Structure detection methods

Complex Networks | @networksvox
CSYS/MATH 303, Spring, 2016

Prof. Peter Dodds | @peterdodds

Dept. of Mathematics & Statistics | Vermont Complex Systems Center
Vermont Advanced Computing Core | University of Vermont

CocoNuTs
Complex Networks
@networksvox
Everything is connected

These slides are brought to you by:

Sealie & Lambie Productions

Outline

Overview

Methods

Hierarchy by aggregation

Hierarchy by division

Hierarchy by shuffling

Spectral methods

Hierarchies & Missing Links

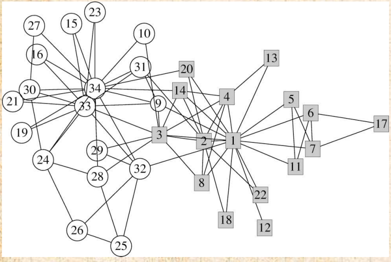
Overlapping communities

Link-based methods

General structure detection

References

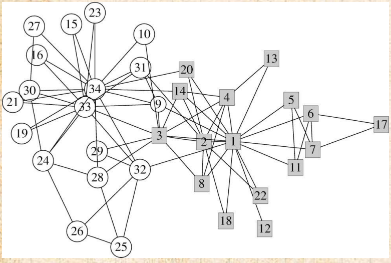
Structure detection



▲ Zachary's karate club [18, 12]

- **The issue:** how do we elucidate the internal structure of large networks across many scales?

Structure detection



▲ Zachary's karate club [18, 12]

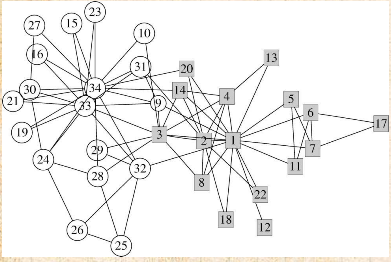
- The issue: how do we elucidate the internal structure of large networks across many scales?

- Possible substructures: hierarchies, cliques, rings, ...

- Plus: All combinations of substructures.

- Much focus on hierarchies...

Structure detection

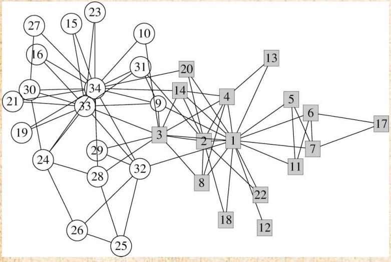


▲ Zachary's karate club [18, 12]

- The issue: how do we elucidate the internal structure of large networks across many scales?

- Possible substructures: hierarchies, cliques, rings, ...
- Plus: All combinations of substructures.
- Much focus on hierarchies...

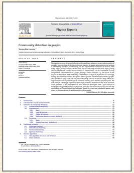
Structure detection



▲ Zachary's karate club [18, 12]

- **The issue:** how do we elucidate the internal structure of large networks across many scales?

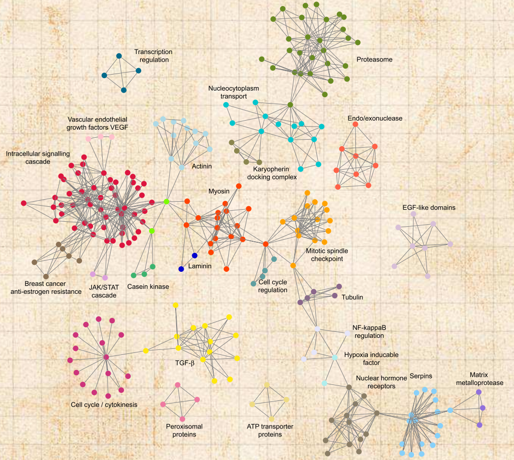
- Possible substructures: hierarchies, cliques, rings, ...
- Plus: All combinations of substructures.
- Much focus on hierarchies...



“Community detection in graphs”

Santo Fortunato,

Physics Reports, **486**, 75–174, 2010. [6]



Outline

Overview

Methods

- Hierarchy by aggregation
- Hierarchy by division
- Hierarchy by shuffling
- Spectral methods
- Hierarchies & Missing Links
- Overlapping communities
- Link-based methods
- General structure detection

References

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method ↗ [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [?]).

CocoNuTs
Complex Networks
@networksvox
Everything is connected

The UNIVERSITY of VERMONT

Hierarchy by aggregation—Bottom up:

- Idea: Extract hierarchical classification scheme for N objects by an agglomeration process.
- Need a measure of distance between all pairs of objects.
- Example: Ward's method [?]
- Procedure:
  1. Order pair-based distances.
  2. Sequentially add links between nodes based on closeness.
  3. Use additional criteria to determine when clusters are meaningful.
- Clusters gradually emerge, likely with clusters inside of clusters.
- Call above property Modularity.
- Works well for data sets where a distance between all objects can be specified (e.g., Aussie Rules [9]).

Hierarchy by aggregation

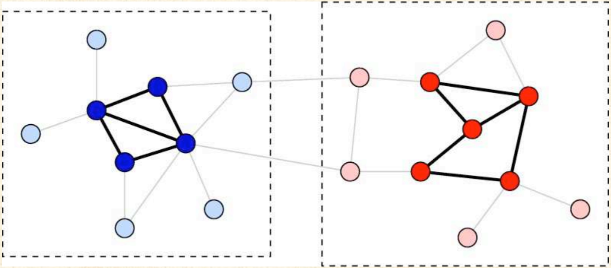
Bottom up problems:

- Tend to plainly not work on data sets representing networks with known modular structures.
- Good at finding cores of well-connected (or similar) nodes... but fail to cope well with peripheral, in-between nodes.

Hierarchy by aggregation

Bottom up problems:

- Tend to plainly not work on data sets representing networks with known modular structures.
- Good at finding cores of well-connected (or similar) nodes... but fail to cope well with peripheral, in-between nodes.



Outline

Overview

Methods

Hierarchy by aggregation

Hierarchy by division

Hierarchy by shuffling

Spectral methods

Hierarchies & Missing Links

Overlapping communities

Link-based methods

General structure detection

References

Hierarchy by division

Top down:

- Idea: Identify global structure first and recursively uncover more detailed structure.
- Basic objective: find dominant components that have significantly more links within than without, as compared to randomized version.
- We'll first work through "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[12]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[10, 11]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002).[7]

Hierarchy by division

Top down:

- Idea: Identify global structure first and recursively uncover more detailed structure.
- Basic objective: find dominant components that have significantly more links within than without, as compared to randomized version.
- We'll first work through "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[12]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[10, 11]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002).[7]

Hierarchy by division

Top down:

- Idea: Identify global structure first and recursively uncover more detailed structure.
- Basic objective: find dominant components that have significantly more links within than without, as compared to randomized version.
- We'll first work through "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[12]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[10, 11]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002).[7]

Hierarchy by division

Top down:

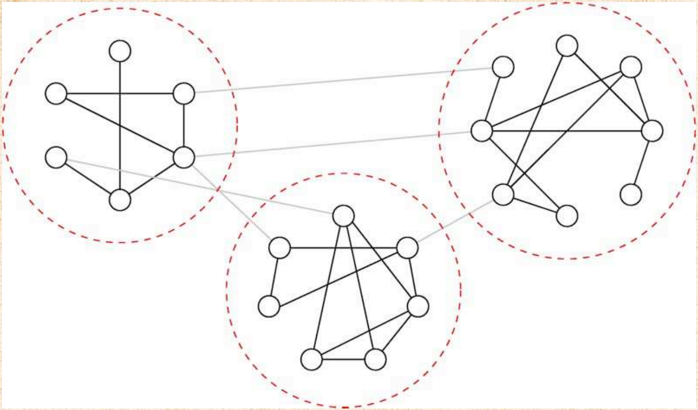
- Idea: Identify global structure first and recursively uncover more detailed structure.
- Basic objective: find dominant components that have significantly more links within than without, as compared to randomized version.
- We'll first work through "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004). [12]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001). [10, 11]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002). [7]

Hierarchy by division

Top down:

- Idea: Identify global structure first and recursively uncover more detailed structure.
- Basic objective: find dominant components that have significantly more links within than without, as compared to randomized version.
- We'll first work through "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004). [12]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001). [10, 11]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002). [7]

Hierarchy by division



Idea: Edges that connect communities have higher betweenness than edges within communities.

Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.

5 Record when components appear as a function of # edges removed.

6 Generate dendogram revealing hierarchical structure.

Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.

5. Record when components appear as a function of # edges removed.
6. Generate dendogram revealing hierarchical structure.

Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.
5. Record when components appear as a function of # edges removed.
6. Generate dendogram revealing hierarchical structure.

Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.
5. Record when components appear as a function of # edges removed.
6. Generate dendogram revealing hierarchical structure.

Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.

5 Record when components appear as a function of # edges removed.

6 Generate dendogram revealing hierarchical structure.

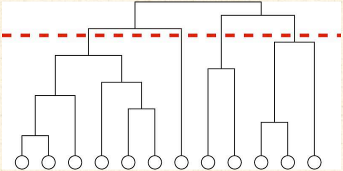
Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.

5 Record when components appear as a function of # edges removed.

6 Generate dendogram revealing hierarchical structure.



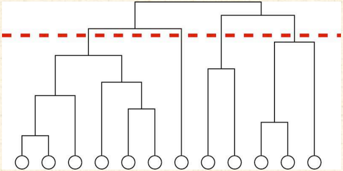
Red line indicates appearance of four (4) components at a certain level.

Hierarchy by division

One class of structure-detection algorithms:

1. Compute edge betweenness for whole network.
2. Remove edge with highest betweenness.
3. Recompute edge betweenness
4. Repeat steps 2 and 3 until all edges are removed.

5. Record when components appear as a function of # edges removed.
6. Generate dendogram revealing hierarchical structure.



Red line indicates appearance of four (4) components at a certain level.

Key element for division approach:

- Recomputing betweenness.
- Reason: Possible to have a low betweenness in links that connect large communities if other links carry majority of shortest paths.

When to stop?:

- How do we know which divisions are meaningful?
- Modularity measure: difference in fraction of within component nodes to that expected for randomized version:

  $Q = \sum_i [e_{ii} - a_i^2]$

  where e_{ij} is the fraction of (undirected) edges travelling between identified communities i and j, and $a_i = \sum_j e_{ij}$ is the fraction of edges with at least one end in community i. ▯

Key element for division approach:

- Recomputing betweenness.
- Reason: Possible to have a low betweenness in links that connect large communities if other links carry majority of shortest paths.

When to stop?:

- How do we know which divisions are meaningful?
- Modularity measure: difference in fraction of within component nodes to that expected for randomized version:

  $Q = \sum_i [e_{ii} - a_i^2]$

  where e_{ij} is the fraction of (undirected) edges travelling between identified communities i and j, and $a_i = \sum_j e_{ij}$ is the fraction of edges with at least one end in community i. □

Key element for division approach:

- Recomputing betweenness.
- Reason: Possible to have a low betweenness in links that connect large communities if other links carry majority of shortest paths.

When to stop?:

- How do we know which divisions are meaningful?
- Modularity measure: difference in fraction of within component nodes to that expected for randomized version:

$$Q = \sum_i [e_{ii} - a_i^2],$$

where e_{ij} is the fraction of (undirected) edges travelling between identified communities i and j, and $a_i = \sum_j e_{ij}$ is the fraction of edges with at least one end in community i. □

Key element for division approach:

- Recomputing betweenness.
- Reason: Possible to have a low betweenness in links that connect large communities if other links carry majority of shortest paths.

When to stop?:

- How do we know which divisions are meaningful?
- Modularity measure: difference in fraction of within component nodes to that expected for randomized version:

$$Q = \sum_i [e_{ii} - a_i^2],$$

where e_{ij} is the fraction of (undirected) edges travelling between identified communities i and j and $a_i = \sum_j e_{ij}$ is the fraction of edges with at least one end in community i. □

Key element for division approach:

- Recomputing betweenness.
- Reason: Possible to have a low betweenness in links that connect large communities if other links carry majority of shortest paths.

When to stop?:

- How do we know which divisions are meaningful?
- Modularity measure: difference in fraction of within component nodes to that expected for randomized version:

  $Q = \sum_i [e_{ii} - a_i^2]$

  where e_{ij} is the fraction of (undirected) edges travelling between identified communities i and j, and $a_i = \sum_j e_{ij}$ is the fraction of edges with at least one end in community i. □

Key element for division approach:

- Recomputing betweenness.
- Reason: Possible to have a low betweenness in links that connect large communities if other links carry majority of shortest paths.

When to stop?:

- How do we know which divisions are meaningful?
- Modularity measure: difference in fraction of within component nodes to that expected for randomized version:

  $Q = \sum_i [e_{ii} - a_i^2]$

  where e_{ij} is the fraction of (undirected) edges travelling between identified communities i and j, and $a_i = \sum_j e_{ij}$ is the fraction of edges with at least one end in community i. □

Measuring modularity:

Hierarchy by division

Test case:

- Generate random community-based networks.
- $N = 128$ with four communities of size 32.
- Add edges randomly within and across communities.
- Example:

  $\langle k \rangle_{\text{in}} = 6$ and $\langle k \rangle_{\text{out}} = 2$.

Hierarchy by division

Test case:

- Generate random community-based networks.
- $N = 128$ with four communities of size 32.
- Add edges randomly within and across communities.
- Example:

$\langle k \rangle_{\text{in}} = 6$ and $\langle k \rangle_{\text{out}} = 2$.

Hierarchy by division

Test case:

- Generate random community-based networks.
- $N = 128$ with four communities of size 32.
- Add edges randomly within and across communities.
- Example:

  $\langle k \rangle_{\text{in}} = 6$ and $\langle k \rangle_{\text{out}} = 2$.

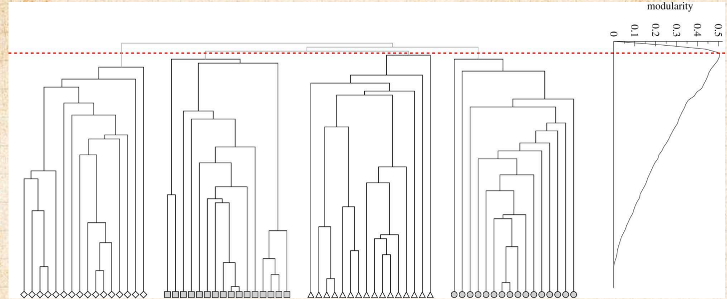
Hierarchy by division

Test case:

- Generate random community-based networks.
- $N = 128$ with four communities of size 32.
- Add edges randomly within and across communities.
- Example:

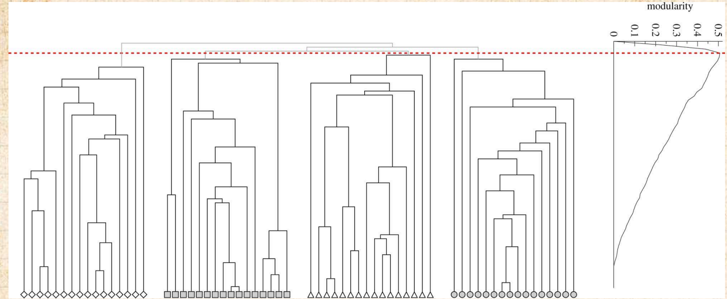
$$\langle k \rangle_{\text{in}} = 6 \text{ and } \langle k \rangle_{\text{out}} = 2.$$

Hierarchy by division



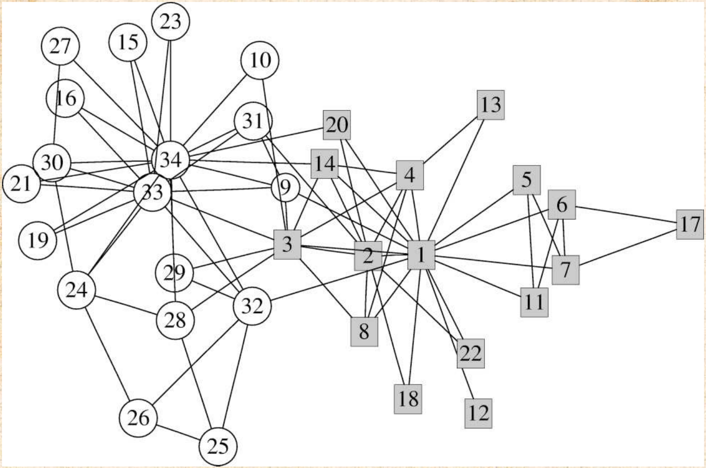
- Maximum modularity $Q \simeq 0.5$ obtained when four communities are uncovered.
- Further 'discovery' of internal structure is somewhat meaningless, as any communities arise accidentally.

Hierarchy by division



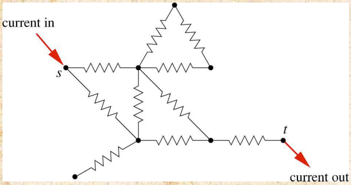
- Maximum modularity $Q \simeq 0.5$ obtained when four communities are uncovered.
- Further ‘discovery’ of internal structure is somewhat meaningless, as any communities arise accidentally.

Hierarchy by division



Factions in Zachary’s karate club network. [18]

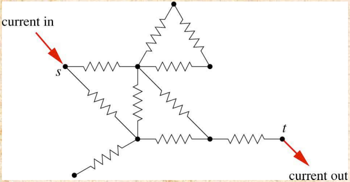
Betweenness for electrons:



- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random walk betweenness.)
- Contributing electronic betweenness for edge between nodes i and j:

$$B^{\text{elec}}_{ij,st} = a_{ij}|V_{i,st} - V_{j,st}|.$$

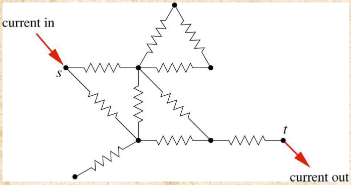
Betweenness for electrons:



- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random walk betweenness.)
- Contributing electronic betweenness for edge between nodes i and j:

$$B^{\text{elec}}_{ij,st} = a_{ij}|V_{i,st} - V_{j,st}|.$$

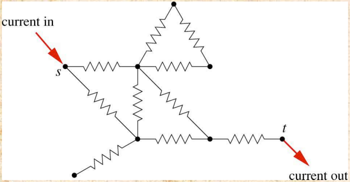
Betweenness for electrons:



- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random walk betweenness.)
- Contributing electronic betweenness for edge between nodes i and j:

$$B^{\text{elec}}_{ij,st} = a_{ij}|V_{i,st} - V_{j,st}|.$$

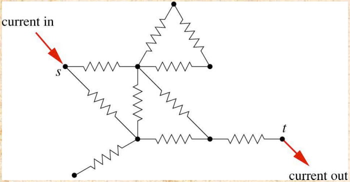
Betweenness for electrons:



- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random walk betweenness.)
- Contributing electronic betweenness for edge between nodes i and j:

$$B^{\text{elec}}_{ij,st} = a_{ij}|V_{i,st} - V_{j,st}|$$

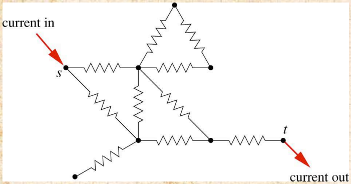
Betweenness for electrons:



- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random walk betweenness.)
- Contributing electronic betweenness for edge between nodes i and j:

$$B^{\text{elec}}_{ij,st} = a_{ij}|V_{i,st} - V_{j,st}|$$

Betweenness for electrons:



- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random walk betweenness.)
- Contributing electronic betweenness for edge between nodes i and j:

$$B^{\text{elec}}_{ij,st} = a_{ij}|V_{i,st} - V_{j,st}|.$$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
  $\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
  $= V_i k_i - \sum_j a_{ij} V_j = \sum_j [k_i \delta_{ij} V_j - a_{ij} V_j]$
  $= [(\mathbf{K} - \mathbf{A})\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff’s laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:

$\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij}V_j$

$= V_i k_i - \sum_j a_{ij}V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij}V_j\right]$

$= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
$\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
$= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j\right]$
$= \left[(K - A)\vec{V}\right]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
  $\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
  $= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j\right]$
  $= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
$\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
$= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j\right]$
$= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
  $\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
  $= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j\right]$
  $= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
  $\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
  $= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j\right]$
  $= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
  $\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
  $= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j \right]$
  $= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
  $\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij} V_j$
  $= V_i k_i - \sum_j a_{ij} V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij} V_j \right]$
  $= [(K - A)\vec{V}]_i$

Electronic betweenness

- Define some arbitrary voltage reference.
- Kirchhoff's laws: current flowing out of node i must balance:

$$\sum_{j=1}^{N} \frac{1}{R_{ij}}(V_j - V_i) = \delta_{is} - \delta_{it}.$$

- Between connected nodes, $R_{ij} = 1 = a_{ij} = 1/a_{ij}$.
- Between unconnected nodes, $R_{ij} = \infty = 1/a_{ij}$.
- We can therefore write:

$$\sum_{j=1}^{N} a_{ij}(V_i - V_j) = \delta_{is} - \delta_{it}.$$

- Some gentle jiggery-pokery on the left hand side:
$\sum_j a_{ij}(V_i - V_j) = V_i \sum_j a_{ij} - \sum_j a_{ij}V_j$
$= V_i k_i - \sum_j a_{ij}V_j = \sum_j \left[k_i \delta_{ij} V_j - a_{ij}V_j\right]$
$= [(K - A)\vec{V}]_i$

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I_{st}^{ext} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I_{st}^{\text{ext}}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\text{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\text{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\text{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\text{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\text{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\text{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\text{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Electronic betweenness

- Write right hand side as $[I^{\mathrm{ext}}]_{i,st} = \delta_{is} - \delta_{it}$, where I^{ext}_{st} holds external source and sink currents.
- Matrixingly then:

$$(K - A)\vec{V} = I^{\mathrm{ext}}_{st}.$$

- $L = K - A$ is a beast of some utility—known as the Laplacian.
- Solve for voltage vector $\vec{V}$ by LU decomposition (Gaussian elimination).
- Do not compute an inverse!
- Note: voltage offset is arbitrary so no unique solution.
- Presuming network has one component, null space of $K - A$ is one dimensional.
- In fact, $\mathcal{N}(K - A) = \{c\vec{1}, c \in R\}$ since $(K - A)\vec{1} = \vec{0}$.

Alternate betweenness measures:

Random walk betweenness:

- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Alternate betweenness measures:

Random walk betweenness:

- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Alternate betweenness measures:

Random walk betweenness:

- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Alternate betweenness measures:

Random walk betweenness:

- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Alternate betweenness measures:

Random walk betweenness:

- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Alternate betweenness measures:

Random walk betweenness:

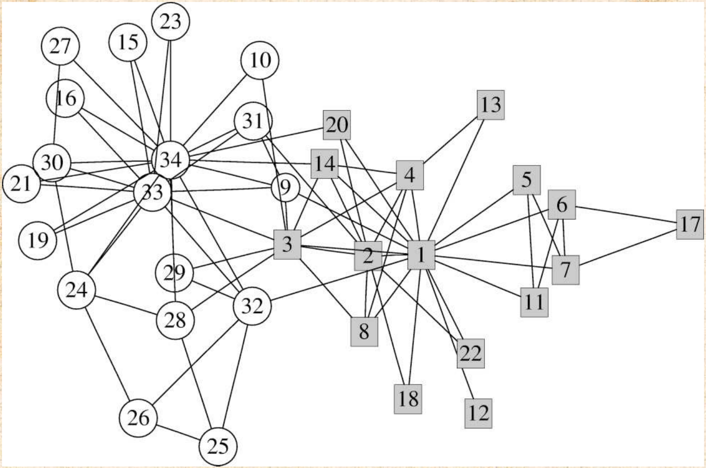
- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Alternate betweenness measures:

Random walk betweenness:

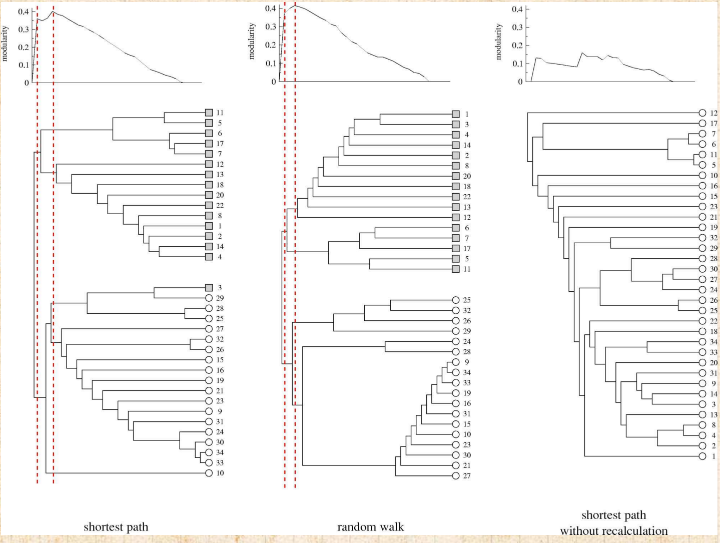
- Asking too much: Need full knowledge of network to travel along shortest paths.
- One of many alternatives: consider all random walks between pairs of nodes i and j.
- Walks starts at node i, traverses the network randomly, ending as soon as it reaches j.
- Record the number of times an edge is followed by a walk.
- Consider all pairs of nodes.
- Random walk betweenness of an edge = absolute difference in probability a random walk travels one way versus the other along the edge.
- Equivalent to electronic betweenness (see also diffusion).

Hierarchy by division



- Factions in Zachary's karate club network.[18]

Hierarchy by division



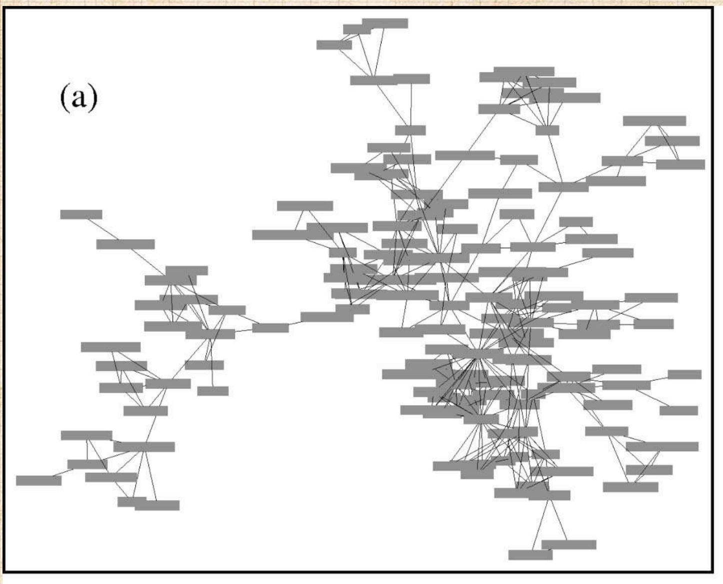
shortest path

random walk

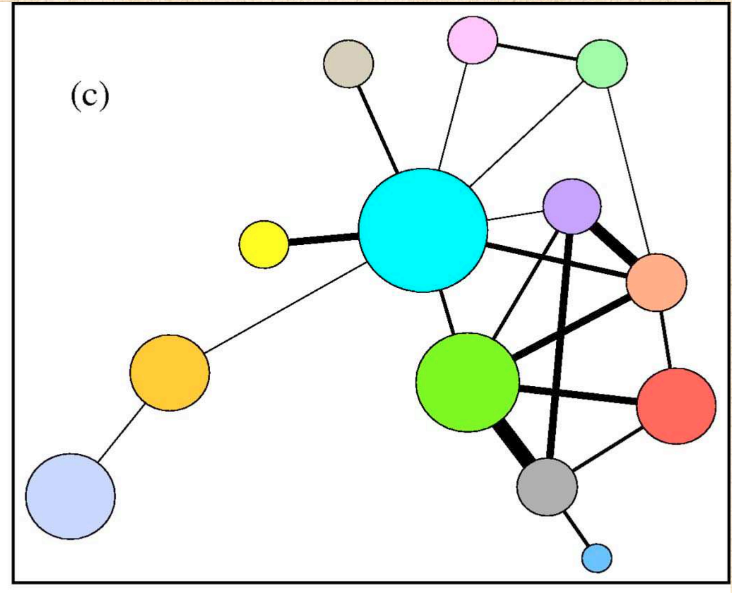
shortest path without recalculation

- Third column shows what happens if we don't recompute betweenness after each edge removal.

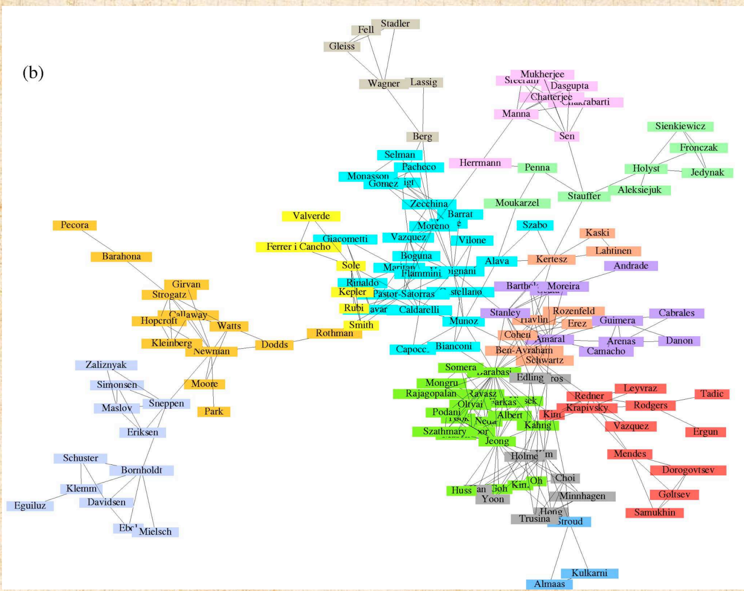
Scientists working on networks (2004)



Scientists working on networks (2004)

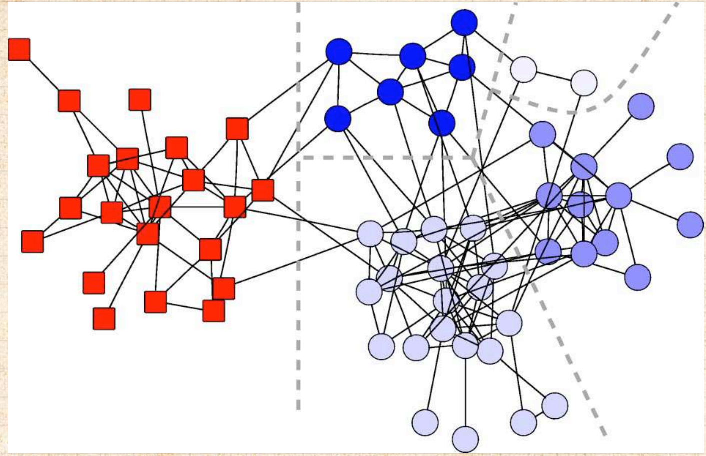


Scientists working on networks (2004)

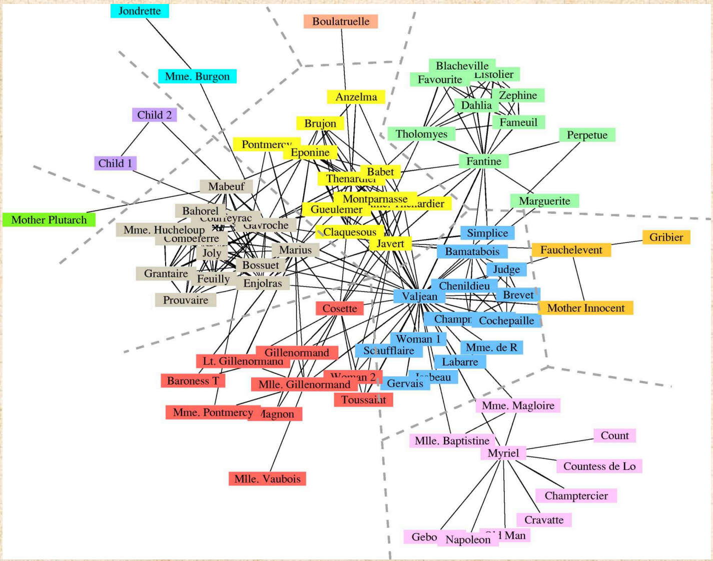


CocoNuTs
Complex Networks
@networksvox
Everything is connected

Dolphins!



Les Miserables



- More network analyses for Les Miserables here and here.

Outline

Overview

Methods

Hierarchy by aggregation

Hierarchy by division

Hierarchy by shuffling

Spectral methods

Hierarchies & Missing Links

Overlapping communities

Link-based methods

General structure detection

References

Shuffling for structure

- "Extracting the hierarchical organization of complex systems"
  Sales-Pardo *et al.*, PNAS (2007) [14, 15]
- Consider all partitions of networks into m groups
- As for Newman and Girvan approach, aim is to find partitions with maximum modularity:

$$Q = \sum_i [e_{ii} - (\sum_j e_{ij})^2] = \text{Tr}E - ||E^2||_1.$$

Shuffling for structure

- "Extracting the hierarchical organization of complex systems"
  Sales-Pardo *et al.*, PNAS (2007) [14, 15]
- Consider all partitions of networks into m groups
- As for Newman and Girvan approach, aim is to find partitions with maximum modularity:

$$Q = \sum_i [e_{ii} - (\sum_j e_{ij})^2] = \mathrm{Tr}E - ||E^2||_1$$

Shuffling for structure

- “Extracting the hierarchical organization of complex systems”
  Sales-Pardo *et al.*, PNAS (2007) [14, 15]
- Consider all partitions of networks into m groups
- As for Newman and Girvan approach, aim is to find partitions with maximum modularity:

$$Q = \sum_i [e_{ii} - (\sum_j e_{ij})^2] = \mathrm{Tr} E - ||E^2||_1 .$$

Shuffling for structure

- Consider partition network, i.e., the network of all possible partitions.
- Defn: Two partitions are connected if they differ only by the reassignment of a single node.
- Look for local maxima in partition network.
- Construct an affinity matrix with entries M^{aff}_{ij}.
- $M^{\mathrm{aff}}_{ij} = \mathbf{Pr}$ random walker on modularity network ends up at a partition with i and j in the same group.
- C.f. topological overlap between i and j = # matching neighbors for i and j divided by maximum of k_i and k_j.

Shuffling for structure

- Consider partition network, i.e., the network of all possible partitions.
- Defn: Two partitions are connected if they differ only by the reassignment of a single node.
- Look for local maxima in partition network.
- Construct an affinity matrix with entries M_{ij}^{aff}.
- $M_{ij}^{\text{aff}} = \mathbf{Pr}$ random walker on modularity network ends up at a partition with i and j in the same group.
- C.f. topological overlap between i and j = # matching neighbors for i and j divided by maximum of k_i and k_j.

Shuffling for structure

- Consider partition network, i.e., the network of all possible partitions.
- Defn: Two partitions are connected if they differ only by the reassignment of a single node.
- Look for local maxima in partition network.
- Construct an affinity matrix with entries M_{ij}^{aff}.
- $M_{ij}^{\text{aff}} = \mathbf{Pr}$ random walker on modularity network ends up at a partition with i and j in the same group.
- C.f. topological overlap between i and j = # matching neighbors for i and j divided by maximum of k_i and k_j.

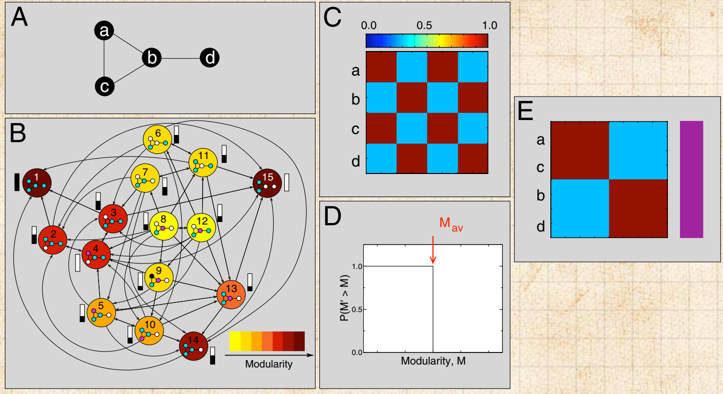
Shuffling for structure

- Consider partition network, i.e., the network of all possible partitions.
- Defn: Two partitions are connected if they differ only by the reassignment of a single node.
- Look for local maxima in partition network.
- Construct an affinity matrix with entries M_{ij}^{aff}.
- $M_{ij}^{\text{aff}} = \mathbf{Pr}$ random walker on modularity network ends up at a partition with i and j in the same group.
- C.f. topological overlap between i and j = # matching neighbors for i and j divided by maximum of k_i and k_j.

Shuffling for structure

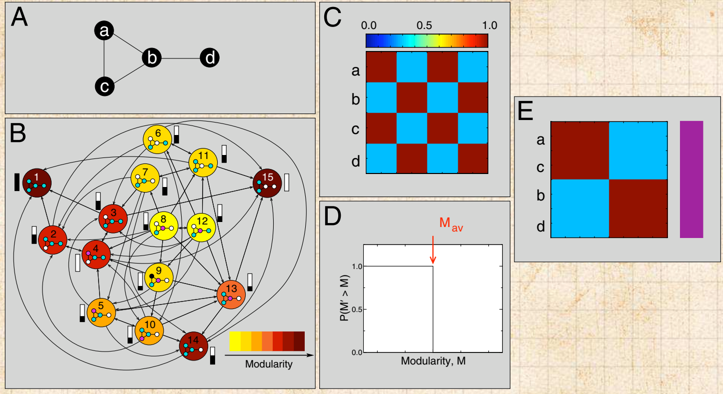
- Consider partition network, i.e., the network of all possible partitions.
- Defn: Two partitions are connected if they differ only by the reassignment of a single node.
- Look for local maxima in partition network.
- Construct an affinity matrix with entries M_{ij}^{aff}.
- M_{ij}^{aff} = $\mathbf{Pr}$ random walker on modularity network ends up at a partition with i and j in the same group.
- C.f. topological overlap between i and j = # matching neighbors for i and j divided by maximum of k_i and k_j.

Shuffling for structure



- **A:** Base network; **B:** Partition network; **C:** Coclassification matrix; **D:** Comparison to random networks (all the same!); **E:** Ordered coclassification matrix;

Shuffling for structure



- **A:** Base network; **B:** Partition network; **C:** Coclassification matrix; **D:** Comparison to random networks (all the same!); **E:** Ordered coclassification matrix; Conclusion: no structure...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\text{aff}}|i-j|.$$

- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\text{aff}} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M^{\text{aff}}_{ij}|i-j|.$$

- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M^{\text{aff}}_{ij} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\mathrm{aff}}|i-j|.$$

- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\mathrm{aff}} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M^{\text{aff}}_{ij}|i-j|.$$

- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M^{\text{aff}}_{ij} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\mathrm{aff}}|i-j|.$$

- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\mathrm{aff}} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} M_{ij}^{\text{aff}} |i - j|.$$

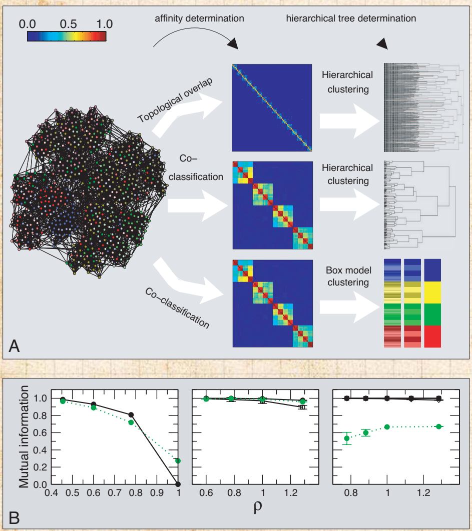
- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} M_{ij}^{\text{aff}} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

- Method obtains a distribution of classification hierarchies.
- Note: the hierarchy with the highest modularity score isn't chosen.
- Idea is to weight possible hierarchies according to their basin of attraction's size in the partition network.
- Next step: Given affinities, now need to sort nodes into modules, submodules, and so on.
- Idea: permute nodes to minimize following cost

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\textrm{aff}}|i-j|.$$

- Use simulated annealing (slow).
- Observation: should achieve same results for more general cost function: $C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}^{\textrm{aff}} f(|i-j|)$ where f is a strictly monotonically increasing function of 0, 1, 2, ...

Shuffling for structure



- $N = 640$,
- $\langle k \rangle = 16$,
- 3 tiered hierarchy.

Shuffling for structure

- Define cost matrix as T with entries $T_{ij} = f(|i-j|)$.
- Weird observation: if $T_{ij} = (i-j)^2$ then T is of rank 3, independent of N.
- Discovered by numerical inspection ...
- The eigenvalues are

$$\lambda_1 = -\frac{1}{6}n(n^2-1),$$

$$\lambda_2 = +\sqrt{nS_{n,4}} + S_{n,2}, \text{ and}$$

$$\lambda_3 = -\sqrt{nS_{n,4}} + S_{n,2}.$$

where

$$S_{n,2} = \frac{1}{12}n(n^2-1), \text{ and}$$

$$S_{n,4} = \frac{1}{240}n(n^2-1)(3n^2-7).$$

Shuffling for structure

- Define cost matrix as T with entries $T_{ij} = f(|i - j|)$.
- Weird observation: if $T_{ij} = (i - j)^2$ then T is of rank 3, independent of N.
- Discovered by numerical inspection ...
- The eigenvalues are

$$\lambda_1 = -\frac{1}{6}n(n^2-1),$$

$$\lambda_2 = +\sqrt{nS_{n,4} + S_{n,2}}, \text{ and}$$

$$\lambda_3 = -\sqrt{nS_{n,4} + S_{n,2}}.$$

where

$$S_{n,2} = \frac{1}{12}n(n^2-1), \text{ and}$$

$$S_{n,4} = \frac{1}{240}n(n^2-1)(3n^2-7).$$

Shuffling for structure

- Define cost matrix as T with entries $T_{ij} = f(|i-j|)$.
- Weird observation: if $T_{ij} = (i-j)^2$ then T is of rank 3, independent of N.
- Discovered by numerical inspection ...
- The eigenvalues are

$$\lambda_1 = -\frac{1}{6}n(n^2-1),$$

$$\lambda_2 = +\sqrt{nS_{n,4} + S_{n,2}}, \text{ and}$$

$$\lambda_3 = -\sqrt{nS_{n,4} + S_{n,2}}.$$

where

$$S_{n,2} = \frac{1}{12}n(n^2-1), \text{ and}$$

$$S_{n,4} = \frac{1}{240}n(n^2-1)(3n^2-7).$$

Shuffling for structure

- Define cost matrix as T with entries $T_{ij} = f(|i - j|)$.
- Weird observation: if $T_{ij} = (i - j)^2$ then T is of rank 3, independent of N.
- Discovered by numerical inspection ...
- The eigenvalues are

$$\lambda_1 = -\frac{1}{6}n(n^2 - 1),$$

$$\lambda_2 = +\sqrt{nS_{n,4}} + S_{n,2}, \text{ and}$$

$$\lambda_3 = -\sqrt{nS_{n,4}} + S_{n,2}.$$

where

$$S_{n,2} = \frac{1}{12}n(n^2 - 1), \text{ and}$$

$$S_{n,4} = \frac{1}{240}n(n^2 - 1)(3n^2 - 7).$$

Shuffling for structure

- Eigenvectors

$$(\vec{v}_1)_i = \left(i - \frac{n+1}{2}\right),$$

$$(\vec{v}_2)_i = \left(i - \frac{n+1}{2}\right)^2 + \sqrt{S_{n,4}/n}, \text{ and}$$

$$(\vec{v}_3)_i = \left(i - \frac{n+1}{2}\right)^2 - \sqrt{S_{n,4}/n}.$$

- Remarkably,

$$T = \lambda_1 \hat{v}_1 \hat{v}_1^{\mathsf{T}} + \lambda_2 \hat{v}_2 \hat{v}_2^{\mathsf{T}} + \lambda_3 \hat{v}_3 \hat{v}_3^{\mathsf{T}}.$$

- The next step: figure out how to capitalize on this...

Shuffling for structure

- Eigenvectors

$$
\begin{aligned}
(\vec{v}_1)_i &= \left(i - \frac{n+1}{2}\right), \\
(\vec{v}_2)_i &= \left(i - \frac{n+1}{2}\right)^2 + \sqrt{S_{n,4}/n}, \text{ and} \\
(\vec{v}_3)_i &= \left(i - \frac{n+1}{2}\right)^2 - \sqrt{S_{n,4}/n}.
\end{aligned}
$$

- Remarkably,

$$T = \lambda_1 \hat{v}_1 \hat{v}_1^{\mathsf{T}} + \lambda_2 \hat{v}_2 \hat{v}_2^{\mathsf{T}} + \lambda_3 \hat{v}_3 \hat{v}_3^{\mathsf{T}}.$$

- The next step: figure out how to capitalize on this...

Shuffling for structure

- Eigenvectors

$$
\begin{aligned}
(\vec{v}_1)_i &= \left(i - \frac{n+1}{2}\right), \\
(\vec{v}_2)_i &= \left(i - \frac{n+1}{2}\right)^2 + \sqrt{S_{n,4}/n}, \text{ and} \\
(\vec{v}_3)_i &= \left(i - \frac{n+1}{2}\right)^2 - \sqrt{S_{n,4}/n}.
\end{aligned}
$$

- Remarkably,

$$T = \lambda_1 \hat{v}_1 \hat{v}_1^{\mathsf{T}} + \lambda_2 \hat{v}_2 \hat{v}_2^{\mathsf{T}} + \lambda_3 \hat{v}_3 \hat{v}_3^{\mathsf{T}}.$$

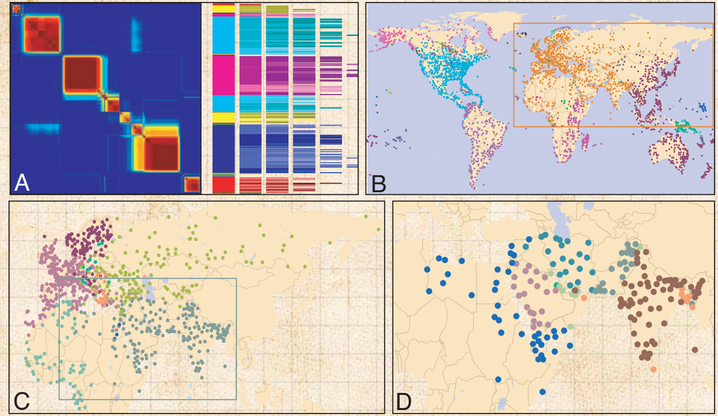
- The next step: figure out how to capitalize on this...

Shuffling for structure

Table 1. Top-level structure of real-world networks

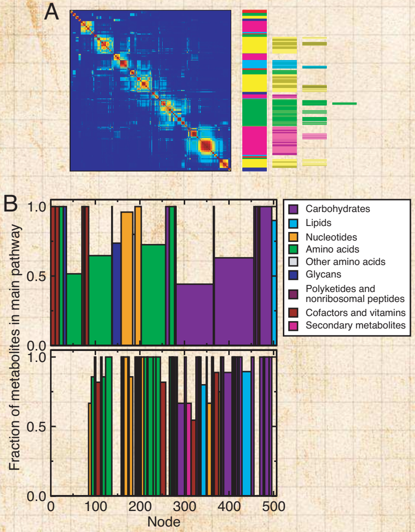
Network	Nodes	Edges	Modules	Main modules
Air transportation	3,618	28,284	57	8
E-mail	1,133	10,902	41	8
Electronic circuit	516	686	18	11
Escherichia coli KEGG	739	1,369	39	13
E. coli UCSD	507	947	28	17

Shuffling for structure



- Modules found match up with geopolitical units.

Shuffling for structure



- Modularity structure for metabolic network of E. coli (UCSD reconstruction).

Outline

Overview

Methods

- Hierarchy by aggregation
- Hierarchy by division
- Hierarchy by shuffling
- Spectral methods
- Hierarchies & Missing Links
- Overlapping communities
- Link-based methods
- General structure detection

References

General structure detection

- “Detecting communities in large networks” Capocci *et al.* (2005) [4]
- Consider normal matrix $K^{-1}A$, random walk matrix $A^{\mathsf{T}}K^{-1}$, Laplacian $K-A$, and AA^{T}.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Builds on Kleinberg's HITS algorithm.

General structure detection

- “Detecting communities in large networks” Capocci *et al.* (2005) [4]
- Consider normal matrix $K^{-1}A$, random walk matrix $A^{\mathsf{T}}K^{-1}$, Laplacian $K - A$, and AA^{T}.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Builds on Kleinberg's HITS algorithm.

General structure detection

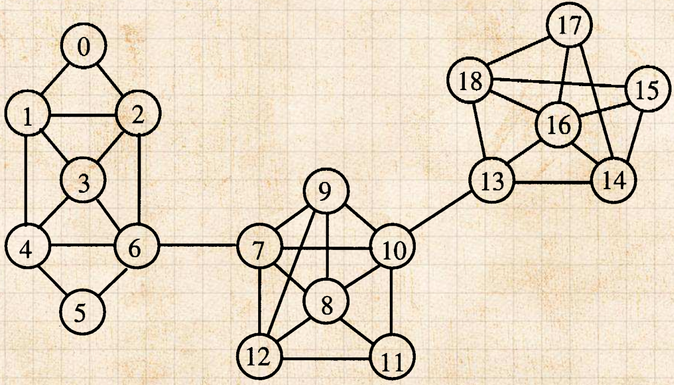
- "Detecting communities in large networks" Capocci *et al.* (2005) [4]
- Consider normal matrix $K^{-1}A$, random walk matrix $A^{\mathsf{T}}K^{-1}$, Laplacian $K - A$, and AA^{T}.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Builds on Kleinberg's HITS algorithm.

General structure detection

- "Detecting communities in large networks" Capocci *et al.* (2005) [4]
- Consider normal matrix $K^{-1}A$, random walk matrix $A^{\mathsf{T}}K^{-1}$, Laplacian $K - A$, and AA^{T}.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Builds on Kleinberg's HITS algorithm.

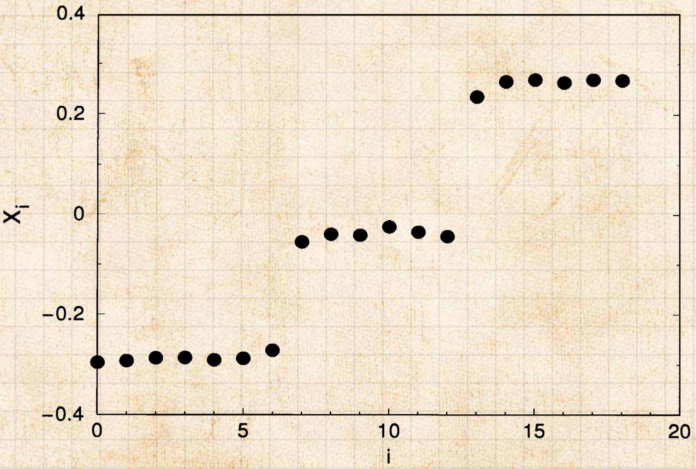
General structure detection

- Example network:



General structure detection

- Second eigenvector's components:



General structure detection

- Network of word associations for 10616 words.
- Average in-degree of 7.
- Using 2nd to 11th evectors of a modified version of AA^{T}:

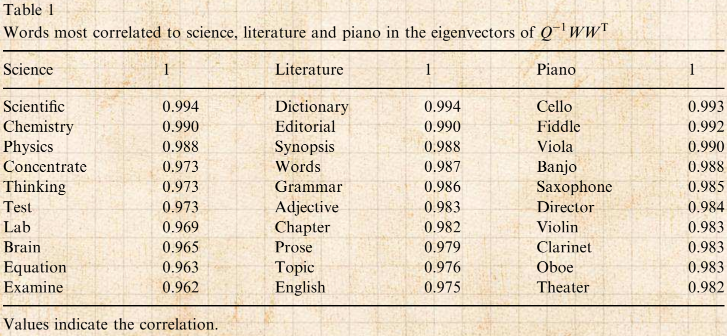


Table 1

Words most correlated to science, literature and piano in the eigenvectors of $Q^{-1}WW^{\mathrm{T}}$

Science	1	Literature	1	Piano	1
Scientific	0.994	Dictionary	0.994	Cello	0.993
Chemistry	0.990	Editorial	0.990	Fiddle	0.992
Physics	0.988	Synopsis	0.988	Viola	0.990
Concentrate	0.973	Words	0.987	Banjo	0.988
Thinking	0.973	Grammar	0.986	Saxophone	0.985
Test	0.973	Adjective	0.983	Director	0.984
Lab	0.969	Chapter	0.982	Violin	0.983
Brain	0.965	Prose	0.979	Clarinet	0.983
Equation	0.963	Topic	0.976	Oboe	0.983
Examine	0.962	English	0.975	Theater	0.982

Values indicate the correlation.

Outline

Overview

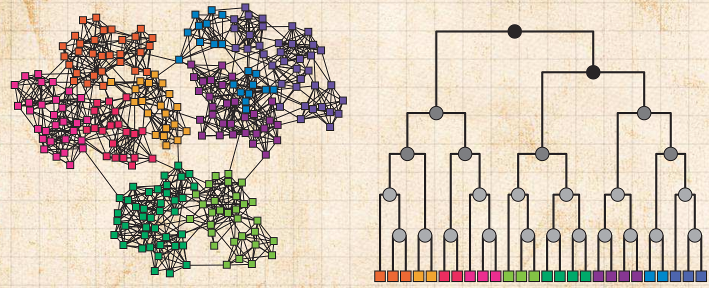
Methods

- Hierarchy by aggregation
- Hierarchy by division
- Hierarchy by shuffling
- Spectral methods
- Hierarchies & Missing Links
- Overlapping communities
- Link-based methods
- General structure detection

References

Hierarchies and missing links

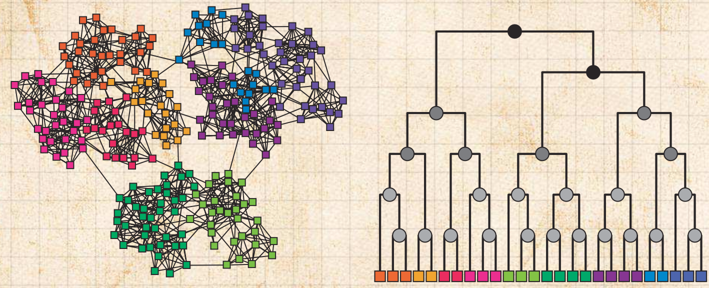
Clauset *et al.*, Nature (2008) [5]



- Idea: Shades indicate probability that nodes in left and right subtrees of dendogram are connected.
- Handle: Hierarchical random graph models.
- Plan: Infer consensus dendogram for a given real network.
- Obtain probability that links are missing (big problem...).

Hierarchies and missing links

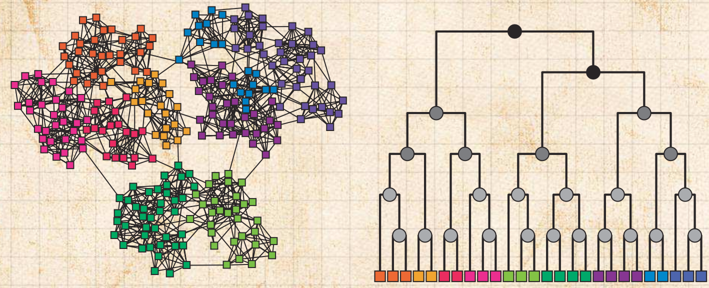
Clauset *et al.*, Nature (2008) [5]



- Idea: Shades indicate probability that nodes in left and right subtrees of dendogram are connected.
- Handle: Hierarchical random graph models.
- Plan: Infer consensus dendogram for a given real network.
- Obtain probability that links are missing (big problem...).

Hierarchies and missing links

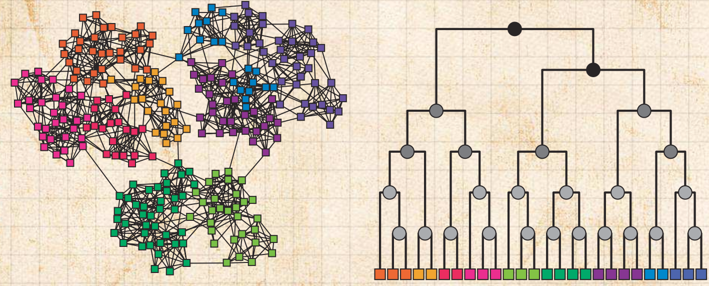
Clauset *et al.*, Nature (2008) [5]



- Idea: Shades indicate probability that nodes in left and right subtrees of dendogram are connected.
- Handle: Hierarchical random graph models.
- Plan: Infer consensus dendogram for a given real network.
- Obtain probability that links are missing (big problem...).

Hierarchies and missing links

Clauset *et al.*, Nature (2008) [5]



- Idea: Shades indicate probability that nodes in left and right subtrees of dendogram are connected.
- Handle: Hierarchical random graph models.
- Plan: Infer consensus dendogram for a given real network.
- Obtain probability that links are missing (big problem...).

Hierarchies and missing links

- Model also predicts reasonably well
  1. average degree,
  2. clustering,
  3. and average shortest path length.

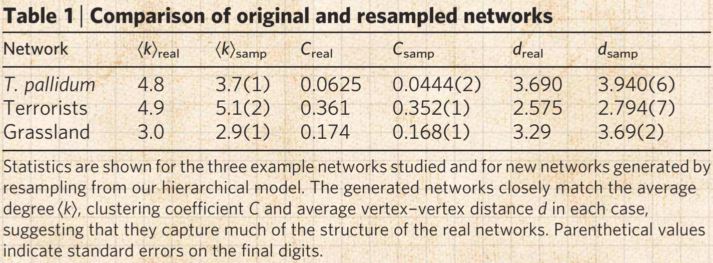
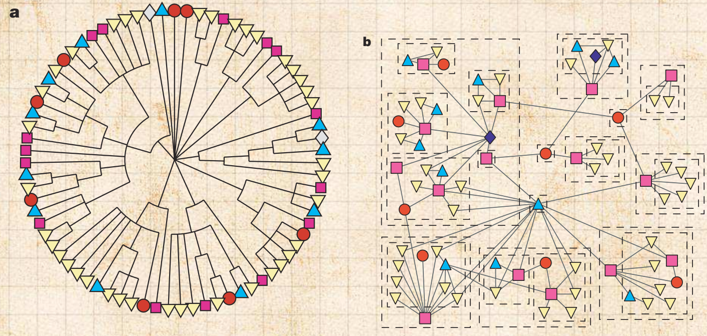


Table 1 | Comparison of original and resampled networks

Network	$\langle k\rangle_{\text{real}}$	$\langle k\rangle_{\text{samp}}$	C_{real}	C_{samp}	d_{real}	d_{samp}
T. pallidum	4.8	3.7(1)	0.0625	0.0444(2)	3.690	3.940(6)
Terrorists	4.9	5.1(2)	0.361	0.352(1)	2.575	2.794(7)
Grassland	3.0	2.9(1)	0.174	0.168(1)	3.29	3.69(2)

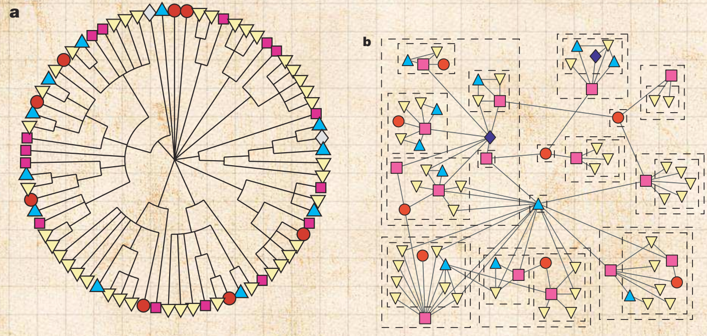
Statistics are shown for the three example networks studied and for new networks generated by resampling from our hierarchical model. The generated networks closely match the average degree $\langle k\rangle$, clustering coefficient C and average vertex–vertex distance d in each case, suggesting that they capture much of the structure of the real networks. Parenthetical values indicate standard errors on the final digits.

Hierarchies and missing links



- Consensus dendogram for grassland species.
- Copes with disassortative and assortative communities.

Hierarchies and missing links



- Consensus dendogram for grassland species.
- Copes with disassortative and assortative communities.

Outline

Overview

Methods

- Hierarchy by aggregation
- Hierarchy by division
- Hierarchy by shuffling
- Spectral methods
- Hierarchies & Missing Links
- Overlapping communities
- Link-based methods
- General structure detection

References

From PoCS: Small-worldness and social searchability

Social networks and identity:

From PoCS:
Small-worldness and social searchability

Social networks and identity:

Identity is formed from attributes such as:

- Geographic location
- Type of employment
- Religious beliefs
- Recreational activities.

From PoCS:
Small-worldness and social searchability

Social networks and identity:

Identity is formed from attributes such as:

- Geographic location
- Type of employment
- Religious beliefs
- Recreational activities.

Groups are formed by people with at least one similar attribute.

From PoCS:
Small-worldness and social searchability

Social networks and identity:

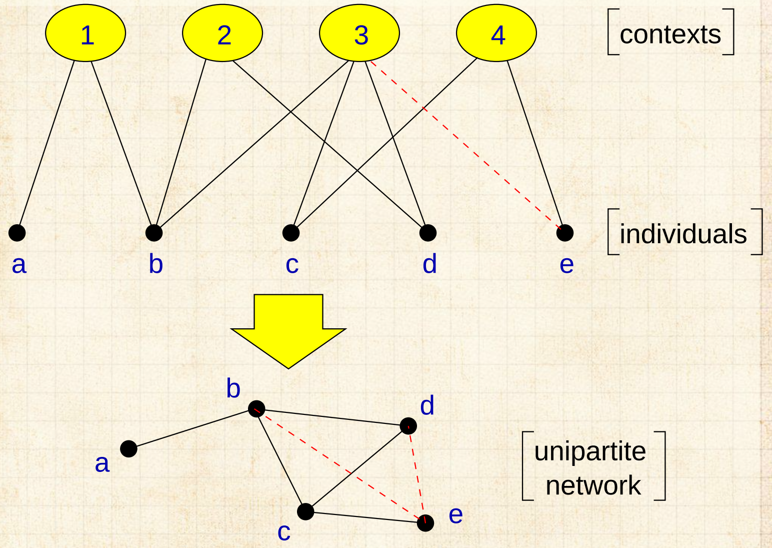
Identity is formed from attributes such as:

- Geographic location
- Type of employment
- Religious beliefs
- Recreational activities.

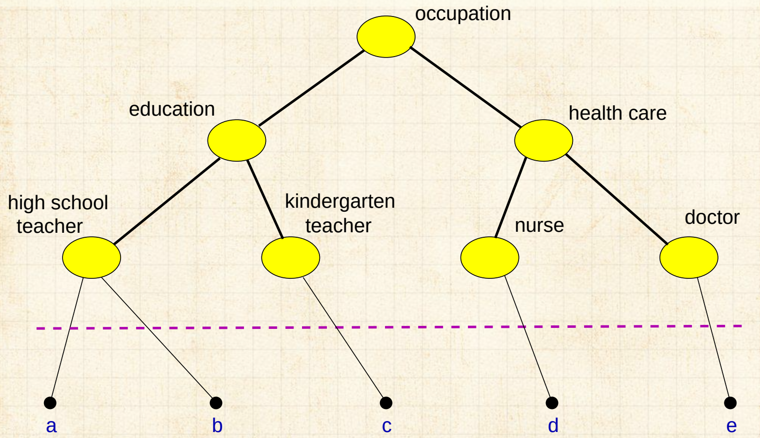
Groups are formed by people with at least one similar attribute.

Attributes ⇔ Contexts ⇔ Interactions ⇔ Networks.

Social distance—Bipartite affiliation networks

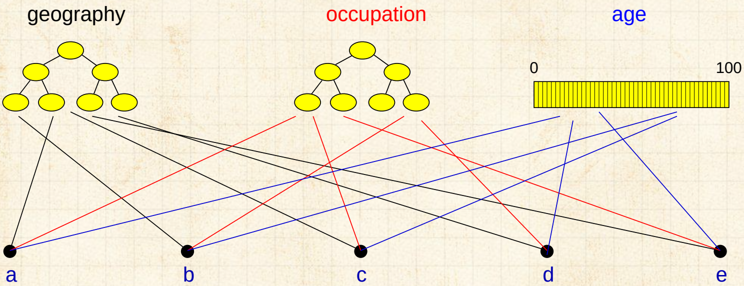


Social distance—Context distance



Models

Generalized affiliation networks



- Blau & Schwartz [2], Simmel [16], Breiger [3], Watts *et al.* [17]; see also Google+ Circles.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m: number of a communities a node belongs to.
  - $s^{\text{ov}}_{\alpha,\beta}$: number of nodes shared between two given communities, α and β.
  - d^{com}_{α}: degree of community α.
  - s^{com}_{α}: community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_{>}(m)$, $P_{>}(s^{\text{ov}}_{\alpha,\beta})$, $P_{>}(d^{\text{com}}_{\alpha})$, and $P_{>}(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to;
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β;
  - d^{com}_{α}, degree of community α;
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

CocoNuTs
Complex Networks
@networksvox
Everything is connected

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to.
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to.
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to.
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions:
  $P_{>}(m)$, $P_{>}(s^{\text{ov}}_{\alpha,\beta})$, $P_{>}(d^{\text{com}}_{\alpha})$, and $P_{>}(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to.
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to.
  - $s^{\text{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - d^{com}_{α}, degree of community α.
  - s^{com}_{α}, community α's size.
- Associated distributions: $P_>(m)$, $P_>(s^{\text{ov}}_{\alpha,\beta})$, $P_>(d^{\text{com}}_{\alpha})$, and $P_>(s^{\text{com}}_{\alpha})$.

Dealing with community overlap:

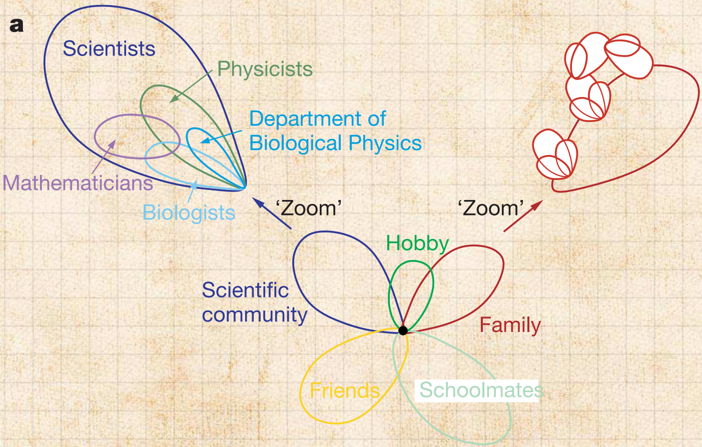
- Earlier structure detection algorithms, agglomerative or divisive, force communities to be purely distinct.
- Overlap: Acknowledge nodes can belong to multiple communities.
- Palla et al. [13] detect communities as sets of adjacent k-cliques (must share $k-1$ nodes).
- One of several issues: how to choose k?
- Four new quantities:
  - m, number of a communities a node belongs to.
  - $s^{\textrm{ov}}_{\alpha,\beta}$, number of nodes shared between two given communities, α and β.
  - $d^{\textrm{com}}_{\alpha}$, degree of community α.
  - $s^{\textrm{com}}_{\alpha}$, community α's size.
- Associated distributions: $P_{>}(m)$, $P_{>}(s^{\textrm{ov}}_{\alpha,\beta})$, $P_{>}(d^{\textrm{com}}_{\alpha})$, and $P_{>}(s^{\textrm{com}}_{\alpha})$.

LETTERS

Uncovering the overlapping community structure of complex networks in nature and society

"Uncovering the overlapping community structure of complex networks in nature and society"

Palla et al.,
Nature, **435**, 814–818, 2005. [13]



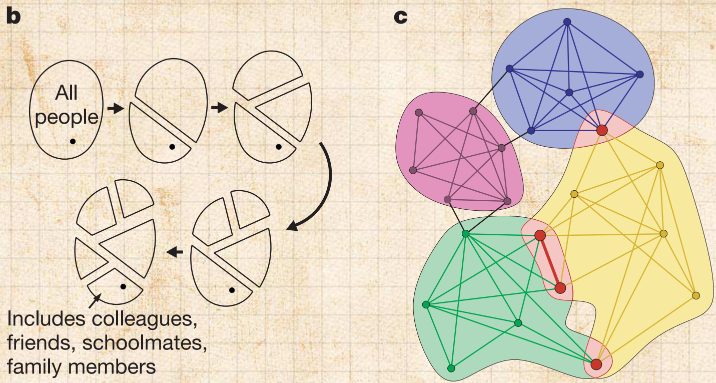


Figure 1 | Illustration of the concept of overlapping communities. a, The black dot in the middle represents either of the authors of this paper, with several of his communities around. Zooming in on the scientific community demonstrates the nested and overlapping structure of the communities, and depicting the cascades of communities starting from some members exemplifies the interwoven structure of the network of communities. **b**, Divisive and agglomerative methods grossly fail to identify the communities when overlaps are significant. **c**, An example of overlapping *k*-clique communities at $k = 4$. The yellow community overlaps the blue one in a single node, whereas it shares two nodes and a link with the green one. These overlapping regions are emphasized in red. Notice that any *k*-clique (complete subgraph of size *k*) can be reached only from the *k*-cliques of the same community through a series of adjacent *k*-cliques. Two *k*-cliques are adjacent if they share $k - 1$ nodes.

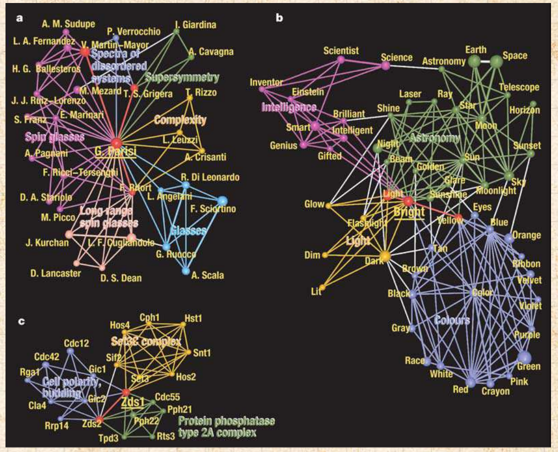


Figure 2 | **The community structure around a particular node in three different networks.** The communities are colour coded, the overlapping nodes and links between them are emphasized in red, and the volume of the balls and the width of the links are proportional to the total number of communities they belong to. For each network the value of k has been set to 4. **a**, The communities of G. Parisi in the co-authorship network of the Los Alamos Condensed Matter archive (for threshold weight $w^* = 0.75$) can be associated with his fields of interest. **b**, The communities of the word 'bright' in the South Florida Free Association norms list (for $w^* = 0.025$) represent the different meanings of this word. **c**, The communities of the protein Zds1 in the DIP core list of the protein–protein interactions of *S. cerevisiae* can be associated with either protein complexes or certain functions.

- Two tunable parameters: w^*, the link weight threshold, and k, the clique size.

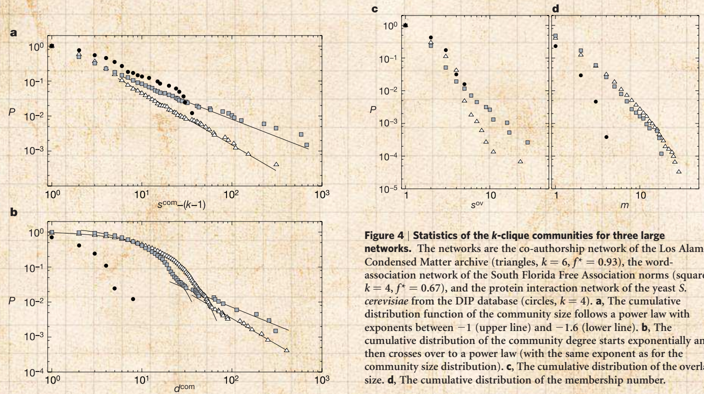


Figure 4 | **Statistics of the *k*-clique communities for three large networks.** The networks are the co-authorship network of the Los Alamos Condensed Matter archive (triangles, $k = 6$, $f^* = 0.93$), the word-association network of the South Florida Free Association norms (squares, $k = 4$, $f^* = 0.67$), and the protein interaction network of the yeast *S. cerevisiae* from the DIP database (circles, $k = 4$). **a**, The cumulative distribution function of the community size follows a power law with exponents between -1 (upper line) and -1.6 (lower line). **b**, The cumulative distribution of the community degree starts exponentially and then crosses over to a power law (with the same exponent as for the community size distribution). **c**, The cumulative distribution of the overlap size. **d**, The cumulative distribution of the membership number.

Outline

Overview

Methods

Hierarchy by aggregation

Hierarchy by division

Hierarchy by shuffling

Spectral methods

Hierarchies & Missing Links

Overlapping communities

Link-based methods

General structure detection

References

A link-based approach:

- What we know now: Many network analyses profit from focusing on links.
- Idea: form communities of links rather than communities of nodes.
- Observation: Links typically of one flavor, while nodes may have many flavors.
- Link communities induce overlapping and still hierarchically structured communities of nodes.
- [Applause.]

A link-based approach:

- What we know now: Many network analyses profit from focusing on links.
- Idea: form communities of links rather than communities of nodes.
- Observation: Links typically of one flavor, while nodes may have many flavors.
- Link communities induce overlapping and still hierarchically structured communities of nodes.
- [Applause.]

A link-based approach:

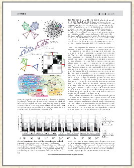
- What we know now: Many network analyses profit from focusing on links.
- Idea: form communities of links rather than communities of nodes.
- Observation: Links typically of one flavor, while nodes may have many flavors.
- Link communities induce overlapping and still hierarchically structured communities of nodes.
- [Applause.]

A link-based approach:

- What we know now: Many network analyses profit from focusing on links.
- Idea: form communities of links rather than communities of nodes.
- Observation: Links typically of one flavor, while nodes may have many flavors.
- Link communities induce overlapping and still hierarchically structured communities of nodes.
- [Applause.]

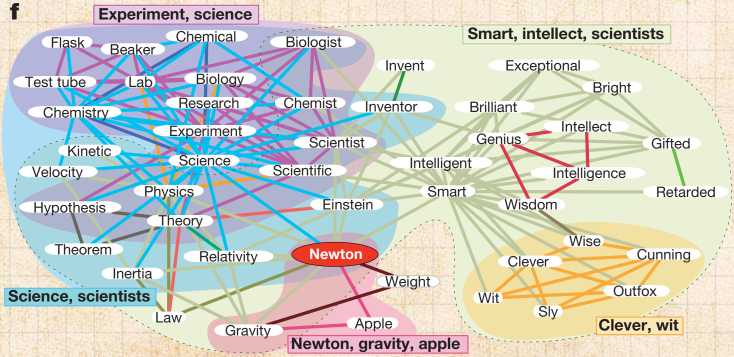
A link-based approach:

- What we know now: Many network analyses profit from focusing on links.
- Idea: form communities of links rather than communities of nodes.
- Observation: Links typically of one flavor, while nodes may have many flavors.
- Link communities induce overlapping and still hierarchically structured communities of nodes.
- [Applause.]



“Link communities reveal multiscale complexity in networks”

Ahn, Bagrow, and Lehmann, Nature, **466**, 761–764, 2010. [1]



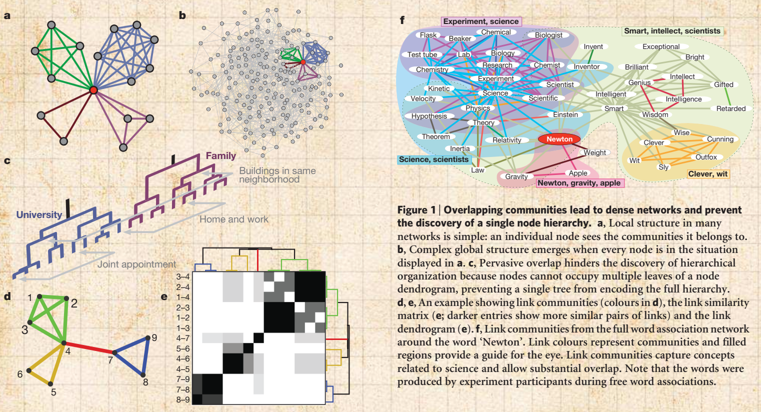


Figure 1 | Overlapping communities lead to dense networks and prevent the discovery of a single node hierarchy. a, Local structure in many networks is simple: an individual node sees the communities it belongs to. **b**, Complex global structure emerges when every node is in the situation displayed in **a**. **c**, Pervasive overlap hinders the discovery of hierarchical organization because nodes cannot occupy multiple leaves of a node dendrogram, preventing a single tree from encoding the full hierarchy. **d**, **e**, An example showing link communities (colours in **d**), the link similarity matrix (**e**; darker entries show more similar pairs of links) and the link dendrogram (**e**). **f**, Link communities from the full word association network around the word 'Newton'. Link colours represent communities and filled regions provide a guide for the eye. Link communities capture concepts related to science and allow substantial overlap. Note that the words were produced by experiment participants during free word associations.

Note: See details of paper on how to choose link communities well based on partition density D.

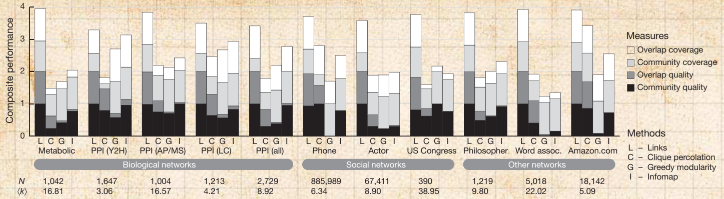
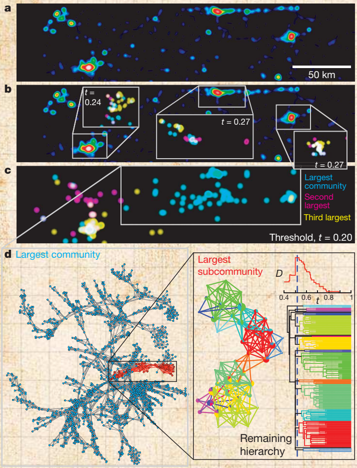


Figure 2 | Assessing the relevance of link communities using real-world networks. Composite performance (Methods and Supplementary Information) is a data-driven measure of the quality (relevance of discovered memberships) and coverage (fraction of network classified) of community and overlap. Tested algorithms are link clustering, introduced here; clique percolation[9]; greedy modularity optimization[26]; and Infomap[21]. Test networks were chosen for their varied sizes and topologies and to represent the different domains where network analysis is used. Shown for each are the number of nodes, N, and the average number of neighbours per node, $\langle k \rangle$. Link clustering finds the most relevant community structure in real-world networks. AP/MS, affinity-purification/mass spectrometry; LC, literature curated; PPI, protein–protein interaction; Y2H, yeast two-hybrid.

- Comparison of structure detection algorithms using four measures over many networks.
- Revealed communities are matched against ‘known’ communities recorded in network metadata.
- Link approach particularly good for dense, overlapful networks.



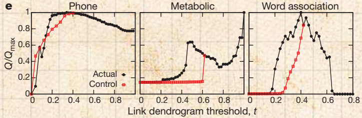


Figure 4 | Meaningful communities at multiple levels of the link dendrogram. **a**–**c**, The social network of mobile phone users displays co-located, overlapping communities on multiple scales. **a**, Heat map of the most likely locations of all users in the region, showing several cities. **b**, Cutting the dendrogram above the optimum threshold yields small, intra-city communities (insets). **c**, Below the optimum threshold, the largest communities become spatially extended but still show correlation. **d**, The social network within the largest community in **c**, with its largest subcommunity highlighted. The highlighted subcommunity is shown along with its link dendrogram and partition density, D, as a function of threshold, t. Link colours correspond to dendrogram branches. **e**, Community quality, Q, as a function of dendrogram level, compared with random control (Methods).

Outline

Overview

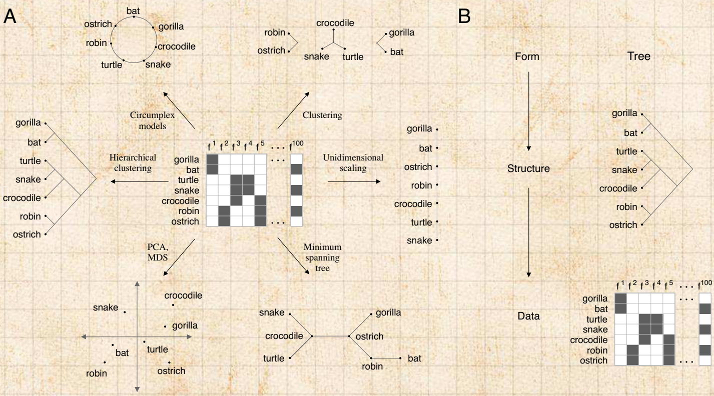
Methods

- Hierarchy by aggregation
- Hierarchy by division
- Hierarchy by shuffling
- Spectral methods
- Hierarchies & Missing Links
- Overlapping communities
- Link-based methods
- General structure detection

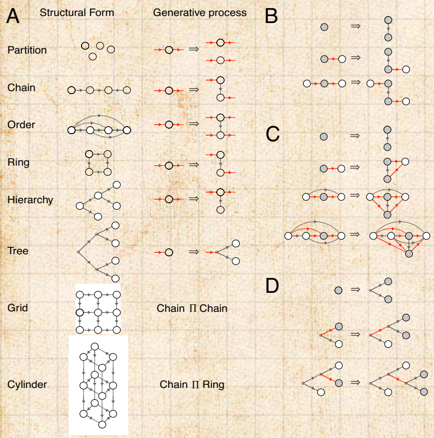
References

General structure detection

- "The discovery of structural form"
  Kemp and Tenenbaum, PNAS (2008) [8]

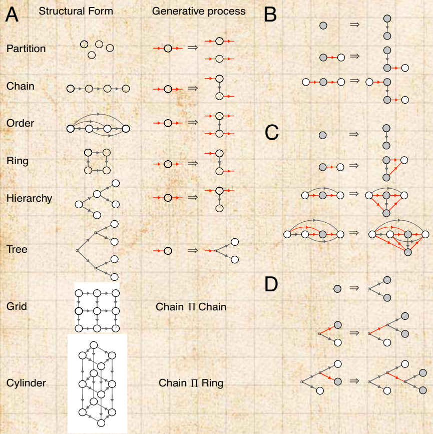


General structure detection



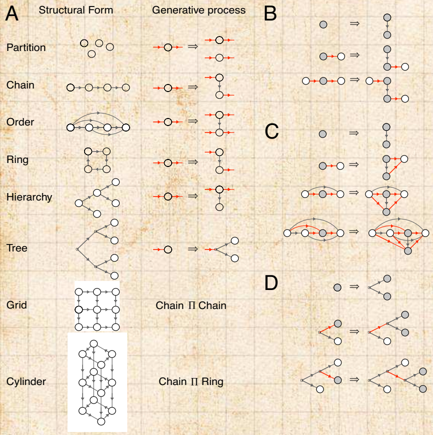
- Top down description of form.
- Node replacement graph grammar: parent node becomes two child nodes.
- B-D: Growing chains, orders, and trees.

General structure detection



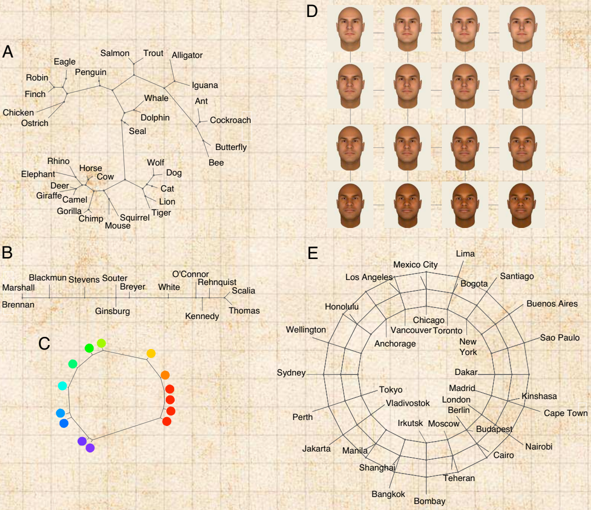
- Top down description of form.
- Node replacement graph grammar: parent node becomes two child nodes.
- B-D: Growing chains, orders, and trees.

General structure detection



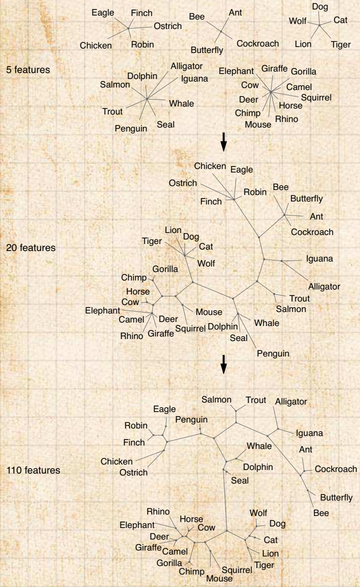
- Top down description of form.
- Node replacement graph grammar: parent node becomes two child nodes.
- B-D: Growing chains, orders, and trees.

Example learned structures:



- Biological features; Supreme Court votes; perceived color differences; face differences; & distances between cities.

General structure detection



- Effect of adding features on detected form.

Straight partition

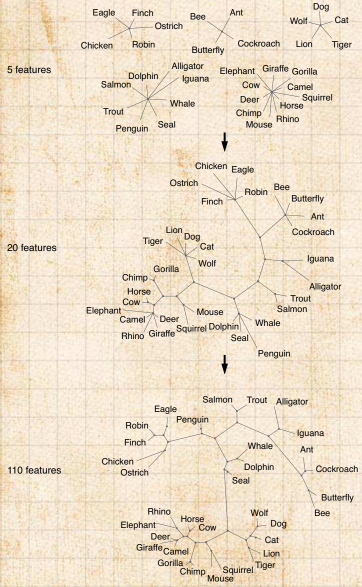
⇓

simple tree

⇓

complex tree

General structure detection



- Effect of adding features on detected form.

Straight partition

⇓

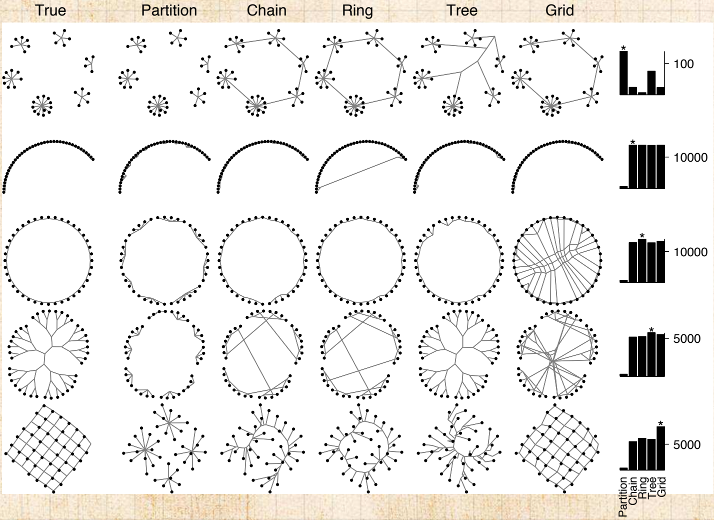
simple tree

⇓

complex tree

General structure detection

Performance for test networks.



References I

[1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann.
Link communities reveal multiscale complexity in networks.
Nature, 466(7307):761–764, 2010. pdf

[2] P. M. Blau and J. E. Schwartz.
Crosscutting Social Circles.
Academic Press, Orlando, FL, 1984.

[3] R. L. Breiger.
The duality of persons and groups.
Social Forces, 53(2):181–190, 1974. pdf

[4] A. Capocci, V. Servedio, G. Caldarelli, and F. Colaiori.
Detecting communities in large networks.
Physica A: Statistical Mechanics and its Applications, 352:669–676, 2005. pdf

References II

[5] A. Clauset, C. Moore, and M. E. J. Newman.
Hierarchical structure and the prediction of missing links in networks.
Nature, 453:98–101, 2008. pdf

[6] S. Fortunato.
Community detection in graphs.
Physics Reports, 486:75–174, 2010. pdf

[7] M. Girvan and M. E. J. Newman.
Community structure in social and biological networks.
Proc. Natl. Acad. Sci., 99:7821–7826, 2002. pdf

[8] C. Kemp and J. B. Tenenbaum.
The discovery of structural form.
Proc. Natl. Acad. Sci., 105:10687–10692, 2008. pdf

References III

[9] D. P. Kiley, A. J. Reagan, L. Mitchell, C. M. Danforth, and P. S. Dodds.
The game story space of professional sports: Australian Rules Football.
Draft version of the present paper using pure random walk null model. Available online at `http://arxiv.org/abs/1507.03886v1`. Accesssed January 17, 2016, 2015. pdf

[10] M. E. J. Newman.
Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality.
Phys. Rev. E, 64(1):016132, 2001. pdf

References IV

[11] M. E. J. Newman.
Erratum: Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality [Phys. Rev. E 64, 016132 (2001)].
Phys. Rev. E, 73:039906(E), 2006. pdf

[12] M. E. J. Newman and M. Girvan.
Finding and evaluating community structure in networks.
Phys. Rev. E, 69(2):026113, 2004. pdf

[13] G. Palla, I. Derényi, I. Farkas, and T. Vicsek.
Uncovering the overlapping community structure of complex networks in nature and society.
Nature, 435(7043):814–818, 2005. pdf

References V

[14] M. Sales-Pardo, R. Guimerà, A. A. Moreira, and L. A. N. Amaral.
Extracting the hierarchical organization of complex systems.
Proc. Natl. Acad. Sci., 104:15224–15229, 2007. pdf

[15] M. Sales-Pardo, R. Guimerà, A. A. Moreira, and L. A. N. Amaral.
Extracting the hierarchical organization of complex systems: Correction.
Proc. Natl. Acad. Sci., 104:18874, 2007. pdf

[16] G. Simmel.
The number of members as determining the sociological form of the group. I.
American Journal of Sociology, 8:1–46, 1902.

References VI

[17] D. J. Watts, P. S. Dodds, and M. E. J. Newman.
Identity and search in social networks.
Science, 296:1302–1305, 2002. pdf

[18] W. W. Zachary.
An information flow model for conflict and fission in small groups.
J. Anthropol. Res., 33:452–473, 1977.