $d^{h_*} \approx N_v$ $\Rightarrow \bar{\ell} \approx h_* = O(\log N_v)$

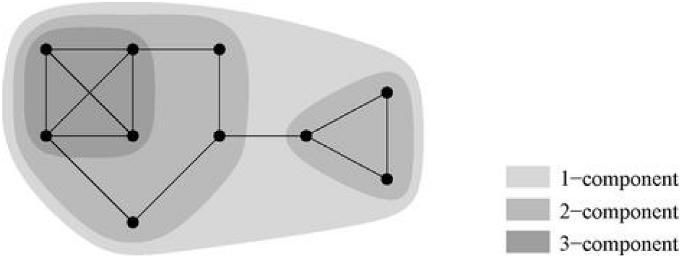
Connectivity of directed graphs

- Connectivity is more subtle with directed graphs. Two notions
- **Def:** Digraph is strongly connected if for every pair $u, v \in V$, u is reachable from v (via a directed walk) and vice versa
- **Def:** Digraph is weakly connected if connected after disregarding arc directions, i.e., the underlying undirected graph is connected



- Above graph is weakly connected but not strongly connected
  $\Rightarrow$ Strong connectivity obviously implies weak connectivity

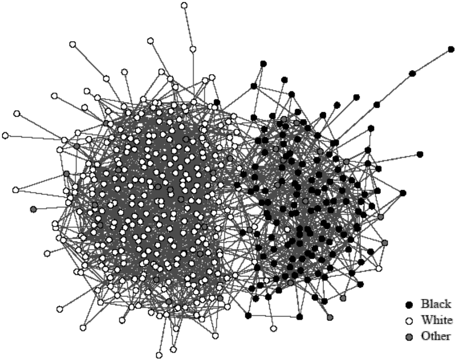
- Generalization of the notion of connected component
- **Def:** A *k*-(connected) component is a maximal subset of vertices such that each is reachable by each other by at least *k* vertex-independent paths
  - ⇒ Paths not sharing any intermediate nodes



- 1-component coincides with the definition of a connected component

Assortative mixing

- People have a stronger tendency to associate with equals
  - $\Rightarrow$ Tendency is called homophily or assortative mixing



- Ex: high-school students by race, bloggers by political party, ...
  - $\Rightarrow$ Can have disassortative mixing e.g., romantic relationships

Quantifying assortative mixing

- Suppose that vertex characteristics are categorical, e.g., male/female
- Let f_{ij} be the fraction of edges joining vertices of categories C_i, C_j

  $\Rightarrow$ $f_{i+} = \sum_j f_{ij}$ is the i-th marginal row sum (similarly for f_{+i})

Quantifying assortative mixing

- Suppose that vertex characteristics are categorical, e.g., male/female
- Let f_{ij} be the fraction of edges joining vertices of categories C_i, C_j

  $\Rightarrow$ $f_{i+} = \sum_j f_{ij}$ is the i-th marginal row sum (similarly for f_{+i})
- Define the assortativity coefficient [Newman'03]

$$r_a = \frac{\sum_i f_{ii} - \sum_i f_{i+} f_{+i}}{1 - \sum_i f_{i+} f_{+i}}$$

$\Rightarrow$ $f_{i+} f_{+i}$ is the expected fraction of edges joining nodes in C_i

$\Rightarrow$ Random edges preserving degree distribution yields $r_a = 0$

Quantifying assortative mixing

- Suppose that vertex characteristics are categorical, e.g., male/female
- Let f_{ij} be the fraction of edges joining vertices of categories C_i, C_j

  $\Rightarrow f_{i+} = \sum_j f_{ij}$ is the i-th marginal row sum (similarly for f_{+i})
- Define the assortativity coefficient [Newman'03]

$$r_a = \frac{\sum_i f_{ii} - \sum_i f_{i+} f_{+i}}{1 - \sum_i f_{i+} f_{+i}}$$

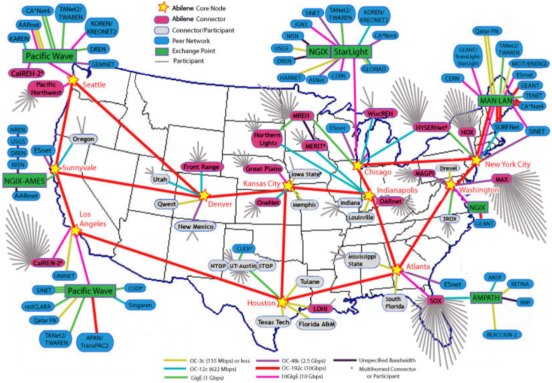
  $\Rightarrow f_{i+} f_{+i}$ is the expected fraction of edges joining nodes in C_i

  $\Rightarrow$ Random edges preserving degree distribution yields $r_a = 0$
- Perfectly assortative mixing yields $r_a^{\max} = 1$, while the minimum is

$$r_a^{\min} = -\frac{\sum_i f_{i+} f_{+i}}{1 - \sum_i f_{i+} f_{+i}} > -1$$

Example: Abilene network

- Abilene network for US universities and research labs
  - ‘Core’ nodes, as well as e.g., ‘Connector’ nodes and ‘Exchange points’



- Hierarchical structure, suggestive of disassortative mixing

- Tabulated counts of inter-category edges in Abilene

	Core	**Exchange**	**Peer**	**Conn.**	**Part.**	**Conn./Part.**
Core	14	6	5	17	0	16
Exchange	6	1	46	2	0	0
Peer	5	46	0	0	0	1
Conn.	17	2	0	0	203	0
Part.	0	0	0	203	0	34
Conn./Part.	16	0	1	34	34	0

- Fractions f_{ij} obtained by scaling table entries by the total of 675
- Assortativity coefficient $r_a = -0.3162$, close to $r_a^{\min} = -0.3461$

  $\Rightarrow$ Strongly supports our suspicion of disassortative mixing

Local density, clustering coefficient and group centrality

Case study: Diagnosing Alzheimer's disease

Network connectivity and assortativity mixing

Case study: Analysis of an epileptic seizure

Network community detection

Modularity maximization

Spectral graph partitioning

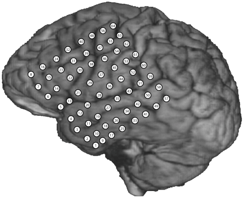
Network analysis and epilepsy

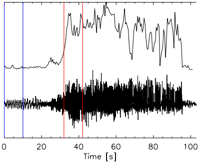
- Epilepsy is the world's most common serious brain disorder

  $\Rightarrow$ Seizures, i.e., recurrent abnormal neuronal activity

- Ex: Network-oriented analysis of epileptic seizure data in humans
- M. A. Kramer et al, "Emergent network topology at seizure onset in humans," *Epilepsy Res.*, vol. 79, pp. 173-186, 2008
- Leverage few summaries of network characteristics we learnt so far

- Electrode grid (8x8) implanted in the cortical surface of the brain
  - ⇒ Also implanted two strips of 6 electrodes (deeper, not shown)
- Electrocorticogram (ECoG) data; voltages indicative of brain activity





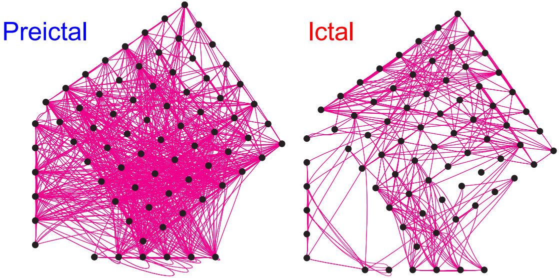
- Two 10 sec. intervals of interest for comparison:
  - ⇒ Preictal period: prior to the epileptic seizure
  - ⇒ Ictal period: immediately after start of seizure
- Top time-series is smoothed, averaged over 8 seizure signals

Network construction

- Network $\rightarrow$ represent couplings among brain regions
  - $\Rightarrow$ Graphs for the preictal and ictal periods, for 8 seizures
- Vertices: correspond to the 76 electrodes (cortical brain regions)
- Edges: threshold correlations between pairwise 10 sec. time series

Network construction

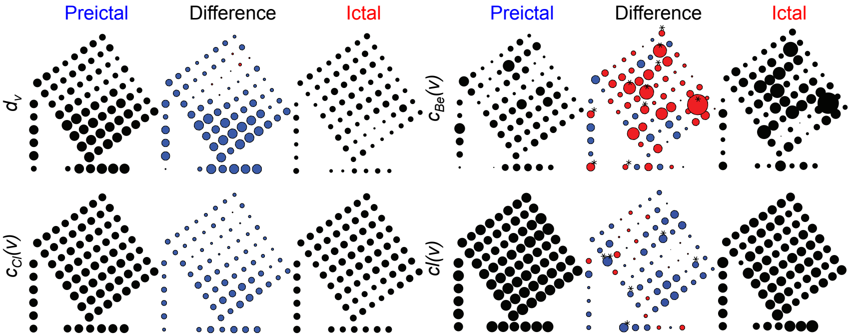
- Network $\rightarrow$ represent couplings among brain regions
  - $\Rightarrow$ Graphs for the preictal and ictal periods, for 8 seizures
- Vertices: correspond to the 76 electrodes (cortical brain regions)
- Edges: threshold correlations between pairwise 10 sec. time series



- Brain is in two very different states before and during seizure
  - $\Rightarrow$ Thinning of edges, coupling reduction at seizure onset
  - $\Rightarrow$ Closest to the strips, where seizure was suspected to emanate

Summaries of network characteristics

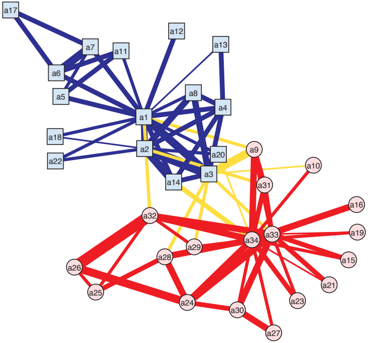
- Evaluated degree, closeness, betweenness centrality; clustering coeff.
  - $\Rightarrow$ Show preictal and ictal periods, as well as their difference



- Identifies spatially localized brain regions that may facilitate seizures
  - $\Rightarrow$ May serve to more precisely guide surgical intervention

Network of the week

- Zachary's Karate club
- Wayne Zachary. "An information flow model for conflict and fission in small groups", *J. of Anthropological Research*, 1977.



- The club split into two, and Zachary could explain the split (almost) perfectly

"Zachary's Karate Club" club

- *"The first scientist at any conference on networks who uses Zachary's karate club as an example is inducted into the Zachary Karate Club Club, and awarded a prize"*

"Zachary's Karate Club" club

- *"The first scientist at any conference on networks who uses Zachary's karate club as an example is inducted into the Zachary Karate Club Club, and awarded a prize"*

Some anecdotes

- "Indeed, the ZKCC trophy is one of my proudest achievements ;) I had been trying to get it for some time. Bonus info about the trophy – I believe it originally cost about $13, but much more has been spent on it since then for internationally shipping it around the world (e.g. overnight FedEx for a price much more than $13) when current holders have left it behind instead of bringing it to a conference. As far as I'm aware this has been done more than once and I heard it got stuck in Spanish customs at one point and sent back (who knows why) meaning that it ultimately had to be shipped again." - Leto Peel
- "... yeah the mysterious ZKCC trophy. My favorite anecdote related to this is how Renaud Lambiotte send a physical (3D printed) Karate Club network to Mason Porter in Oxford (it might have been after Mason got the Trophy). However, her Majesty's postal service did not treat the shipment gently enough, resulting in a poorly partitioned Karate club – a fact which Mason complained about afterwards" - Michael Schaub

Local density, clustering coefficient and group centrality

Case study: Diagnosing Alzheimer's disease

Network connectivity and assortativity mixing

Case study: Analysis of an epileptic seizure

Network community detection

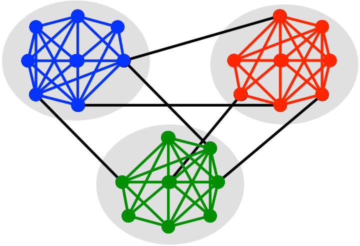
Modularity maximization

Spectral graph partitioning

- Nodes in real-world networks organize into communities

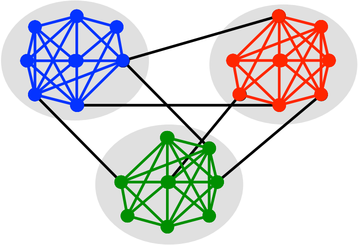
  Ex: families, clubs, political organizations, proteins by function, . . .

- Supported by Granovetter's strength of weak ties theory



Unveiling network communities

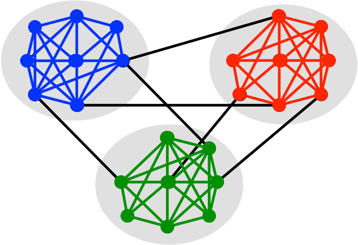
- Nodes in real-world networks organize into communities
  Ex: families, clubs, political organizations, proteins by function, . . .
- Supported by Granovetter's strength of weak ties theory



- Community (a.k.a. group, cluster, module) members are:
  $\Rightarrow$ Well connected among themselves
  $\Rightarrow$ Relatively well separated from the rest

Unveiling network communities

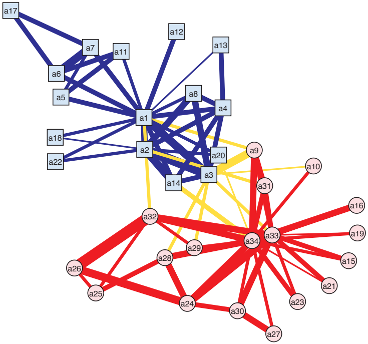
- Nodes in real-world networks organize into communities
  Ex: families, clubs, political organizations, proteins by function, . . .
- Supported by Granovetter's strength of weak ties theory



- Community (a.k.a. group, cluster, module) members are:
  - $\Rightarrow$ Well connected among themselves
  - $\Rightarrow$ Relatively well separated from the rest
- Exhibit high cohesiveness w.r.t. the underlying relational patterns

Zachary's karate club

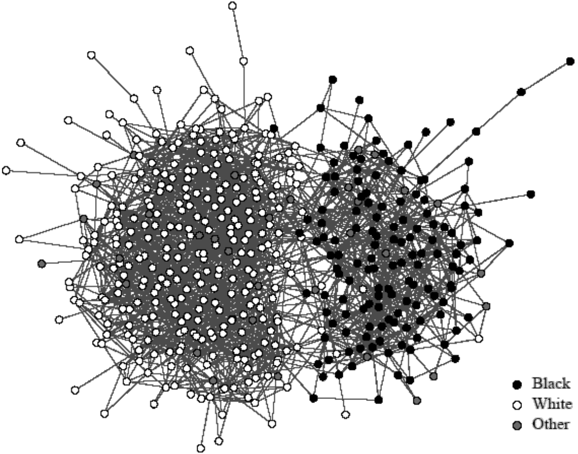
- Social interactions among members of a karate club in the 70s



- Zachary witnessed the club split in two during his study
  - ⇒ Toy network, yet canonical for community detection algorithms
  - ⇒ Offers "ground truth" community membership (a rare luxury)

High-school students

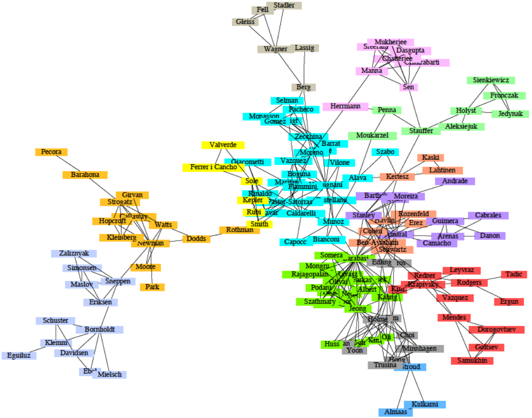
- Network of social interactions among high-school students



- Strong assortative mixing, with race as latent characteristic

Physicists working on Network Science

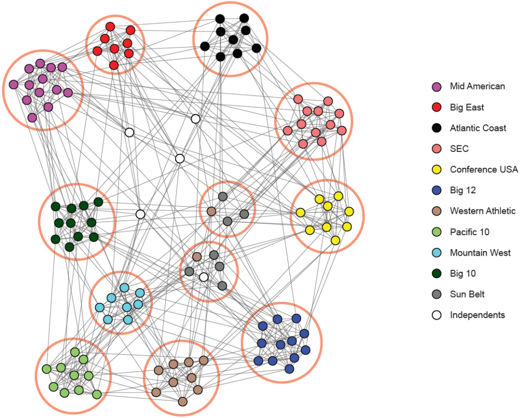
- Coauthorship network of physicists publishing networks' research



- Tightly-knit subgroups are evident from the network structure

College football

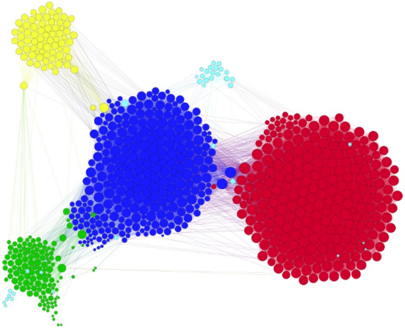
- Vertices are NCAA football teams, edges are games during Fall'00



- Communities are the NCAA conferences and independent teams

Facebook friendships

- Facebook egonet with 744 vertices and 30K edges



- Asked "ego" to identify social circles to which friends belong
  - $\Rightarrow$ Company, high-school, basketball club, squash club, family

Community detection and graph partitioning

- Community detection is a challenging clustering problem
  - C1) No consensus on the structural definition of community
  - C2) Node subset selection often intractable
  - C3) Lack of ground-truth for validation
- Useful for exploratory analysis of network data

  Ex: clues about social interactions, content-related web pages

Community detection and graph partitioning

- Community detection is a challenging clustering problem
  - C1) No consensus on the structural definition of community
  - C2) Node subset selection often intractable
  - C3) Lack of ground-truth for validation
- Useful for exploratory analysis of network data

  Ex: clues about social interactions, content-related web pages

Graph partitioning

Split V into given number of non-overlapping groups of given sizes

- **Criterion:** number of edges between groups is minimized (more soon)

- Community detection is a challenging clustering problem
  - C1) No consensus on the structural definition of community
  - C2) Node subset selection often intractable
  - C3) Lack of ground-truth for validation
- Useful for exploratory analysis of network data

  Ex: clues about social interactions, content-related web pages

Graph partitioning

Split V into given number of non-overlapping groups of given sizes

- **Criterion:** number of edges between groups is minimized (more soon)
- Number and sizes of groups unspecified in community detection

  $\Rightarrow$ Identify the natural fault lines along which a network separates

Graph partitioning is hard

- Ex: Graph bisection problem, i.e., partition V into two groups
  - Suppose the groups V_1 and V_2 are non-overlapping
  - Suppose groups have equal size, i.e., $|V_1| = |V_2| = N_v/2$
  - Minimize edges running between vertices in different groups

Graph partitioning is hard

- Ex: Graph bisection problem, i.e., partition V into two groups
  - Suppose the groups V_1 and V_2 are non-overlapping
  - Suppose groups have equal size, i.e., $|V_1| = |V_2| = N_v/2$
  - Minimize edges running between vertices in different groups
- Simple problem to describe, but hard to solve

$$\text{Number of ways to partition } V: \quad \binom{N_v}{N_v/2} \approx \frac{2^{N_v}}{\sqrt{N_v}}$$

$\Rightarrow$ Used Stirling's formula $N_v! \approx \sqrt{2\pi N_v}(N_v/e)^{N_v}$

$\Rightarrow$ Exhaustive search intractable beyond toy small-sized networks

Graph partitioning is hard

- Ex: Graph bisection problem, i.e., partition V into two groups
  - Suppose the groups V_1 and V_2 are non-overlapping
  - Suppose groups have equal size, i.e., $|V_1| = |V_2| = N_v/2$
  - Minimize edges running between vertices in different groups
- Simple problem to describe, but hard to solve

$$\text{Number of ways to partition } V: \quad \binom{N_v}{N_v/2} \approx \frac{2^{N_v}}{\sqrt{N_v}}$$

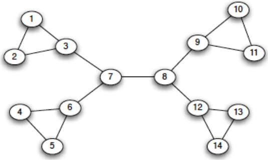
$\Rightarrow$ Used Stirling's formula $N_v! \approx \sqrt{2\pi N_v}(N_v/e)^{N_v}$

$\Rightarrow$ Exhaustive search intractable beyond toy small-sized networks

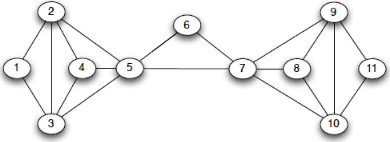
- No smart (i.e., polynomial time) algorithm, NP-hard problem

$\Rightarrow$ Seek good heuristics, e.g., relaxations of natural criteria

- Local bridges connect weakly interacting parts of the network



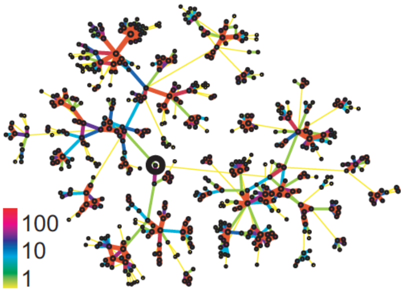
- Q: What about removing those to reveal communities?



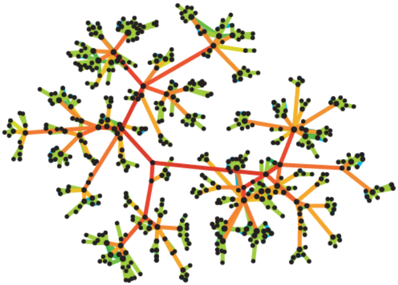
- **Challenges**
  - Multiple local bridges. Some better that others? Which one first?
  - There might be no local bridge, yet an apparent natural division

Edge betweenness centrality

- Idea: high edge betweenness centrality to identify weak ties
  - High $c_{Be}(e)$ edges carry large traffic volume over shortest paths
  - Position at the interface between tightly-knit groups
- Ex: cell-phone network with colored edge strength and betwenness



Edge strength



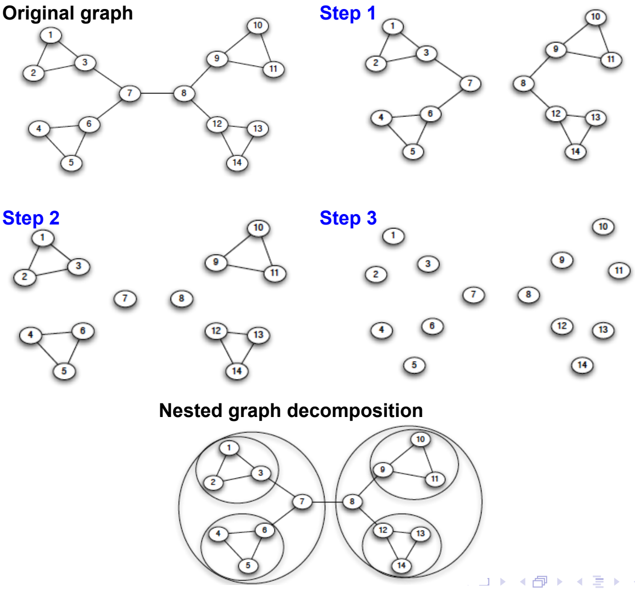
Edge betweenness

Girvan-Newman's method

- Girvan-Newmann's method extremely simple conceptually
  - $\Rightarrow$ Find and remove "spanning links" between cohesive subgroups
- **Algorithm:** Repeat until there are no edges left
  - $\Rightarrow$ Calculate the betweenness centrality $c_{Be}(e)$ of all edges
  - $\Rightarrow$ Remove edge(s) with highest $c_{Be}(e)$

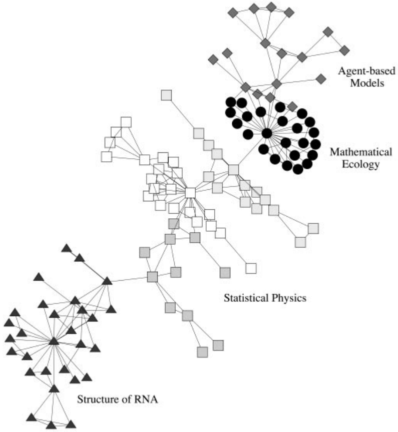
- Girvan-Newmann's method extremely simple conceptually
  - $\Rightarrow$ Find and remove "spanning links" between cohesive subgroups
- **Algorithm:** Repeat until there are no edges left
  - $\Rightarrow$ Calculate the betweenness centrality $c_{Be}(e)$ of all edges
  - $\Rightarrow$ Remove edge(s) with highest $c_{Be}(e)$
- Connected components are the communities identified
  - Divisive method: network falls apart into pieces as we go
  - Nested partition: larger communities potentially host denser groups
  - Recompute edge betweenness in $O(N_v N_e)$-time per step
- M. Girvan and M. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, pp. 7821-7826, 2002

Example: The algorithm in action



Scientific collaboration network

- Ex: Coauthorship network of scientists at the Santa Fe Institute



- Communities found can be traced to different disciplines

- Greedy approach to iteratively modify successive candidate partitions
  - Agglomerative: successive coarsening of partitions through merging
  - Divisive: successive refinement of partitions through splitting

- Greedy approach to iteratively modify successive candidate partitions
  - Agglomerative: successive coarsening of partitions through merging
  - Divisive: successive refinement of partitions through splitting
- Per step, partitions are modified in a way that minimizes a cost
  - Measures of (dis)similarity x_{ij} between pairs of vertices v_i and v_j
  - Ex: Euclidean distance dissimilarity

$$x_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}$$

- Greedy approach to iteratively modify successive candidate partitions
  - Agglomerative: successive coarsening of partitions through merging
  - Divisive: successive refinement of partitions through splitting
- Per step, partitions are modified in a way that minimizes a cost
  - Measures of (dis)similarity x_{ij} between pairs of vertices v_i and v_j
  - Ex: Euclidean distance dissimilarity

$$x_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}$$

- Method returns an entire hierarchy of nested partitions of the graph

$\Rightarrow$ Can range fully from $\{\{v_1\}, \ldots, \{v_{N_v}\}\}$ to V

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

  S4: Compute the dissimilarity between the new group and all others

Agglomerative clustering

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

  S4: Compute the dissimilarity between the new group and all others

  S5: Repeat from S3 until all vertices belong to a single group

Agglomerative clustering

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

  S4: Compute the dissimilarity between the new group and all others

  S5: Repeat from S3 until all vertices belong to a single group

- Need to define group dissimilarity from pairwise vertex counterparts

Agglomerative clustering

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

  S4: Compute the dissimilarity between the new group and all others

  S5: Repeat from S3 until all vertices belong to a single group

- Need to define group dissimilarity from pairwise vertex counterparts
  - Single linkage: group dissimilarity $x^{SL}_{G_i,G_j}$ follows single least dissimilar pair

$$x^{SL}_{G_i,G_j} = \min_{u \in G_i, v \in G_j} x_{uv}$$

- An agglomerative hierarchical clustering algorithm proceeds as follows

  S1: Choose a dissimilarity metric and compute it for all vertex pairs

  S2: Assign each vertex to a group of its own

  S3: Merge the pair of groups with smallest dissimilarity

  S4: Compute the dissimilarity between the new group and all others

  S5: Repeat from S3 until all vertices belong to a single group

- Need to define group dissimilarity from pairwise vertex counterparts
  - Single linkage: group dissimilarity $x^{SL}_{G_i,G_j}$ follows single least dissimilar pair

$$x^{SL}_{G_i,G_j} = \min_{u \in G_i, v \in G_j} x_{uv}$$

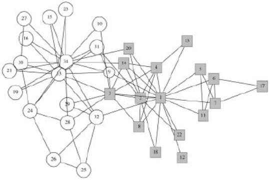
  - Complete linkage: every vertex pair highly similar to have small $x^{CL}_{G_i,G_j}$

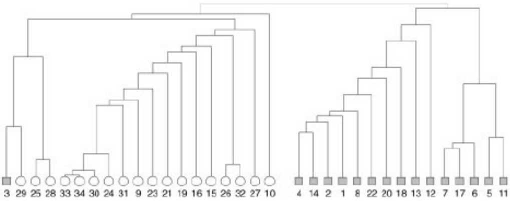
$$x^{CL}_{G_i,G_j} = \max_{u \in G_i, v \in G_j} x_{uv}$$

Dendrogram

- Hierarchical partitions often represented with a dendrogram
- Shows groups found in the network at all algorithmic steps

  ⇒ Split the network at different resolutions
- Ex: Girvan-Newman's algorithm for the Zachary's karate club

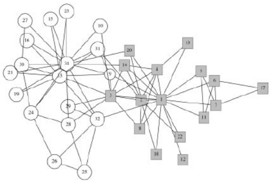


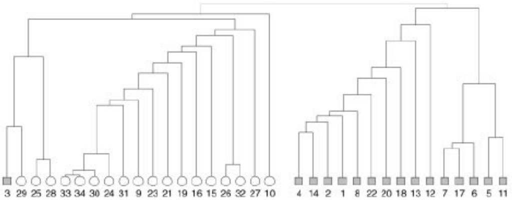


Dendrogram

- Hierarchical partitions often represented with a dendrogram
- Shows groups found in the network at all algorithmic steps

  $\Rightarrow$ Split the network at different resolutions
- Ex: Girvan-Newman's algorithm for the Zachary's karate club



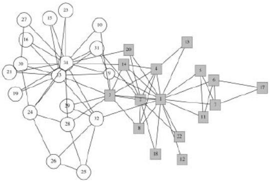


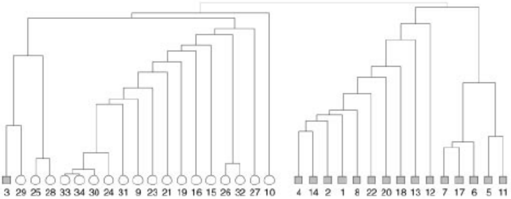
- Q: Which of the divisions is the most useful/optimal in some sense?

Dendrogram

- Hierarchical partitions often represented with a dendrogram
- Shows groups found in the network at all algorithmic steps

  ⇒ Split the network at different resolutions
- Ex: Girvan-Newman's algorithm for the Zachary's karate club





- Q: Which of the divisions is the most useful/optimal in some sense?
- A: Need to define metrics of graph clustering quality

Modularity

- Size of communities typically unknown $\Rightarrow$ Identify automatically

Modularity

- Size of communities typically unknown $\Rightarrow$ Identify automatically
- Modularity measures how well a network is partitioned in communities
  - Intuition: density of edges in communities higher than expected

Modularity

- Size of communities typically unknown $\Rightarrow$ Identify automatically
- Modularity measures how well a network is partitioned in communities
  - Intuition: density of edges in communities higher than expected
- Consider a graph G and a partition into groups $s \in S$. Modularity:

$$Q(G,S) \propto \sum_{s \in S} [(\#\text{ of edges within group } s) - \mathbb{E}[\#\text{ of such edges}]]$$

Modularity

- Size of communities typically unknown $\Rightarrow$ Identify automatically
- Modularity measures how well a network is partitioned in communities
  - Intuition: density of edges in communities higher than expected
- Consider a graph G and a partition into groups $s \in S$. Modularity:

$$Q(G,S) \propto \sum_{s \in S} [(\#\text{ of edges within group } s) - \mathbb{E}[\#\text{ of such edges}]]$$

- Formally, after normalization such that $Q(G,S) \in [-1,1]$

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

$\Rightarrow$ Null model: randomize edges, preserving degree distribution

Expected connectivity among nodes

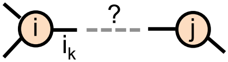
- Null model: randomize edges preserving degree distribution in G

  $\Rightarrow$ Random variable $A_{ij} := \mathbb{I}\{(i,j) \in E\}$

  $\Rightarrow$ Compute $\mathbb{E}[A_{ij}]$

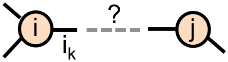
Expected connectivity among nodes

- Null model: randomize edges preserving degree distribution in G
  - $\Rightarrow$ Random variable $A_{ij} := \mathbb{I}\{(i,j) \in E\}$
  - $\Rightarrow$ Compute $\mathbb{E}[A_{ij}]$
- Suppose node i has degree d_i, node j has degree d_j
  - $\Rightarrow$ Degree is "# of spokes" per node, $2N_e$ spokes in G



Expected connectivity among nodes

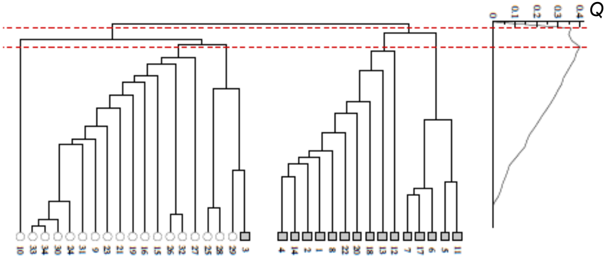
- Null model: randomize edges preserving degree distribution in G
  - $\Rightarrow$ Random variable $A_{ij} := \mathbb{I}\{(i,j) \in E\}$
  - $\Rightarrow$ Compute $\mathbb{E}[A_{ij}]$
- Suppose node i has degree d_i, node j has degree d_j
  - $\Rightarrow$ Degree is "# of spokes" per node, $2N_e$ spokes in G



- Probability spoke i_k connected to j is $\frac{d_j}{2N_e-1} \approx \frac{d_j}{2N_e}$, hence

$$\mathbb{E}[A_{ij}] = \sum_{i_k=1}^{d_i} \mathrm{P}\left[\text{spoke } i_k \text{ connected to } j\right] = \frac{d_i d_j}{2N_e}$$

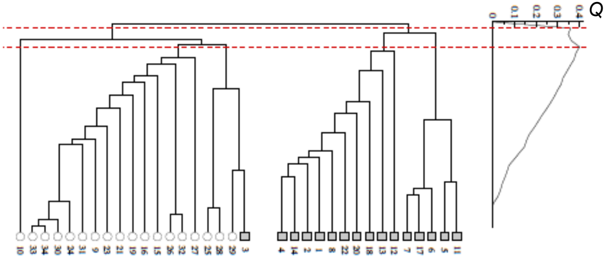
- Can evaluate the modularity of each partition in a dendrogram
  - $\Rightarrow$ Maximum value gives the "best" community structure
- Ex: Girvan-Newman's algorithm for the Zachary's karate club



- Can evaluate the modularity of each partition in a dendrogram

  $\Rightarrow$ Maximum value gives the "best" community structure

- Ex: Girvan-Newman's algorithm for the Zachary's karate club



- Q: Why not optimize $Q(G, S)$ directly over possible partitions S?

Modularity revisited

- Recall our definition of modularity

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

Modularity revisited

- Recall our definition of modularity

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

- Let g_i be the group membership of vertex i, and rewrite

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

Modularity revisited

- Recall our definition of modularity

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

- Let g_i be the group membership of vertex i, and rewrite

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

- Define for convenience the summands $B_{ij} := A_{ij} - \frac{d_i d_j}{2N_e}$

  $\Rightarrow$ Both marginal sums of B_{ij} vanish, since e.g.,

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

Graph bisection

- Consider (for simplicity) dividing the network in two groups
- Binary community membership variables per vertex

$$s_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to group 1} \\ -1, & \text{vertex } i \text{ belongs to group 2} \end{cases}$$

Graph bisection

- Consider (for simplicity) dividing the network in two groups
- Binary community membership variables per vertex

$$s_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to group 1} \\ -1, & \text{vertex } i \text{ belongs to group 2} \end{cases}$$

- Using the identity $\frac{1}{2}(s_i s_j + 1) = \mathbb{I}\{g_i = g_j\}$, the modularity is

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

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

Graph bisection

- Consider (for simplicity) dividing the network in two groups
- Binary community membership variables per vertex

$$s_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to group 1} \\ -1, & \text{vertex } i \text{ belongs to group 2} \end{cases}$$

- Using the identity $\frac{1}{2}(s_i s_j + 1) = \mathbb{I}\{g_i = g_j\}$, the modularity is

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

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

- Recall $\sum_j B_{ij} = 0$ to obtain the simpler expression

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

Optimizing modularity

- Let $\mathbf{B} \in \mathbb{R}^{N_v \times N_v}$ be the modularity matrix with entries $B_{ij} := A_{ij} - \frac{d_i d_j}{2N_e}$

  $\Rightarrow$ Any partition S is defined by the vector $\mathbf{s} = [s_1, \ldots, s_{N_v}]^\top$

Optimizing modularity

- Let $\mathbf{B} \in \mathbb{R}^{N_v \times N_v}$ be the modularity matrix with entries $B_{ij} := A_{ij} - \frac{d_i d_j}{2N_e}$

  $\Rightarrow$ Any partition S is defined by the vector $\mathbf{s} = [s_1, \ldots, s_{N_v}]^\top$

- Modularity is a quadratic form

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

Optimizing modularity

- Let $\mathbf{B} \in \mathbb{R}^{N_v \times N_v}$ be the modularity matrix with entries $B_{ij} := A_{ij} - \frac{d_i d_j}{2N_e}$

  $\Rightarrow$ Any partition S is defined by the vector $\mathbf{s} = [s_1, \ldots, s_{N_v}]^\top$

- Modularity is a quadratic form

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

- Modularity as criterion for graph bisection yields the formulation

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

  $\Rightarrow$ Nasty binary constraints $\mathbf{s} \in \{\pm 1\}^{N_v}$ (hypercube vertices)

  $\Rightarrow$ Modularity optimization is NP-hard [Brandes et al '06]

Just relax!

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \mathbf{s}^\top \mathbf{B} \mathbf{s}, \quad \text{s. to} \;\; \mathbf{s}^\top \mathbf{s} = N_v$$

Just relax!

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \mathbf{s}^\top \mathbf{B}\mathbf{s}, \quad \text{s. to} \;\; \mathbf{s}^\top \mathbf{s} = N_v$$

- Associate a Langrange multiplier λ to the constraint $\mathbf{s}^\top \mathbf{s} = N_v$

  $\Rightarrow$ Optimality conditions yields

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

- Conclusion is that $\mathbf{s}$ is an eigenvector of $\mathbf{B}$ with eigenvalue λ

Just relax!

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \mathbf{s}^\top \mathbf{B}\mathbf{s}, \quad \text{s. to} \;\; \mathbf{s}^\top \mathbf{s} = N_v$$

- Associate a Langrange multiplier λ to the constraint $\mathbf{s}^\top \mathbf{s} = N_v$
  $\Rightarrow$ Optimality conditions yields

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

- Conclusion is that $\mathbf{s}$ is an eigenvector of $\mathbf{B}$ with eigenvalue λ
- Q: Which eigenvector should we pick?
  $\Rightarrow$ At optimum $\mathbf{B}\mathbf{s} = \lambda \mathbf{s}$ so objective becomes

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

Just relax!

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \mathbf{s}^\top \mathbf{B}\mathbf{s}, \quad \text{s. to} \;\; \mathbf{s}^\top \mathbf{s} = N_v$$

- Associate a Langrange multiplier λ to the constraint $\mathbf{s}^\top \mathbf{s} = N_v$
  - $\Rightarrow$ Optimality conditions yields

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

- Conclusion is that $\mathbf{s}$ is an eigenvector of $\mathbf{B}$ with eigenvalue λ
- Q: Which eigenvector should we pick?
  - $\Rightarrow$ At optimum $\mathbf{B}\mathbf{s} = \lambda \mathbf{s}$ so objective becomes

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

- A: To maximize modularity pick the dominant eigenvector of $\mathbf{B}$

Spectral modularity maximization

- Let $\mathbf{u}_1$ be the dominant eigenvector of $\mathbf{B}$, with i-th entry $[\mathbf{u}_1]_i$

  $\Rightarrow$ Cannot just set $\mathbf{s} = \sqrt{N_v}\mathbf{u}_1$ because $\mathbf{u}_1 \neq \{\pm 1\}^{N_v}$

  $\Rightarrow$ Best effort: maximize their similarity $\mathbf{s}^\top \mathbf{u}_1$ which gives

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

Spectral modularity maximization

- Let $\mathbf{u}_1$ be the dominant eigenvector of $\mathbf{B}$, with i-th entry $[\mathbf{u}_1]_i$
  - $\Rightarrow$ Cannot just set $\mathbf{s} = \sqrt{N_v}\mathbf{u}_1$ because $\mathbf{u}_1 \neq \{\pm 1\}^{N_v}$
  - $\Rightarrow$ Best effort: maximize their similarity $\mathbf{s}^\top \mathbf{u}_1$ which gives

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

- Spectral modularity maximization algorithm

S1: Compute modularity matrix $\mathbf{B}$ with entries $B_{ij} = A_{ij} - \frac{d_i d_j}{2N_e}$

S2: Find dominant eigenvector $\mathbf{u}_1$ of $\mathbf{B}$ (e.g., power method)

S3: Cluster membership of vertex i is $s_i = \text{sign}([\mathbf{u}_1]_i)$

- Let $\mathbf{u}_1$ be the dominant eigenvector of $\mathbf{B}$, with i-th entry $[\mathbf{u}_1]_i$
  - $\Rightarrow$ Cannot just set $\mathbf{s} = \sqrt{N_v}\mathbf{u}_1$ because $\mathbf{u}_1 \neq \{\pm 1\}^{N_v}$
  - $\Rightarrow$ Best effort: maximize their similarity $\mathbf{s}^\top \mathbf{u}_1$ which gives

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

- Spectral modularity maximization algorithm

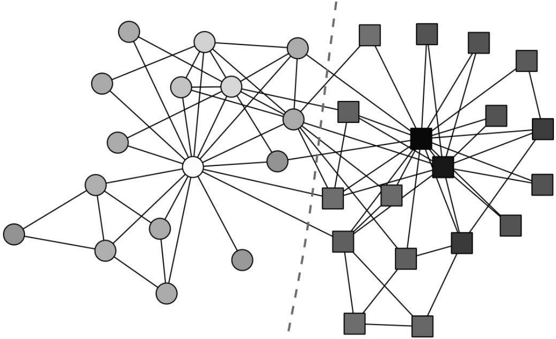
  S1: Compute modularity matrix $\mathbf{B}$ with entries $B_{ij} = A_{ij} - \frac{d_i d_j}{2N_e}$

  S2: Find dominant eigenvector $\mathbf{u}_1$ of $\mathbf{B}$ (e.g., power method)

  S3: Cluster membership of vertex i is $s_i = \text{sign}([\mathbf{u}_1]_i)$

- Multiple (> 2) communities through e.g., repeated graph bisection

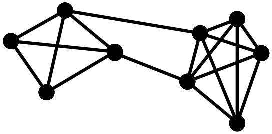
Example: Zachary's karate club

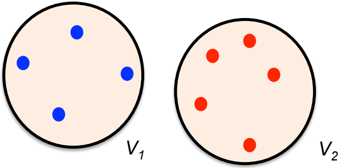


- Spectral modularity maximization
  - Shapes of vertices indicate community membership
  - Dotted line indicates partition found by the algorithm
  - Vertex colors indicate the strength of their membership

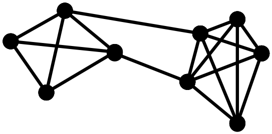
Graph bisection

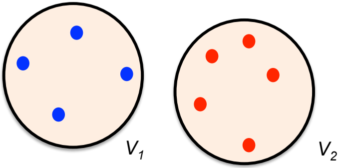
- Consider an undirected graph $G(V, E)$
- Ex: Graph bisection problem, i.e., partition V into two groups
  - Groups V_1 and $V_2 = V_1^C$ are non-overlapping
  - Groups have given size, i.e., $|V_1| = N_1$ and $|V_2| = N_2$





- Consider an undirected graph $G(V, E)$
- Ex: Graph bisection problem, i.e., partition V into two groups
  - Groups V_1 and $V_2 = V_1^C$ are non-overlapping
  - Groups have given size, i.e., $|V_1| = N_1$ and $|V_2| = N_2$



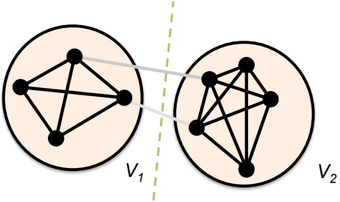


- Q: What is a good criterion to partition the graph?
- A: We have already seen modularity. Let's see a different one

Graph cut

- Desiderata: Community members should be
  - $\Rightarrow$ Well connected among themselves; and
  - $\Rightarrow$ Relatively well separated from the rest of the nodes

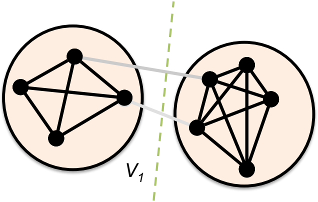
- Desiderata: Community members should be
  - $\Rightarrow$ Well connected among themselves; and
  - $\Rightarrow$ Relatively well separated from the rest of the nodes



- **Def:** A cut C is the number of edges between groups V_1 and $V \setminus V_1$

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

- Desiderata: Community members should be
  - $\Rightarrow$ Well connected among themselves; and
  - $\Rightarrow$ Relatively well separated from the rest of the nodes



- **Def:** A cut C is the number of edges between groups V_1 and $V \setminus V_1$

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

- Natural criterion: minimize cut, i.e., edges across groups V_1 and V_2

From graph cuts ...

- Binary community membership variables per vertex

$$s_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to } V_1 \\ -1, & \text{vertex } i \text{ belongs to } V_2 \end{cases}$$

From graph cuts . . .

- Binary community membership variables per vertex

$$s_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to } V_1 \\ -1, & \text{vertex } i \text{ belongs to } V_2 \end{cases}$$

- Let g_i be the group membership of vertex i, such that

$$\mathbb{I}\{g_i \neq g_j\} = \frac{1}{2}(1 - s_i s_j) = \begin{cases} 1, & i \text{ and } j \text{ in different groups} \\ 0, & i \text{ and } j \text{ in the same group} \end{cases}$$

From graph cuts ...

- Binary community membership variables per vertex

$$s_i = \begin{cases} +1, & \text{vertex } i \text{ belongs to } V_1 \\ -1, & \text{vertex } i \text{ belongs to } V_2 \end{cases}$$

- Let g_i be the group membership of vertex i, such that

$$\mathbb{I}\{g_i \neq g_j\} = \frac{1}{2}(1 - s_i s_j) = \begin{cases} 1, & i \text{ and } j \text{ in different groups} \\ 0, & i \text{ and } j \text{ in the same group} \end{cases}$$

- Cut expressible in terms of the variables s_i as

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

- First summand in $C = \frac{1}{2}\sum_{i,j} A_{ij}(1 - s_i s_j)$ is

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

- First summand in $C = \frac{1}{2}\sum_{i,j} A_{ij}(1 - s_i s_j)$ is

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

- Used $s_i^2 = 1$ since $s_i \in \{\pm 1\}$. The cut becomes

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

. . . to the graph Laplacian matrix

- First summand in $C = \frac{1}{2}\sum_{i,j} A_{ij}(1 - s_i s_j)$ is

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

- Used $s_i^2 = 1$ since $s_i \in \{\pm 1\}$. The cut becomes

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

- Cut in terms of L_{ij}, entries of the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, i.e.,

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

- First summand in $C = \frac{1}{2}\sum_{i,j} A_{ij}(1 - s_i s_j)$ is

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

- Used $s_i^2 = 1$ since $s_i \in \{\pm 1\}$. The cut becomes

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

- Cut in terms of L_{ij}, entries of the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, i.e.,

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

- Maximize modularity $Q(\mathbf{s}) \propto \mathbf{s}^\top \mathbf{B} \mathbf{s}$ vs. Minimze cut $C(\mathbf{s}) \propto \mathbf{s}^\top \mathbf{L} \mathbf{s}$

- Since $|V_1| = N_1$ and $|V_2| = N_2 = N - N_1$, we have the constraint

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

Graph cut minimization

- Since $|V_1| = N_1$ and $|V_2| = N_2 = N - N_1$, we have the constraint

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

- Minimum-cut criterion for graph bisection yields the formulation

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \{\pm 1\}^{N_v}} \mathbf{s}^\top \mathbf{L} \mathbf{s}, \quad \text{s. to } \mathbf{1}^\top \mathbf{s} = N_1 - N_2$$

- Again, binary constraints $\mathbf{s} \in \{\pm 1\}^{N_v}$ render cut minimization hard

  $\Rightarrow$ Relax binary constraints as with modularity maximization

Laplacian matrix properties revisited

- Smoothness: For any vector $\mathbf{x} \in \mathbb{R}^{N_v}$ of "vertex values", one has

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j \in V} L_{ij} x_i x_j = \sum_{(i,j) \in E} (x_i - x_j)^2$$

which can be minimized to enforce smoothness of functions on G

Laplacian matrix properties revisited

- Smoothness: For any vector $\mathbf{x} \in \mathbb{R}^{N_v}$ of "vertex values", one has

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j \in V} L_{ij} x_i x_j = \sum_{(i,j) \in E} (x_i - x_j)^2$$

  which can be minimized to enforce smoothness of functions on G

- Positive semi-definiteness: Follows since $\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^{N_v}$

Laplacian matrix properties revisited

- Smoothness: For any vector $\mathbf{x} \in \mathbb{R}^{N_v}$ of "vertex values", one has

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j \in V} L_{ij} x_i x_j = \sum_{(i,j) \in E} (x_i - x_j)^2$$

  which can be minimized to enforce smoothness of functions on G

- Positive semi-definiteness: Follows since $\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^{N_v}$
- Spectrum: All eigenvalues of $\mathbf{L}$ are real and non-negative

  $\Rightarrow$ Eigenvectors form an orthonormal basis of $\mathbb{R}^{N_v}$

Laplacian matrix properties revisited

- Smoothness: For any vector $\mathbf{x} \in \mathbb{R}^{N_v}$ of "vertex values", one has

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j \in V} L_{ij} x_i x_j = \sum_{(i,j) \in E} (x_i - x_j)^2$$

  which can be minimized to enforce smoothness of functions on G
- Positive semi-definiteness: Follows since $\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^{N_v}$
- Spectrum: All eigenvalues of $\mathbf{L}$ are real and non-negative

  $\Rightarrow$ Eigenvectors form an orthonormal basis of $\mathbb{R}^{N_v}$
- Rank deficiency: Since $\mathbf{L}\mathbf{1} = \mathbf{0}$, $\mathbf{L}$ is rank deficient

Laplacian matrix properties revisited

- Smoothness: For any vector $\mathbf{x} \in \mathbb{R}^{N_v}$ of "vertex values", one has

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j \in V} L_{ij} x_i x_j = \sum_{(i,j) \in E} (x_i - x_j)^2$$

  which can be minimized to enforce smoothness of functions on G

- Positive semi-definiteness: Follows since $\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^{N_v}$
- Spectrum: All eigenvalues of $\mathbf{L}$ are real and non-negative

  $\Rightarrow$ Eigenvectors form an orthonormal basis of $\mathbb{R}^{N_v}$

- Rank deficiency: Since $\mathbf{L}\mathbf{1} = \mathbf{0}$, $\mathbf{L}$ is rank deficient
- Spectrum and connectivity: The smallest eigenvalue λ_1 of $\mathbf{L}$ is 0
  - If the second-smallest eigenvalue $\lambda_2 \neq 0$, then G is connected
  - If $\mathbf{L}$ has n zero eigenvalues, G has n connected components

Further intuition

- Since $\mathbf{s}^\top \mathbf{L}\mathbf{s} = \sum_{(i,j)\in E}(s_i - s_j)^2$, the minimum-cut formulation is

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

- Since $\mathbf{s}^\top \mathbf{L}\mathbf{s} = \sum_{(i,j)\in E}(s_i - s_j)^2$, the minimum-cut formulation is

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

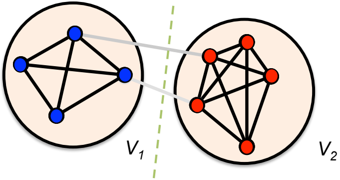
- Q: Does this equivalent cost function make sense?

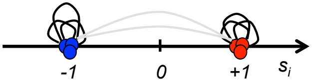
Further intuition

- Since $\mathbf{s}^\top \mathbf{L}\mathbf{s} = \sum_{(i,j)\in E}(s_i - s_j)^2$, the minimum-cut formulation is

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

- Q: Does this equivalent cost function make sense? A: Absolutely!
  - $\Rightarrow$ Edges joining vertices in the same group do not add to the sum
  - $\Rightarrow$ Edges joining vertices in different groups add 4 to the sum



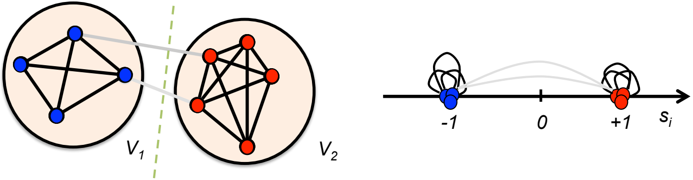


Further intuition

- Since $\mathbf{s}^\top \mathbf{L}\mathbf{s} = \sum_{(i,j)\in E}(s_i - s_j)^2$, the minimum-cut formulation is

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

- Q: Does this equivalent cost function make sense? A: Absolutely!
  - $\Rightarrow$ Edges joining vertices in the same group do not add to the sum
  - $\Rightarrow$ Edges joining vertices in different groups add 4 to the sum



- Minimize cut: assign values s_i to nodes i such that few edges cross 0

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \mathbf{s}^\top \mathbf{L}\mathbf{s}, \quad \text{s. to } \mathbf{1}^\top \mathbf{s} = N_1 - N_2 \text{ and } \mathbf{s}^\top \mathbf{s} = N_v$$

$\Rightarrow$ Straightforward to solve using Lagrange multipliers

Minimum-cut relaxation

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \mathbf{s}^\top \mathbf{L} \mathbf{s}, \quad \text{s. to } \mathbf{1}^\top \mathbf{s} = N_1 - N_2 \text{ and } \mathbf{s}^\top \mathbf{s} = N_v$$

$\Rightarrow$ Straightforward to solve using Lagrange multipliers

- Characterization of the solution $\hat{\mathbf{s}}$ [Fiedler '73]:

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

$\Rightarrow$ The 'second-smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$ satisfies $\mathbf{1}^\top \mathbf{v}_2 = 0$

$\Rightarrow$ Minimum cut is $C(\hat{\mathbf{s}}) = \hat{\mathbf{s}}^\top \mathbf{L} \hat{\mathbf{s}} = \mathbf{v}_2^\top \mathbf{L} \mathbf{v}_2 \propto \lambda_2$

Minimum-cut relaxation

- Relax the constraint $\mathbf{s} \in \{\pm 1\}^{N_v}$ to $\mathbf{s} \in \mathbb{R}^{N_v}$, $\|\mathbf{s}\|_2 = \sqrt{N_v}$

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \mathbf{s}^\top \mathbf{L}\mathbf{s}, \quad \text{s. to } \mathbf{1}^\top \mathbf{s} = N_1 - N_2 \text{ and } \mathbf{s}^\top \mathbf{s} = N_v$$

$\Rightarrow$ Straightforward to solve using Lagrange multipliers

- Characterization of the solution $\hat{\mathbf{s}}$ [Fiedler '73]:

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

$\Rightarrow$ The 'second-smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$ satisfies $\mathbf{1}^\top \mathbf{v}_2 = 0$

$\Rightarrow$ Minimum cut is $C(\hat{\mathbf{s}}) = \hat{\mathbf{s}}^\top \mathbf{L}\hat{\mathbf{s}} = \mathbf{v}_2^\top \mathbf{L}\mathbf{v}_2 \propto \lambda_2$

- If the graph G is disconnected then we know $\lambda_2 = 0 = C(\hat{\mathbf{s}})$

$\Rightarrow$ If G is amenable to bisection, the cut is small and so is λ_2

- Q: How to obtain the binary cluster labels $\mathbf{s} \in \{\pm 1\}^{N_v}$ from $\hat{\mathbf{s}} \in \mathbb{R}^{N_v}$?
  - $\Rightarrow$ Again, maximize the similarity measure $\mathbf{s}^\top \hat{\mathbf{s}}$

$$s_i = f(\mathbf{v}_2) := \begin{cases} +1, & [\mathbf{v}_2]_i \text{ among the } N_1 \text{ largest entries of } \mathbf{v}_2 \\ -1, & \text{otherwise} \end{cases}$$

- Q: How to obtain the binary cluster labels $\mathbf{s} \in \{\pm 1\}^{N_v}$ from $\hat{\mathbf{s}} \in \mathbb{R}^{N_v}$?

  $\Rightarrow$ Again, maximize the similarity measure $\mathbf{s}^\top \hat{\mathbf{s}}$

$$s_i = f(\mathbf{v}_2) := \begin{cases} +1, & [\mathbf{v}_2]_i \text{ among the } N_1 \text{ largest entries of } \mathbf{v}_2 \\ -1, & \text{otherwise} \end{cases}$$

- Spectral graph bisection algorithm

S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$

S2: Find 'second smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$

S3: Candidate membership of vertex i is $\bar{s}_i = f([\mathbf{v}_2])$ (or $\underline{s}_i = f([-\mathbf{v}_2])$)

S4: Among $\bar{\mathbf{s}}$ and $\underline{\mathbf{s}}$ pick the one that minimizes $C(\mathbf{s})$

Spectral graph partitioning

- Q: How to obtain the binary cluster labels $\mathbf{s} \in \{\pm 1\}^{N_v}$ from $\hat{\mathbf{s}} \in \mathbb{R}^{N_v}$?
  $\Rightarrow$ Again, maximize the similarity measure $\mathbf{s}^\top \hat{\mathbf{s}}$

$$s_i = f(\mathbf{v}_2) := \begin{cases} +1, & [\mathbf{v}_2]_i \text{ among the } N_1 \text{ largest entries of } \mathbf{v}_2 \\ -1, & \text{otherwise} \end{cases}$$

- Spectral graph bisection algorithm

  S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$

  S2: Find 'second smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$

  S3: Candidate membership of vertex i is $\bar{s}_i = f([\mathbf{v}_2])$ (or $\underline{s}_i = f([-\mathbf{v}_2])$)

  S4: Among $\bar{\mathbf{s}}$ and $\underline{\mathbf{s}}$ pick the one that minimizes $C(\mathbf{s})$

- Complexity: efficient Lanczos algorithm variant in $O(\frac{N_e}{\lambda_3 - \lambda_2})$ time
- Nomenclature: $\mathbf{v}_2$ is known as the Fiedler vector
  $\Rightarrow$ Eigenvalue λ_2 is Fiedler value, or algebraic connectivity of G

Spectral gap in Fiedler vector entries

- Suppose G is disconnected and has two connected components
  - $\mathbf{L}$ is block diagonal, two smallest eigenvectors indicate groups, i.e.,

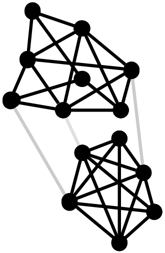
$$\mathbf{v}_1 = [1, 1, \ldots, 1, 0, \ldots, 0]^\top \text{ and } \mathbf{v}_2 = [0, 0, \ldots, 0, 1, \ldots, 1]^\top$$

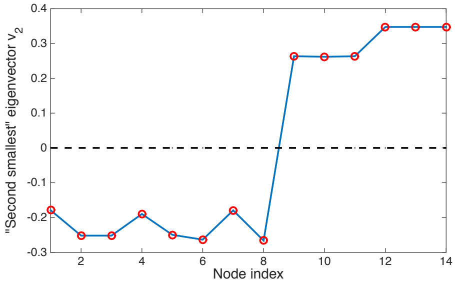
Spectral gap in Fiedler vector entries

- Suppose G is disconnected and has two connected components
  - $\mathbf{L}$ is block diagonal, two smallest eigenvectors indicate groups, i.e.,

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

- If G is connected but amenable to bisection, $\mathbf{v}_1 = \mathbf{1}$ and $\lambda_2 \approx 0$
  - Also, $\mathbf{1}^\top \mathbf{v}_2 = \sum_i [\mathbf{v}_2]_i = 0 \Rightarrow$ Positive and negative entries in $\mathbf{v}_2$





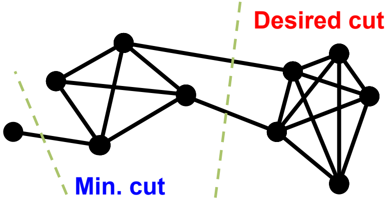
You are here

Wk.	Date	Topic	HW	Project
1	23-Aug	Introduction to course	HW0 out	
2	30-Aug	Graph theory	HW0 solutions posted	
3	6-Sep	LABOR DAY (no class)	HW1 out	
4	13-Sep	Centrality measures / Community detection		
5	20-Sep	Community detection		
6	27-Sep	Signal Processing and Deep learning for graphs	HW1 due	
7	4-Oct	Signal Processing and Deep learning for graphs	HW2 out	
8	11-Oct	FALL BREAK (no class)		
9	18-Oct	Network models	HW2 due	
10	25-Oct	Network models	HW3 out	Project proposal due
11	1-Nov	Epidemics		
12	8-Nov	Inference of network topologies, features, and processes	HW3 due	
13	15-Nov	Inference of network topologies, features, and processes		
14	22-Nov	Inference of network topologies, features, and processes		Project progress report
15	29-Nov	Inference of network topologies, features, and processes		
	13-Dec	Project presentation (video recording) and final report due		

Unknown community sizes

- Consider the graph bisection problem with unknown group sizes

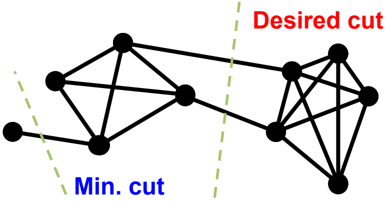
  $\Rightarrow$ Minimizing the graph cut may be no longer meaningfull!



  $\Rightarrow$ Cost $C := \sum_{i \in V_1, j \in V_2} A_{ij}$ agnostic to groups' internal structure

- Consider the graph bisection problem with unknown group sizes

  ⇒ Minimizing the graph cut may be no longer meaningful!



  ⇒ Cost $C := \sum_{i \in V_1, j \in V_2} A_{ij}$ agnostic to groups' internal structure

- Better criterion is the ratio cut R defined as

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

  ⇒ Balanced partitions: small community is penalized by the cost

Ratio-cut minimization

- Fix a bisection S of G into groups V_1 and V_2
- Define $\mathbf{f} : \mathbf{f}(S) = [f_1, \ldots, f_{N_v}]^\top \in \mathbb{R}^{N_v}$ with entries

$$f_i = \begin{cases} \sqrt{\frac{|V_2|}{|V_1|}}, & \text{vertex } i \text{ belongs to } V_1 \\ -\sqrt{\frac{|V_1|}{|V_2|}}, & \text{vertex } i \text{ belongs to } V_2 \end{cases}$$

Ratio-cut minimization

- Fix a bisection S of G into groups V_1 and V_2
- Define $\mathbf{f}$: $\mathbf{f}(S) = [f_1, \ldots, f_{N_v}]^\top \in \mathbb{R}^{N_v}$ with entries

$$f_i = \begin{cases} \sqrt{\frac{|V_2|}{|V_1|}}, & \text{vertex } i \text{ belongs to } V_1 \\ -\sqrt{\frac{|V_1|}{|V_2|}}, & \text{vertex } i \text{ belongs to } V_2 \end{cases}$$

- One can establish the following properties:

P1: $\mathbf{f}^\top \mathbf{L} \mathbf{f} = N_v R(S)$;
P2: $\sum_i f_i = 0$, i.e., $\mathbf{1}^\top \mathbf{f} = 0$; and
P3: $\|\mathbf{f}\|^2 = N_v$

Ratio-cut minimization

- Fix a bisection S of G into groups V_1 and V_2
- Define $\mathbf{f} : \mathbf{f}(S) = [f_1, \ldots, f_{N_v}]^\top \in \mathbb{R}^{N_v}$ with entries

$$f_i = \begin{cases} \sqrt{\frac{|V_2|}{|V_1|}}, & \text{vertex } i \text{ belongs to } V_1 \\ -\sqrt{\frac{|V_1|}{|V_2|}}, & \text{vertex } i \text{ belongs to } V_2 \end{cases}$$

- One can establish the following properties:

P1: $\mathbf{f}^\top \mathbf{L} \mathbf{f} = N_v R(S)$;
P2: $\sum_i f_i = 0$, i.e., $\mathbf{1}^\top \mathbf{f} = 0$; and
P3: $\|\mathbf{f}\|^2 = N_v$

- From P1-P3 it follows that ratio-cut minimization is equivalent to

$$\min_{\mathbf{f} \in \mathcal{F}} \mathbf{f}^\top \mathbf{L} \mathbf{f}, \quad \text{s. to } \mathbf{1}^\top \mathbf{f} = 0 \text{ and } \mathbf{f}^\top \mathbf{f} = N_v$$

Ratio cut and spectral graph bisection

- Ratio-cut minimization is also NP-hard. Relax to obtain

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathbb{R}^{N_v}} \mathbf{s}^\top \mathbf{L}\mathbf{s}, \quad \text{s. to } \mathbf{1}^\top \mathbf{s} = 0 \text{ and } \mathbf{s}^\top \mathbf{s} = N_v$$

- Partition $\hat{S}$ also given by the spectral graph bisection algorithm

S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$

S2: Find 'second smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$

S3: Cluster membership of vertex i is $s_i = \text{sign}([\mathbf{v}_2]_i)$

Ratio cut and spectral graph bisection

- Ratio-cut minimization is also NP-hard. Relax to obtain

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s} \in \mathbb{R}^{N_v}} \mathbf{s}^\top \mathbf{L} \mathbf{s}, \quad \text{s. to } \mathbf{1}^\top \mathbf{s} = 0 \text{ and } \mathbf{s}^\top \mathbf{s} = N_v$$

- Partition $\hat{S}$ also given by the spectral graph bisection algorithm

  S1: Compute Laplacian matrix $\mathbf{L}$ with entries $L_{ij} = D_{ij} - A_{ij}$

  S2: Find 'second smallest' eigenvector $\mathbf{v}_2$ of $\mathbf{L}$

  S3: Cluster membership of vertex i is $s_i = \text{sign}([\mathbf{v}_2]_i)$

- Alternative criterion is the normalized cut NC defined as

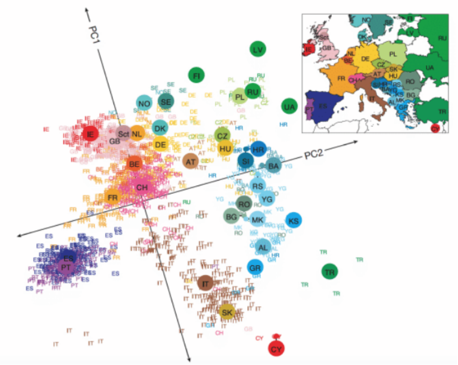
$$NC = \frac{C}{vol(V_1)} + \frac{C}{vol(V_2)}, \quad vol(V_i) := \sum_{v \in V_i} d_v, \; i = 1, 2$$

$\Rightarrow$ Corresponds to using the normalized Laplacian $\mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$

- What if we want to detect more than two groups?
- Opt. 1: Apply the above process iteratively
- Opt. 2: A generic spectral approach
  - $\Rightarrow$ Find eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ of $\mathbf{L}$
  - $\Rightarrow$ Assign $([\mathbf{v}_1]_i, \ldots, [\mathbf{v}_K]_i)$ to node $i \in V$
  - $\Rightarrow$ Use this embedding to apply k-means clustering

Example of more than two groups

- Genes mirror geography within Europe, Novembre et al., Nature (2008)
- Two-dimensional embedding of ‘gene similarity’ matrix
  - $\Rightarrow$ Consistent with origins of individuals in European map



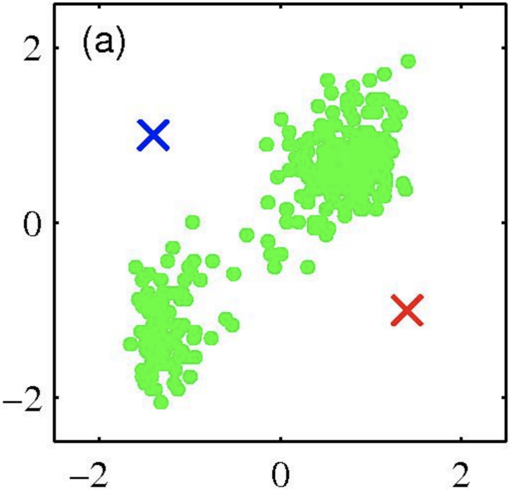
k-means clustering

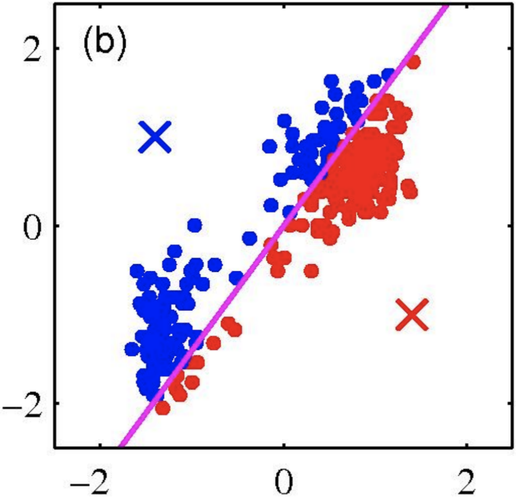
- An iterative clustering algorithm for a point cloud
- Initialize: Pick k random points as cluster centers
- Alternate:
  - $\Rightarrow$ 1. Assign data points to closest cluster center
  - $\Rightarrow$ 2. Change the cluster center to the average of its assigned points
- Finalize: When no assignments change
- Guaranteed to converge in a finite number of iterations

$$\min_{\{\mathbf{u}_i\}_{i=1}^k} \min_{\{C_i\}_{i=1}^k} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{u}_i\|_2^2$$

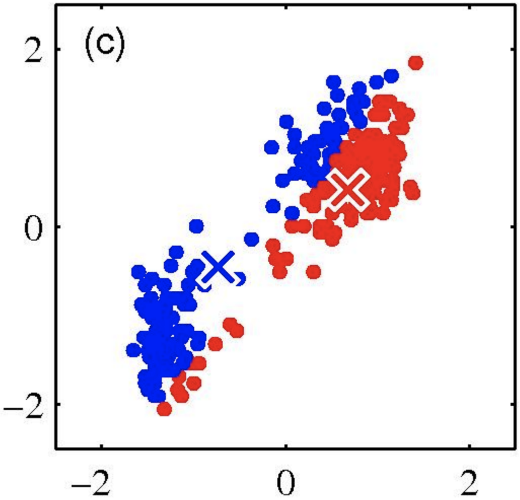
- Block coordinate descent in the above expression

k-means clustering: example

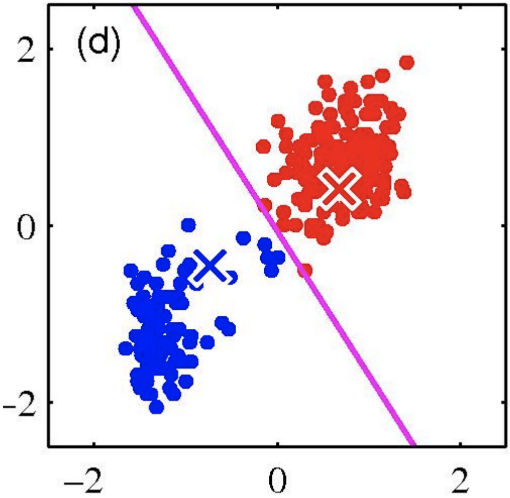



(b)
2
0
−2
−2
0
2

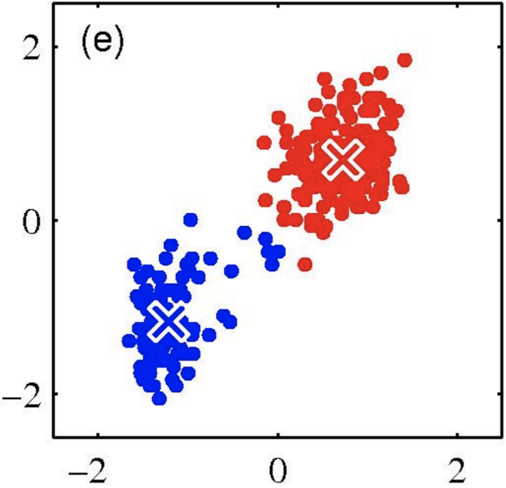
k-means clustering: example

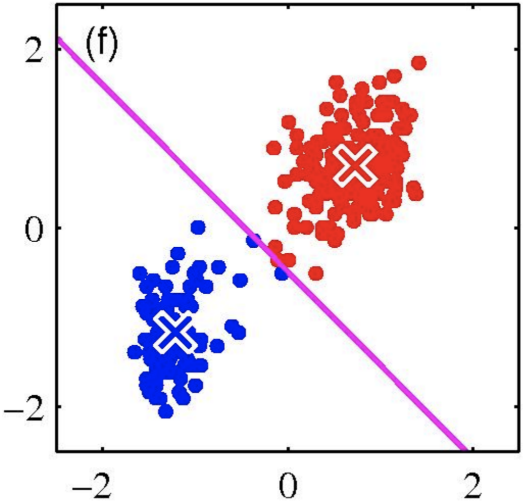


k-means clustering: example

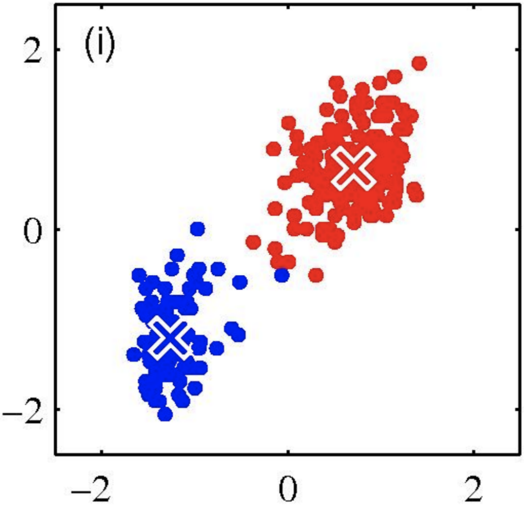


k-means clustering: example



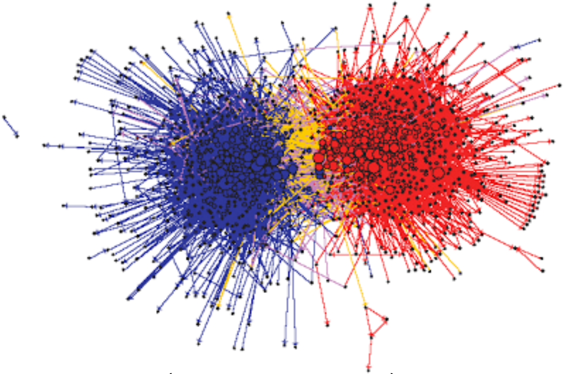

(f)
2
0
−2
−2
0
2

k-means clustering: example



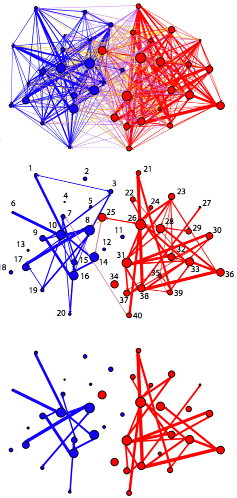
Network of the week

- The political blogosphere for the US 2004 presidential election



- Published in March 2005 (Adamic and Glance) $\Rightarrow$ over 3300 citations
- *Conservative blogs link to each other more frequently and in a denser pattern*
- *62% of online Americans still do not know what a weblog is*
- *Americans are turning to the Internet to stay informed about politics*

Network of the week



(A)

(B)

(C)

1 Digbys Blog
2 James Walcott
3 Pandagon
4 blog.johnkerry.com
5 Oliver Willis
6 America Blog
7 Crooked Timber
8 Daily Kos
9 American Prospect
10 Eschaton
11 Wonkette
12 Talk Left
13 Political Wire
14 Talking Points Memo
15 Matthew Yglesias
16 Washington Monthly
17 MyDD
18 Juan Cole
19 Left Coaster
20 Bradford DeLong

21 JawaReport
22 Voka Pundit
23 Roger L Simon
24 Tim Blair
25 Andrew Sullivan
26 Instapundit
27 Blogs for Bush
28 Little Green Footballs
29 Belmont Club
30 Captain's Quarters
31 Powerline
32 Hugh Hewitt
33 INDC Journal
34 Real Clear Politics
35 Winds of Change
36 Allahpundit
37 Michelle Malkin
38 WizBang
39 Dean's World
40 Volokh